

RIP Feature Guide for the OCX Series

Release

14.1X53



Modified: 2015-06-17

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Juniper Networks, Junos, Steel-Belted Radius, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. The Juniper Networks Logo, the Junos logo, and JunosE are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

RIP Feature Guide for the OCX Series

14.1X53

Copyright © 2015, Juniper Networks, Inc.

All rights reserved.

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at <http://www.juniper.net/support/eula.html>. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

	About the Documentation	xi
	Documentation and Release Notes	xi
	Supported Platforms	xi
	Using the Examples in This Manual	xi
	Merging a Full Example	xii
	Merging a Snippet	xii
	Documentation Conventions	xiii
	Documentation Feedback	xv
	Requesting Technical Support	xv
	Self-Help Online Tools and Resources	xv
	Opening a Case with JTAC	xvi
Part 1	Using RIP	
	RIP Overview	1
	Distance-Vector Routing Protocols	2
	RIP Protocol Overview	2
	RIP Packets	3
	Maximizing Hop Count	4
	Split Horizon and Poison Reverse Efficiency Techniques	4
	Limitations of Unidirectional Connectivity	6
Chapter 1	Example: Configuring RIP	7
	Understanding Basic RIP Routing	7
	Example: Configuring a Basic RIP Network	7
Chapter 2	Example: Configuring Authentication for RIP Routes	15
	Understanding RIP Authentication	15
	Example: Configuring Route Authentication for RIP	15
	Enabling Authentication with Plain-Text Passwords (CLI Procedure)	20
	Enabling Authentication with MD5 Authentication (CLI Procedure)	21
Chapter 3	Example: Configuring BFD for RIP	23
	Understanding BFD for RIP	23
	Example: Configuring BFD for RIP	24
Chapter 4	Example: Configuring BFD Authentication for RIP	31
	Understanding BFD Authentication for RIP	31
	BFD Authentication Algorithms	31
	Security Authentication Keychains	32
	Strict Versus Loose Authentication	32
	Example: Configuring BFD Authentication for RIP	33

Chapter 5	Example: Applying Policies to RIP Routes Imported from Neighbors	41
	Understanding RIP Import Policy	41
	Example: Applying Policies to RIP Routes Imported from Neighbors	41
Chapter 6	Examples: Controlling Traffic with Metrics in a RIP Network	47
	Understanding Traffic Control with Metrics in a RIP Network	47
	Example: Controlling Traffic in a RIP Network with an Incoming Metric	48
	Example: Controlling Traffic in a RIP Network with an Outgoing Metric	50
	Example: Configuring the Metric Value Added to Imported RIP Routes	52
Chapter 7	Example: Configuring the Sending and Receiving of RIPv1 and RIPv2 Packets	57
	Understanding the Sending and Receiving of RIPv1 and RIPv2 Packets	57
	Example: Configuring the Sending and Receiving of RIPv1 and RIPv2 Packets	57
Chapter 8	Example: Redistributing Routes Among RIP Instances	63
	Understanding Route Redistribution Among RIP instances	63
	Example: Redistributing Routes Between Two RIP Instances	64
Chapter 9	Example: Configuring RIP Timers	71
	Understanding RIP Timers	71
	Example: Configuring RIP Timers	72
Chapter 10	Example: Tracing RIP Protocol Traffic	79
	Understanding RIP Trace Operations	79
	Example: Tracing RIP Protocol Traffic	80
	Monitoring RIP Routing Information	?
Part 2	Configuration Statements and Operational Commands	
Chapter 11	Configuration Statements for RIP	87
	any-sender	88
	authentication-key (Protocols RIP)	89
	authentication-type (Protocols RIP)	90
	bfd-liveness-detection (Protocols RIP)	91
	check-zero	94
	export (Protocols RIP)	95
	group (Protocols RIP)	96
	holddown (Protocols RIP)	98
	import (Protocols RIP)	99
	message-size	100
	metric-in (Protocols RIP)	101
	metric-out (Protocols RIP)	102
	neighbor (Protocols RIP)	103
	preference (Protocols RIP)	104
	receive (Protocols RIP)	105
	rib-group (Protocols RIP)	106
	rip	106
	route-timeout (Protocols RIP)	107
	send (Protocols RIP)	108
	traceoptions (Protocols RIP)	109

	update-interval (Protocols RIP)	112
Chapter 12	RIP Operational Commands	113
	clear rip general-statistics	114
	clear rip statistics	115
	show rip general-statistics	116
	show rip neighbor	118
	show rip statistics	120

List of Figures

Part 1	Using RIP	
	Figure 1: Distance-Vector Protocol	2
	Figure 2: Split Horizon Example	5
	Figure 3: Poison Reverse Example	5
	Figure 4: Limitations of Unidirectional Connectivity	6
Chapter 1	Example: Configuring RIP	7
	Figure 5: Sample RIP Network Topology	8
Chapter 2	Example: Configuring Authentication for RIP Routes	15
	Figure 6: RIP Authentication Network Topology	16
Chapter 3	Example: Configuring BFD for RIP	23
	Figure 7: RIP BFD Network Topology	26
Chapter 4	Example: Configuring BFD Authentication for RIP	31
	Figure 8: RIP BFD Authentication Network Topology	33
Chapter 5	Example: Applying Policies to RIP Routes Imported from Neighbors	41
	Figure 9: RIP Import Policy Network Topology	42
Chapter 6	Examples: Controlling Traffic with Metrics in a RIP Network	47
	Figure 10: Controlling Traffic in a RIP Network with the Incoming Metric	49
	Figure 11: Controlling Traffic in a RIP Network with the Outgoing Metric	51
	Figure 12: RIP Incoming Metrics Network Topology	52
Chapter 7	Example: Configuring the Sending and Receiving of RIPv1 and RIPv2 Packets	57
	Figure 13: Sending and Receiving RIPv1 and RIPv2 Packets Network Topology	58
Chapter 8	Example: Redistributing Routes Among RIP Instances	63
	Figure 14: Redistributing Routes Between RIP Instances Network Topology	65
Chapter 9	Example: Configuring RIP Timers	71
	Figure 15: RIP Timers Network Topology	73
Chapter 10	Example: Tracing RIP Protocol Traffic	79
	Figure 16: RIP Trace Operations Network Topology	81

List of Tables

	About the Documentation	xi
	Table 1: Notice Icons	xiii
	Table 2: Text and Syntax Conventions	xiii
Part 1	Using RIP	
Chapter 2	Example: Configuring Authentication for RIP Routes	15
	Table 3: Configuring Simple RIP Authentication	20
	Table 4: Configuring MD5 RIP Authentication	21
Part 2	Configuration Statements and Operational Commands	
Chapter 12	RIP Operational Commands	113
	Table 5: show rip general-statistics Output Fields	116
	Table 6: show rip neighbor Output Fields	119
	Table 7: show rip statistics Output Fields	121

About the Documentation

- Documentation and Release Notes on page xi
- Supported Platforms on page xi
- Using the Examples in This Manual on page xi
- Documentation Conventions on page xiii
- Documentation Feedback on page xv
- Requesting Technical Support on page xv

Documentation and Release Notes

To obtain the most current version of all Juniper Networks[®] technical documentation, see the product documentation page on the Juniper Networks website at <http://www.juniper.net/techpubs/>.

If the information in the latest release notes differs from the information in the documentation, follow the product Release Notes.

Juniper Networks Books publishes books by Juniper Networks engineers and subject matter experts. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration. The current list can be viewed at <http://www.juniper.net/books>.

Supported Platforms

For the features described in this document, the following platforms are supported:

- OCX1100

Using the Examples in This Manual

If you want to use the examples in this manual, you can use the **load merge** or the **load merge relative** command. These commands cause the software to merge the incoming configuration into the current candidate configuration. The example does not become active until you commit the candidate configuration.

If the example configuration contains the top level of the hierarchy (or multiple hierarchies), the example is a *full example*. In this case, use the **load merge** command.

If the example configuration does not start at the top level of the hierarchy, the example is a *snippet*. In this case, use the **load merge relative** command. These procedures are described in the following sections.

Merging a Full Example

To merge a full example, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration example into a text file, save the file with a name, and copy the file to a directory on your routing platform.

For example, copy the following configuration to a file and name the file **ex-script.conf**. Copy the **ex-script.conf** file to the **/var/tmp** directory on your routing platform.

```
system {
  scripts {
    commit {
      file ex-script.xml;
    }
  }
}
interfaces {
  fxp0 {
    disable;
    unit 0 {
      family inet {
        address 10.0.0.1/24;
      }
    }
  }
}
```

2. Merge the contents of the file into your routing platform configuration by issuing the **load merge** configuration mode command:

```
[edit]
user@host# load merge /var/tmp/ex-script.conf
load complete
```

Merging a Snippet

To merge a snippet, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration snippet into a text file, save the file with a name, and copy the file to a directory on your routing platform.

For example, copy the following snippet to a file and name the file **ex-script-snippet.conf**. Copy the **ex-script-snippet.conf** file to the **/var/tmp** directory on your routing platform.

```
commit {
  file ex-script-snippet.xml; }
```

2. Move to the hierarchy level that is relevant for this snippet by issuing the following configuration mode command:

```
[edit]
user@host# edit system scripts
[edit system scripts]
```

3. Merge the contents of the file into your routing platform configuration by issuing the **load merge relative** configuration mode command:

```
[edit system scripts]
user@host# load merge relative /var/tmp/ex-script-snippet.conf
load complete
```

For more information about the **load** command, see the *CLI User Guide*.

Documentation Conventions

Table 1 on page xiii defines notice icons used in this guide.

Table 1: Notice Icons

Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.
	Tip	Indicates helpful information.
	Best practice	Alerts you to a recommended use or implementation.

Table 2 on page xiii defines the text and syntax conventions used in this guide.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the configure command: user@host> configure

Table 2: Text and Syntax Conventions (*continued*)

Convention	Description	Examples
Fixed-width text like this	Represents output that appears on the terminal screen.	<code>user@host> show chassis alarms</code> <code>No alarms currently active</code>
<i>Italic text like this</i>	<ul style="list-style-type: none">Introduces or emphasizes important new terms.Identifies guide names.Identifies RFC and Internet draft titles.	<ul style="list-style-type: none">A policy <i>term</i> is a named structure that defines match conditions and actions.<i>Junos OS CLI User Guide</i>RFC 1997, <i>BGP Communities Attribute</i>
<i>Italic text like this</i>	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# set system domain-name <i>domain-name</i>
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	<ul style="list-style-type: none">To configure a stub area, include the stub statement at the [edit protocols ospf area area-id] hierarchy level.The console port is labeled CONSOLE.
< > (angle brackets)	Encloses optional keywords or variables.	stub <default-metric <i>metric</i>>;
(pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	broadcast multicast (<i>string1</i> <i>string2</i> <i>string3</i>)
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	rsvp { # Required for dynamic MPLS only
[] (square brackets)	Encloses a variable for which you can substitute one or more values.	community name members [<i>community-ids</i>]
Indentation and braces ({ })	Identifies a level in the configuration hierarchy.	[edit] routing-options { static { route default { nexthop <i>address</i> ; retain; } } }
;(semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
GUI Conventions		
Bold text like this	Represents graphical user interface (GUI) items you click or select.	<ul style="list-style-type: none">In the Logical Interfaces box, select All Interfaces.To cancel the configuration, click Cancel.

Table 2: Text and Syntax Conventions (*continued*)

Convention	Description	Examples
> (bold right angle bracket)	Separates levels in a hierarchy of menu selections.	In the configuration editor hierarchy, select Protocols>Ospf .

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can provide feedback by using either of the following methods:

- Online feedback rating system—On any page at the Juniper Networks Technical Documentation site at <http://www.juniper.net/techpubs/index.html>, simply click the stars to rate the content, and use the pop-up form to provide us with information about your experience. Alternately, you can use the online feedback form at <https://www.juniper.net/cgi-bin/docbugreport/>.
- E-mail—Send your comments to techpubs-comments@juniper.net. Include the document or topic name, URL or page number, and software version (if applicable).

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or Partner Support Service support contract, or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at <http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf>.
- Product warranties—For product warranty information, visit <http://www.juniper.net/support/warranty/>.
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: <http://www.juniper.net/customers/support/>
- Search for known bugs: <http://www2.juniper.net/kb/>
- Find product documentation: <http://www.juniper.net/techpubs/>
- Find solutions and answer questions using our Knowledge Base: <http://kb.juniper.net/>

- Download the latest versions of software and review release notes:
<http://www.juniper.net/customers/csc/software/>
- Search technical bulletins for relevant hardware and software notifications:
<http://kb.juniper.net/InfoCenter/>
- Join and participate in the Juniper Networks Community Forum:
<http://www.juniper.net/company/communities/>
- Open a case online in the CSC Case Management tool: <http://www.juniper.net/cm/>

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: <https://tools.juniper.net/SerialNumberEntitlementSearch/>

Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Management tool in the CSC at <http://www.juniper.net/cm/>.
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, see <http://www.juniper.net/support/requesting-support.html>.

PART 1

Using RIP

- [RIP Overview on page 1](#)
- [Example: Configuring RIP on page 7](#)
- [Example: Configuring Authentication for RIP Routes on page 15](#)
- [Example: Configuring BFD for RIP on page 23](#)
- [Example: Configuring BFD Authentication for RIP on page 31](#)
- [Example: Applying Policies to RIP Routes Imported from Neighbors on page 41](#)
- [Examples: Controlling Traffic with Metrics in a RIP Network on page 47](#)
- [Example: Configuring the Sending and Receiving of RIPv1 and RIPv2 Packets on page 57](#)
- [Example: Redistributing Routes Among RIP Instances on page 63](#)
- [Example: Configuring RIP Timers on page 71](#)
- [Example: Tracing RIP Protocol Traffic on page 79](#)
- [Monitoring RIP Routing Information on page ?](#)

RIP Overview

RIP is an interior gateway protocol (IGP) that uses a distance-vector algorithm to determine the best route to a destination, using the hop count as the metric.

In a RIP network, each router's forwarding table is distributed among the nodes through the flooding of routing table information. Because topology changes are flooded throughout the network, every node maintains the same list of destinations. Packets are then routed to these destinations based on path-cost calculations done at each node in the network.



NOTE: In general, the term *RIP* refers to RIP version 1 and RIP version 2.

This topic contains the following sections:

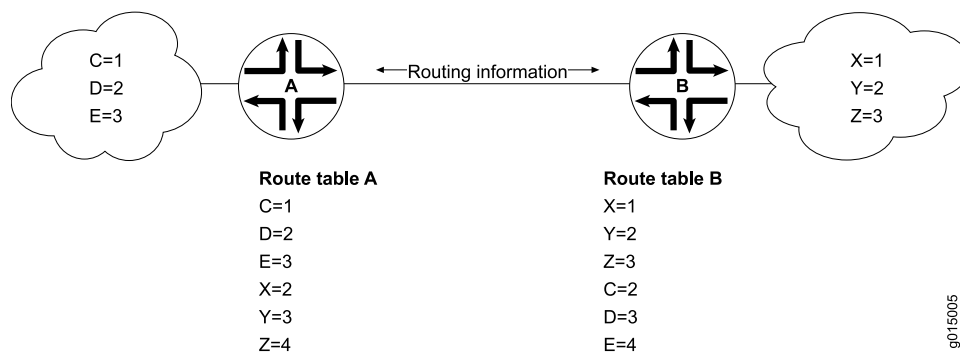
- [Distance-Vector Routing Protocols on page 2](#)
- [RIP Protocol Overview on page 2](#)
- [RIP Packets on page 3](#)
- [Maximizing Hop Count on page 4](#)

- [Split Horizon and Poison Reverse Efficiency Techniques on page 4](#)
- [Limitations of Unidirectional Connectivity on page 6](#)

Distance-Vector Routing Protocols

Distance-vector routing protocols transmit routing information that includes a distance vector, typically expressed as the number of hops to the destination. This information is flooded out all protocol-enabled interfaces at regular intervals (every 30 seconds in the case of RIP) to create a network map that is stored in each node's local topology database. [Figure 1 on page 2](#) shows how distance-vector routing works.

Figure 1: Distance-Vector Protocol



In [Figure 1 on page 2](#), Routers A and B have RIP enabled on adjacent interfaces. Router A has known RIP neighbors Routers C, D, and E, which are 1, 2, and 3 hops away, respectively. Router B has known RIP neighbors Routers X, Y, and Z, which are 1, 2, and 3 hops away, respectively. Every 30 seconds, each router floods its entire routing table information out all RIP-enabled interfaces. In this case, flooding exchanges routing table information across the RIP link.

When Router A receives routing information from Router B, it adds 1 to the hop count to determine the new hop count. For example, Router X has a hop count of 1, but when Router A imports the route to X, the new hop count is 2. The imported route also includes information about where the route was learned, so that the original route is imported as a route to Router X through Router B with a hop count of 2.

When multiple routes to the same host are received, RIP uses the distance-vector algorithm to determine which path to import into the forwarding table. The route with the smallest hop count is imported. If there are multiple routes with the same hop count, all are imported into the forwarding table, and traffic is sent along the paths in round-robin fashion.

RIP Protocol Overview

The RIP IGP uses the Bellman-Ford, or *distance-vector*, algorithm to determine the best route to a destination. RIP uses the hop count as the metric. RIP enables hosts and routers to exchange information for computing routes through an IP-based network. RIP is intended to be used as an IGP in reasonably homogeneous networks of moderate size.

The Junos[®] operating system (Junos OS) supports RIP versions 1 and 2.



NOTE: RIP is not supported for multipoint interfaces.

RIP version 1 packets contain the minimal information necessary to route packets through a network. However, this version of RIP does not support authentication or subnetting.

RIP uses User Datagram Protocol (UDP) port 520.

RIP has the following architectural limitations:

- The longest network path cannot exceed 15 hops (assuming that each network, or hop, has a cost of 1).
- RIP depends on counting to infinity to resolve certain unusual situations—When the network consists of several hundred routers, and when a routing loop has formed, the amount of time and network bandwidth required to resolve a next hop might be great.
- RIP uses only a fixed metric to select a route. Other IGP's use additional parameters, such as measured delay, reliability, and load.

RIP Packets

RIP packets contain the following fields:

- **Command**—Indicates whether the packet is a request or response message. Request messages seek information for the router's routing table. Response messages are sent periodically and also when a request message is received. Periodic response messages are called *update messages*. Update messages contain the command and version fields and 25 destinations (by default), each of which includes the destination IP address and the metric to reach that destination.



NOTE: Beginning with Junos OS Release 11.1, three additional command field types are available to support RIP demand circuits. When you configure an interface for RIP demand circuits, the command field indicates whether the packet is an update request, update response, or update acknowledge message. Neighbor interfaces send updates on demand, not periodically. These command field types are only valid on interfaces configured for RIP demand circuits. For more detailed information, see *RIP Demand Circuits Overview*.

- **Version number**—Version of RIP that the originating router is running.
- **Address family identifier**—Address family used by the originating router. The family is always IP.
- **Address**—IP address included in the packet.
- **Metric**—Value of the metric advertised for the address.
- **Mask**—Mask associated with the IP address (RIP version 2 only).
- **Next hop**—IP address of the next-hop router (RIP version 2 only).

Routing information is exchanged in a RIP network by RIP request and RIP response packets. A router that has just booted can broadcast a RIP request on all RIP-enabled interfaces. Any routers running RIP on those links receive the request and respond by sending a RIP response packet immediately to the router. The response packet contains the routing table information required to build the local copy of the network topology map.

In the absence of RIP request packets, all RIP routers broadcast a RIP response packet every 30 seconds on all RIP-enabled interfaces. The RIP broadcast is the primary way in which topology information is flooded throughout the network.

Once a router learns about a particular destination through RIP, it starts a timer. Every time it receives a new response packet with information about the destination, the router resets the timer to zero. However, if the router receives no updates about a particular destination for 180 seconds, it removes the destination from its RIP routing table.

In addition to the regular transmission of RIP packets every 30 seconds, if a router detects a new neighbor or detects that an interface is unavailable, it generates a triggered update. The new routing information is immediately broadcast out all RIP-enabled interfaces, and the change is reflected in all subsequent RIP response packets.

Maximizing Hop Count

The successful routing of traffic across a RIP network requires that every node in the network maintain the same view of the topology. Topology information is broadcast between RIP neighbors every 30 seconds. If Router A is many hops away from a new host, Router B, the route to B might take significant time to propagate through the network and be imported into Router A's routing table. If the two routers are 5 hops away from each other, Router A cannot import the route to Router B until 2.5 minutes after Router B is online (30 seconds per hop). For large numbers of hops, the delay becomes prohibitive. To help prevent this delay from growing arbitrarily large, RIP enforces a maximum hop count of 15 hops. Any prefix that is more than 15 hops away is treated as unreachable and assigned a hop count equal to infinity. This maximum hop count is called the *network diameter*.

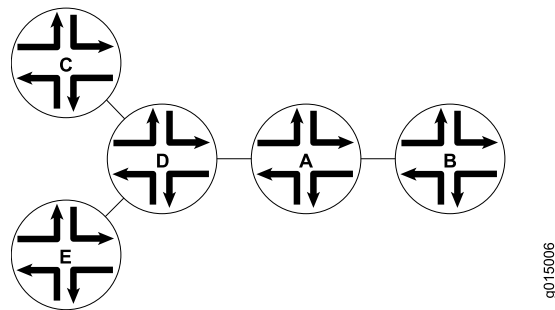
Split Horizon and Poison Reverse Efficiency Techniques

Because RIP functions by periodically flooding the entire routing table out to the network, it generates a lot of traffic. The split horizon and poison reverse techniques can help reduce the amount of network traffic originated by RIP hosts and make the transmission of routing information more efficient.

If a router receives a set of route advertisements on a particular interface, RIP determines that those advertisements do not need to be retransmitted out the same interface. This technique, known as *split horizon*, helps limit the amount of RIP routing traffic by eliminating information that other neighbors on that interface have already learned.

[Figure 2 on page 5](#) shows an example of the split horizon technique.

Figure 2: Split Horizon Example

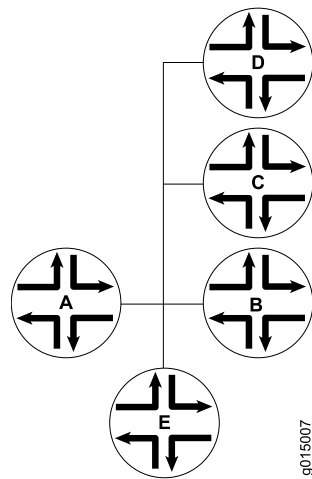


In [Figure 2 on page 5](#), Router A advertises routes to Routers C, D, and E to Router B. In this example, Router A can reach Router C in 2 hops. When Router A advertises the route to Router B, Router B imports it as a route to Router C through Router A in 3 hops. If Router B then readvertised this route to Router A, Router A would import it as a route to Router C through Router B in 4 hops. However, the advertisement from Router B to Router A is unnecessary, because Router A can already reach the route in 2 hops. The split horizon technique helps reduce extra traffic by eliminating this type of route advertisement.

Similarly, the poison reverse technique helps to optimize the transmission of routing information and improve the time to reach network convergence. If Router A learns about unreachable routes through one of its interfaces, it advertises those routes as unreachable (hop count of 16) out the same interface.

[Figure 3 on page 5](#) shows an example of the poison reverse technique.

Figure 3: Poison Reverse Example

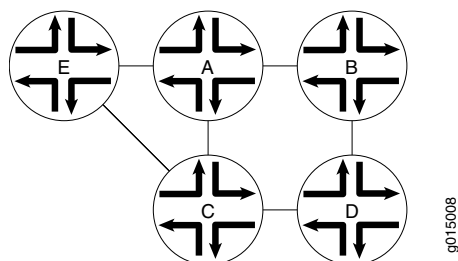


In [Figure 3 on page 5](#), Router A learns through one of its interfaces that routes to Routers C, D, and E are unreachable. Router A readvertises those routes out the same interface as unreachable. The advertisement informs Router B that Routers C, D, and E are definitely not reachable through Router A.

Limitations of Unidirectional Connectivity

Because RIP processes routing information based solely on the receipt of routing table updates, it cannot ensure bidirectional connectivity. As [Figure 4 on page 6](#) shows, RIP networks are limited by their unidirectional connectivity.

Figure 4: Limitations of Unidirectional Connectivity



In [Figure 4 on page 6](#), Routers A and D flood their routing table information to Router B. Because the path to Router E has the fewest hops when routed through Router A, that route is imported into Router B's forwarding table. However, suppose that Router A can transmit traffic but is not receiving traffic from Router B because of an unavailable link or invalid routing policy. If the only route to Router E is through Router A, any traffic destined for Router A is lost, because bidirectional connectivity was never established.

OSPF establishes bidirectional connectivity with a three-way handshake.

Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [RIP Configuration Overview](#)
- [Example: Configuring RIP on page 7](#)

CHAPTER 1

Example: Configuring RIP

- [Understanding Basic RIP Routing on page 7](#)
- [Example: Configuring a Basic RIP Network on page 7](#)

Understanding Basic RIP Routing

RIP is an interior gateway protocol (IGP) that routes packets within a single autonomous system (AS). By default, RIP does not advertise the subnets that are directly connected through the device's interfaces. For traffic to pass through a RIP network, you must create a routing policy to export these routes. Advertising only the direct routes propagates the routes to the immediately adjacent RIP-enabled router only. To propagate all routes through the entire RIP network, you must configure the routing policy to export the routes learned through RIP.

Related Documentation

- [*Junos OS Feature Support Reference for SRX Series and J Series Devices*](#)
- [RIP Overview on page 1](#)
- [Example: Configuring a Basic RIP Network on page 7](#)

Example: Configuring a Basic RIP Network

This example shows how to configure a basic RIP network.

- [Requirements on page 7](#)
- [Overview on page 8](#)
- [Configuration on page 8](#)
- [Verification on page 10](#)

Requirements

No special configuration beyond device initialization is required before configuring this example.

Overview

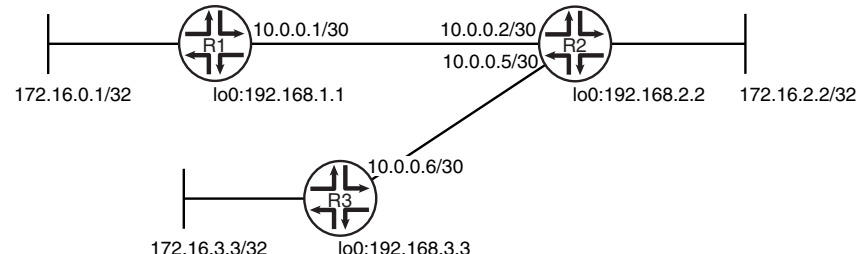
In this example, you configure a basic RIP network, create a RIP group called **rip-group**, and add the directly connected interfaces to the RIP group. Then you configure a routing policy to advertise direct routes using policy statement **advertise-routes-through-rip**.

By default, Junos OS does not advertise RIP routes, not even routes that are learned through RIP. To advertise RIP routes, you must configure and apply an export routing policy that advertises RIP-learned and direct routes.

In Junos OS, you do not need to configure the RIP version. RIP version 2 is used by default.

To use RIP on the device, you must configure RIP on all of the RIP interfaces within the network. [Figure 5 on page 8](#) shows the topology used in this example.

Figure 5: Sample RIP Network Topology



“CLI Quick Configuration” on [page 8](#) shows the configuration for all of the devices in [Figure 5 on page 8](#). The section “Step-by-Step Procedure” on [page 9](#) describes the steps on Device R1.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device R1

```
set interfaces fe-1/2/0 unit 1 family inet address 10.0.0.1/30
set interfaces lo0 unit 1 family inet address 172.16.0.1/32
set interfaces lo0 unit 1 family inet address 192.168.1.1/32
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.1
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept
```

Device R2

```
set interfaces fe-1/2/0 unit 2 family inet address 10.0.0.2/30
set interfaces fe-1/2/1 unit 5 family inet address 10.0.0.5/30
set interfaces lo0 unit 2 family inet address 192.168.2.2/32
set interfaces lo0 unit 2 family inet address 172.16.2.2/32
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.2
```



```

set protocols rip group rip-group neighbor fe-1/2/1.5
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept

```

Device R3

```

set interfaces fe-1/2/0 unit 6 family inet address 10.0.0.6/30
set interfaces lo0 unit 3 family inet address 192.168.3.3/32
set interfaces lo0 unit 3 family inet address 172.16.3.3/32
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.6
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept

```

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure a basic RIP network:

1. Configure the network interfaces.

This example shows multiple loopback interface addresses to simulate attached networks.

```

[edit interfaces]
user@R1# set fe-1/2/0 unit 1 family inet address 10.0.0.1/30

```

```

user@R1# set lo0 unit 1 family inet address 172.16.0.1/32
user@R1# set lo0 unit 1 family inet address 192.168.1.1/32

```

2. Create the RIP group and add the interface.

To configure RIP in Junos OS, you must configure a group that contains the interfaces on which RIP is enabled. You do not need to enable RIP on the loopback interface.

```

[edit protocols rip group rip-group]
user@R1# set neighbor fe-1/2/0.1

```

3. Create the routing policy to advertise both direct and RIP-learned routes.

```

[edit policy-options policy-statement advertise-routes-through-rip term 1]
user@R1# set from protocol direct
user@R1# set from protocol rip
user@R1# set then accept

```

4. Apply the routing policy.

In Junos OS, you can only apply RIP export policies at the group level.

```

[edit protocols rip group rip-group]
user@R1# set export advertise-routes-through-rip

```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, and **show policy-options** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@R1# show interfaces
fe-1/2/0 {
  unit 1 {
    family inet {
      address 10.0.0.1/30;
    }
  }
}
lo0 {
  unit 1 {
    family inet {
      address 172.16.0.1/32;
      address 192.168.1.1/32;
    }
  }
}

user@R1# show protocols
rip {
  group rip-group {
    export advertise-routes-through-rip;
    neighbor fe-1/2/0.1;
  }
}

user@R1# show policy-options
policy-statement advertise-routes-through-rip {
  term 1 {
    from protocol [ direct rip ];
    then accept;
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

- [Checking the Routing Table on page 10](#)
- [Looking at the Routes That Device R1 Is Advertising to Device R2 on page 11](#)
- [Looking at the Routes That Device R1 Is Receiving from Device R2 on page 11](#)
- [Verifying the RIP-Enabled Interfaces on page 12](#)
- [Verifying the Exchange of RIP Messages on page 12](#)
- [Verifying Reachability of All Hosts in the RIP Network on page 13](#)

Checking the Routing Table

Purpose Verify that the routing table is populated with the expected routes..

Action From operational mode, enter the **show route protocol rip** command.

```
user@R1> show route protocol rip
inet.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.4/30      *[RIP/100] 00:59:15, metric 2, tag 0
                 > to 10.0.0.2 via fe-1/2/0.1
172.16.2.2/32   *[RIP/100] 02:52:48, metric 2, tag 0
                 > to 10.0.0.2 via fe-1/2/0.1
172.16.3.3/32   *[RIP/100] 00:45:05, metric 3, tag 0
                 > to 10.0.0.2 via fe-1/2/0.1
192.168.2.2/32  *[RIP/100] 02:52:48, metric 2, tag 0
                 > to 10.0.0.2 via fe-1/2/0.1
192.168.3.3/32  *[RIP/100] 00:45:05, metric 3, tag 0
                 > to 10.0.0.2 via fe-1/2/0.1
224.0.0.9/32    *[RIP/100] 00:45:09, metric 1
                 MultiRecv
```

Meaning The output shows that the routes have been learned from Device R2 and Device R3.

If you were to delete the **from protocol rip** condition in the routing policy on Device R2, the remote routes from Device R3 would not be learned on Device R1.

Looking at the Routes That Device R1 Is Advertising to Device R2

Purpose Verify that Device R1 is sending the expected routes.

Action From operational mode, enter the **show route advertising-protocol rip 10.0.0.1** command.

```
user@R1> show route advertising-protocol rip 10.0.0.1
inet.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

172.16.0.1/32   *[Direct/0] 05:18:26
                 > via lo0.1
192.168.1.1/32  *[Direct/0] 05:18:25
                 > via lo0.1
```

Meaning Device R1 is sending routes to its directly connected networks.

Looking at the Routes That Device R1 Is Receiving from Device R2

Purpose Verify that Device R1 is receiving the expected routes.

Action From operational mode, enter the **show route receive-protocol rip** command.

```
user@R1> show route receive-protocol rip 10.0.0.2
inet.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.4/30      *[RIP/100] 02:31:22, metric 2, tag 0
                 > to 10.0.0.2 via fe-1/2/0.1
172.16.2.2/32   *[RIP/100] 04:24:55, metric 2, tag 0
                 > to 10.0.0.2 via fe-1/2/0.1
172.16.3.3/32   *[RIP/100] 02:17:12, metric 3, tag 0
                 > to 10.0.0.2 via fe-1/2/0.1
```

```

192.168.2.2/32    *[RIP/100] 04:24:55, metric 2, tag 0
                  > to 10.0.0.2 via fe-1/2/0.1
192.168.3.3/32    *[RIP/100] 02:17:12, metric 3, tag 0
                  > to 10.0.0.2 via fe-1/2/0.1

```

Meaning Device R1 is receiving from Device R2 all of Device R2's directly connected networks. Device R1 is also receiving from Device R2 all of Device R3's directly connected networks, which Device R2 learned from Device R3 through RIP.

Verifying the RIP-Enabled Interfaces

Purpose Verify that all RIP-enabled Interfaces are available and active.

Action From operational mode, enter the **show rip neighbor** command.

```

user@R1> show rip neighbor

```

Neighbor	Local State	Source Address	Destination Address	Send Mode	Receive Mode	In Met
fe-1/2/0.1	Up	10.0.0.1	224.0.0.9	mcast	both	1

Meaning The output shows that the RIP-enabled interface on Device R1 is operational.

In general for this command, the output shows a list of the RIP neighbors that are configured on the device. Verify the following information:

- Each configured interface is present. Interfaces are listed in alphabetical order.
- Each configured interface is up. The state of the interface is listed in the **Local State** column. A state of **Up** indicates that the link is passing RIP traffic. A state of **Dn** indicates that the link is not passing RIP traffic. In a point-to-point link, this state generally means that either the end point is not configured for RIP or the link is unavailable.

Verifying the Exchange of RIP Messages

Purpose Verify that RIP messages are being sent and received on all RIP-enabled interfaces.

Action From operational mode, enter the **show rip statistics** command.

```

user@R1> show rip statistics
RIPv2 info: port 520; holddown 120s.
      rts learned  rts held down  rqsts dropped  resps dropped
              5              0              0              0

fe-1/2/0.1: 5 routes learned; 2 routes advertised; timeout 180s; update interval
30s

```

Counter	Total	Last 5 min	Last minute
Updates Sent	2669	10	2
Triggered Updates Sent	2	0	0
Responses Sent	0	0	0
Bad Messages	0	0	0
RIPv1 Updates Received	0	0	0
RIPv1 Bad Route Entries	0	0	0
RIPv1 Updates Ignored	0	0	0
RIPv2 Updates Received	2675	11	2

RIPv2 Bad Route Entries	0	0	0
RIPv2 Updates Ignored	0	0	0
Authentication Failures	0	0	0
RIP Requests Received	0	0	0
RIP Requests Ignored	0	0	0
none	0	0	0

Meaning The output shows the number of RIP routes learned. It also shows the number of RIP updates sent and received on the RIP-enabled interfaces. Verify the following information:

- The number of RIP routes learned matches the number of expected routes learned. Subnets learned by direct connectivity through an outgoing interface are not listed as RIP routes.
- RIP updates are being sent on each RIP-enabled interface. If no updates are being sent, the routing policy might not be configured to export routes.
- RIP updates are being received on each RIP-enabled interface. If no updates are being received, the routing policy might not be configured to export routes on the host connected to that subnet. The lack of updates might also indicate an authentication error.

Verifying Reachability of All Hosts in the RIP Network

Purpose Use the **traceroute** command on each loopback address in the network to verify that all hosts in the RIP network are reachable from each Juniper Networks device.

Action From operational mode, enter the **traceroute** command.

```
user@R1> traceroute 192.168.3.3
traceroute to 192.168.3.3 (192.168.3.3), 30 hops max, 40 byte packets
 1  10.0.0.2 (10.0.0.2)  1.094 ms  1.028 ms  0.957 ms
 2  192.168.3.3 (192.168.3.3)  1.344 ms  2.245 ms  2.125 ms
```

Meaning Each numbered row in the output indicates a routing hop in the path to the host. The three-time increments indicate the round-trip time (RTT) between the device and the hop for each traceroute packet.

To ensure that the RIP network is healthy, verify the following information:

- The final hop in the list is the host you want to reach.
- The number of expected hops to the host matches the number of hops in the traceroute output. The appearance of more hops than expected in the output indicates that a network segment is probably unreachable. It might also indicate that the incoming or outgoing metric on one or more hosts has been set unexpectedly.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- [Understanding Basic RIP Routing on page 7](#)
- *RIP Configuration Overview*

- Related Documentation**
- *Example: Configuring Point-to-Multipoint RIP Networks*

CHAPTER 2

Example: Configuring Authentication for RIP Routes

- [Understanding RIP Authentication on page 15](#)
- [Example: Configuring Route Authentication for RIP on page 15](#)
- [Enabling Authentication with Plain-Text Passwords \(CLI Procedure\) on page 20](#)
- [Enabling Authentication with MD5 Authentication \(CLI Procedure\) on page 21](#)

Understanding RIP Authentication

RIPv2 provides authentication support so that RIP links can require authentication keys (passwords) before they become active. Authentication provides an additional layer of security on the network beyond the other security features. By default, this authentication is disabled.

Authentication keys can be specified in either plain-text or MD5 form. Authentication requires all routers within the RIP network or subnetwork to have the same authentication type and key (password) configured.

This type of authentication is not supported on RIPv1 networks.

Related Documentation

- [*Junos OS Feature Support Reference for SRX Series and J Series Devices*](#)
- [RIP Overview on page 1](#)
- [Enabling Authentication with Plain-Text Passwords \(CLI Procedure\) on page 20](#)
- [Enabling Authentication with MD5 Authentication \(CLI Procedure\) on page 21](#)

Example: Configuring Route Authentication for RIP

This example shows how to configure authentication for a RIP network.

- [Requirements on page 16](#)
- [Overview on page 16](#)
- [Configuration on page 16](#)
- [Verification on page 19](#)

Requirements

No special configuration beyond device initialization is required before configuring this example.

Overview

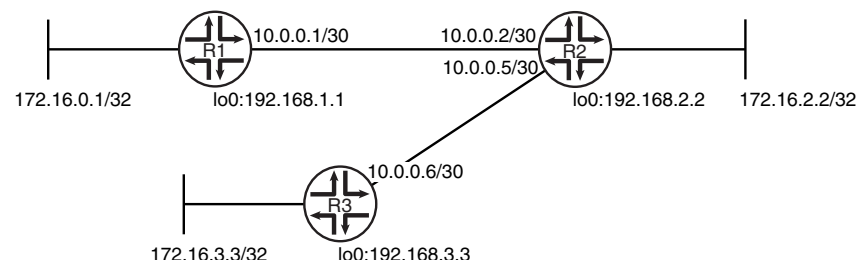
You can configure the router to authenticate RIP route queries. By default, authentication is disabled. You can use one of the following authentication methods:

- Simple authentication—Uses a text password that is included in the transmitted packet. The receiving router uses an authentication key (password) to verify the packet.
- MD5 authentication—Creates an encoded checksum that is included in the transmitted packet. The receiving router uses an authentication key (password) to verify the packet's MD5 checksum.

This example shows MD5 authentication.

Figure 6 on page 16 shows the topology used in this example.

Figure 6: RIP Authentication Network Topology



"CLI Quick Configuration" on page 16 shows the configuration for all of the devices in Figure 6 on page 16. The section "Step-by-Step Procedure" on page 17 describes the steps on Device R1.

Configuration

CLI Quick Configuration To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device R1

```

set interfaces fe-1/2/0 unit 1 family inet address 10.0.0.1/30
set interfaces lo0 unit 1 family inet address 172.16.0.1/32
set interfaces lo0 unit 1 family inet address 192.168.1.1/32
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.1
set protocols rip authentication-type md5
set protocols rip authentication-key "$9$ONLRBhreK87dsM8i.5FAAtM8XxNb"
set protocols rip traceoptions file rip-authentication-messages
set protocols rip traceoptions flag auth
set protocols rip traceoptions flag packets

```



```

set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept

```

Device R2

```

set interfaces fe-1/2/0 unit 2 family inet address 10.0.0.2/30
set interfaces fe-1/2/1 unit 5 family inet address 10.0.0.5/30
set interfaces lo0 unit 2 family inet address 192.168.2.2/32
set interfaces lo0 unit 2 family inet address 172.16.2.2/32
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.2
set protocols rip group rip-group neighbor fe-1/2/1.5
set protocols rip authentication-type md5
set protocols rip authentication-key "$9$Lf1Xds2gJDHmoJCu1hKvoJGUjq"
set protocols rip traceoptions file rip-authentication-messages
set protocols rip traceoptions flag auth
set protocols rip traceoptions flag packets
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept

```

Device R3

```

set interfaces fe-1/2/0 unit 6 family inet address 10.0.0.6/30
set interfaces lo0 unit 3 family inet address 192.168.3.3/32
set interfaces lo0 unit 3 family inet address 172.16.3.3/32
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.6
set protocols rip authentication-type md5
set protocols rip authentication-key "$9$G.UkP5T39tOz3K87V4oz36/Cu"
set protocols rip traceoptions file rip-authentication-messages
set protocols rip traceoptions flag auth
set protocols rip traceoptions flag packets
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept

```

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure RIP authentication:

1. Configure the network interfaces.

This example shows multiple loopback interface addresses to simulate attached networks.

```

[edit interfaces]
user@R1# set fe-1/2/0 unit 1 family inet address 10.0.0.1/30

user@R1# set lo0 unit 1 family inet address 172.16.0.1/32

```

```
user@R1# set lo0 unit 1 family inet address 192.168.1.1/32
```

2. Create the RIP group and add the interface.

To configure RIP in Junos OS, you must configure a group that contains the interfaces on which RIP is enabled. You do not need to enable RIP on the loopback interface.

```
[edit protocols rip group rip-group]
user@R1# set neighbor fe-1/2/0.1
```

3. Create the routing policy to advertise both direct and RIP-learned routes.

```
[edit policy-options policy-statement advertise-routes-through-rip term 1]
user@R1# set from protocol direct
user@R1# set from protocol rip
user@R1# set then accept
```

4. Apply the routing policy.

In Junos OS, you can only apply RIP export policies at the group level.

```
[edit protocols rip group rip-group]
user@R1# set export advertise-routes-through-rip
```

5. Require MD5 authentication for RIP route queries received on an interface.

The passwords must match on neighboring RIP routers. If the password does not match, the packet is rejected. The password can be from 1 through 16 contiguous characters long and can include any ASCII strings.

Do not enter the password as shown here. The password shown here is the encrypted password that is displayed in the configuration after the actual password is already configured.

```
[edit protocols rip]
user@R1# set authentication-type md5
user@R1# set authentication-key "$9$ONLRBhreK87dsM8i.5FAtm8XxNb"
```

6. Configure tracing operations to track authentication.

```
[edit protocols rip traceoptions]
user@R1# set file rip-authentication-messages
user@R1# set flag auth
user@R1# set flag packets
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, and **show policy-options** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@R1# show interfaces
fe-1/2/0 {
  unit 1 {
    family inet {
      address 10.0.0.1/30;
    }
  }
}
lo0 {
```

```

unit 1 {
    family inet {
        address 172.16.0.1/32;
        address 192.168.1.1/32;
    }
}

user@R1# show protocols
rip {
    traceoptions {
        file rip-authentication-messages;
        flag auth;
        flag packets;
    }
    authentication-type md5;
    authentication-key "$9$ONLRBhreK87dsM8i.5FAtM8XxNb"; ## SECRET-DATA
    group rip-group {
        export advertise-routes-through-rip;
        neighbor fe-1/2/0.1;
    }
}

user@R1# show policy-options
policy-statement advertise-routes-through-rip {
    term 1 {
        from protocol [ direct rip ];
        then accept;
    }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

- [Checking for Authentication Failures on page 19](#)
- [Verifying That MD5 Authentication Is Enabled in RIP Update Packets on page 20](#)

Checking for Authentication Failures

Purpose Verify that there are no authentication failures.

Action From operational mode, enter the **show rip statistics** command.

```

user@R1> show rip statistics
RIPv2 info: port 520; holddown 120s.
      rts learned  rts held down  rqsts dropped  resps dropped
              5              0              0              0

fe-1/2/0.1: 5 routes learned; 2 routes advertised; timeout 180s; update interval
30s
Counter              Total      Last 5 min  Last minute
-----
Updates Sent          2669         10         2
Triggered Updates Sent      2         0         0

```

Responses Sent	0	0	0
Bad Messages	0	0	0
RIPv1 Updates Received	0	0	0
RIPv1 Bad Route Entries	0	0	0
RIPv1 Updates Ignored	0	0	0
RIPv2 Updates Received	2675	11	2
RIPv2 Bad Route Entries	0	0	0
RIPv2 Updates Ignored	0	0	0
Authentication Failures	0	0	0
RIP Requests Received	0	0	0
RIP Requests Ignored	0	0	0
none	0	0	0

Meaning The output shows that there are no authentication failures.

Verifying That MD5 Authentication Is Enabled in RIP Update Packets

Purpose Use tracing operations to verify that MD5 authentication is enabled in RIP updates.

Action From operational mode, enter the **show log** command.

```
user@R1> show log rip-authentication-messages | match md5
Feb 15 15:45:13.969462      sending msg 0xb9a8c04, 3 rtes (needs MD5)
Feb 15 15:45:43.229867      sending msg 0xb9a8c04, 3 rtes (needs MD5)
Feb 15 15:46:13.174410      sending msg 0xb9a8c04, 3 rtes (needs MD5)
Feb 15 15:46:42.716566      sending msg 0xb9a8c04, 3 rtes (needs MD5)
Feb 15 15:47:11.425076      sending msg 0xb9a8c04, 3 rtes (needs MD5)
...
```

Meaning The **(needs MD5)** output shows that all route updates require MD5 authentication.

Related Documentation

- [Understanding Basic RIP Routing on page 7](#)

Enabling Authentication with Plain-Text Passwords (CLI Procedure)

To configure authentication that requires a plain-text password to be included in the transmitted packet, enable simple authentication by performing these steps on all RIP devices in the network:

1. Navigate to the top of the configuration hierarchy.
2. Perform the configuration tasks described in [Table 3 on page 20](#).
3. If you are finished configuring the router, commit the configuration.

Table 3: Configuring Simple RIP Authentication

Task	CLI Configuration Editor
Navigate to Rip level in the configuration hierarchy.	From the [edit] hierarchy level, enter edit protocols rip

Table 3: Configuring Simple RIP Authentication (*continued*)

Task	CLI Configuration Editor
Set the authentication type to simple .	Set the authentication type to simple : set authentication-type simple
Set the authentication key to a simple-text password. The password can be from 1 through 16 contiguous characters long and can include any ASCII strings.	Set the authentication key to a simple-text password: set authentication-key password

- Related Documentation**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - [Understanding RIP Authentication on page 15](#)
 - *RIP Configuration Overview*
 - [Enabling Authentication with MD5 Authentication \(CLI Procedure\) on page 21](#)

Enabling Authentication with MD5 Authentication (CLI Procedure)

To configure authentication that requires an MD5 password to be included in the transmitted packet, enable MD5 authentication by performing these steps on all RIP devices in the network:

1. Navigate to the top of the configuration hierarchy.
2. Perform the configuration tasks described in [Table 4 on page 21](#).
3. If you are finished configuring the router, commit the configuration.

Table 4: Configuring MD5 RIP Authentication

Task	CLI Configuration Editor
Navigate to Rip level in the configuration hierarchy.	From the [edit] hierarchy level, enter edit protocols rip
Set the authentication type to MD5 .	Set the authentication type to md5 : set authentication-type md5
Set the MD5 authentication key (password). The key can be from 1 through 16 contiguous characters long and can include any ASCII strings.	Set the MD5 authentication key: set authentication-key password

- Related Documentation**
- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
 - [Understanding RIP Authentication on page 15](#)
 - *RIP Configuration Overview*

- [Enabling Authentication with Plain-Text Passwords \(CLI Procedure\) on page 20](#)

**Related
Documentation**

- [Example: Configuring RIP on page 7](#)

CHAPTER 3

Example: Configuring BFD for RIP

- [Understanding BFD for RIP on page 23](#)
- [Example: Configuring BFD for RIP on page 24](#)

Understanding BFD for RIP

The Bidirectional Forwarding Detection (BFD) Protocol is a simple hello mechanism that detects failures in a network. Hello packets are sent at a specified, regular interval. A neighbor failure is detected when the routing device stops receiving a reply after a specified interval. BFD works with a wide variety of network environments and topologies. BFD failure detection times are shorter than RIP detection times, providing faster reaction times to various kinds of failures in the network. Instead of waiting for the routing protocol neighbor timeout, BFD provides rapid detection of link failures. BFD timers are adaptive and can be adjusted to be more or less aggressive. For example, a timer can adapt to a higher value if the adjacency fails, or a neighbor can negotiate a higher value for a timer than the one configured.

BFD enables quick failover between a primary and a secondary routed path. The protocol tests the operational status of the interface multiple times per second. BFD provides for configuration timers and thresholds for failure detection. For example, if the minimum interval is set for 50 milliseconds and the threshold uses the default value of three missed messages, a failure is detected on an interface within 200 milliseconds of the failure.

Intervening devices (for example, an Ethernet LAN switch) hide link-layer failures from routing protocol peers, such as when two routers are connected by way of a LAN switch, where the local interface status remains up even when a physical fault happens on the remote link. Link-layer failure detection times vary, depending on the physical media and the Layer 2 encapsulation. BFD can provide fast failure detection times for all media types, encapsulations, topologies, and routing protocols.

To enable BFD for RIP, both sides of the connection must receive an update message from the peer. By default, RIP does not export any routes. Therefore, you must enable update messages to be sent by configuring an export policy for routes before a BFD session is triggered.

Related Documentation

- [Example: Configuring BFD for RIP on page 24](#)

Example: Configuring BFD for RIP

This example shows how to configure Bidirectional Forwarding Detection (BFD) for a RIP network.

- [Requirements on page 24](#)
- [Overview on page 24](#)
- [Configuration on page 26](#)
- [Verification on page 28](#)

Requirements

No special configuration beyond device initialization is required before configuring this example.

Overview

To enable failure detection, include the **bfd-liveness-detection** statement:

```
bfd-liveness-detection {  
  detection-time {  
    threshold milliseconds;  
  }  
  minimum-interval milliseconds;  
  minimum-receive-interval milliseconds;  
  multiplier number;  
  no-adaptation;  
  transmit-interval {  
    threshold milliseconds;  
    minimum-interval milliseconds;  
  }  
  version (1 | automatic);  
}
```

Optionally, you can specify the threshold for the adaptation of the detection time by including the **threshold** statement. When the BFD session detection time adapts to a value equal to or greater than the threshold, a single trap and a system log message are sent.

To specify the minimum transmit and receive interval for failure detection, include the **minimum-interval** statement. This value represents the minimum interval at which the local routing device transmits hello packets as well as the minimum interval at which the routing device expects to receive a reply from a neighbor with which it has established a BFD session. You can configure a value in the range from 1 through 255,000 milliseconds. This examples sets a minimum interval of 600 milliseconds.



NOTE: BFD is an intensive protocol that consumes system resources. Specifying a minimum interval for BFD of less than 100 ms for Routing Engine-based sessions and 10 ms for distributed BFD sessions can cause undesired BFD flapping.

Depending on your network environment, these additional recommendations might apply:

- For large-scale network deployments with a large number of BFD sessions, specify a minimum interval of 300 ms for Routing Engine-based sessions and 100 ms for distributed BFD sessions.
- For very large-scale network deployments with a large number of BFD sessions, contact Juniper Networks customer support for more information.
- For BFD sessions to remain up during a Routing Engine switchover event when nonstop active routing (NSR) is configured, specify a minimum interval of 2500 ms for Routing Engine-based sessions. For distributed BFD sessions with nonstop active routing configured, the minimum interval recommendations are unchanged and depend only on your network deployment.

You can optionally specify the minimum transmit and receive intervals separately.

To specify only the minimum receive interval for failure detection, include the **minimum-receive-interval** statement. This value represents the minimum interval at which the local routing device expects to receive a reply from a neighbor with which it has established a BFD session. You can configure a value in the range from 1 through 255,000 milliseconds.

To specify only the minimum transmit interval for failure detection, include the **transmit-interval minimum-interval** statement. This value represents the minimum interval at which the local routing device transmits hello packets to the neighbor with which it has established a BFD session. You can configure a value in the range from 1 through 255,000 milliseconds.

To specify the number of hello packets not received by a neighbor that causes the originating interface to be declared down, include the **multiplier** statement. The default is 3, and you can configure a value in the range from 1 through 255.

To specify the threshold for detecting the adaptation of the transmit interval, include the **transmit-interval threshold** statement. The threshold value must be greater than the transmit interval.

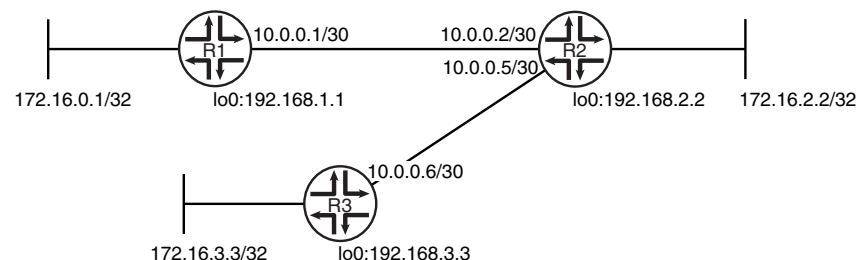
To specify the BFD version used for detection, include the **version** statement. The default is to have the version detected automatically.

You can trace BFD operations by including the **traceoptions** statement at the **[edit protocols bfd]** hierarchy level.

In Junos OS Release 9.0 and later, you can configure BFD sessions not to adapt to changing network conditions. To disable BFD adaptation, include the **no-adaptation** statement. We recommend that you not disable BFD adaptation unless it is preferable not to have BFD adaptation enabled in your network.

Figure 7 on page 26 shows the topology used in this example.

Figure 7: RIP BFD Network Topology



"CLI Quick Configuration" on page 26 shows the configuration for all of the devices in Figure 7 on page 26. The section "Step-by-Step Procedure" on page 27 describes the steps on Device R1.

Configuration

CLI Quick Configuration	To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the [edit] hierarchy level.
Device R1	<pre> set interfaces fe-1/2/0 unit 1 family inet address 10.0.0.1/30 set protocols bfd traceoptions file bfd-trace set protocols bfd traceoptions flag all set protocols rip group rip-group export advertise-routes-through-rip set protocols rip group rip-group neighbor fe-1/2/0.1 set protocols rip group rip-group bfd-liveness-detection minimum-interval 600 set policy-options policy-statement advertise-routes-through-rip term 1 from protocol direct set policy-options policy-statement advertise-routes-through-rip term 1 from protocol rip set policy-options policy-statement advertise-routes-through-rip term 1 then accept </pre>
Device R2	<pre> set interfaces fe-1/2/0 unit 2 family inet address 10.0.0.2/30 set interfaces fe-1/2/1 unit 5 family inet address 10.0.0.5/30 set protocols rip group rip-group export advertise-routes-through-rip set protocols rip group rip-group neighbor fe-1/2/0.2 set protocols rip group rip-group neighbor fe-1/2/1.5 set protocols rip group rip-group bfd-liveness-detection minimum-interval 600 set policy-options policy-statement advertise-routes-through-rip term 1 from protocol direct set policy-options policy-statement advertise-routes-through-rip term 1 from protocol rip set policy-options policy-statement advertise-routes-through-rip term 1 then accept </pre>
Device R3	<pre> set interfaces fe-1/2/0 unit 6 family inet address 10.0.0.6/30 set protocols rip group rip-group export advertise-routes-through-rip </pre>

```

set protocols rip group rip-group neighbor fe-1/2/0.6
set protocols rip group rip-group bfd-liveness-detection minimum-interval 600
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept

```

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure a BFD for a RIP network:

1. Configure the network interfaces.

```

[edit interfaces]
user@R1# set fe-1/2/0 unit 1 family inet address 10.0.0.1/30

```

2. Create the RIP group and add the interface.

To configure RIP in Junos OS, you must configure a group that contains the interfaces on which RIP is enabled. You do not need to enable RIP on the loopback interface.

```

[edit protocols rip group rip-group]
user@R1# set neighbor fe-1/2/0.1

```

3. Create the routing policy to advertise both direct and RIP-learned routes.

```

[edit policy-options policy-statement advertise-routes-through-rip term 1]
user@R1# set from protocol direct
user@R1# set from protocol rip
user@R1# set then accept

```

4. Apply the routing policy.

In Junos OS, you can only apply RIP export policies at the group level.

```

[edit protocols rip group rip-group]
user@R1# set export advertise-routes-through-rip

```

5. Enable BFD.

```

[edit protocols rip group rip-group]
user@R1# set bfd-liveness-detection minimum-interval 600

```

6. Configure tracing operations to track BFD messages.

```

[edit protocols bfd traceoptions]
user@R1# set file bfd-trace
user@R1# set flag all

```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, and **show policy-options** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

user@R1# show interfaces
fe-1/2/0 {

```

```

unit 1 {
    family inet {
        address 10.0.0.1/30;
    }
}

user@R1# show protocols
bfd {
    traceoptions {
        file bfd-trace;
        flag all;
    }
}
rip {
    group rip-group {
        export advertise-routes-through-rip;
        bfd-liveness-detection {
            minimum-interval 600;
        }
        neighbor fe-1/2/0.1;
    }
}

user@R1# show policy-options
policy-statement advertise-routes-through-rip {
    term 1 {
        from protocol [ direct rip ];
        then accept;
    }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

- [Verifying That the BFD Sessions Are Up on page 28](#)
- [Checking the BFD Trace File on page 29](#)

Verifying That the BFD Sessions Are Up

Purpose Make sure that the BFD sessions are operating.

Action From operational mode, enter the **show bfd session** command.

```

user@R1> show bfd session

```

Address	State	Interface	Detect Time	Transmit Interval	Multiplier
10.0.0.2	Up	fe-1/2/0.1	1.800	0.600	3

```

1 sessions, 1 clients
Cumulative transmit rate 1.7 pps, cumulative receive rate 1.7 pps

```

Meaning The output shows that there are no authentication failures.

Checking the BFD Trace File

Purpose Use tracing operations to verify that BFD packets are being exchanged.

Action From operational mode, enter the **show log** command.

```
user@R1> show log bfd-trace
Feb 16 10:26:32 PPM Trace: BFD periodic xmit to 10.0.0.2 (IFL 124, rtbl 53,
single-hop port)
Feb 16 10:26:32 Received Downstream TraceMsg (24) len 86:
Feb 16 10:26:32   IfIndex (3) len 4: 0
Feb 16 10:26:32   Protocol (1) len 1: BFD
Feb 16 10:26:32   Data (9) len 61: (hex) 42 46 44 20 70 61 63 6b 65 74 20 66 72
6f 6d 20 31 30 2e
Feb 16 10:26:32 PPM Trace: BFD packet from 10.0.0.1 (IFL 73, rtbl 56, ttl 255)
absorbed
Feb 16 10:26:32 Received Downstream TraceMsg (24) len 60:
Feb 16 10:26:32   IfIndex (3) len 4: 0
Feb 16 10:26:32   Protocol (1) len 1: BFD
Feb 16 10:26:32   Data (9) len 35: (hex) 42 46 44 20 70 65 72 69 6f 64 69 63 20
78 6d 69 74 20 6f
...
```

Meaning The output shows the normal functioning of BFD.

Related Documentation

- [Understanding BFD for RIP on page 23](#)

Related Documentation

- [Example: Configuring RIP on page 7](#)
- [Example: Configuring Authentication for RIP Routes on page 15](#)
- [Example: Configuring Point-to-Multipoint RIP Networks](#)

CHAPTER 4

Example: Configuring BFD Authentication for RIP

- [Understanding BFD Authentication for RIP on page 31](#)
- [Example: Configuring BFD Authentication for RIP on page 33](#)

Understanding BFD Authentication for RIP

BFD enables rapid detection of communication failures between adjacent systems. By default, authentication for BFD sessions is disabled. However, when running BFD over Network Layer protocols, the risk of service attacks can be significant. We strongly recommend using authentication if you are running BFD over multiple hops or through insecure tunnels. Beginning with Junos OS Release 9.6, Junos OS supports authentication for BFD sessions running over RIP. BFD authentication is only supported in the domestic image and is not available in the export image.

You authenticate BFD sessions by specifying an authentication algorithm and keychain, and then associating that configuration information with a security authentication keychain using the keychain name.

The following sections describe the supported authentication algorithms, security keychains, and the level of authentication that can be configured:

- [BFD Authentication Algorithms on page 31](#)
- [Security Authentication Keychains on page 32](#)
- [Strict Versus Loose Authentication on page 32](#)

BFD Authentication Algorithms

Junos OS supports the following algorithms for BFD authentication:

- **simple-password**—Plain-text password. One to 16 bytes of plain text are used to authenticate the BFD session. One or more passwords can be configured. This method is the least secure and should be used only when BFD sessions are not subject to packet interception.
- **keyed-md5**—Keyed Message Digest 5 hash algorithm for sessions with transmit and receive intervals greater than 100 ms. To authenticate the BFD session, keyed MD5 uses one or more secret keys (generated by the algorithm) and a sequence number

that is updated periodically. With this method, packets are accepted at the receiving end of the session if one of the keys matches and the sequence number is greater than or equal to the last sequence number received. Although more secure than a simple password, this method is vulnerable to replay attacks. Increasing the rate at which the sequence number is updated can reduce this risk.

- **meticulous-keyed-md5**—Meticulous keyed Message Digest 5 hash algorithm. This method works in the same manner as keyed MD5, but the sequence number is updated with every packet. Although more secure than keyed MD5 and simple passwords, this method might take additional time to authenticate the session.
- **keyed-sha-1**—Keyed Secure Hash Algorithm I for sessions with transmit and receive intervals greater than 100 ms. To authenticate the BFD session, keyed SHA uses one or more secret keys (generated by the algorithm) and a sequence number that is updated periodically. The key is not carried within the packets. With this method, packets are accepted at the receiving end of the session if one of the keys matches and the sequence number is greater than the last sequence number received.
- **meticulous-keyed-sha-1**—Meticulous keyed Secure Hash Algorithm I. This method works in the same manner as keyed SHA, but the sequence number is updated with every packet. Although more secure than keyed SHA and simple passwords, this method might take additional time to authenticate the session.



NOTE: Nonstop active routing is not supported with meticulous-keyed-md5 and meticulous-keyed-sha-1 authentication algorithms. BFD sessions using these algorithms might go down after a switchover.

Security Authentication Keychains

The security authentication keychain defines the authentication attributes used for authentication key updates. When the security authentication keychain is configured and associated with a protocol through the keychain name, authentication key updates can occur without interrupting routing and signaling protocols.

The authentication keychain contains one or more keychains. Each keychain contains one or more keys. Each key holds the secret data and the time at which the key becomes valid. The algorithm and keychain must be configured on both ends of the BFD session, and they must match. Any mismatch in configuration prevents the BFD session from being created.

BFD allows multiple clients per session, and each client can have its own keychain and algorithm defined. To avoid confusion, we recommend specifying only one security authentication keychain.

Strict Versus Loose Authentication

By default, strict authentication is enabled and authentication is checked at both ends of each BFD session. Optionally, to smooth migration from nonauthenticated sessions to authenticated sessions, you can configure *loose checking*. When loose checking is

configured, packets are accepted without authentication being checked at each end of the session. This feature is intended for transitional periods only.

Related Documentation

- [Example: Configuring BFD Authentication for RIP on page 33](#)
- [bfd-liveness-detection on page 91](#)
- **authentication-key-chains** statement in the *Junos OS Administration Library for Routing Devices*
- **show bfd session** command in the [CLI Explorer](#)
- [Example: Configuring BFD for RIP on page 24](#)

Example: Configuring BFD Authentication for RIP

This example shows how to configure Bidirectional Forwarding Detection (BFD) authentication for a RIP network.

- [Requirements on page 33](#)
- [Overview on page 33](#)
- [Configuration on page 34](#)
- [Verification on page 37](#)

Requirements

No special configuration beyond device initialization is required before configuring this example.

The devices must be running Junos OS Release 9.6 or later.

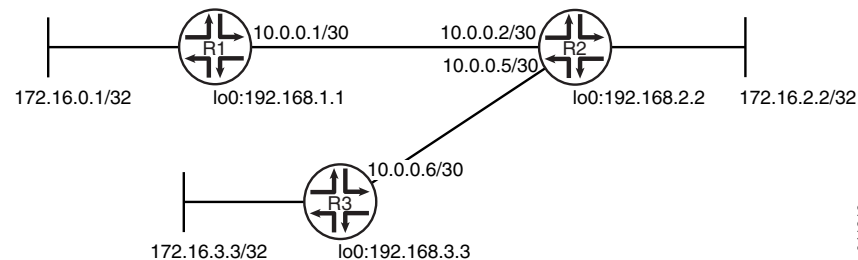
Overview

Only three steps are needed to configure authentication on a BFD session:

1. Specify the BFD authentication algorithm for the RIP protocol.
2. Associate the authentication keychain with the RIP protocol.
3. Configure the related security authentication keychain.

[Figure 8 on page 33](#) shows the topology used in this example.

Figure 8: RIP BFD Authentication Network Topology



"CLI Quick Configuration" on page 34 shows the configuration for all of the devices in Figure 8 on page 33. The section "Step-by-Step Procedure" on page 35 describes the steps on Device R1.

Configuration

CLI Quick Configuration	To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the [edit] hierarchy level.
Device R1	<pre> set interfaces fe-1/2/0 unit 1 family inet address 10.0.0.1/30 set protocols bfd traceoptions file bfd-trace set protocols bfd traceoptions flag all set protocols rip group rip-group export advertise-routes-through-rip set protocols rip group rip-group neighbor fe-1/2/0.1 set protocols rip group rip-group bfd-liveness-detection minimum-interval 600 set protocols rip group rip-group bfd-liveness-detection authentication key-chain bfd-rip set protocols rip group rip-group bfd-liveness-detection authentication algorithm keyed-md5 set protocols rip group rip-group bfd-liveness-detection authentication loose-check set policy-options policy-statement advertise-routes-through-rip term 1 from protocol direct set policy-options policy-statement advertise-routes-through-rip term 1 from protocol rip set policy-options policy-statement advertise-routes-through-rip term 1 then accept set security authentication-key-chains key-chain bfd-rip key 53 secret "\$9\$d1V2aZGi.fzDiORSeXxDikqmT" set security authentication-key-chains key-chain bfd-rip key 53 start-time "2012-2-16.12:00:00 -0800" </pre>
Device R2	<pre> set interfaces fe-1/2/0 unit 2 family inet address 10.0.0.2/30 set interfaces fe-1/2/1 unit 5 family inet address 10.0.0.5/30 set protocols rip group rip-group export advertise-routes-through-rip set protocols rip group rip-group neighbor fe-1/2/0.2 set protocols rip group rip-group neighbor fe-1/2/1.5 set protocols rip group rip-group bfd-liveness-detection minimum-interval 600 set protocols rip group rip-group bfd-liveness-detection authentication key-chain bfd-rip set protocols rip group rip-group bfd-liveness-detection authentication algorithm keyed-md5 set protocols rip group rip-group bfd-liveness-detection authentication loose-check set policy-options policy-statement advertise-routes-through-rip term 1 from protocol direct set policy-options policy-statement advertise-routes-through-rip term 1 from protocol rip set policy-options policy-statement advertise-routes-through-rip term 1 then accept set security authentication-key-chains key-chain bfd-rip key 53 secret "\$9\$d1V2aZGi.fzDiORSeXxDikqmT" set security authentication-key-chains key-chain bfd-rip key 53 start-time "2012-2-16.12:00:00 -0800" </pre>
Device R3	<pre> set interfaces fe-1/2/0 unit 6 family inet address 10.0.0.6/30 set protocols rip group rip-group export advertise-routes-through-rip set protocols rip group rip-group neighbor fe-1/2/0.6 set protocols rip group rip-group bfd-liveness-detection minimum-interval 600 </pre>

```

set protocols rip group rip-group bfd-liveness-detection authentication key-chain bfd-rip
set protocols rip group rip-group bfd-liveness-detection authentication algorithm
  keyed-md5
set protocols rip group rip-group bfd-liveness-detection authentication loose-check
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept
set security authentication-key-chains key-chain bfd-rip key 53 secret
  "$9$d1V2aZGi.fzDiORSeXxDikqmT"
set security authentication-key-chains key-chain bfd-rip key 53 start-time
  "2012-2-16.12:00:00 -0800"

```

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure a BFD authentication:

1. Configure the network interfaces.

```

[edit interfaces]
user@R1# set fe-1/2/0 unit 1 family inet address 10.0.0.1/30

```

2. Create the RIP group and add the interface.

To configure RIP in Junos OS, you must configure a group that contains the interfaces on which RIP is enabled. You do not need to enable RIP on the loopback interface.

```

[edit protocols rip group rip-group]
user@R1# set neighbor fe-1/2/0.1

```

3. Create the routing policy to advertise both direct and RIP-learned routes.

```

[edit policy-options policy-statement advertise-routes-through-rip term 1]
user@R1# set from protocol direct
user@R1# set from protocol rip
user@R1# set then accept

```

4. Apply the routing policy.

In Junos OS, you can only apply RIP export policies at the group level.

```

[edit protocols rip group rip-group]
user@R1# set export advertise-routes-through-rip

```

5. Enable BFD.

```

[edit protocols rip group rip-group]
user@R1# set bfd-liveness-detection minimum-interval 600

```

6. Specify the algorithm (`keyed-md5`, `keyed-sha-1`, `meticulous-keyed-md5`, `meticulous-keyed-sha-1`, or `simple-password`) to use.



NOTE: Nonstop active routing is not supported with meticulous-keyed-md5 and meticulous-keyed-sha-1 authentication algorithms. BFD sessions using these algorithms might go down after a switchover.

```
[edit protocols rip group rip-group]
user@R1# set bfd-liveness-detection authentication algorithm keyed-md5
```

7. Specify the keychain to be used to associate BFD sessions on RIP with the unique security authentication keychain attributes.

The keychain you specify must match a keychain name configured at the **[edit security authentication key-chains]** hierarchy level.

The algorithm and keychain must be configured on both ends of the BFD session, and they must match. Any mismatch in configuration prevents the BFD session from being created.

```
[edit protocols rip group rip-group]
user@R1# set bfd-liveness-detection authentication key-chain bfd-rip
```

8. (Optional) Specify loose authentication checking if you are transitioning from nonauthenticated sessions to authenticated sessions.

```
[edit protocols rip group rip-group]
user@R1# set bfd-liveness-detection authentication loose-check
```

9. Specify the unique security authentication information for BFD sessions:

- The matching keychain name as specified in Step 7.
- At least one key, a unique integer between 0 and 63. Creating multiple keys allows multiple clients to use the BFD session.
- The secret data used to allow access to the session.
- The time at which the authentication key becomes active, in the format *yyyy-mm-dd.hh:mm:ss*.

```
[edit security authentication-key-chains key-chain bfd-rip]
user@R1# set key 53 secret "$9$d1V2aZGi.fzDiORSeXxDikqmT"
user@R1# set key 53 start-time "2012-2-16.12:00:00 -0800"
```

10. Configure tracing operations to track BFD authentication.

```
[edit protocols bfd traceoptions]
user@R1# set file bfd-trace
user@R1# set flag all
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, **show policy-options**, and **show security** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@R1# show interfaces
```

```

fe-1/2/0 {
  unit 1 {
    family inet {
      address 10.0.0.1/30;
    }
  }
}

user@R1# show protocols
bfd {
  traceoptions {
    file bfd-trace;
    flag all;
  }
}
rip {
  group rip-group {
    export advertise-routes-through-rip;
    bfd-liveness-detection {
      minimum-interval 600;
    }
    neighbor fe-1/2/0.1;
  }
}

user@R1# show policy-options
policy-statement advertise-routes-through-rip {
  term 1 {
    from protocol [ direct rip ];
    then accept;
  }
}

user@R1# show security
authentication-key-chains {
  key-chain bfd-rip {
    key 53 {
      secret "$9$d1V2aZGi.fzDiORSeXxDikqmT"; ## SECRET-DATA
      start-time "2012-2-16.12:00:00 -0800";
    }
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

- [Verifying That the BFD Sessions Are Authenticated on page 37](#)
- [Viewing Extensive Information About the BFD Authentication on page 38](#)
- [Checking the BFD Trace File on page 38](#)

Verifying That the BFD Sessions Are Authenticated

Purpose Make sure that the BFD sessions are authenticated.

Action From operational mode, enter the **show bfd session detail** command.

```
user@R1> show bfd session detail
```

Address	State	Interface	Detect Time	Transmit Interval	Multiplier
10.0.0.2	Up	fe-1/2/0.1	1.800	0.600	3

Client RIP, TX interval 0.600, RX interval 0.600, **Authenticate**
 Session up time 01:39:34
 Local diagnostic None, remote diagnostic None
 Remote state Up, version 1
 Logical system 6, routing table index 53

1 sessions, 1 clients
 Cumulative transmit rate 1.7 pps, cumulative receive rate 1.7 pps

Meaning **Authenticate** is displayed to indicate that BFD authentication is configured.

Viewing Extensive Information About the BFD Authentication

Purpose View the keychain name, the authentication algorithm and mode for each client in the session, and the BFD authentication configuration status.

Action From operational mode, enter the **show bfd session extensive** command.

```
user@R1> show bfd session extensive
```

Address	State	Interface	Detect Time	Transmit Interval	Multiplier
10.0.0.2	Up	fe-1/2/0.1	1.800	0.600	3

Client RIP, TX interval 0.600, RX interval 0.600, **Authenticate**
keychain bfd-rip, algo keyed-md5, mode loose
 Session up time 01:46:29
 Local diagnostic None, remote diagnostic None
 Remote state Up, version 1
 Logical system 6, routing table index 53
 Min async interval 0.600, min slow interval 1.000
 Adaptive async TX interval 0.600, RX interval 0.600
 Local min TX interval 0.600, minimum RX interval 0.600, multiplier 3
 Remote min TX interval 0.600, min RX interval 0.600, multiplier 3
 Local discriminator 225, remote discriminator 226
 Echo mode disabled/inactive
Authentication enabled/active, keychain bfd-rip, algo keyed-md5, mode loose
 Session ID: 0x300501

1 sessions, 1 clients
 Cumulative transmit rate 1.7 pps, cumulative receive rate 1.7 pps

Meaning The output shows the keychain name, the authentication algorithm and mode for the client in the session, and the BFD authentication configuration status.

Checking the BFD Trace File

Purpose Use tracing operations to verify that BFD packets are being exchanged.

Action From operational mode, enter the **show log** command.

```
user@R1> show log bfd-trace
```

```
Feb 16 10:26:32 PPM Trace: BFD periodic xmit to 10.0.0.2 (IFL 124, rtbl 53,
single-hop port)
Feb 16 10:26:32 Received Downstream TraceMsg (24) len 86:
Feb 16 10:26:32   IfIndex (3) len 4: 0
Feb 16 10:26:32   Protocol (1) len 1: BFD
Feb 16 10:26:32   Data (9) len 61: (hex) 42 46 44 20 70 61 63 6b 65 74 20 66 72
6f 6d 20 31 30 2e
Feb 16 10:26:32 PPM Trace: BFD packet from 10.0.0.1 (IFL 73, rtbl 56, ttl 255)
absorbed
Feb 16 10:26:32 Received Downstream TraceMsg (24) len 60:
Feb 16 10:26:32   IfIndex (3) len 4: 0
Feb 16 10:26:32   Protocol (1) len 1: BFD
Feb 16 10:26:32   Data (9) len 35: (hex) 42 46 44 20 70 65 72 69 6f 64 69 63 20
78 6d 69 74 20 6f
...
```

Meaning The output shows the normal functioning of BFD.

Related Documentation

- [Understanding BFD Authentication for RIP on page 31](#)

Related Documentation

- [Example: Configuring BFD for RIP on page 23](#)
- [Example: Configuring Authentication for RIP Routes on page 15](#)
- [Example: Configuring RIP on page 7](#)

CHAPTER 5

Example: Applying Policies to RIP Routes Imported from Neighbors

- [Understanding RIP Import Policy on page 41](#)
- [Example: Applying Policies to RIP Routes Imported from Neighbors on page 41](#)

Understanding RIP Import Policy

The default RIP import policy is to accept all received RIP routes that pass a sanity check. To filter routes being imported by the local routing device from its neighbors, include the **import** statement, and list the names of one or more policies to be evaluated. If you specify more than one policy, they are evaluated in order (first to last) and the first matching policy is applied to the route. If no match is found, the local routing device does not import any routes.

Related Documentation

- [Example: Applying Policies to RIP Routes Imported from Neighbors on page 41](#)

Example: Applying Policies to RIP Routes Imported from Neighbors

This example shows how to configure an import policy in a RIP network.

- [Requirements on page 41](#)
- [Overview on page 41](#)
- [Configuration on page 42](#)
- [Verification on page 45](#)

Requirements

No special configuration beyond device initialization is required before configuring this example.

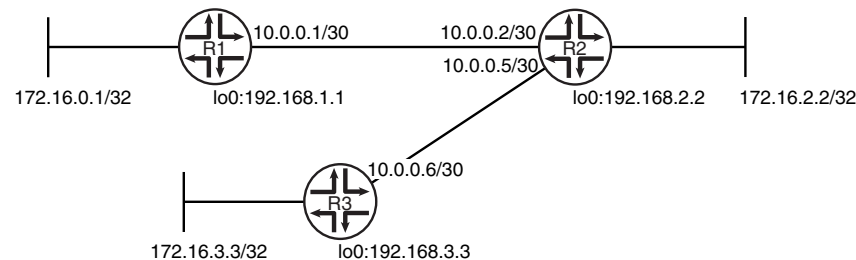
Overview

In this example, Device R1 has an import policy that accepts the 10/8 and 192.168/16 RIP routes and rejects all other RIP routes. This means that the 172.16/16 RIP routes are excluded from Device R1's routing table.

An export policy is also shown because an export policy is required as part of the minimum configuration for RIP.

Figure 9 on page 42 shows the topology used in this example.

Figure 9: RIP Import Policy Network Topology



"CLI Quick Configuration" on page 42 shows the configuration for all of the devices in Figure 9 on page 42. The section "Step-by-Step Procedure" on page 43 describes the steps on Device R1.

Configuration

CLI Quick Configuration	To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the [edit] hierarchy level.
Device R1	<pre> set interfaces fe-1/2/0 unit 1 family inet address 10.0.0.1/30 set interfaces lo0 unit 1 family inet address 172.16.0.1/32 set interfaces lo0 unit 1 family inet address 192.168.1.1/32 set protocols rip import rip-import set protocols rip group rip-group export advertise-routes-through-rip set protocols rip group rip-group neighbor fe-1/2/0.1 set policy-options policy-statement advertise-routes-through-rip term 1 from protocol direct set policy-options policy-statement advertise-routes-through-rip term 1 from protocol rip set policy-options policy-statement advertise-routes-through-rip term 1 then accept set policy-options policy-statement rip-import term 1 from protocol rip set policy-options policy-statement rip-import term 1 from route-filter 10.0.0.0/8 orlonger set policy-options policy-statement rip-import term 1 from route-filter 192.168.0.0/16 orlonger set policy-options policy-statement rip-import term 1 then accept set policy-options policy-statement rip-import term 2 then reject </pre>
Device R2	<pre> set interfaces fe-1/2/0 unit 2 family inet address 10.0.0.2/30 set interfaces fe-1/2/1 unit 5 family inet address 10.0.0.5/30 set interfaces lo0 unit 2 family inet address 192.168.2.2/32 set interfaces lo0 unit 2 family inet address 172.16.2.2/32 set protocols rip group rip-group export advertise-routes-through-rip set protocols rip group rip-group neighbor fe-1/2/0.2 set protocols rip group rip-group neighbor fe-1/2/1.5 set policy-options policy-statement advertise-routes-through-rip term 1 from protocol direct </pre>

```

set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept

```

Device R3

```

set interfaces fe-1/2/0 unit 6 family inet address 10.0.0.6/30
set interfaces lo0 unit 3 family inet address 192.168.3.3/32
set interfaces lo0 unit 3 family inet address 172.16.3.3/32
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.6
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept

```

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure a RIP import policy:

1. Configure the network interfaces.

This example shows multiple loopback interface addresses to simulate attached networks.

```

[edit interfaces]
user@R1# set fe-1/2/0 unit 1 family inet address 10.0.0.1/30

user@R1# set lo0 unit 1 family inet address 172.16.0.1/32
user@R1# set lo0 unit 1 family inet address 192.168.1.1/32

```

2. Create the RIP group and add the interface.

To configure RIP in Junos OS, you must configure a group that contains the interfaces on which RIP is enabled.

You do not need to enable RIP on the loopback interface.

```

[edit protocols rip group rip-group]
user@R1# set neighbor fe-1/2/0.1

```

3. Create the routing policy to advertise both direct and RIP-learned routes.

```

[edit policy-options policy-statement advertise-routes-through-rip term 1]
user@R1# set from protocol direct
user@R1# set from protocol rip
user@R1# set then accept

```

4. Apply the routing policy.

In Junos OS, you can only apply RIP export policies at the group level.

```

[edit protocols rip group rip-group]
user@R1# set export advertise-routes-through-rip

```

5. Configure the import policy.

```

[edit policy-options policy-statement rip-import]

```

```
user@R1# set term 1 from protocol rip
user@R1# set term 1 from route-filter 10.0.0.0/8 orlonger
user@R1# set term 1 from route-filter 192.168.0.0/16 orlonger
user@R1# set term 1 then accept
user@R1# set term 2 then reject
```

6. Apply the import policy.

```
[edit protocols rip]
user@R1# set import rip-import
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, and **show policy-options** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@R1# show interfaces
fe-1/2/0 {
  unit 1 {
    family inet {
      address 10.0.0.1/30;
    }
  }
}
lo0 {
  unit 1 {
    family inet {
      address 172.16.0.1/32;
      address 192.168.1.1/32;
    }
  }
}

user@R1# show protocols
rip {
  import rip-import;
  group rip-group {
    export advertise-routes-through-rip;
    neighbor fe-1/2/0.1;
  }
}

user@R1# show policy-options
policy-statement advertise-routes-through-rip {
  term 1 {
    from protocol [ direct rip ];
    then accept;
  }
}
policy-statement rip-import {
  term 1 {
    from {
      protocol rip;
      route-filter 10.0.0.0/8 orlonger;
      route-filter 192.168.0.0/16 orlonger;
    }
  }
}
```

```

        then accept;
    }
    term 2 {
        then reject;
    }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

- [Looking at the Routes That Device R2 Is Advertising to Device R1 on page 45](#)
- [Looking at the Routes That Device R1 Is Receiving from Device R2 on page 45](#)
- [Checking the Routing Table on page 46](#)
- [Testing the Import Policy on page 46](#)

Looking at the Routes That Device R2 Is Advertising to Device R1

Purpose Verify that Device R2 is sending the expected routes.

Action From operational mode, enter the **show route advertising-protocol rip 10.0.0.2** command.

```
user@R2> show route advertising-protocol rip 10.0.0.2
```

```
inet.0: 11 destinations, 11 routes (11 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```

10.0.0.4/30      *[Direct/0] 2d 01:17:44
                  >   via fe-1/2/0.5
172.16.2.2/32    *[Direct/0] 2d 04:09:52
                  >   via lo0.2
172.16.3.3/32    *[RIP/100] 23:40:02, metric 2, tag 0
                  > to 10.0.0.6 via fe-1/2/0.5
192.168.2.2/32   *[Direct/0] 2d 04:09:52
                  >   via lo0.2
192.168.3.3/32   *[RIP/100] 23:40:02, metric 2, tag 0
                  > to 10.0.0.6 via fe-1/2/0.5

```

Meaning Device R2 is sending 172.16/16 routes to Device R1.

Looking at the Routes That Device R1 Is Receiving from Device R2

Purpose Verify that Device R1 is receiving the expected routes.

Action From operational mode, enter the **show route receive-protocol rip 10.0.0.2** command.

```
user@R1> show route receive-protocol rip 10.0.0.2
```

```
inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```

10.0.0.4/30      *[RIP/100] 01:06:03, metric 2, tag 0
                  > to 10.0.0.2 via fe-1/2/0.1
192.168.2.2/32   *[RIP/100] 01:06:03, metric 2, tag 0

```

```
192.168.3.3/32      > to 10.0.0.2 via fe-1/2/0.1
                   *[RIP/100] 01:06:03, metric 3, tag 0
                   > to 10.0.0.2 via fe-1/2/0.1
```

Meaning The output shows that the 172.16/16 routes are excluded.

Checking the Routing Table

Purpose Verify that the routing table is populated with the expected routes.

Action From operational mode, enter the **show route protocol rip** command.

```
user@R1> show route protocol rip

inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.4/30      *[RIP/100] 00:54:34, metric 2, tag 0
                 > to 10.0.0.2 via fe-1/2/0.1
192.168.2.2/32   *[RIP/100] 00:54:34, metric 2, tag 0
                 > to 10.0.0.2 via fe-1/2/0.1
192.168.3.3/32   *[RIP/100] 00:54:34, metric 3, tag 0
                 > to 10.0.0.2 via fe-1/2/0.1
224.0.0.9/32     *[RIP/100] 00:49:00, metric 1
                 MultiRecv
```

Meaning The output shows that the routes have been learned from Device R2 and Device R3.

If you delete or deactivate the import policy, the routing table contains the 172.16/16 routes.

Testing the Import Policy

Purpose By using the **test policy** command, monitor the number of rejected prefixes.

Action From operational mode, enter the **test policy rip-import 172.16/16** command.

```
user@R1> test policy rip-import 172.16/16
Policy rip-import: 0 prefix accepted, 1 prefix rejected
```

Meaning The output shows that the policy rejected one prefix.

Related Documentation

- [Understanding RIP Import Policy on page 41](#)

Related Documentation

- [Example: Configuring RIP on page 7](#)

CHAPTER 6

Examples: Controlling Traffic with Metrics in a RIP Network

- [Understanding Traffic Control with Metrics in a RIP Network on page 47](#)
- [Example: Controlling Traffic in a RIP Network with an Incoming Metric on page 48](#)
- [Example: Controlling Traffic in a RIP Network with an Outgoing Metric on page 50](#)
- [Example: Configuring the Metric Value Added to Imported RIP Routes on page 52](#)

Understanding Traffic Control with Metrics in a RIP Network

To tune a RIP network and to control traffic flowing through the network, you increase or decrease the cost of the paths through the network. RIP provides two ways to modify the path cost: an incoming metric and an outgoing metric, which are each set to 1 by default. In other words, by default, the metric of routes that RIP imports from a neighbor or exports to a neighbor is incremented by 1. These routes include those learned from RIP as well as those learned from other protocols. The metrics are attributes that specify the cost of any route advertised through a host. By increasing or decreasing the metrics—and thus the cost—of links throughout the network, you can control packet transmission across the network.

The incoming metric modifies the cost of an individual segment when a route across the segment is imported into the routing table. For example, if you set the incoming metric on the segment to 3, the individual segment cost along the link is changed from 1 to 3. The increased cost affects all route calculations through that link. Other routes that were previously excluded because of a high hop count might now be included in the router's forwarding table.

The outgoing metric modifies the path cost for all the routes advertised out of a particular interface. Unlike the incoming metric, the outgoing metric modifies the routes that other routers are learning and thereby controls the way they send traffic.

If an exported route was learned from a member of the same RIP group, the metric associated with that route is the normal RIP metric. For example, a RIP route with a metric of 5 learned from a neighbor configured with an incoming metric of 2 is advertised with a combined metric of 7 when advertised to neighbors in the same group. However, if this route was learned from a RIP neighbor in a different group or from a different protocol, the route is advertised with the metric value configured in the outgoing metric for that group.

You might want to increase the metric of routes to decrease the likelihood that a particular route is selected and installed in the routing table. This process is sometimes referred to as *route poisoning*. Some reasons that you might want to poison a route are that the route is relatively expensive to use, or it has relatively low bandwidth.

A route with a higher metric than another route becomes the active route only when the lower-metric route becomes unavailable. In this way, the higher-metric route serves as a backup path.

One way to increase the metric of imported routes is to configure an import policy. Another way is to include the **metric-in** statement in the RIP neighbor configuration. One way to increase the metric of export routes is to configure an export policy. Another way is to include the **metric-out** statement in the RIP neighbor configuration.

Related Documentation

- [Junos OS Feature Support Reference for SRX Series and J Series Devices](#)
- [RIP Overview on page 1](#)
- [Example: Controlling Traffic in a RIP Network with an Incoming Metric on page 48](#)
- [Example: Controlling Traffic in a RIP Network with an Outgoing Metric on page 50](#)
- [Example: Configuring the Metric Value Added to Imported RIP Routes on page 52](#)

Example: Controlling Traffic in a RIP Network with an Incoming Metric

This example shows how to control traffic with an incoming metric.

- [Requirements on page 48](#)
- [Overview on page 48](#)
- [Configuration on page 49](#)
- [Verification on page 49](#)

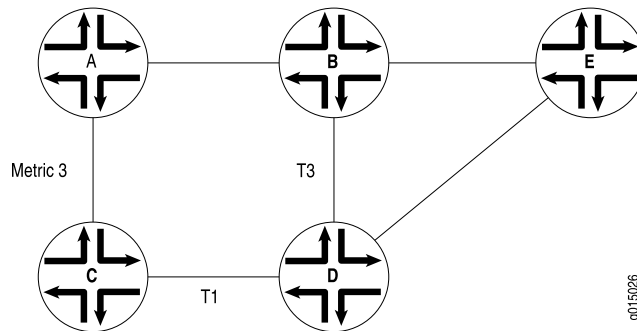
Requirements

Before you begin, define RIP groups, and add interfaces to the groups. Then configure a routing policy to export directly connected routes and routes learned through the RIP routing exchanges. See [“Example: Configuring a Basic RIP Network” on page 7](#).

Overview

In this example, routes to Router D are received by Router A across both of its RIP-enabled interfaces as shown in [Figure 10 on page 49](#). Because the route through Router B and the route through Router C have the same number of hops, both routes are imported into the forwarding table. However, because the T3 link from Router B to Router D has a higher bandwidth than the T1 link from Router C to Router D, you want traffic to flow from Router A through Router B to Router D.

Figure 10: Controlling Traffic in a RIP Network with the Incoming Metric



To force this flow, you can modify the route metrics as they are imported into Router A's routing table. By setting the incoming metric on the interface from Router A to Router C, you modify the metric on all routes received through that interface. Setting the incoming route metric on Router A changes only the routes in Router A's routing table, and affects only how Router A sends traffic to Router D. Router D's route selection is based on its own routing table, which, by default, includes no adjusted metric values.

In the example, Router C receives a route advertisement from Router D and readvertises the route to Router A. When Router A receives the route, it applies the incoming metric on the interface. Instead of incrementing the metric by 1 (the default), Router A increments it by 3 (the configured incoming metric), giving the route from Router A to Router D through Router C a total path metric of 4. Because the route through Router B has a metric of 2, it becomes the preferred route for all traffic from Router A to Router D.

This example uses a RIP group called **alpha 1** on interface **g3-0/0/0**.

Configuration

Step-by-Step Procedure

To control traffic with an incoming metric:

1. Enable RIP on the interface.

```
[edit protocols rip]
user@host# set group alpha1 neighbor ge-0/0/0
```
2. Set the incoming metric.

```
[edit protocols rip]
user@host# set metric-in 3
```
3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Verification

To verify that the configuration is working properly, enter the **show route protocols rip** command.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*

- [Understanding Traffic Control with Metrics in a RIP Network on page 47 on page 29](#)
- [RIP Configuration Overview](#)
- [Example: Controlling Traffic in a RIP Network with an Outgoing Metric on page 50](#)
- [Verifying a RIP Configuration](#)

Example: Controlling Traffic in a RIP Network with an Outgoing Metric

This example shows how to control traffic with an outgoing metric.

- [Requirements on page 50](#)
- [Overview on page 50](#)
- [Configuration on page 51](#)
- [Verification on page 51](#)

Requirements

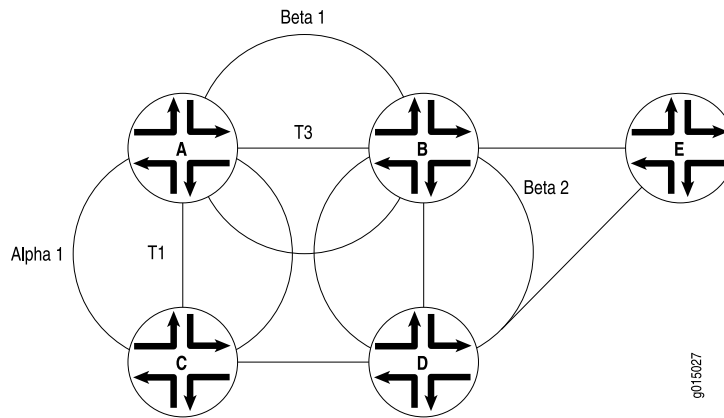
Before you begin:

- Define RIP groups, and add interfaces to the groups. Then configure a routing policy to export directly connected routes and routes learned through RIP routing exchanges. See [“Example: Configuring a Basic RIP Network” on page 7](#).
- Control traffic with an incoming metric. See [“Example: Controlling Traffic in a RIP Network with an Incoming Metric” on page 48](#).

Overview

In this example, each route from Router A to Router D has two hops as shown in [Figure 11 on page 51](#). However, because the link from Router A to Router B in the RIP group has a higher bandwidth than the link from Router A to Router C in RIP group Alpha 1, you want traffic from Router D to Router A to flow through Router B. To control the way Router D sends traffic to Router A, you can alter the routes that Router D receives by configuring the outgoing metric on Router A's interfaces in the Alpha 1 RIP group.

Figure 11: Controlling Traffic in a RIP Network with the Outgoing Metric



If the outgoing metric for the Alpha 1 RIP group—the A-to-C link—is changed to 3, Router D calculates the total path metric from Router A through Router C as 4. In contrast, the unchanged default total path metric to Router A through Router B in the RIP group is 2. The fact that Router A's interfaces belong to two different RIP groups allows you to configure two different outgoing metrics on its interfaces, because you configure path metrics at the group level.

By configuring the outgoing metric, you control the way Router A sends traffic to Router D. By configuring the outgoing metric on the same router, you control the way Router D sends traffic to Router A.

This example uses an outgoing metric of 3.

Configuration

Step-by-Step Procedure

To control traffic with an outgoing metric:

1. Set the outgoing metric.


```
[edit protocols rip group alpha1]
user@host# set metric-out 3
```
2. If you are done configuring the device, commit the configuration.


```
[edit]
user@host# commit
```

Verification

To verify that the configuration is working properly, enter the **show protocols rip** command.

Related Documentation

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- [Understanding Traffic Control with Metrics in a RIP Network on page 47](#)
- *RIP Configuration Overview*
- *Verifying a RIP Configuration*

Example: Configuring the Metric Value Added to Imported RIP Routes

This example shows how to change the default metric to be added to incoming routes to control the route selection process.

- [Requirements on page 52](#)
- [Overview on page 52](#)
- [Configuration on page 52](#)
- [Verification on page 55](#)

Requirements

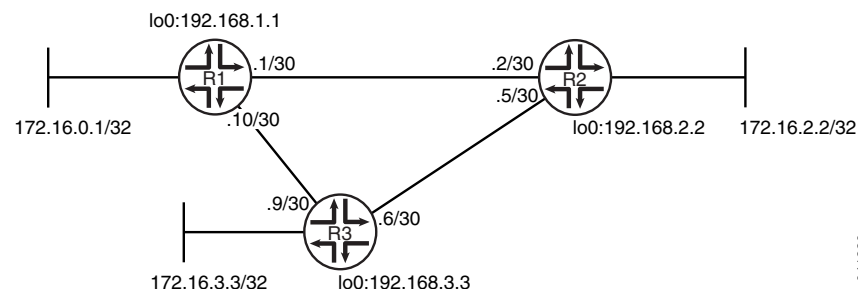
No special configuration beyond device initialization is required before configuring this example.

Overview

Normally, when multiple routes are available, RIP selects the route with the lowest hop count. Changing the default metric enables you to control the route selection process such that a route with a higher hop count can be preferred over of a route with a lower hop count.

[Figure 12 on page 52](#) shows the topology used in this example.

Figure 12: RIP Incoming Metrics Network Topology



Device R1 has two potential paths to reach 172.16.2.2/32. The default behavior is to send traffic out the 0.1/30 interface facing Device R2. Suppose, though, that the path through Device R3 is less expensive to use or has higher bandwidth links. This example shows how to use the **metric-in** statement to ensure that Device R1 uses the path through Device R3 to reach 172.16.2.2/32. “[CLI Quick Configuration](#)” on [page 52](#) shows the configuration for all of the devices in [Figure 12 on page 52](#). The section “[Step-by-Step Procedure](#)” on [page 53](#) describes the steps on Device R1.

Configuration

CLI Quick Configuration To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device R1 **set interfaces fe-1/2/0 unit 1 description to-R2**

```

set interfaces fe-1/2/0 unit 1 family inet address 10.0.0.1/30
set interfaces ge-1/2/1 unit 10 description to-R3
set interfaces ge-1/2/1 unit 10 family inet address 10.0.0.10/30
set interfaces lo0 unit 1 family inet address 172.16.0.1/32
set interfaces lo0 unit 1 family inet address 192.168.1.1/32
set protocols rip group primary export advertise-routes-through-rip
set protocols rip group primary neighbor ge-1/2/1.10
set protocols rip group secondary export advertise-routes-through-rip
set protocols rip group secondary neighbor fe-1/2/0.1 metric-in 4
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept

```

Device R2

```

set interfaces fe-1/2/0 unit 2 family inet address 10.0.0.2/30
set interfaces ge-1/2/1 unit 5 family inet address 10.0.0.5/30
set interfaces lo0 unit 2 family inet address 192.168.2.2/32
set interfaces lo0 unit 2 family inet address 172.16.2.2/32
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.2
set protocols rip group rip-group neighbor ge-1/2/1.5
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept

```

Device R3

```

set interfaces fe-1/2/0 unit 6 family inet address 10.0.0.6/30
set interfaces ge-1/2/1 unit 9 family inet address 10.0.0.9/30
set interfaces lo0 unit 3 family inet address 192.168.3.3/32
set interfaces lo0 unit 3 family inet address 172.16.3.3/32
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.6
set protocols rip group rip-group neighbor ge-1/2/1.9
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept

```

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure a RIP metrics:

1. Configure the network interfaces.

```

[edit interfaces]
user@R1# set fe-1/2/0 unit 1 description to-R2
user@R1# set fe-1/2/0 unit 1 family inet address 10.0.0.1/30

user@R1# set ge-1/2/1 unit 10 description to-R3
user@R1# set ge-1/2/1 unit 10 family inet address 10.0.0.10/30

```

```

user@R1# set lo0 unit 1 family inet address 172.16.0.1/32
user@R1# set lo0 unit 1 family inet address 192.168.1.1/32

```

2. Create the RIP groups and add the interfaces.

To configure RIP in Junos OS, you must configure one or more groups that contain the interfaces on which RIP is enabled. You do not need to enable RIP on the loopback interface.

For the interface that is facing Device R2, the **metric-in 4** setting causes this route to be less likely to be chosen as the active route.

```

[edit protocols rip]
user@R1# set group primary neighbor ge-1/2/1.10
user@R1# set group secondary neighbor fe-1/2/0.1 metric-in 4

```

3. Create the routing policy to advertise both direct and RIP-learned routes.

```

[edit policy-options policy-statement advertise-routes-through-rip term 1]
user@R1# set from protocol direct
user@R1# set from protocol rip
user@R1# set then accept

```

4. Apply the routing policy.

In Junos OS, you can only apply RIP export policies at the group level.

```

[edit protocols rip]
user@R1# set group primary export advertise-routes-through-rip
user@R1# set group secondary export advertise-routes-through-rip

```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, and **show policy-options** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

user@R1# show interfaces
fe-1/2/0 {
  unit 1 {
    description to-R2;
    family inet {
      address 10.0.0.1/30;
    }
  }
}
ge-1/2/1 {
  unit 10 {
    description to-R3;
    family inet {
      address 10.0.0.10/30;
    }
  }
}
lo0 {
  unit 1 {
    family inet {
      address 172.16.0.1/32;
      address 192.168.1.1/32;
    }
  }
}

```

```

    }
  }
}

user@R1# show protocols
rip {
  group primary {
    export advertise-routes-through-rip;
    neighbor ge-1/2/1.10;
  }
  group secondary {
    export advertise-routes-through-rip;
    neighbor fe-1/2/0.1 {
      metric-in 4;
    }
  }
}

user@R1# show policy-options
policy-statement advertise-routes-through-rip {
  term 1 {
    from protocol [ direct rip ];
    then accept;
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

- [Verifying That the Expected Route Is Active on page 55](#)
- [Removing the metric-in Statement on page 55](#)

Verifying That the Expected Route Is Active

Purpose Make sure that to reach 172.16.2.2/32, Device R1 uses the path through Device R3.

Action From operational mode, enter the **show route 172.16.2.2** command.

```

user@R1> show route 172.16.2.2
inet.0: 12 destinations, 12 routes (12 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

172.16.2.2/32      *[RIP/100] 00:15:46, metric 3, tag 0
> to 10.0.0.9 via ge-1/2/1.10

```

Meaning The **to 10.0.0.9 via ge-1/2/1.10** output shows that Device R1 uses the path through Device R3 to reach 172.16.2.2/32. The metric for this route is 3.

Removing the metric-in Statement

Purpose Delete or deactivate the **metric-in** statement to see what happens to the 172.16.2.2/32 route.

- Action** 1. From configuration mode, deactivate the **metric-in** statement.

```
[edit protocols rip group secondary neighbor fe-1/2/0.1]
user@R1# deactivate metric-in
user@R1# commit
```

2. From operational mode, enter the **show route 172.16.2.2** command.

```
user@R1> show route 172.16.2.2
inet.0: 12 destinations, 12 routes (12 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

172.16.2.2/32      *[RIP/100] 00:00:06, metric 2, tag 0
> to 10.0.0.2 via fe-1/2/0.1
```

Meaning The **to 10.0.0.2 via fe-1/2/0.1** output shows that Device R1 uses the path through Device R2 to reach 172.16.2.2/32. The metric for this route is 2.

Related Documentation

- [Understanding Traffic Control with Metrics in a RIP Network on page 47](#)

Related Documentation

- [Example: Applying Policies to RIP Routes Imported from Neighbors on page 41](#)

CHAPTER 7

Example: Configuring the Sending and Receiving of RIPv1 and RIPv2 Packets

- [Understanding the Sending and Receiving of RIPv1 and RIPv2 Packets on page 57](#)
- [Example: Configuring the Sending and Receiving of RIPv1 and RIPv2 Packets on page 57](#)

Understanding the Sending and Receiving of RIPv1 and RIPv2 Packets

RIP version 1 (RIPv1) and RIP version 2 (RIPv2) can run simultaneously. This might make sense when you are migrating a RIPv1 network to a RIPv2 network. This also allows interoperation with a device that supports RIPv1 but not RIPv2.

By default, when RIP is enabled on an interface, Junos OS receives both RIPv1 and RIPv2 packets and sends only RIPv2 packets. You can configure this behavior by including the [send](#) and [receive](#) statements in the RIP configuration.

Related Documentation

- [Example: Configuring the Sending and Receiving of RIPv1 and RIPv2 Packets on page 57](#)

Example: Configuring the Sending and Receiving of RIPv1 and RIPv2 Packets

This example shows how to configure whether the RIP update messages conform to RIP version 1 (RIPv1) only, to RIP version 2 (RIPv2) only, or to both versions. You can also disable the sending or receiving of update messages.

- [Requirements on page 57](#)
- [Overview on page 58](#)
- [Configuration on page 58](#)
- [Verification on page 60](#)

Requirements

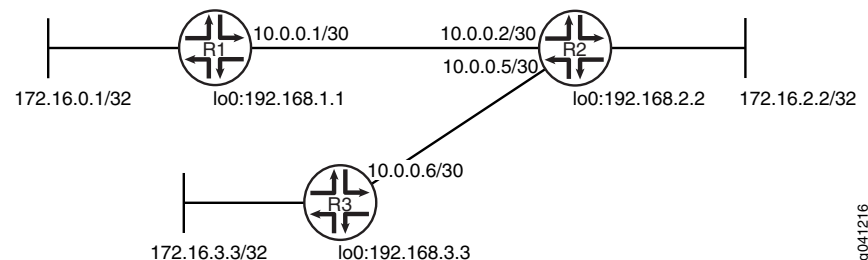
No special configuration beyond device initialization is required before configuring this example.

Overview

By default, when RIP is enabled on an interface, Junos OS receives both RIPv1 and RIPv2 packets and sends only RIPv2 packets.

Figure 13 on page 58 shows the topology used in this example.

Figure 13: Sending and Receiving RIPv1 and RIPv2 Packets Network Topology



In this example, Device R1 is configured to receive only RIPv2 packets.

“CLI Quick Configuration” on page 58 shows the configuration for all of the devices in Figure 13 on page 58. The section “Step-by-Step Procedure” on page 59 describes the steps on Device R1.

Configuration

CLI Quick Configuration	To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the [edit] hierarchy level.
Device R1	<pre> set interfaces fe-1/2/0 unit 1 family inet address 10.0.0.1/30 set interfaces lo0 unit 1 family inet address 172.16.0.1/32 set interfaces lo0 unit 1 family inet address 192.168.1.1/32 set protocols rip group rip-group export advertise-routes-through-rip set protocols rip group rip-group neighbor fe-1/2/0.1 receive version-2 set policy-options policy-statement advertise-routes-through-rip term 1 from protocol direct set policy-options policy-statement advertise-routes-through-rip term 1 from protocol rip set policy-options policy-statement advertise-routes-through-rip term 1 then accept </pre>
Device R2	<pre> set interfaces fe-1/2/0 unit 2 family inet address 10.0.0.2/30 set interfaces fe-1/2/1 unit 5 family inet address 10.0.0.5/30 set interfaces lo0 unit 2 family inet address 192.168.2.2/32 set interfaces lo0 unit 2 family inet address 172.16.2.2/32 set protocols rip group rip-group export advertise-routes-through-rip set protocols rip group rip-group neighbor fe-1/2/0.2 set protocols rip group rip-group neighbor fe-1/2/1.5 set policy-options policy-statement advertise-routes-through-rip term 1 from protocol direct set policy-options policy-statement advertise-routes-through-rip term 1 from protocol rip </pre>

```
set policy-options policy-statement advertise-routes-through-rip term 1 then accept
```

Device R3

```
set interfaces fe-1/2/0 unit 6 family inet address 10.0.0.6/30
set interfaces lo0 unit 3 family inet address 192.168.3.3/32
set interfaces lo0 unit 3 family inet address 172.16.3.3/32
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.6
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept
```

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure a RIP packet versions that can be received:

1. Configure the network interfaces.

```
[edit interfaces]
user@R1# set fe-1/2/0 unit 1 family inet address 10.0.0.1/30

user@R1# set lo0 unit 1 family inet address 172.16.0.1/32
user@R1# set lo0 unit 1 family inet address 192.168.1.1/32
```

2. Create the RIP groups and add the interfaces.

To configure RIP in Junos OS, you must configure one or more groups that contain the interfaces on which RIP is enabled. You do not need to enable RIP on the loopback interface.

For the interface that is facing Device R2, the **receive version-2** setting causes this interface to accept only RIPv2 packets.

```
[edit protocols rip group rip-group]
user@R1# set neighbor fe-1/2/0.1 receive version-2
```

3. Create the routing policy to advertise both direct and RIP-learned routes.

```
[edit policy-options policy-statement advertise-routes-through-rip term 1]
user@R1# set from protocol direct
user@R1# set from protocol rip
user@R1# set then accept
```

4. Apply the routing policy.

In Junos OS, you can only apply RIP export policies at the group level.

```
[edit protocols rip group rip-group]
user@R1# set export advertise-routes-through-rip
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, and **show policy-options** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

user@R1# show interfaces
fe-1/2/0 {
  unit 1 {
    family inet {
      address 10.0.0.1/30;
    }
  }
}
lo0 {
  unit 1 {
    family inet {
      address 172.16.0.1/32;
      address 192.168.1.1/32;
    }
  }
}

user@R1# show protocols
rip {
  group rip-group {
    export advertise-routes-through-rip;
    neighbor fe-1/2/0.1 {
      receive version-2;
    }
  }
}

user@R1# show policy-options
policy-statement advertise-routes-through-rip {
  term 1 {
    from protocol [ direct rip ];
    then accept;
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

Verifying That the Receive Mode Is Set to RIPv2 Only

Purpose Make sure that the interfacing Device R2 is configured to receive only RIPv2 packets, instead of both RIPv1 and RIPv2 packets.

Action From operational mode, enter the **show rip neighbor** command.

```
user@R1> show rip neighbor
```

Neighbor	Local State	Source Address	Destination Address	Send Mode	Receive Mode	In Met
fe-1/2/0.1	Up	10.0.0.1	224.0.0.9	mcast	v2 only	1

Meaning In the output, the **Receive Mode** field displays **v2 only**. The default **Receive Mode** is **both**.

Related Documentation • [Understanding the Sending and Receiving of RIPv1 and RIPv2 Packets on page 57](#)

Related Documentation • [Example: Configuring RIP on page 7](#)

Example: Redistributing Routes Among RIP Instances

- [Understanding Route Redistribution Among RIP instances on page 63](#)
- [Example: Redistributing Routes Between Two RIP Instances on page 64](#)

Understanding Route Redistribution Among RIP instances

You can redistribute routes among RIP processes. Another way to say this is to export RIP routes from one RIP instance to other RIP instances.

In Junos OS, route redistribution among routing instances is accomplished by using routing table groups, also called RIB groups. Routing table groups allow you to import and export routes from a protocol within one routing table into another routing table.



NOTE: In contrast, the policy-based import and export functions allow you import and export routes between different protocols within the same routing table.

Consider the following partial example:

```
protocols {
  rip {
    rib-group inet-to-voice;
  }
}
routing-instances {
  voice {
    protocols {
      rip {
        rib-group voice-to-inet;
      }
    }
  }
}
routing-options {
  rib-groups {
    inet-to-voice {
      import-rib [ inet.0 voice.inet.0 ];
    }
  }
}
```

```
    }  
    voice-to-inet {  
        import-rib [ voice.inet.0 inet.0 ];  
    }  
}
```

The way to read the **import-rib** statement is as follows. Take the routes from the protocol (RIP, in this case), and import them into the primary (or local) routing table and also into any other routing tables listed after this. The primary routing table is the routing table where the routing table group is being used. That would be either **inet.0** if used in the main routing instance or **voice.inet.0** if used within the routing instance. In the **inet-to-voice** routing table group, **inet.0** is listed first because this routing table group is used in the main routing instance. In the **voice-to-inet** routing table group, **voice.inet.0** is listed first because this routing table group is used in the voice routing instance.

Related Documentation

- [Example: Redistributing Routes Between Two RIP Instances on page 64](#)

Example: Redistributing Routes Between Two RIP Instances

This example shows how to configure a RIP routing instance and control the redistribution of RIP routes between the routing instance and the master instance.

- [Requirements on page 64](#)
- [Overview on page 64](#)
- [Configuration on page 65](#)
- [Verification on page 68](#)

Requirements

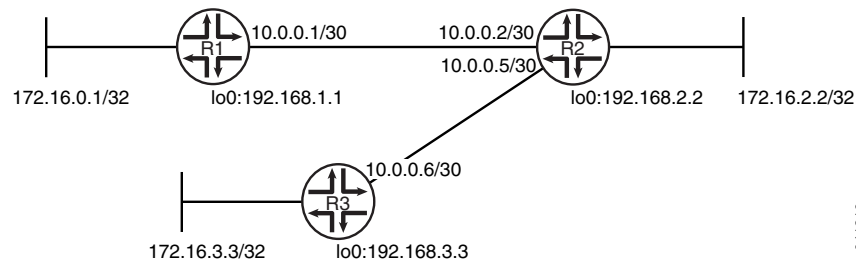
No special configuration beyond device initialization is required before configuring this example.

Overview

When you create a routing instance called **voice**, Junos OS creates a routing table called **voice.inet.0**. The example shows how to install routes learned through the master RIP instance into the **voice.inet.0** routing table. The example also shows how to install routes learned through the voice routing instance into **inet.0**. This is done by configuring routing table groups. RIP routes are installed into each routing table that belongs to a routing table group.

[Figure 14 on page 65](#) shows the topology used in this example.

Figure 14: Redistributing Routes Between RIP Instances Network Topology



"CLI Quick Configuration" on page 65 shows the configuration for all of the devices in Figure 14 on page 65. The section "Step-by-Step Procedure" on page 66 describes the steps on Device R2.

Configuration

CLI Quick Configuration To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device R1

```

set interfaces fe-1/2/0 unit 1 family inet address 10.0.0.1/30
set interfaces lo0 unit 1 family inet address 172.16.0.1/32
set interfaces lo0 unit 1 family inet address 192.168.1.1/32
set protocols rip group to-R2 export advertise-routes-through-rip
set protocols rip group to-R2 neighbor fe-1/2/0.1
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept

```

Device R2

```

set interfaces fe-1/2/0 unit 2 family inet address 10.0.0.2/30
set interfaces fe-1/2/1 unit 5 family inet address 10.0.0.5/30
set interfaces lo0 unit 2 family inet address 192.168.2.2/32
set interfaces lo0 unit 2 family inet address 172.16.2.2/32
set protocols rip rib-group inet-to-voice
set protocols rip group to-R3 export advertise-routes-through-rip
set protocols rip group to-R3 neighbor fe-1/2/1.5
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept
set routing-instances voice protocols rip group to-R1 export advertise-routes-through-rip
set routing-instances voice interface fe-1/2/0.2
set routing-instances voice protocols rip rib-group voice-to-inet
set routing-instances voice protocols rip group to-R1 neighbor fe-1/2/0.2
set routing-options rib-groups inet-to-voice import-rib inet.0
set routing-options rib-groups inet-to-voice import-rib voice.inet.0
set routing-options rib-groups voice-to-inet import-rib voice.inet.0
set routing-options rib-groups voice-to-inet import-rib inet.0

```

Device R3

```

set interfaces fe-1/2/0 unit 6 family inet address 10.0.0.6/30

```

```
set interfaces lo0 unit 3 family inet address 192.168.3.3/32
set interfaces lo0 unit 3 family inet address 172.16.3.3/32
set protocols rip group to-R2 export advertise-routes-through-rip
set protocols rip group to-R2 neighbor fe-1/2/0.6
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept
```

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To redistribute RIP routes between routing instances:

1. Configure the network interfaces.

```
[edit interfaces]
user@R2# set fe-1/2/0 unit 2 family inet address 10.0.0.2/30

user@R2# set fe-1/2/1 unit 5 family inet address 10.0.0.5/30
```

```
user@R2# set lo0 unit 2 family inet address 192.168.2.2/32
user@R2# set lo0 unit 2 family inet address 172.16.2.2/32
```

2. Create the routing instance, and add one or more interfaces to the routing instance.

```
[edit routing-instances voice]
user@R2# set interface fe-1/2/0.2
```

3. Create the RIP groups and add the interfaces.

```
[edit protocols rip group to-R3]
user@R2# set neighbor fe-1/2/1.5

[edit routing-instances voice protocols rip group to-R1]
user@R2# set neighbor fe-1/2/0.2
```

4. Create the routing table groups.

```
[edit routing-options rib-groups]
user@R2# set inet-to-voice import-rib inet.0
user@R2# set inet-to-voice import-rib voice.inet.0

user@R2# set voice-to-inet import-rib voice.inet.0
user@R2# set voice-to-inet import-rib inet.0
```

5. Apply the routing table groups.

```
[edit protocols rip]
user@R2# set rib-group inet-to-voice

[edit routing-instances voice protocols rip]
user@R2# set rib-group voice-to-inet
```

6. Create the routing policy to advertise both direct and RIP-learned routes.

```
[edit policy-options policy-statement advertise-routes-through-rip term 1]
user@R2# set from protocol direct
user@R2# set from protocol rip
user@R2# set then accept
```

7. Apply the routing policy.

In Junos OS, you can only apply RIP export policies at the group level.

```
[edit protocols rip group to-R3]
user@R2# set export advertise-routes-through-rip
```

```
[edit routing-instances voice protocols rip group to-R1]
user@R2# set export advertise-routes-through-rip
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, **show policy-options**, **show routing-instances**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@R2# show interfaces
fe-1/2/0 {
  unit 2 {
    family inet {
      address 10.0.0.2/30;
    }
  }
}
fe-1/2/1 {
  unit 5 {
    family inet {
      address 10.0.0.5/30;
    }
  }
}
lo0 {
  unit 2 {
    family inet {
      address 192.168.2.2/32;
      address 172.16.2.2/32;
    }
  }
}

user@R2# show protocols
rip {
  rib-group inet-to-voice;
  group to-R3 {
    export advertise-routes-through-rip;
    neighbor fe-1/2/1.5;
  }
}

user@R2# show policy-options
policy-statement advertise-routes-through-rip {
  term 1 {
```

```

        from protocol [ direct rip ];
        then accept;
    }
}

user@R2# show routing-instances
voice {
    interface fe-1/2/0.2;
    protocols {
        rip {
            rib-group voice-to-inet;
            group to-R1 {
                export advertise-routes-through-rip;
                neighbor fe-1/2/0.2;
            }
        }
    }
}

user@R2# show routing-options
rib-groups {
    inet-to-voice {
        import-rib [ inet.0 voice.inet.0 ];
    }
    voice-to-inet {
        import-rib [ voice.inet.0 inet.0 ];
    }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

Checking the Routing Tables

Purpose Make sure that the routing tables contain the expected routes.

Action From operational mode, enter the **show route protocol rip** command.

```

user@R2> show route protocol rip
inet.0: 9 destinations, 9 routes (9 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

172.16.0.1/32      * [RIP/100] 01:58:14, metric 2, tag 0
> to 10.0.0.1 via fe-1/2/0.2
172.16.3.3/32      * [RIP/100] 02:06:03, metric 2, tag 0
> to 10.0.0.6 via fe-1/2/0.5
192.168.1.1/32     * [RIP/100] 01:58:14, metric 2, tag 0
> to 10.0.0.1 via fe-1/2/0.2
192.168.3.3/32     * [RIP/100] 02:06:03, metric 2, tag 0
> to 10.0.0.6 via fe-1/2/0.5
224.0.0.9/32       * [RIP/100] 01:44:13, metric 1
MultiRecv

voice.inet.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

```

```
172.16.0.1/32      *[RIP/100] 02:06:03, metric 2, tag 0
                  > to 10.0.0.1 via fe-1/2/0.2
172.16.3.3/32      *[RIP/100] 01:58:14, metric 2, tag 0
                  > to 10.0.0.6 via fe-1/2/0.5
192.168.1.1/32     *[RIP/100] 02:06:03, metric 2, tag 0
                  > to 10.0.0.1 via fe-1/2/0.2
192.168.3.3/32     *[RIP/100] 01:58:14, metric 2, tag 0
                  > to 10.0.0.6 via fe-1/2/0.5
224.0.0.9/32       *[RIP/100] 01:44:13, metric 1
                  MultiRecv
```

Meaning The output shows that both routing tables contain all of the RIP routes.

Related Documentation

- [Understanding Route Redistribution Among RIP instances on page 63](#)

Related Documentation

- [Example: Configuring RIP on page 7](#)
- [Example: Applying Policies to RIP Routes Imported from Neighbors on page 41](#)

CHAPTER 9

Example: Configuring RIP Timers

- [Understanding RIP Timers on page 71](#)
- [Example: Configuring RIP Timers on page 72](#)

Understanding RIP Timers

RIP uses several timers to regulate its operation.

The update interval is the interval at which routes that are learned by RIP are advertised to neighbors. This timer controls the interval between routing updates. The update interval is set to 30 seconds, by default, with a small random amount of time added when the timer is reset. This added time prevents congestion that can occur if all routing devices update their neighbors simultaneously.

To configure the update time interval, include the **update-interval** statement:

```
update-interval seconds;
```

seconds can be a value from 10 through 60.

You can set a route timeout interval. If a route is not refreshed after being installed in the routing table by the specified time interval, the route is marked as invalid and is removed from the routing table after the hold-down period expires.

To configure the route timeout for RIP, include the **route-timeout** statement:

```
route-timeout seconds;
```

seconds can be a value from 30 through 360. The default value is 180 seconds.

RIP routes expire when either a route timeout limit is met or a route metric reaches infinity, and the route is no longer valid. However, the expired route is retained in the routing table for a specified period so that neighbors can be notified that the route has been dropped. This time period is set by configuring the hold-down timer. Upon expiration of the hold-down timer, the route is removed from the routing table.

To configure the hold-down timer for RIP, include the **holddown** statement:

```
holddown seconds;
```

seconds can be a value from 10 through 180. The default value is 120 seconds.



NOTE: In Junos OS Release 11.1 and later, a retransmission timer is available for RIP demand circuits.

Generally, we recommend against changing the RIP timers, unless the effects of a change are well understood. The route timeout should be at least three times the update interval. Normally, the default values are best left in effect for standard operations.

**Related
Documentation**

- [Example: Configuring RIP Timers on page 72](#)
- [Example: Configuring RIP Demand Circuits](#)

Example: Configuring RIP Timers

This example shows how to configure the RIP update interval and how to monitor the impact of the change.

- [Requirements on page 72](#)
- [Overview on page 72](#)
- [Configuration on page 73](#)
- [Verification on page 75](#)

Requirements

No special configuration beyond device initialization is required before configuring this example.

Overview

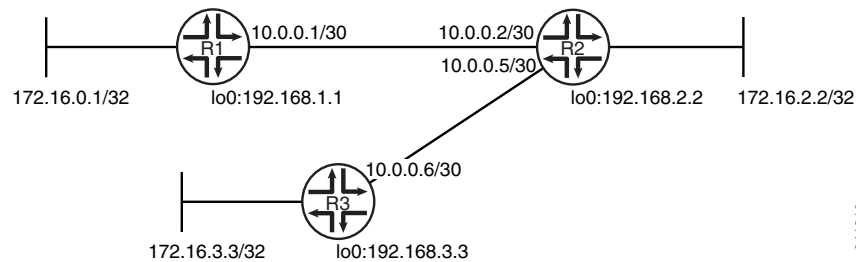
In this example, Device R2 has an update interval of 60 seconds for its neighbor, Device R1, and an update interval of 10 seconds for its neighbor, Device R3.

This example is not necessarily practical, but it is shown for demonstration purposes. Generally, we recommend against changing the RIP timers, unless the effects of a change are well understood. Normally, the default values are best left in effect for standard operations.

An export policy is also shown because an export policy is required as part of the minimum configuration for RIP.

[Figure 15 on page 73](#) shows the topology used in this example.

Figure 15: RIP Timers Network Topology



"CLI Quick Configuration" on page 73 shows the configuration for all of the devices in Figure 15 on page 73. The section "Step-by-Step Procedure" on page 74 describes the steps on Device R2.

Configuration

CLI Quick Configuration To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device R1

```

set interfaces fe-1/2/0 unit 1 family inet address 10.0.0.1/30
set interfaces lo0 unit 1 family inet address 172.16.0.1/32
set interfaces lo0 unit 1 family inet address 192.168.1.1/32
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.1
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept

```

Device R2

```

set interfaces fe-1/2/0 unit 2 family inet address 10.0.0.2/30
set interfaces fe-1/2/1 unit 5 family inet address 10.0.0.5/30
set interfaces lo0 unit 2 family inet address 192.168.2.2/32
set interfaces lo0 unit 2 family inet address 172.16.2.2/32
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.2 update-interval 60
set protocols rip group rip-group neighbor fe-1/2/1.5 update-interval 10
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept

```

Device R3

```

set interfaces fe-1/2/0 unit 6 family inet address 10.0.0.6/30
set interfaces lo0 unit 3 family inet address 192.168.3.3/32
set interfaces lo0 unit 3 family inet address 172.16.3.3/32
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.6
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip

```

```
set policy-options policy-statement advertise-routes-through-rip term 1 then accept
```

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure the RIP update interval:

1. Configure the network interfaces.

This example shows multiple loopback interface addresses to simulate attached networks.

```
[edit interfaces]
user@R2# set fe-1/2/0 unit 2 family inet address 10.0.0.2/30

user@R2# set fe-1/2/1 unit 5 family inet address 10.0.0.5/30
```

```
user@R2# set lo0 unit 2 family inet address 192.168.2.2/32
user@R2# set lo0 unit 2 family inet address 172.16.2.2/32
```

2. Configure different update intervals for the two RIP neighbors.

To configure RIP in Junos OS, you must configure a group that contains the interfaces on which RIP is enabled. You do not need to enable RIP on the loopback interface.

```
[edit protocols rip group rip-group]
user@R2# set neighbor fe-1/2/0.2 update-interval 60
user@R2# set neighbor fe-1/2/1.5 update-interval 10
```

3. Create the routing policy to advertise both direct and RIP-learned routes.

```
[edit policy-options policy-statement advertise-routes-through-rip term 1]
user@R2# set from protocol direct
user@R2# set from protocol rip
user@R2# set then accept
```

4. Apply the routing policy.

In Junos OS, you can only apply RIP export policies at the group level.

```
[edit protocols rip group rip-group]
user@R2# set export advertise-routes-through-rip
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, and **show policy-options** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@R2# show interfaces
fe-1/2/0 {
  unit 2 {
    family inet {
      address 10.0.0.2/30;
    }
  }
}
```

```

fe-1/2/1 {
  unit 5 {
    family inet {
      address 10.0.0.5/30;
    }
  }
}
lo0 {
  unit 2 {
    family inet {
      address 192.168.2.2/32;
      address 172.16.2.2/32;
    }
  }
}

user@R2# show protocols
rip {
  group rip-group {
    export advertise-routes-through-rip;
    neighbor fe-1/2/0.2 {
      update-interval 60;
    }
    neighbor fe-1/2/1.5 {
      update-interval 10;
    }
  }
}

user@R2# show policy-options
policy-statement advertise-routes-through-rip {
  term 1 {
    from protocol [ direct rip ];
    then accept;
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

- [Checking the RIP Updates Sent by Device R2 on page 75](#)
- [Checking the RIP Updates Received by Device R2 on page 76](#)
- [Checking the RIP Updates Received by Device R3 on page 77](#)

Checking the RIP Updates Sent by Device R2

Purpose Make sure that the RIP update packets are sent at the expected interval.

Action From operational mode, enter the **show rip statistics** command.

```

user@R2> show rip statistics
RIPv2 info: port 520; holddown 120s.
      rts learned  rts held down  rqsts dropped  resps dropped

```

```

4          2          0          0

fe-1/2/0.2: 2 routes learned; 5 routes advertised; timeout 180s; update interval
60s

```

Counter	Total	Last 5 min	Last minute
-----	-----	-----	-----
Updates Sent	123	5	1
Triggered Updates Sent	0	0	0
Responses Sent	0	0	0
Bad Messages	0	0	0
RIPv1 Updates Received	0	0	0
RIPv1 Bad Route Entries	0	0	0
RIPv1 Updates Ignored	0	0	0
RIPv2 Updates Received	244	10	2
RIPv2 Bad Route Entries	0	0	0
RIPv2 Updates Ignored	0	0	0
Authentication Failures	0	0	0
RIP Requests Received	0	0	0
RIP Requests Ignored	0	0	0
none	0	0	0

```

fe-1/2/1.5: 2 routes learned; 5 routes advertised; timeout 180s; update interval
10s

```

Counter	Total	Last 5 min	Last minute
-----	-----	-----	-----
Updates Sent	734	32	6
Triggered Updates Sent	0	0	0
Responses Sent	0	0	0
Bad Messages	0	0	0
RIPv1 Updates Received	0	0	0
RIPv1 Bad Route Entries	0	0	0
RIPv1 Updates Ignored	0	0	0
RIPv2 Updates Received	245	11	2
RIPv2 Bad Route Entries	0	0	0
RIPv2 Updates Ignored	0	0	0
Authentication Failures	0	0	0
RIP Requests Received	0	0	0
RIP Requests Ignored	0	0	0
none	0	0	0

Meaning The **update interval** field shows that the interval is 60 seconds for Neighbor R1 and 10 seconds for Neighbor R3. The **Updates Sent** field shows that Device R2 is sending updates to Device R1 at roughly 1/6 of the rate that it is sending updates to Device R3.

Checking the RIP Updates Received by Device R2

Purpose Make sure that the RIP update packets are sent at the expected interval.

Action From operational mode, enter the **show rip statistics** command.

```

user@R1> show rip statistics
RIPv2 info: port 520; holddown 120s.
    rts learned  rts held down  rqsts dropped  resps dropped
      5          0          0          0

```

```

fe-1/2/0.1: 5 routes learned; 2 routes advertised; timeout 180s; update interval
30s

```

Counter	Total	Last 5 min	Last minute
-----	-----	-----	-----

Updates Sent	312	10	2
Triggered Updates Sent	2	0	0
Responses Sent	0	0	0
Bad Messages	0	0	0
RIPv1 Updates Received	0	0	0
RIPv1 Bad Route Entries	0	0	0
RIPv1 Updates Ignored	0	0	0
RIPv2 Updates Received	181	5	1
RIPv2 Bad Route Entries	0	0	0
RIPv2 Updates Ignored	0	0	0
Authentication Failures	0	0	0
RIP Requests Received	1	0	0
RIP Requests Ignored	0	0	0
none	0	0	0

Meaning The **RIPv2 Updates Received** field shows the number of updates received from Device R2.

Checking the RIP Updates Received by Device R3

Purpose Make sure that the RIP update packets are sent at the expected interval.

Action From operational mode, enter the **show rip statistics** command.

```
user@R3> show rip statistics
```

```
RIPv2 info: port 520; holddown 120s.
```

```
    rts learned   rts held down   rqsts dropped   resps dropped
          5             0             0             0
```

```
fe-1/2/0.6: 5 routes learned; 2 routes advertised; timeout 180s; update interval 30s
```

Counter	Total	Last 5 min	Last minute
-----	-----	-----	-----
Updates Sent	314	11	2
Triggered Updates Sent	1	0	0
Responses Sent	0	0	0
Bad Messages	0	0	0
RIPv1 Updates Received	0	0	0
RIPv1 Bad Route Entries	0	0	0
RIPv1 Updates Ignored	0	0	0
RIPv2 Updates Received	827	31	6
RIPv2 Bad Route Entries	0	0	0
RIPv2 Updates Ignored	0	0	0
Authentication Failures	0	0	0
RIP Requests Received	0	0	0
RIP Requests Ignored	0	0	0
none	0	0	0

Meaning The **RIPv2 Updates Received** field shows the number of updates received from Device R2.

Related Documentation

- [Understanding RIP Timers on page 71](#)

Related Documentation

- [Example: Configuring RIP on page 7](#)
- [Example: Configuring RIP Demand Circuits](#)

CHAPTER 10

Example: Tracing RIP Protocol Traffic

- [Understanding RIP Trace Operations on page 79](#)
- [Example: Tracing RIP Protocol Traffic on page 80](#)

Understanding RIP Trace Operations

You can trace various types of RIP protocol traffic to help debug RIP protocol issues.

To trace RIP protocol traffic, include the **traceoptions** statement at the **[edit protocols rip]** hierarchy level:

```
traceoptions {  
  file filename <files number> <size size> <world-readable | no-world-readable>;  
  flag flag <flag-modifier> <disable>;  
}
```

You can specify the following RIP protocol-specific trace options using the **flag** statement:

- **auth**—RIP authentication
- **error**—RIP error packets
- **expiration**—RIP route expiration processing
- **holddown**—RIP hold-down processing
- **nsr-synchronization**—Nonstop active routing synchronization events
- **packets**—All RIP packets
- **request**—RIP information packets
- **trigger**—RIP triggered updates
- **update**—RIP update packets

You can optionally specify one or more of the following flag modifiers:

- **detail**—Detailed trace information
- **receive**—Packets being received
- **send**—Packets being transmitted



NOTE: Use the **detail** flag modifier with caution as this may cause the CPU to become very busy.

Global tracing options are inherited from the configuration set by the **traceoptions** statement at the **[edit routing-options]** hierarchy level. You can override the following global trace options for the RIP protocol using the **traceoptions flag** statement included at the **[edit protocols rip]** hierarchy level:

- **all**—All tracing operations
- **general**—All normal operations and routing table changes (a combination of the normal and route trace operations)
- **normal**—Normal events
- **policy**—Policy processing
- **route**—Routing information
- **state**—State transitions
- **task**—Routing protocol task processing
- **timer**—Routing protocol timer processing



NOTE: Use the trace flag **all** with caution because this may cause the CPU to become very busy.

Related Documentation

- [Example: Tracing RIP Protocol Traffic on page 80](#)

Example: Tracing RIP Protocol Traffic

This example shows how to trace RIP protocol operations.

- [Requirements on page 80](#)
- [Overview on page 80](#)
- [Configuration on page 81](#)
- [Verification on page 83](#)

Requirements

No special configuration beyond device initialization is required before configuring this example.

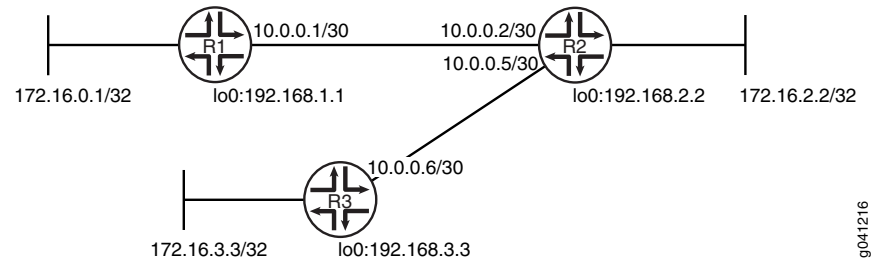
Overview

In this example, Device R1 is set to trace routing information updates.

An export policy is also shown because an export policy is required as part of the minimum configuration for RIP.

Figure 16 on page 81 shows the topology used in this example.

Figure 16: RIP Trace Operations Network Topology



"CLI Quick Configuration" on page 81 shows the configuration for all of the devices in Figure 16 on page 81. The section "Step-by-Step Procedure" on page 82 describes the steps on Device R1.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device R1

```
set interfaces fe-1/2/0 unit 1 family inet address 10.0.0.1/30
set interfaces lo0 unit 1 family inet address 172.16.0.1/32
set interfaces lo0 unit 1 family inet address 192.168.1.1/32
set protocols rip traceoptions file rip-trace-file
set protocols rip traceoptions flag route
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.1
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept
```

Device R2

```
set interfaces fe-1/2/0 unit 2 family inet address 10.0.0.2/30
set interfaces fe-1/2/1 unit 5 family inet address 10.0.0.5/30
set interfaces lo0 unit 2 family inet address 192.168.2.2/32
set interfaces lo0 unit 2 family inet address 172.16.2.2/32
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.2
set protocols rip group rip-group neighbor fe-1/2/1.5
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept
```

Device R3

```
set interfaces fe-1/2/0 unit 6 family inet address 10.0.0.6/30
set interfaces lo0 unit 3 family inet address 192.168.3.3/32
```

```
set interfaces lo0 unit 3 family inet address 172.16.3.3/32
set protocols rip group rip-group export advertise-routes-through-rip
set protocols rip group rip-group neighbor fe-1/2/0.6
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  direct
set policy-options policy-statement advertise-routes-through-rip term 1 from protocol
  rip
set policy-options policy-statement advertise-routes-through-rip term 1 then accept
```

Step-by-Step Procedure The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure the RIP update interval:

1. Configure the network interfaces.

This example shows multiple loopback interface addresses to simulate attached networks.

```
[edit interfaces]
user@R1# set fe-1/2/0 unit 1 family inet address 10.0.0.1/30
```

```
user@R1# set lo0 unit 1 family inet address 172.16.0.1/32
user@R1# set lo0 unit 1 family inet address 192.168.1.1/32
```

2. Configure the RIP group, and add the interface to the group.

To configure RIP in Junos OS, you must configure a group that contains the interfaces on which RIP is enabled. You do not need to enable RIP on the loopback interface.

```
[edit protocols rip group rip-group]
user@R1# set neighbor fe-1/2/0.1
```

3. Configure RIP tracing operations.

```
[edit protocols rip traceoptions]
user@R1# set file rip-trace-file
user@R1# set flag route
```

4. Create the routing policy to advertise both direct and RIP-learned routes.

```
[edit policy-options policy-statement advertise-routes-through-rip term 1]
user@R1# set from protocol direct
user@R1# set from protocol rip
user@R1# set then accept
```

5. Apply the routing policy.

In Junos OS, you can only apply RIP export policies at the group level.

```
[edit protocols rip group rip-group]
user@R1# set export advertise-routes-through-rip
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, and **show policy-options** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```

user@R1# show interfaces
fe-1/2/0 {
  unit 1 {
    family inet {
      address 10.0.0.1/30;
    }
  }
}
lo0 {
  unit 1 {
    family inet {
      address 172.16.0.1/32;
      address 192.168.1.1/32;
    }
  }
}

user@R1# show protocols
rip {
  traceoptions {
    file rip-trace-file;
    flag route;
  }
  group rip-group {
    export advertise-routes-through-rip;
    neighbor fe-1/2/0.1;
  }
}

user@R1# show policy-options
policy-statement advertise-routes-through-rip {
  term 1 {
    from protocol [ direct rip ];
    then accept;
  }
}

```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

Checking the Log File

Purpose Make sure that the RIP route updates are logged in the configured log file.

- Action**
1. Deactivate the extra loopback interface address on Device R3.

```

[edit interfaces lo0 unit 3 family inet]
user@R3# deactivate address 172.16.3.3/32
user@R3# commit

```
 2. From operational mode on Device R1, enter the **show log rip-trace-file** command with the **| match 172.16.3.3** option.

```

user@R1> show log rip-trace-file | match 172.16.3.3

```

```
Mar 1 11:39:53.975192 Setting RIPv2 rtbit on route 172.16.3.3/32, tsi =
0xbb69228
Mar 1 11:39:59.847118 172.16.3.3/32: metric-in: 16, change: 3 -> 16; # gw:
1, pkt_upd_src 10.0.0.2, inx: 0, rte_upd_src 10.0.0.2
Mar 1 11:39:59.847568 CHANGE 172.16.3.3/32 nhid 591 gw 10.0.0.2
RIP pref 100/0 metric 3/0 fe-1/2/0.1 <Delete Int>
Mar 1 11:39:59.847629 Best route to 172.16.3.3/32 got deleted. Doing route calculation
on the stored rte-info
```

Meaning The output shows that the route to 172.16.3.3/32 was deleted.

Related Documentation

- [Understanding RIP Trace Operations on page 79](#)

Related Documentation

- [Example: Configuring RIP on page 7](#)

PART 2

Configuration Statements and Operational Commands


- [Configuration Statements for RIP on page 87](#)
- [RIP Operational Commands on page 113](#)

CHAPTER 11

Configuration Statements for RIP

- [any-sender](#) on page 88
- [authentication-key \(Protocols RIP\)](#) on page 89
- [authentication-type \(Protocols RIP\)](#) on page 90
- [bfd-liveness-detection \(Protocols RIP\)](#) on page 91
- [check-zero](#) on page 94
- [export \(Protocols RIP\)](#) on page 95
- [group \(Protocols RIP\)](#) on page 96
- [holddown \(Protocols RIP\)](#) on page 98
- [import \(Protocols RIP\)](#) on page 99
- [message-size](#) on page 100
- [metric-in \(Protocols RIP\)](#) on page 101
- [metric-out \(Protocols RIP\)](#) on page 102
- [neighbor \(Protocols RIP\)](#) on page 103
- [preference \(Protocols RIP\)](#) on page 104
- [receive \(Protocols RIP\)](#) on page 105
- [rib-group \(Protocols RIP\)](#) on page 106
- [rip](#) on page 106
- [route-timeout \(Protocols RIP\)](#) on page 107
- [send \(Protocols RIP\)](#) on page 108
- [traceoptions \(Protocols RIP\)](#) on page 109
- [update-interval \(Protocols RIP\)](#) on page 112

any-sender

Syntax	<code>any-sender;</code>
Hierarchy Level	<code>[edit logical-systems <i>logical-system-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</code> <code>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols</code> <code>rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</code> <code>[edit protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</code> <code>[edit routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor</code> <code><i>neighbor-name</i>]</code>
Release Information	Statement introduced in Junos OS Release 8.0. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	<p>Disable strict sender address checks.</p> <p>If the sender of a RIP message does not belong to the subnet of the interface, the message is discarded. This situation might cause problems with dropped packets when RIP is running on point-to-point interfaces, or when the addresses on the interfaces do not fall in the same subnet. You can resolve this by disabling strict address checks on the RIP traffic.</p>
<div> NOTE: The <code>any-sender</code> statement is supported only for peer-to-peer interfaces.</div>	
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Example: Configuring RIP on page 7

authentication-key (Protocols RIP)

Syntax	<code>authentication-key password;</code>
Hierarchy Level	<p>[edit logical-systems <i>logical-system-name</i> protocols rip],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit protocols rip],</p> <p>[edit protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>]</p>
Release Information	<p>Statement introduced before Junos OS Release 7.4.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 12.1 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.</p>
Description	Require authentication for RIP route queries received on an interface.
Options	<i>password</i> —Authentication password. If the password does not match, the packet is rejected. The password can be from 1 through 16 contiguous characters long and can include any ASCII strings.
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • Example: Configuring Route Authentication for RIP on page 15

authentication-type (Protocols RIP)

Syntax	<code>authentication-type type;</code>
Hierarchy Level	<p>[edit logical-systems <i>logical-system-name</i> protocols rip],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit protocols rip],</p> <p>[edit protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>]</p>
Release Information	<p>Statement introduced before Junos OS Release 7.4.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 12.1 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.</p>
Description	Configure the type of authentication for RIP route queries received on an interface.
Default	If you do not include this statement and the authentication-key statement, RIP authentication is disabled.
Options	<p>type—Authentication type:</p> <ul style="list-style-type: none"> md5—Use the MD5 algorithm to create an encoded checksum of the packet. The encoded checksum is included in the transmitted packet. The receiving routing device uses the authentication key to verify the packet, discarding it if the digest does not match. This algorithm provides a more secure authentication scheme. none—Disable authentication. If none is configured, the configured authentication key is ignored. simple—Use a simple password. The password is included in the transmitted packet, which makes this method of authentication relatively insecure. The password can be from 1 through 16 contiguous letters or digits long.
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> Example: Configuring Route Authentication for RIP on page 15 authentication-key on page 89

bfd-liveness-detection (Protocols RIP)

Syntax	<pre> bfd-liveness-detection { authentication { algorithm <i>algorithm-name</i>; key-chain <i>key-chain-name</i>; loose-check; } detection-time { threshold <i>milliseconds</i>; } minimum-interval <i>milliseconds</i>; minimum-receive-interval <i>milliseconds</i>; multiplier <i>number</i>; no-adaptation; transmit-interval { minimum-interval <i>milliseconds</i>; threshold <i>milliseconds</i>; } version (1 automatic); }</pre>
Hierarchy Level	<p>[edit logical-systems <i>logical-system-name</i> protocols rip group <i>group-name</i>], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>], [edit protocols rip group <i>group-name</i>], [edit routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>]</p>
Release Information	<p>Statement introduced in Junos OS Release 8.0.</p> <p>Options detection-time threshold and transmit-interval threshold introduced in Junos OS Release 8.2.</p> <p>Support for logical systems introduced in Junos OS Release 8.3.</p> <p>Option no-adaptation introduced in Junos OS Release 9.0.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Options authentication algorithm, authentication key-chain, and authentication loose-check introduced in Junos OS Release 9.6.</p> <p>Options authentication algorithm, authentication key-chain, and authentication loose-check introduced in Junos OS Release 9.6 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 12.1 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.</p>
Description	Configure bidirectional failure detection timers and authentication.
Options	<p>authentication algorithm <i>algorithm-name</i> —Configure the algorithm used to authenticate the specified BFD session: simple-password, keyed-md5, keyed-sha-1, meticulous-keyed-md5, or meticulous-keyed-sha-1.</p> <p>authentication key-chain <i>key-chain-name</i> —Associate a security key with the specified BFD session using the name of the security keychain. The name you specify must</p>

match one of the keychains configured in the **authentication-key-chains key-chain** statement at the **[edit security]** hierarchy level.

authentication loose-check—(Optional) Configure loose authentication checking on the BFD session. Use only for transitional periods when authentication is not configured at both ends of the BFD session.

detection-time threshold *milliseconds*—Configure a threshold for the adaptation of the BFD session detection time. When the detection time adapts to a value equal to or greater than the threshold, a single trap and a single system log message are sent.

minimum-interval *milliseconds*—Configure the minimum interval after which the local routing device transmits a hello packet and then expects to receive a reply from the neighbor with which it has established a BFD session. Optionally, instead of using this statement, you can specify the minimum transmit and receive intervals separately using the **transmit-interval minimum-interval** and **minimum-receive-interval** statements.

Range: 1 through 255,000 milliseconds

minimum-receive-interval *milliseconds*—Configure the minimum interval after which the local routing device expects to receive a reply from a neighbor with which it has established a BFD session. Optionally, instead of using this statement, you can configure the minimum receive interval using the **minimum-interval** statement.

Range: 1 through 255,000 milliseconds

multiplier *number*—Configure the number of hello packets not received by a neighbor that causes the originating interface to be declared down.

Range: 1 through 255

Default: 3

no-adaptation—Configure BFD sessions not to adapt to changing network conditions. We recommend that you not disable BFD adaptation unless it is preferable not to have BFD adaptation enabled in your network.

transmit-interval threshold *milliseconds*—Configure the threshold for the adaptation of the BFD session transmit interval. When the transmit interval adapts to a value greater than the threshold, a single trap and a single system message are sent. The interval threshold must be greater than the minimum transmit interval.

Range: 0 through 4,294,967,295 ($2^{32} - 1$)

transmit-interval minimum-interval *milliseconds*—Configure a minimum interval after which the local routing device transmits hello packets to a neighbor. Optionally, instead of using this statement, you can configure the minimum transmit interval using the **minimum-interval** statement.

Range: 1 through 255,000

version—Configure the BFD version to detect: **1** (BFD version 1) or **automatic** (autodetect the BFD version).


Default: automatic

Required Privilege	routing—To view this statement in the configuration.
Level	routing-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Example: Configuring BFD for RIP on page 24• Example: Configuring BFD Authentication for RIP on page 33

check-zero

Syntax	(check-zero no-check-zero);
Hierarchy Level	<p>[edit logical-systems <i>logical-system-name</i> protocols rip],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit protocols rip],</p> <p>[edit protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>]</p>
Release Information	<p>Statement introduced before Junos OS Release 7.4.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 12.1 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.</p>
Description	<p>Some of the reserved fields in RIP version 1 packets must be zero, whereas in RIP version 2 packets, most of these reserved fields can contain nonzero values. By default, RIP discards version 1 packets that have nonzero values in the reserved fields and version 2 packets that have nonzero values in the fields that must be zero. This default behavior implements the RIP version 1 and version 2 specifications.</p> <p>If you find that you are receiving RIP version 1 packets with nonzero values in the reserved fields or RIP version 2 packets with nonzero values in the fields that must be zero, you can configure RIP to receive these packets even though they are being sent in violation of the specifications in RFC 1058 and RFC 2453.</p> <p>Check whether the reserved fields in a RIP packet are zero:</p> <ul style="list-style-type: none"> • check-zero—Discard version 1 packets that have nonzero values in the reserved fields and version 2 packets that have nonzero values in the fields that must be zero. This default behavior implements the RIP version 1 and version 2 specifications. • no-check-zero—Receive RIP version 1 packets with nonzero values in the reserved fields or RIP version 2 packets with nonzero values in the fields that must be zero. This is in spite of the fact that they are being sent in violation of the specifications in RFC 1058 and RFC 2453.
Default	check-zero
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • Example: Configuring RIP on page 7

export (Protocols RIP)

Syntax	<code>export [<i>policy-names</i>];</code>
Hierarchy Level	<p>[edit logical-systems <i>logical-system-name</i> protocols rip group <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i>],</p> <p>[edit protocols rip group <i>group-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i>]</p>
Release Information	<p>Statement introduced before Junos OS Release 7.4.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p>
Description	<p>Apply a policy to routes being exported to the neighbors.</p> <p>By default, RIP does not export routes it has learned to its neighbors. To enable RIP to export routes, apply one or more export policies.</p> <p>If no routes match the policies, the local routing device does not export any routes to its neighbors. Export policies override any metric values determined through calculations involving the values configured with the metric-in and metric-out statements.</p>
<div>  NOTE: The export policy on RIP does not support manipulating routing information of the next hop. </div>	
Options	<i>policy-names</i> —Name of one or more policies.
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • Example: Configuring RIP on page 7 • import on page 99

group (Protocols RIP)

Syntax `group group-name {`
 `bfd-liveness-detection {`
 `authentication {`
 `algorithm algorithm-name;`
 `key-chain key-chain-name;`
 `loose-check;`
 `}`
 `detection-time {`
 `threshold milliseconds;`
 `}`
 `minimum-interval milliseconds;`
 `minimum-receive-interval milliseconds;`
 `transmit-interval {`
 `threshold milliseconds;`
 `minimum-interval milliseconds;`
 `}`
 `multiplier number;`
 `version (0 | 1 | automatic);`
 `}`
 `demand-circuit;`
 `export policy;`
 `max-retrans-time seconds;`
 `metric-out metric;`
 `preference number;`
 `route-timeout seconds;`
 `update-interval seconds;`
 `neighbor neighbor-name {`
 `authentication-key password;`
 `authentication-type type;`
 `bfd-liveness-detection {`
 `authentication {`
 `algorithm algorithm-name;`
 `key-chain key-chain-name;`
 `loose-check;`
 `}`
 `detection-time {`
 `threshold milliseconds;`
 `}`
 `minimum-interval milliseconds;`
 `minimum-receive-interval milliseconds;`
 `transmit-interval {`
 `threshold milliseconds;`
 `minimum-interval milliseconds;`
 `}`
 `multiplier number;`
 `version (0 | 1 | automatic);`
 `}`
 `(check-zero | no-check-zero);`
 `demand-circuit;`
 `import policy-name;`
 `max-retrans-time seconds;`
 `message-size number;`


```

metric-in metric;
metric-out metric;
receive receive-options;
route-timeout seconds;
send send-options;
update-interval seconds;
}
}

```

Hierarchy Level	<p>[edit logical-systems <i>logical-system-name</i> protocols rip],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip],</p> <p>[edit protocols rip],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip]</p>
Release Information	<p>Statement introduced before Junos OS Release 7.4.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 12.1 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.</p>
Description	<p>Configure a set of RIP neighbors that share an export policy and metric. The export policy and metric govern what routes to advertise to neighbors in a given group. Each group must contain at least one neighbor. You should create a group for every export policy.</p>
Options	<p><i>group-name</i>—Name of a group, up to 16 characters long.</p> <p>The remaining statements are explained separately.</p>
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • Example: Configuring RIP on page 7


holddown (Protocols RIP)

Syntax	<code>holddown seconds;</code>
Hierarchy Level	[edit logical-systems <i>logical-system-name</i> protocols rip], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip], [edit protocols rip], [edit routing-instances <i>routing-instance-name</i> protocols rip]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 12.1 for the QFX Series. Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.
Description	<p>Configure how long the expired route is retained in the routing table before being removed.</p> <p>When the hold-down timer runs on RIP demand circuits, routes are advertised as unreachable on other interfaces. When the hold-down timer expires, the route is removed from the routing table if all destinations detect that the route is unreachable or the remaining destinations are down.</p>
Options	seconds —Estimated time to wait before making updates to the routing table. Range: 10 through 180 seconds Default: 120 seconds
Required Privilege Level	routing —To view this statement in the configuration. routing-control —To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Example: Configuring RIP Timers on page 72• RIP Demand Circuits Overview

import (Protocols RIP)

Syntax	<code>import [<i>policy-names</i>];</code>
Hierarchy Level	<p>[edit logical-systems <i>logical-system-name</i> protocols rip],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit protocols rip],</p> <p>[edit protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>]</p>
Release Information	<p>Statement introduced before Junos OS Release 7.4.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 12.1 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.</p>
Description	Apply one or more policies to routes being imported by the local routing device from neighbors.
Options	<i>policy-names</i> —Name of one or more policies.
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • Example: Applying Policies to RIP Routes Imported from Neighbors on page 41 • Routing Policies, Firewall Filters, and Traffic Policers Feature Guide for Routing Devices • export on page 95

message-size

Syntax	<code>message-size <i>number</i>;</code>
Hierarchy Level	<p>[edit logical-systems <i>logical-system-name</i> protocols rip],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit protocols rip],</p> <p>[edit protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>]</p>
Release Information	<p>Statement introduced before Junos OS Release 7.4.</p> <p>Statement for SRX Series devices introduced in Junos OS Release 9.5.</p> <p>Statement for J Series platform introduced in Junos OS Release 8.5.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 12.1 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.</p>
Description	Specify the number of route entries to be included in every RIP update message.
<div>  <p>TIP: To ensure interoperability with other vendors' equipment, use the standard of 25 route entries per message. Do not change the default number of route entries in a RIP update message.</p> </div>	
Options	<p><i>number</i>—Number of route entries per update message.</p> <p>Range: 25 through 255 entries</p> <p>Default: 25 entries</p>
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> Example: Configuring RIP on page 7

metric-in (Protocols RIP)

Syntax	<code>metric-in <i>metric</i>;</code>
Hierarchy Level	<p>[edit logical-systems <i>logical-system-name</i> protocols rip],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit protocols rip],</p> <p>[edit protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>]</p>
Release Information	<p>Statement introduced before Junos OS Release 7.4.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 12.1 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.</p>
Description	Specify the metric to add to incoming routes when the routing device advertises into RIP routes that were learned from other protocols. Use this statement to configure the routing device to prefer RIP routes learned through a specific neighbor.
Options	<p><i>metric</i>—Metric value.</p> <p>Range: 1 through 16</p> <p>Default: 1</p>
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • Example: Configuring the Metric Value Added to Imported RIP Routes on page 52

metric-out (Protocols RIP)

Syntax	<code>metric-out <i>metric</i>;</code>
Hierarchy Level	<code>[edit logical-systems <i>logical-system-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</code> <code>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</code> <code>[edit protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</code> <code>[edit routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>]</code>
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches.
Description	<p>Specify the metric value to add to routes transmitted to the neighbor. Use this statement to control how other routing devices prefer RIP routes sent from this neighbor.</p> <p>If you have included the export statement, RIP exports routes it has learned to the neighbors configured by including the neighbor statement.</p> <p>The metric associated with a RIP route (unless modified by an export policy) is the normal RIP metric. For example, a RIP route with a metric of 5 learned from a neighbor configured with a metric-in value of 2 is advertised with a combined metric of 7 when advertised to RIP neighbors in the same group. However, if this route was learned from a RIP neighbor in a different group or from a different protocol, the route is advertised with the metric value configured for that group with the metric-out statement.</p> <p>The metric for a route can be modified with an export policy. That metric is seen when the route is exported to the next hop.</p> <p>To increase the metric for routes advertised outside a group, include the metric-out statement.</p>
Options	<i>metric</i> —Metric value. Range: 1 through 16 Default: 1
Required Privilege Level	routing —To view this statement in the configuration. routing-control —To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Examples: Controlling Traffic with Metrics in a RIP Network on page 4750

neighbor (Protocols RIP)

Syntax	<pre> neighbor <i>neighbor-name</i> { authentication-key <i>password</i>; authentication-type <i>type</i>; bfd-liveness-detection { authentication { algorithm <i>algorithm-name</i>; key-chain <i>key-chain-name</i>; loose-check; } detection-time { threshold <i>milliseconds</i>; } minimum-interval <i>milliseconds</i>; minimum-receive-interval <i>milliseconds</i>; transmit-interval { threshold <i>milliseconds</i>; minimum-interval <i>milliseconds</i>; } multiplier <i>number</i>; version (0 1 automatic); } (check-zero no-check-zero); demand-circuit; import <i>policy-name</i>; max-retrans-time <i>seconds</i>; message-size <i>number</i>; metric-in <i>metric</i>; metric-out <i>metric</i>; receive <i>receive-options</i>; route-timeout <i>seconds</i>; send <i>send-options</i>; update-interval <i>seconds</i>; } </pre>
Hierarchy Level	<pre> [edit logical-systems <i>logical-system-name</i> protocols rip group <i>group-name</i>], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i>], [edit protocols rip group <i>group-name</i>], [edit routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i>] </pre>
Release Information	<p>Statement introduced before Junos OS Release 7.4.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p>
Description	Configure neighbor-specific RIP parameters, thereby overriding the defaults set for the routing device.
Options	<p><i>neighbor-name</i>—Name of an interface over which a routing device communicates to its neighbors.</p> <p>The remaining statements are explained separately.</p>

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- [Example: Configuring RIP on page 7](#)

preference (Protocols RIP)

Syntax `preference preference;`

Hierarchy Level [edit logical-systems *logical-system-name* protocols rip **group** *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols
rip **group** *group-name*],
[edit protocols rip **group** *group-name*],
[edit routing-instances *routing-instance-name* protocols rip **group** *group-name*]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.

Description Specify the preference of external routes learned by RIP as compared to those learned from other routing protocols.

By default, Junos OS assigns a preference of 100 to routes that originate from RIP. When Junos OS determines a route's preference to become the active route, the software selects the route with the lowest preference and installs this route into the forwarding table.

Options *preference*—Preference value. A lower value indicates a more preferred route.
Range: 0 through 4,294,967,295 ($2^{32} - 1$)
Default: 100

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- [Route Preferences Overview](#)

receive (Protocols RIP)

Syntax	<code>receive receive-options;</code>
Hierarchy Level	<p>[edit logical-systems <i>logical-system-name</i> protocols rip],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols rip group <i>group-name</i> neighbor neighbor-name],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor neighbor-name],</p> <p>[edit protocols rip],</p> <p>[edit protocols rip group <i>group-name</i> neighbor neighbor-name],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor neighbor-name]</p>
Release Information	<p>Statement introduced before Junos OS Release 7.4.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 12.1 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.</p>
Description	Configure RIP receive options.
Options	<p><i>receive-options</i>—One of the following:</p> <ul style="list-style-type: none"> • both—Accept both RIP version 1 and version 2 packets. • none—Do not receive RIP packets. • version-1—Accept only RIP version 1 packets. • version-2—Accept only RIP version 2 packets. <p>Default: both</p>
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • Example: Configuring the Sending and Receiving of RIPv1 and RIPv2 Packets on page 57 • send on page 108

rib-group (Protocols RIP)

Syntax	<code>rib-group group-name;</code>
Hierarchy Level	[edit logical-systems <i>logical-system-name</i> protocols rip], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip], [edit protocols rip], [edit routing-instances <i>routing-instance-name</i> protocols rip]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 12.1 for the QFX Series. Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.
Description	Install RIP routes into multiple routing tables by configuring a routing table group.
Options	<i>group-name</i> —Name of the routing table group.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Example: Redistributing Routes Between Two RIP Instances on page 64

rip

Syntax	<code>rip {...}</code>
Hierarchy Level	[edit logical-systems <i>logical-system-name</i> protocols], [edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols], [edit protocols], [edit routing-instances <i>routing-instance-name</i> protocols]
Release Information	Statement introduced before Junos OS Release 7.4. Statement introduced in Junos OS Release 9.0 for EX Series switches. Statement introduced in Junos OS Release 12.1 for the QFX Series. Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.
Description	Enable RIP routing on the routing device.
Default	RIP is disabled on the routing device.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Example: Configuring RIP on page 7

route-timeout (Protocols RIP)

Syntax	<code>route-timeout seconds;</code>
Hierarchy Level	<p>[edit logical-systems <i>logical-system-name</i> protocols rip],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols rip group <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit protocols rip],</p> <p>[edit protocols rip group <i>group-name</i>],</p> <p>[edit protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>]</p>
Release Information	<p>Statement introduced in Junos OS Release 7.6.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 12.1 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.</p>
Description	Configure the route timeout interval for RIP. If a route is not refreshed after being installed in the routing table by the specified timeout interval, the route is marked as invalid and is removed from the routing table after the hold-down period expires.
Options	<p>seconds—Estimated time to wait before making updates to the routing table.</p> <p>Range: 30 through 360 seconds</p> <p>Default: 180 seconds</p>
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • Example: Configuring RIP Timers on page 72 • RIP Demand Circuits Overview

send (Protocols RIP)

Syntax	<code>send <i>send-options</i>;</code>
Hierarchy Level	<p>[edit logical-systems <i>logical-system-name</i> protocols rip],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit protocols rip],</p> <p>[edit protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>]</p>
Release Information	<p>Statement introduced before Junos OS Release 7.4.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 12.1 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.</p>
Description	Configure RIP send options.
Options	<p><i>send-options</i>—One of the following:</p> <ul style="list-style-type: none"> • broadcast—Broadcast RIP version 2 packets (RIP version 1 compatible). • multicast—Multicast RIP version 2 packets. This is the default. • none—Do not send RIP updates. • version-1—Broadcast RIP version 1 packets. <p>Default: multicast</p>
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • Example: Configuring the Sending and Receiving of RIPv1 and RIPv2 Packets on page 57 • receive on page 105

traceoptions (Protocols RIP)

Syntax	<pre>traceoptions { file <i>filename</i> <files <i>number</i>> <size <i>size</i>> <world-readable no-world-readable>; flag <i>flag</i> <flag-modifier> <disable>; }</pre>
Hierarchy Level	<p>[edit logical-systems <i>logical-system-name</i> protocols rip],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip],</p> <p>[edit protocols rip],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip]</p>
Release Information	<p>Statement introduced before Junos OS Release 7.4.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 12.1 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.</p>
Description	Set RIP protocol-level tracing options.



NOTE: The `traceoptions` statement is not supported on QFabric systems.

Default The default RIP protocol-level trace options are inherited from the global `traceoptions` statement.

Options **disable**—(Optional) Disable the tracing operation. One use of this option is to disable a single operation when you have defined a broad group of tracing operations, such as **all**.

file *filename*—Name of the file to receive the output of the tracing operation. Enclose the name in quotation marks. We recommend that you place RIP tracing output in the file `/var/log/rip-log`.

files *number*—(Optional) Maximum number of trace files. When a trace file named ***trace-file*** reaches its maximum size, it is renamed ***trace-file.0***, then ***trace-file.1***, and so on, until the maximum number of trace files is reached. Then, the oldest trace file is overwritten. If you specify a maximum number of files, you must also specify a maximum file size with the **size** option.

Range: 2 through 1000 files

Default: 10 files

flag *flag*—Tracing operation to perform. To specify more than one tracing operation, include multiple **flag** statements.

RIP Tracing Options

- **auth**—RIP authentication

- **error**—RIP error packets
- **expiration**—RIP route expiration processing
- **holddown**—RIP hold-down processing
- **nsr-synchronization**—Nonstop routing synchronization events
- **packets**—All RIP packets
- **request**—RIP information packets such as request, poll, and poll entry packets
- **trigger**—RIP triggered updates
- **update**—RIP update packets

Global Tracing Options

- **all**—All tracing operations
- **general**—A combination of the **normal** and **route** trace operations
- **normal**—All normal operations

Default: If you do not specify this option, only unusual or abnormal operations are traced.

- **policy**—Policy operations and actions
- **route**—Routing table changes
- **state**—State transitions
- **task**—Routing protocol task processing
- **timer**—Routing protocol timer processing

flag-modifier—(Optional) Modifier for the tracing flag. You can specify one or more of these modifiers:

- **detail**—Provide detailed trace information.
- **receive**—Trace the packets being received.
- **receive-detail**—Provide detailed trace information for packets being received.
- **send**—Trace the packets being transmitted.
- **send-detail**—Provide detailed trace information for packets being transmitted.

no-world-readable—(Optional) Prevent any user from reading the log file.

size *size*—(Optional) Maximum size of each trace file, in kilobytes (KB) or megabytes (MB). When a trace file named **trace-file** reaches this size, it is renamed **trace-file.0**. When the **trace-file** again reaches its maximum size, **trace-file.0** is renamed **trace-file.1** and **trace-file** is renamed **trace-file.0**. This renaming scheme continues until the maximum number of trace files is reached. Then, the oldest trace file is overwritten. If you specify a maximum file size, you must also specify a maximum number of trace files with the **files** option.

Syntax: *xk* to specify KB, *xm* to specify MB, or *xg* to specify GB

Range: 10 KB through the maximum file size supported on your system

Default: 128 KB

world-readable—(Optional) Allow any user to read the log file.

Required Privilege Level	routing—To view this statement in the configuration.
	routing-control—To add this statement to the configuration.
Related Documentation	<ul style="list-style-type: none">• Example: Tracing RIP Protocol Traffic on page 80

update-interval (Protocols RIP)

Syntax	<code>update-interval seconds;</code>
Hierarchy Level	<p>[edit logical-systems <i>logical-system-name</i> protocols rip],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols rip group <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i>],</p> <p>[edit logical-systems <i>logical-system-name</i> routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit protocols rip],</p> <p>[edit protocols rip group <i>group-name</i>],</p> <p>[edit protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i>],</p> <p>[edit routing-instances <i>routing-instance-name</i> protocols rip group <i>group-name</i> neighbor <i>neighbor-name</i>]</p>
Release Information	<p>Statement introduced in Junos OS Release 7.6.</p> <p>Statement introduced in Junos OS Release 9.0 for EX Series switches.</p> <p>Statement introduced in Junos OS Release 12.1 for the QFX Series.</p> <p>Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.</p>
Description	Configure the interval at which routes learned by RIP are sent to neighbors. This timer controls the interval between routing updates. This timer is set to 30 seconds, by default, with a small random amount of time added when the timer is reset. This added time prevents congestion that can happen if all routing devices update their neighbors simultaneously.
Options	<p>seconds—Estimated time to wait before making updates to the routing table.</p> <p>Range: 10 through 60 seconds</p> <p>Default: 30 seconds</p>
Required Privilege Level	<p>routing—To view this statement in the configuration.</p> <p>routing-control—To add this statement to the configuration.</p>
Related Documentation	<ul style="list-style-type: none"> • Example: Configuring RIP Timers on page 72

CHAPTER 12

RIP Operational Commands

- `clear rip general-statistics`
- `clear rip statistics`
- `show rip general-statistics`
- `show rip neighbor`
- `show rip statistics`

clear rip general-statistics

List of Syntax	Syntax on page 114 Syntax (EX Series Switches and QFX Series) on page 114
Syntax	clear rip general-statistics <logical-system (all <i>logical-system-name</i>)>
Syntax (EX Series Switches and QFX Series)	clear rip general-statistics
Release Information	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches. Command introduced in Junos OS Release 12.1 for the QFX Series. Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.
Description	Clear RIP general statistics.
Options	none —Clear RIP general statistics. logical-system (all <i>logical-system-name</i>) —(Optional) Perform this operation on all logical systems or on a particular logical system.
Required Privilege Level	clear
Related Documentation	<ul style="list-style-type: none">• show rip general-statistics on page 116
List of Sample Output	clear rip general-statistics on page 114
Output Fields	When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear rip general-statistics

```
user@host> clear rip general-statistics
```

clear rip statistics

List of Syntax	Syntax on page 115 Syntax (EX Series Switches and QFX Series) on page 115
Syntax	clear rip statistics <instance (all <i>instance-name</i>)> <logical-system (all <i>logical-system-name</i>)> <neighbor> <peer (all <i>address</i>)>
Syntax (EX Series Switches and QFX Series)	clear rip statistics <instance (all <i>instance-name</i>)> <neighbor>
Release Information	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches. Command introduced in Junos OS Release 12.1 for the QFX Series. Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.
Description	Clear RIP statistics.
Options	<p>none—Reset RIP counters for all neighbors for all routing instances.</p> <p>instance (all <i>instance-name</i>)—(Optional) Clear RIP statistics for all instances or for the specified routing instance only.</p> <p>logical-system (all <i>logical-system-name</i>)—(Optional) Perform this operation on all logical systems or on a particular logical system.</p> <p>neighbor—(Optional) Clear RIP statistics for the specified neighbor only.</p> <p>peer (all <i>address</i>)—(Optional) Clear RIP statistics for a single peer or all peers.</p>
Required Privilege Level	clear
Related Documentation	<ul style="list-style-type: none"> • show rip statistics on page 120
List of Sample Output	clear rip statistics on page 115
Output Fields	When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear rip statistics

```
user@host> clear rip statistics
```

show rip general-statistics

List of Syntax	Syntax on page 116 Syntax (EX Series Switches and QFX Series) on page 116
Syntax	show rip general-statistics <logical-system (all <i>logical-system-name</i>)>
Syntax (EX Series Switches and QFX Series)	show rip general-statistics
Release Information	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches. Command introduced in Junos OS Release 12.1 for the QFX Series. Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.
Description	Display brief RIP statistics.
Options	none —Display brief RIP statistics. logical-system (all <i>logical-system-name</i>) —(Optional) Perform this operation on all logical systems or on a particular logical system.
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none"> • clear rip general-statistics on page 114
List of Sample Output	show rip general-statistics on page 116
Output Fields	Table 5 on page 116 lists the output fields for the show rip general-statistics command. Output fields are listed in the approximate order in which they appear.

Table 5: show rip general-statistics Output Fields

Field Name	Field Description
bad msgs	Number of invalid messages received.
no rcv intf	Number of packets received with no matching interface.
curr memory	Amount of memory currently used by RIP.
max memory	Most memory used by RIP.

Sample Output

show rip general-statistics

```
user@host> show rip general-statistics
```

```
RIPv2 I/O info:
  bad msgs      :      0
  no recv intf  :      0
  curr memory    :      0
  max memory     :      0
```

show rip neighbor

List of Syntax	Syntax on page 118 Syntax (EX Series Switches and QFX Series) on page 118
Syntax	<pre>show rip neighbor <instance (all <i>instance-name</i>)> <logical-system (all <i>logical-system-name</i>)> <name></pre>
Syntax (EX Series Switches and QFX Series)	<pre>show rip neighbor <instance (all <i>instance-name</i>)> <name></pre>
Release Information	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches. Command introduced in Junos OS Release 12.1 for the QFX Series. Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.
Description	Display information about RIP neighbors.
Options	<p>none—Display information about all RIP neighbors for all instances.</p> <p>instance (all <i>instance-name</i>)—(Optional) Display RIP neighbor information for all instances or for only the specified routing instance.</p> <p>logical-system (all <i>logical-system-name</i>)—(Optional) Perform this operation on all logical systems or on a particular logical system.</p> <p><i>name</i>—(Optional) Display detailed information about only the specified RIP neighbor.</p>
Required Privilege Level	view
List of Sample Output	show rip neighbor on page 119 show rip neighbor (With Demand Circuits Configured) on page 119
Output Fields	Table 6 on page 119 lists the output fields for the show rip neighbor command. Output fields are listed in the approximate order in which they appear.

Table 6: show rip neighbor Output Fields

Field Name	Field Description
Neighbor	<p>Name of the RIP neighbor.</p> <p>NOTE: Beginning with Junos OS Release 11.1, when you configure demand circuits, the output displays a demand circuit (DC) flag next to neighbor interfaces configured for demand circuits.</p> <p>If you configure demand circuits at the [edit protocols rip group group-name neighbor neighbor-name] hierarchy level, the output shows only the neighboring interface that you specifically configured as a demand circuit. If you configure demand circuits at the [edit protocols rip group group-name] hierarchy level, all of the interfaces in the group are configured as demand circuits. Therefore, the output shows all of the interfaces in that group as demand circuits.</p>
State	State of the connection: Up or Dn (Down).
Source Address	Address of the port on the local router.
Destination Address	Address of the port on the remote router.
Send Mode	Send options: broadcast , multicast , none , or version 1 .
Receive Mode	Type of packets to accept: both , none , version 1 , or version 2 .
In Met	Metric added to incoming routes when advertising into RIP routes that were learned from other protocols.

Sample Output

show rip neighbor

```

user@host> show rip neighbor
Neighbor      Local  Source      Destination  Send  Receive  In
-----      -
ge-2/3/0.0    Up    192.168.9.105  192.168.9.107  bcast  both      1
at-5/1/1.42    Dn    (null)        (null)        mcast  v2 only   3
at-5/1/0.42    Dn    (null)        (null)        mcast  both      3
at-5/1/0.0     Up    20.0.0.1      224.0.0.9     mcast  both      3
so-0/0/0.0     Up    192.168.9.97  224.0.0.9     mcast  both      3

```

show rip neighbor (With Demand Circuits Configured)

```

user@host> show rip neighbor
Neighbor      Local  Source      Destination  Send  Receive  In
-----      -
so-0/1/0.0(DC) Up    10.10.10.2   224.0.0.9     mcast  both      1
so-0/2/0.0(DC) Up    13.13.13.2   224.0.0.9     mcast  both      1

```

show rip statistics

List of Syntax	Syntax on page 120 Syntax (EX Series Switches and QFX Series) on page 120
Syntax	<pre>show rip statistics <instance (all <i>instance-name</i>)> <logical-system (all <i>logical-system-name</i>)> <<i>name</i>> <peer (all <i>address</i>)></pre>
Syntax (EX Series Switches and QFX Series)	<pre>show rip statistics <instance (all <i>instance-name</i>)> <<i>name</i>></pre>
Release Information	Command introduced before Junos OS Release 7.4. Command introduced in Junos OS Release 9.0 for EX Series switches. Command introduced in Junos OS Release 12.1 for the QFX Series. Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.
Description	Display RIP statistics about messages sent and received on an interface, as well as information received from advertisements from other routing devices.
Options	<p>none—Display RIP statistics for all routing instances.</p> <p>instance (all <i>instance-name</i>)—(Optional) Display RIP statistics for all instances or for only the specified routing instance.</p> <p>logical-system (all <i>logical-system-name</i>)—(Optional) Perform this operation on all logical systems or on a particular logical system.</p> <p><i>name</i>—(Optional) Display detailed information about only the specified RIP neighbor.</p> <p>peer (all <i>address</i>)—(Optional) Display RIP statistics for a single peer or all peers.</p>
Required Privilege Level	view
Related Documentation	<ul style="list-style-type: none">• clear rip statistics on page 115
List of Sample Output	show rip statistics on page 121
Output Fields	Table 7 on page 121 lists the output fields for the show rip statistics command. Output fields are listed in the approximate order in which they appear.

Table 7: show rip statistics Output Fields

Field Name	Field Description
RIP info	<p>Information about RIP on the specified interface:</p> <ul style="list-style-type: none"> • port—UDP port number used for RIP. • update interval—Interval between routing table updates, in seconds. • holddown—Hold-down interval, in seconds. • timeout—Timeout interval, in seconds. • restart in progress—Graceful restart status. Displayed when RIP is or has been in the process of graceful restart. • restart time—Estimated time for the graceful restart to finish, in seconds. • restart will complete in—Remaining time for the graceful restart to finish, in seconds. • rts learned—Number of routes learned through RIP. • rts held down—Number of routes held down by RIP. • rqsts dropped—Number of received request packets that were dropped. • resps dropped—Number of received response packets that were dropped.
logical-interface	<p>Name of the logical interface and its statistics:</p> <ul style="list-style-type: none"> • routes learned—Number of routes learned on the logical interface. • routes advertised—Number of routes advertised by the logical interface.
Counter	<p>List of counter types:</p> <ul style="list-style-type: none"> • Updates Sent—Number of update messages sent. • Triggered Updates Sent—Number of triggered update messages sent. • Responses Sent—Number of response messages sent. • Bad Messages—Number of invalid messages received. • RIPv1 Updates Received—Number of RIPv1 update messages received. • RIPv1 Bad Route Entries—Number of RIPv1 invalid route entry messages received. • RIPv1 Updates Ignored—Number of RIPv1 update messages ignored. • RIPv2 Updates Received—Number of RIPv2 update messages received. • RIPv2 Bad Route Entries—Number of RIPv2 invalid route entry messages received. • RIPv2 Updates Ignored—Number of RIPv2 update messages ignored. • Authentication Failures—Number of received update messages that failed authentication. • RIP Requests Received—Number of RIP request messages received. • RIP Requests Ignored—Number of RIP request messages ignored.
Total	Total number of packets for the selected counter.
Last 5 min	Number of packets for the selected counter in the most recent 5-minute period.
Last minute	Number of packets for the selected counter in the most recent 1-minute period.

Sample Output

show rip statistics

```
user@host> show rip statistics so-0/0/0.0
```

RIP info: port 520; update interval: 30s; holddown 180s; timeout 120s
restart in progress: restart time 60s; restart will complete in 55s
 rts learned rts held down rqsts dropped resps dropped
 0 0 0 0

so-0/0/0.0: 0 routes learned; 501 routes advertised

Counter	Total	Last 5 min	Last minute
-----	-----	-----	-----
Updates Sent	0	0	0
Triggered Updates Sent	0	0	0
Responses Sent	0	0	0
Bad Messages	0	0	0
RIPv1 Updates Received	0	0	0
RIPv1 Bad Route Entries	0	0	0
RIPv1 Updates Ignored	0	0	0
RIPv2 Updates Received	0	0	0
RIPv2 Bad Route Entries	0	0	0
RIPv2 Updates Ignored	0	0	0
Authentication Failures	0	0	0
RIP Requests Received	0	0	0
RIP Requests Ignored	0	0	0