VYATTA, INC. | Vyatta System

RIP

REFERENCE GUIDE

RIP

Vyatta
Suite 200
1301 Shoreway Road
Belmont, CA 94002
vyatta.com
650 413 7200
1 888 VYATTA 1 (US and Canada)

COPYRIGHT

PROPRIETARY NOTICES

RELEASE DATE: February 2011

DOCUMENT REVISION. R6.2 v01

RELEASED WITH: R6.2.0

PART NO. A0-0218-10-0010

Table of Contents

Quick Reference to Commands

Use this section to help you quickly locate a command.

Quick List of Examples

Use this list to help you locate examples you'd like to try or look at.

Preface

This document describes the various deployment, installation, and upgrade options for Vyatta software.

This preface provides information about using this guide. The following topics are presented:

Intended Audience

This guide is intended for experienced system and network administrators. Depending on the functionality to be used, readers should have specific knowledge in the following areas:

- Networking and data communications
- TCP/IP protocols
- General router configuration
- Routing protocols
- Network administration
- Network security
- IP services

Organization of This Guide

This guide has the following aid to help you find the information you are looking for:

- Quick Reference to Commands

  Use this list to help you quickly locate commands.

- Quick List of Examples

  Use this list to help you locate examples you'd like to try or look at.

This guide has the following chapters:

Chapter	Description	Page
Chapter 1: RIP Configuration	This chapter describes how to configure the Routing Information Protocol on the Vyatta System.	1
Chapter 2: Router-Level Configuration	This chapter describes commands for configuring RIP at the router level.	7
Chapter 3: Route Redistribution	This chapter describes commands for redistributing routes from other routing protocols into RIP.	30
Chapter 4: Route Filtering	This chapter describes commands for RIP route filtering.	42

Document Conventions

This guide uses the following advisory paragraphs, as follows.

WARNING *Warnings alert you to situations that may pose a threat to personal safety.*

CAUTION *Cautions alert you to situations that might cause harm to your system or damage to equipment, or that may affect service.*

NOTE *Notes provide information you might need to avoid problems or configuration errors.*

This document uses the following typographic conventions.

`Monospace`	Examples, command-line output, and representations of configuration nodes.
`bold Monospace`	Your input: something you type at a command line.
bold	Commands, keywords, and file names, when mentioned inline. Objects in the user interface, such as tabs, buttons, screens, and panes.
italics	An argument or variable where you supply a value.
<key>	A key on your keyboard, such as <Enter>. Combinations of keys are joined by plus signs (“+”), as in <Ctrl>+c.
[key1 | key2]	Enumerated options for completing a syntax. An example is [enable | disable].
num1–numN	A inclusive range of numbers. An example is 1–65535, which means 1 through 65535, inclusive.

arg1..argN	A range of enumerated values. An example is eth0..eth3, which means eth0, eth1, eth2, or eth3.
arg[*arg*...] *arg*[,*arg*...]	A value that can optionally represent a list of elements (a space-separated list and a comma-separated list, respectively).

Vyatta Publications

Full product documentation is provided in the Vyatta technical library. To see what documentation is available for your release, see the *Guide to Vyatta Documentation*. This guide is posted with every release of Vyatta software and provides a great starting point for finding the information you need.

Additional information is available on www.vyatta.com and www.vyatta.org.

Chapter 1: RIP Configuration

This chapter describes how to configure the Routing Information Protocol on the Vyatta System.

The following topics are covered:

RIP Overview

Routing Information Protocol (RIP) is a dynamic routing protocol suitable for small, homogenous networks. It is classified as an interior gateway protocol (IGP) and employs the distance-vector routing algorithm. RIP determines the best path by counting the hops to the destination. The maximum hop count is 15 (16 is considered an infinite distance), making RIP less suitable for large networks. RIP is considered obsoleted by OSPF.

Supported Standards

The Vyatta implementation of RIP complies with the following standards:

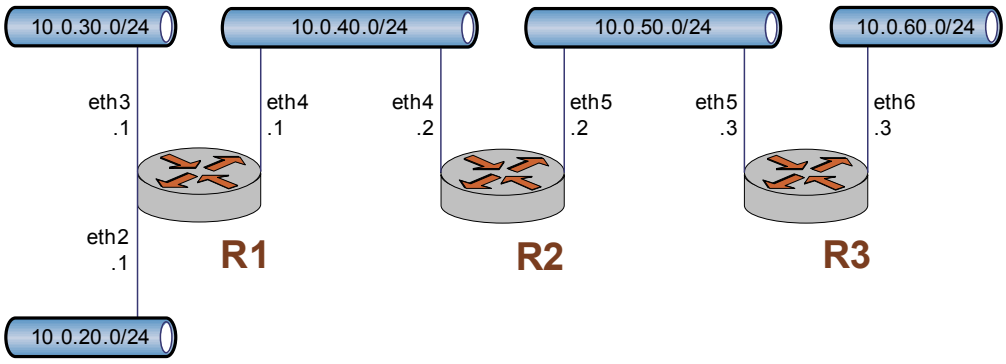
- RFC 1058: Routing Information Protocol
- RFC 2453: RIP Version 2

Configuring RIP

This section presents the following topics:

This section presents a sample configuration for RIP. The configuration example is based on the reference diagram in Figure 1-1.

Figure 1-1 RIP configuration reference diagram



Basic RIP Configuration

In this section, you configure RIP on the routers labeled R1, R2, and R3 in the reference network diagram. The routers are advertising their routes on networks 10.0.40.0/24 and 10.0.50.0/24.

This example assumes that you have already configured the router interfaces; only the steps required to implement RIP are shown.

To create a basic RIP configuration, perform the following steps in configuration mode:

Example 1-1 Basic RIP configuration

Router	Step	Command(s)
R1	Advertise to network 10.0.40.0/24.	`vyatta@R1# set protocols rip network 10.0.40.0/24`
R1	Redistribute connected routes to RIP.	`vyatta@R1# set protocols rip redistribute connected`
R1	Commit the configuration.	`vyatta@R1# commit`

R1	Display the configuration.	`vyatta@R1# show protocols`
		`rip {`
		`network 10.0.40.0/24`
		`redistribute {`
		`connected {`
		`}`
		`}`
		`}`

R2	Advertise to network 10.0.40.0/24.	`vyatta@R2# set protocols rip network 10.0.40.0/24`
R2	Advertise to network 10.0.50.0/24.	`vyatta@R2# set protocols rip network 10.0.50.0/24`
R2	Redistribute connected routes to RIP.	`vyatta@R2# set protocols rip redistribute connected`
R2	Commit the configuration.	`vyatta@R2# commit`

Example 1-1 Basic RIP configuration

R2	Display the configuration.	vyatta@R2# show protocols
		rip {
		network 10.0.40.0/24
		network 10.0.50.0/24
		redistribute {
		connected {
		}
		}
		}

R3	Advertise to network 10.0.50.0/24.	vyatta@R3# **set protocols rip network 10.0.50.0/24**
R3	Redistribute connected routes to RIP.	vyatta@R3# **set protocols rip redistribute connected**
R3	Commit the configuration.	vyatta@R3# commit

R3	Display the configuration.	vyatta@R3# show protocols
		rip {
		network 10.0.50.0/24
		redistribute {
		connected {
		}
		}
		}

Verifying the RIP Configuration

The following operational mode commands can be used to verify the RIP configuration.

R3: show ip route

Example 1-2 shows the output of the **show ip route** command for router R3.

Example 1-2 Verifying RIP on R3: ”show ip route”

```
vyatta@R3:~$ show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

R>* 10.0.20.0/24 [120/3] via 10.0.50.2, eth5, 00:20:16
```

```
R>* 10.0.30.0/24 [120/3] via 10.0.50.2, eth5, 00:34:04
R>* 10.0.40.0/24 [120/2] via 10.0.50.2, eth5, 02:15:26
C>* 10.0.50.0/24 is directly connected, eth5
C>* 10.0.60.0/24 is directly connected, eth6
C>* 127.0.0.0/8 is directly connected, lo
vyatta@R3:~$
```

The output shows that routes to 10.0.20.0/24, 10.0.30.0/24, and 10.0.40.0/24 have been learned via RIP and that packets to those networks will be forwarded out eth5 to 10.0.50.2. Networks 10.0.50.0/24 and 10.0.60.0/24 are directly connected.

R3: show ip rip

The **show ip rip** command for R3 displays similar information in a different format. This is shown in Example 1-3.

Example 1-3 Verifying RIP on R3: “show ip rip”

```
vyatta@R3:~$ show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
      (n) - normal, (s) - static, (d) - default, (r) - redistribute,
      (i) - interface

     Network            Next Hop         Metric From            Tag Time
R(n) 10.0.20.0/24       10.0.50.2             3 10.0.50.2         0 00:23
R(n) 10.0.30.0/24       10.0.50.2             3 10.0.50.2         0 00:23
R(n) 10.0.40.0/24       10.0.50.2             2 10.0.50.2         0 00:23
C(i) 10.0.50.0/24       0.0.0.0               1 self              0
C(r) 10.0.60.0/24       0.0.0.0               1 self (connected:1)  0
vyatta@R3:~$
```

Again, the output shows that networks 10.0.20.0/24, 10.0.30.0/24, and 10.0.40.0/24 have been learned via RIP and that packets to those networks will be forwarded to 10.0.50.2. Networks 10.0.50.0/24 and 10.0.60.0/24 are directly connected.

R3: ping 10.0.20.1

Using the **ping** command from R3 we can confirm that we can reach hosts on remote networks. In this case we ping an IP address on R1. This is shown in Example 1-4.

Example 1-4 Verifying RIP on R3: “ping 10.0.20.1”

```
vyatta@R3:~$ ping 10.0.20.1
PING 10.0.20.1 (10.0.20.1) 56(84) bytes of data.
```

```
64 bytes from 10.0.20.1: icmp_seq=1 ttl=63 time=7.39 ms
64 bytes from 10.0.20.1: icmp_seq=2 ttl=63 time=1.56 ms
64 bytes from 10.0.20.1: icmp_seq=3 ttl=63 time=1.49 ms
^C
--- 10.0.20.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 1.497/3.482/7.390/2.763 ms
vyatta@R3:~$
```

This confirms that the RIP configuration is working and that we are able to reach a remote network.

Chapter 2: Router-Level Configuration

This chapter describes commands for configuring RIP at the router level.

This chapter presents the following topics:

Router-Level Configuration Commands

This chapter contains the following commands.

Configuration Commands	
protocols rip default-distance <distance>	Sets the administrative distance for RIP.
protocols rip default-information originate	Generates a default route into a RIP routing domain.
protocols rip default-metric <metric>	Sets the default metric for external routes redistributed into RIP.
protocols rip interface <ethx>	Enables the Routing Information Protocol (RIP) for an interface.
protocols rip neighbor <ipv4>	Defines a RIP neighbor router.
protocols rip network <ipv4net>	Specifies a network for the Routing Information Protocol (RIP).
protocols rip network-distance <ipv4net>	Specifies the administrative distance for a RIP network.
protocols rip passive-interface <ethx>	Suppresses RIP routing updates on an interface.
protocols rip route <ipv4net>	Specifies a RIP static route.
protocols rip timers garbage-collection <seconds>	Allows you to set timers for RIP garbage collection.
protocols rip timers timeout <seconds>	Allows you to set the interval for RIP time-outs.
protocols rip timers update <seconds>	Allows you to set the timer for RIP routing table updates.
Operational Commands	
debug rip events	Enables or disables debug message generation related to RIP events.
debug rip packet	Enables or disables debug message generation related to all RIP packet types.
debug rip zebra	Enables or disables debug message generation for the Zebra RIP process.
show debugging rip	Displays RIP protocol debugging flags.
show ip route rip	Displays all IP RIP routes.
show ip rip	Displays information for the Routing Information Protocol (RIP).

debug rip events

Enables or disables debug message generation related to RIP events.

Syntax

debug rip events

no debug rip events

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to enable generation of trace-level messages related to Routing Information Protocol (RIP) events.

Use the **no** form of this command to disable debugging of RIP events.

debug rip packet

Enables or disables debug message generation related to all RIP packet types.

Syntax

debug rip packet [**recv** [**detail**] | **send** [**detail**]]

no debug rip packet [**recv** | **send**]

Command Mode

Operational mode.

Parameters

recv	Optional. Provides debugging on all received packets.
recv detail	Optional. Provides detailed debugging on all received packets.
send	Optional. Provides debugging on all sent packets.
send detail	Optional. Provides detailed debugging on all sent packets.

Default

None.

Usage Guidelines

Use this command to enable generation of trace-level messages related to all Routing Information Protocol (RIP) packet types.

Use the **no** form of this command to disable debugging of all RIP packet types.

debug rip zebra

Enables or disables debug message generation for the Zebra RIP process.

Syntax

debug rip zebra

no debug rip zebra

Command Mode

Operational mode.

Parameters

None.

Default

Debug messages are generated for actions related to the Zebra RIP process.

Usage Guidelines

Use this command to enable generation of trace-level messages related to the Zebra Routing Information Protocol (RIP) process.

Use the **no** form of this command to disable debugging for the Zebra RIP process.

protocols rip default-distance <distance>

Sets the administrative distance for RIP.

Syntax

set protocols rip default-distance *distance*

delete protocols rip default-distance

show protocols rip default-distance

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
    rip {
        default-distance distance
    }
}
```

Parameters

distance	Mandatory. Sets the default administrative distance for RIP. The range is 1-255. The default is 120.

Default

The default administrative distance for RIP is 120.

Usage Guidelines

Use the **set** form of this command to set the default administrative distance for RIP.

Use the **delete** form of this command to restore the default administrative distance for RIP.

Use the **show** form of this command to display the administrative distance for RIP.

protocols rip default-information originate

Generates a default route into a RIP routing domain.

Syntax

set protocols rip default-information originate

delete protocols rip default-information originate

show protocols rip default-information originate

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
    rip {
        default-information {
            originate
        }
    }
}
```

Parameters

None.

Default

By default, the system does not generate an external default route into the RIP routing domain.

Usage Guidelines

Use the **set** form of this command to generate a default route into the RIP routing domain.

Use the **delete** form of this command to restore the default behavior for default route generation into RIP.

Use the **show** form of this command to display default route generation configuration.

protocols rip default-metric <metric>

Sets the default metric for external routes redistributed into RIP.

Syntax

set protocols rip default-metric *metric*

delete protocols rip default-metric

show protocols rip default-metric

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
    rip {
        default-metric metric
    }
}
```

Parameters

metric	Mandatory. The metric that will be assigned to external routes imported into RIP for redistribution. The range is 1-16. The default is 1.

Default

Routes being imported into RIP are assigned a metric of 1.

Usage Guidelines

Use the **set** form of this command to set the metric for routes being redistributed into RIP.

Use the **delete** form of this command to restore the default RIP metric to default values.

Use the **show** form of this command to display the default metric for routes being redistributed into RIP.

protocols rip interface <ethx>

Enables the Routing Information Protocol (RIP) for an interface.

Syntax

set protocols rip interface *ethx*

delete protocols rip interface *ethx*

show protocols rip interface *ethx*

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
    rip {
        interface ethx
    }
}
```

Parameters

ethx	Mandatory. Multi-node. The name of a configured Ethernet interface. You can enable RIP on more than one interface by creating multiple **protocols rip interface** configuration nodes.

Default

None.

Usage Guidelines

Use the **set** form of this command to enable RIP on an interface. The interface must be enabled for RIP before you can use it for RIP routing.

Use the **delete** form of this command to disable RIP on an interface.

Use the **show** form of this command to display RIP interface configuration.

protocols rip neighbor <ipv4>

Defines a RIP neighbor router.

Syntax

set protocols rip neighbor *ipv4*

delete protocols rip neighbor *ipv4*

show protocols rip neighbor

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
    rip {
        neighbor ipv4
    }
}
```

Parameters

ipv4	Mandatory. Multi-node. The IP address of the neighbor router. You can define more than one RIP neighbor router by creating multiple **protocols rip neighbor** configuration nodes.

Default

None.

Usage Guidelines

Use the **set** form of this command to define a RIP neighbor router.

Use the **delete** form of this command to remove a neighbor router.

Use the **show** form of this command to display RIP neighbor configuration.

protocols rip network <ipv4net>

Specifies a network for the Routing Information Protocol (RIP).

Syntax

set protocols rip network *ipv4net*

delete protocols rip network *ipv4net*

show protocols rip network

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
    rip {
        network ipv4net
    }
}
```

Parameters

ipv4net	Mandatory. Multi-node. The IP network address of the RIP network. You can identify more than one RIP network by creating multiple **protocols rip network** configuration nodes.

Default

None.

Usage Guidelines

Use the **set** form of this command to specify a RIP network.

Use the **delete** form of this command to remove a RIP network.

Use the **show** form of this command to display RIP network configuration.

protocols rip network-distance <ipv4net>

Specifies the administrative distance for a RIP network.

Syntax

set protocols rip network-distance *ipv4net* {**access-list** *list-name* | **distance** *distance*}

delete protocols rip network-distance *ipv4net* [**access-list** *list-name* | **distance** *distance*]

show protocols rip network-distance *ipv4net* [**access-list | distance**]

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
    rip {
        network-distance ipv4net {
            access-list list-name
            distance distance
    }
}
```

Parameters

ipv4net	Mandatory. The IP network address identifying the network.
access-list	Applies a defined access list to the specified network.
distance	Applies the specified administrative distance to the specified network. The range is 1 to 255. The default is 120.

Default

None.

Usage Guidelines

Use the **set** form of this command to set the default administrative distance for a RIP network or apply an access list to a RIP network.

The administrative distance indicates the trustworthiness of a router or group of routers as a source of routing information. In general, the higher the value, the less trusted the entity. An administrative distance of 1 usually represents a directly connected network, and and an administrative distance of 255 the routing source is unreliable or unknown. The administrative distance conventionally applied to RIP is 120.

Use the **delete** form of this command to restore the default administrative distance to a RIP network or remove an access list.

Use the **show** form of this command to display administrative distance of a RIP network or access list application.

protocols rip passive-interface <ethx>

Suppresses RIP routing updates on an interface.

Syntax

set protocols rip passive-interface *ethx*

delete protocols rip passive-interface *ethx*

show protocols rip passive-interface

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
    rip {
        passive-interface ethx
    }
}
```

Parameters

ethx	Mandatory. Multi-node. The name of a configured Ethernet interface on which to suppress RIP routing updates. You can suppress routing updates on more than one RIP interface by creating multiple **protocols rip passive-interface** configuration nodes.

Default

RIP routing updates are not suppressed.

Usage Guidelines

Use the **set** form of this command to suppress RIP routing updates on an interface

Use the **delete** form of this command to disable RIP routing update suppression on an interface.

Use the **show** form of this command to display RIP route suppression configuration for an interface.

protocols rip route <ipv4net>

Specifies a RIP static route.

Syntax

set protocols rip route *ipv4net*

delete protocols rip route *ipv4net*

show protocols rip route

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
    rip {
        route ipv4net
    }
}
```

Parameters

ipv4net	Mandatory. The network address defining the RIP static route.

Default

None.

Usage Guidelines

Use the **set** form of this command to define a RIP static route.

Use the **delete** form of this command to remove a RIP static route.

Use the **show** form of this command to display RIP static route configuration.

protocols rip timers garbage-collection <seconds>

Allows you to set timers for RIP garbage collection.

Syntax

set protocols rip timers garbage-collection *seconds*

delete protocols rip timers garbage-collection [*seconds*]

show protocols rip timers garbage-collection

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
    rip {
        timers {
            garbage-collection seconds
        }
    }
}
```

Parameters

seconds	Mandatory. The timer interval period in seconds. The range is 5 to 2147483647.

Default

The default is 120.

Usage Guidelines

Use the **set** form of this command to set the garbage collection timer. When the timer expires, the system will scan for stale RIP resources and release them for use.

Use the **delete** form of this command to restore the default value for the RIP garbage collection timer.

Use the **show** form of this command to display RIP garbage collection timer configuration.

protocols rip timers timeout <seconds>

Allows you to set the interval for RIP time-outs.

Syntax

set protocols rip timers timeout *seconds*

delete protocols rip timers timeout [*seconds*]

show protocols rip timers timeout

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
    rip {
        timers {
            timeout seconds
        }
    }
}
```

Parameters

seconds	Mandatory. The RIP timeout interval, in seconds. The range is 5 to 2147483647. The default is 180.

Default

RIP time-outs occur at 180 second.

Usage Guidelines

Use the **set** form of this command to set the value for RIP time-outs.

Use the **delete** form of this command to restore the RIP timeout interval to the default value.

Use the **show** form of this command to display RIP timeout configuration.

protocols rip timers update <seconds>

Allows you to set the timer for RIP routing table updates.

Syntax

set protocols rip timers update *seconds*

delete protocols rip timers update [*seconds*]

show protocols rip timers update

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
    rip {
        timers {
            update seconds
        }
    }
}
```

Parameters

seconds	Mandatory. The interval at which RIP routing table updates will occur. The range is 5 is 2147483647. The default is 30.

Default

The RIP routing table is updated every 30 seconds.

Usage Guidelines

Use the **set** form of this command to set the interval between RIP routing table updates. The shorter this interval, the more accurate the routing information in the tables; however, the more protocol network traffic occurs.

Use the **delete** form of this command to restore the RIP update timer to the default value.

Use the **show** form of this command to display the RIP update time configuration.

show debugging rip

Displays RIP protocol debugging flags.

Syntax

show debug rip

Command Mode

Operational mode.

Parameters

None

Default

None.

Usage Guidelines

Use this command to see how debugging is set for RIP.

show ip route rip

Displays all IP RIP routes.

Syntax

show ip route rip

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display RIP routes contained in the Routing Information Base (RIB).

Examples

Example 2-1 shows all RIP routes from the RIB.

Example 2-1 "show ip route rip": Displaying routes

```
vyatta@vyatta:~$ show ip route rip
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route
vyatta@vyatta:~$
```

show ip rip

Displays information for the Routing Information Protocol (RIP).

Syntax

show ip rip [**status**]

Command Mode

Operational mode.

Parameters

status	Optional. Displays only RIP protocol status information.

Default

Displays all RIP protocol information.

Usage Guidelines

Use this command to see information about the Routing Information Protocol.

Examples

Example 2-2 lists RIP information.

Example 2-2 "show ip rip": Displaying RIP information

```
vyatta@vyatta:~$ show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
      (n) - normal, (s) - static, (d) - default, (r) - redistribute,
      (i) - interface

     Network            Next Hop         Metric From            Tag Time
C(i) 192.168.1.0/24     0.0.0.0               1 self              0
vyatta@vyatta:~$
```

Chapter 3: Route Redistribution

This chapter describes commands for redistributing routes from other routing protocols into RIP.

This chapter presents the following topics:

Route Redistribution Commands

This chapter contains the following commands.

Configuration Commands	
protocols rip redistribute bgp	Allows you to redistribute BGP routes into RIP routing tables.
protocols rip redistribute connected	Allows you to redistribute directly connected routes into RIP routing tables.
protocols rip redistribute kernel	Allows you to redistribute kernel routes into RIP routing tables.
protocols rip redistribute ospf	Allows you to redistribute OSPF routes into RIP routing tables.
protocols rip redistribute static	Allows you to redistribute static routes into RIP routing tables.
Operational Commands	
None	

protocols rip redistribute bgp

Allows you to redistribute BGP routes into RIP routing tables.

Syntax

set protocols rip redistribute bgp [**metric** *metric* | **route-map** *map-name*]

delete protocols rip redistribute bgp [**metric** | **route-map**]

show protocols rip redistribute bgp [**metric** | **route-map**]

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
    rip {
        redistribute {
            bgp {
                metric metric
                route-map map-name
            }
        }
    }
}
```

Parameters

metric *metric*	The routing metric to be applied to BGP routes being imported into RIP routing tables. The range is 1-16. The default is 1.
map-name	Optional. Applies the specified route map to BGP routes being imported into RIP routing tables.

Default

BGP routes being redistributed into RIP are assigned a routing metric of 1. By default, no route map is applied to redistributed BGP routes.

Usage Guidelines

Use the **set** form of this command to set the routing metric for BGP routes being redistributed into RIP, or to specify a route map to be applied to redistributed BGP routes.

Use the **delete** form of this command to remove BGP route redistribution configuration.

Use the **show** form of this command to display BGP route redistribution configuration.

protocols rip redistribute connected

Allows you to redistribute directly connected routes into RIP routing tables.

Syntax

set protocols rip redistribute connected [**metric** *metric* | **route-map** *map-name*]

delete protocols rip redistribute connected [**metric** | **route-map**]

show protocols rip redistribute connected [**metric** | **route-map**]

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
    rip {
        redistribute {
            connected {
                metric metric
                route-map map-name
            }
        }
    }
}
```

Parameters

metric	Optional. The routing metric to be applied to connected routes being imported into RIP routing tables. The range is 1-16. The default is 1.
map-name	Optional. Applies the specified route map to connected routes being imported into RIP routing tables.

Default

Connected routes being redistributed into RIP are assigned a routing metric of 1. By default, no route map is applied to redistributed connected routes.

Usage Guidelines

Use the **set** form of this command to set the routing metric for connected routes being redistributed into RIP, or to specify a route map to be applied to redistributed connected routes.

Use the **delete** form of this command to remove connected route redistribution configuration.

Use the **show** form of this command to display connected route redistribution configuration.

protocols rip redistribute kernel

Allows you to redistribute kernel routes into RIP routing tables.

Syntax

set protocols rip redistribute kernel [**metric** *metric* | **route-map** *map-name*]

delete protocols rip redistribute kernel [**metric** | **route-map**]

show protocols rip redistribute kernel [**metric** | **route-map**]

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
    rip {
        redistribute {
            kernel {
                metric metric
                route-map map-name
            }
        }
    }
}
```

Parameters

metric	Optional. The routing metric to be applied to kernel routes being imported into RIP routing tables. The range is 1-16. The default is 1.
map-name	Optional. Applies the specified route map to kernel routes being imported into RIP routing tables.

Default

Kernel routes being redistributed into RIP are assigned a routing metric of 1. By default, no route map is applied to redistributed kernel routes.

Usage Guidelines

Use the **set** form of this command to set the routing metric for kernel routes being redistributed into RIP, or to specify a route map to be applied to redistributed kernel routes.

Use the **delete** form of this command to remove kernel route redistribution configuration.

Use the **show** form of this command to display kernel route redistribution configuration.

protocols rip redistribute ospf

Allows you to redistribute OSPF routes into RIP routing tables.

Syntax

set protocols rip redistribute ospf [**metric** *metric* | **route-map** *map-name*]

delete protocols rip redistribute ospf [**metric** | **route-map**]

show protocols rip redistribute ospf [**metric** | **route-map**]

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
    rip {
        redistribute {
            ospf {
                metric metric
                route-map map-name
            }
        }
    }
}
```

Parameters

metric	Optional. The routing metric to be applied to OSPF routes being imported into RIP routing tables. The range is 1-16. The default is 1.
map-name	Optional. Applies the specified route map to OSPF routes being imported into RIP routing tables.

Default

OSPF routes being redistributed into RIP are assigned a routing metric of 1. By default, no route map is applied to redistributed OSPF routes.

Usage Guidelines

Use the **set** form of this command to set the routing metric for OSPF routes being redistributed into RIP, or to specify a route map to be applied to redistributed OSPF routes.

Use the **delete** form of this command to remove OSPF route redistribution configuration.

Use the **show** form of this command to display OSPF route redistribution configuration.

protocols rip redistribute static

Allows you to redistribute static routes into RIP routing tables.

Syntax

set protocols rip redistribute static [**metric** *metric* | **route-map** *map-name*]

delete protocols rip redistribute static [**metric** | **route-map**]

show protocols rip redistribute static [**metric** | **route-map**]

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
    rip {
        redistribute {
            static {
                metric metric
                route-map map-name
            }
        }
    }
}
```

Parameters

metric	Optional. The routing metric to be applied to static routes being imported into RIP routing tables. The range is 1-16. The default is 1.
map-name	Optional. Applies the specified route map to static routes being imported into RIP routing tables.

Default

Static routes being redistributed into RIP are assigned a routing metric of 1. By default, no route map is applied to redistributed static routes.

Usage Guidelines

Use the **set** form of this command to set the routing metric for static routes being redistributed into RIP, or to specify a route map to be applied to redistributed static routes.

Use the **delete** form of this command to remove static route redistribution configuration.

Use the **show** form of this command to display static route redistribution configuration.

Chapter 4: Route Filtering

This chapter describes commands for RIP route filtering.

This chapter presents the following topics:

RIP Route Filtering Commands

This chapter contains the following commands.

Configuration Commands	
protocols rip distribute-list access-list	Applies an access list for filtering inbound or outbound RIP packets.
protocols rip distribute-list interface <ethx> access-list	Applies an access list to a specific interface for filtering inbound or outbound RIP packets.
protocols rip distribute-list interface <ethx> prefix-list	Applies a prefix list to a specific interface for filtering inbound or outbound RIP packets.
protocols rip distribute-list prefix-list	Applies a prefix list for filtering inbound or outbound RIP packets.
Operational Commands	
None.	

protocols rip distribute-list access-list

Applies an access list for filtering inbound or outbound RIP packets.

Syntax

set protocols rip distribute-list access-list {**in** *in-list* | **out** *out-list*}

delete protocols rip distribute-list access-list {**in** | **out**}

show protocols rip distribute-list access-list {**in** | **out**}

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
    rip {
        distribute-list {
            access-list {
                in in-list
                out out-list
            }
        }
    }
}
```

Parameters

in-list	The identifier of a defined access list. The access list will be applied to filter inbound RIP packets.
out-list	The identifier of a defined access list. The access list will be applied to filter outbound RIP packets.

Default

None.

Usage Guidelines

Use the **set** form of this command to apply an access list for filtering inbound or outbound RIP packets.

Use the **delete** form of this command to remove access list packet filtering from RIP packets.

Use the **show** form of this command to display RIP access list filtering configuration.

protocols rip distribute-list interface <ethx> access-list

Applies an access list to a specific interface for filtering inbound or outbound RIP packets.

Syntax

set protocols rip distribute-list interface *ethx* **access-list** {**in** *in-list* | **out** *out-list*]

delete protocols rip distribute-list interface *ethx* **access-list** {**in** | **out**}

show protocols rip distribute-list interface *ethx* **access-list** {**in** | **out**}

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
    rip {
        distribute-list {
            interface ethx
                access-list {
                    in in-list
                    out out-list
                }
            }
        }
    }
}
```

Parameters

ethx	Mandatory. Interface on which to filter packets.
in-list	The identifier of a defined access list. The access list will be applied to the specified interface to filter inbound RIP packets.
out-list	The identifier of a defined access list. The access list will be applied to the specified interface to filter outbound RIP packets.

Default

None.

Usage Guidelines

Use the **set** form of this command to apply an access list to a specific interface for filtering inbound or outbound RIP packets.

Use the **delete** form of this command to remove RIP access list packet filtering from an interface.

Use the **show** form of this command to display RIP access list filtering configuration for an interface.

protocols rip distribute-list interface <ethx> prefix-list

Applies a prefix list to a specific interface for filtering inbound or outbound RIP packets.

Syntax

set protocols rip distribute-list interface *ethx* **prefix-list** {**in** *in-list* | **out** *out-list*}

delete protocols rip distribute-list interface *ethx* **prefix-list** {**in** | **out**}

show protocols rip distribute-list interface *ethx* **prefix-list** {**in** | **out**}

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
    rip {
        distribute-list {
            interface ethx
                prefix-list {
                    in in-list
                    out out-list
                }
            }
        }
    }
}
```

Parameters

ethx	Mandatory. Interface on which to apply the access list filter.
in-list	The identifier of a defined prefix list. The prefix list will be applied to the specified interface to filter inbound RIP packets.
out-list	The identifier of a defined prefix list. The prefix list will be applied to the specified interface to filter outbound RIP packets.

Default

None.

Usage Guidelines

Use the **set** form of this command to apply a prefix list to a specific interface for filtering inbound or outbound RIP packets.

Use the **delete** form of this command to remove RIP prefix list packet filtering from an interface.

Use the **show** form of this command to display RIP prefix list filtering configuration for an interface.

protocols rip distribute-list prefix-list

Applies a prefix list for filtering inbound or outbound RIP packets.

Syntax

set protocols rip distribute-list prefix-list {**in** *in-list* | **out** *out-list*}

delete protocols rip distribute-list prefix-list {**in** | **out**}

show protocols rip distribute-list prefix-list {**in** | **out**}

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
    rip {
        distribute-list {
            prefix-list {
                in in-list
                out out-list
            }
        }
    }
}
```

Parameters

in-list	The identifier of a defined prefix list. The prefix list will be applied to filter inbound RIP packets.
out-list	The identifier of a defined prefix list. The prefix list will be applied to filter outbound RIP packets.

Default

None.

Usage Guidelines

Use the **set** form of this command to apply a prefix list for filtering inbound or outbound RIP packets.

Use the **delete** form of this command to remove RIP prefix list packet filtering.

Use the **show** form of this command to display RIP prefix list filtering configuration.

Chapter 5: RIP Interface Commands

This chapter describes commands for configuring RIP on various interfaces.

This chapter presents the following topics:

RIP Interface Commands

This chapter contains the following commands.

Configuration Commands	
interfaces <interface> ip rip	Enables RIP on an interface.
interfaces <interface> ip rip authentication	Specify RIP authentication for the interface.
interfaces <interface> ip rip split-horizon	Configures split-horizon in RIP updates coming from this interface.
Operational Commands	
None.	

interfaces <interface> ip rip

Enables RIP on an interface.

Syntax

set interfaces *interface* **ip rip**

delete interfaces *interface* **ip rip**

show interfaces *interface* **ip rip**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces interface {
    ip {
        rip
    }
}
```

Parameters

interface	Mandatory. The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in the Usage Guidelines below.

Default

None.

Usage Guidelines

Use this command to enable Routing Information Protocol (RIP) on an interface.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
ADSL Bridged Ethernet	adsl *adslx* pvc *pvc-id* bridged-ethernet	*adslx* The name of a Bridged Ethernet- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
ADSL Classical IPOA	adsl *adslx* pvc *pvc-id* classical-ipoa	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
ADSL PPPoA	adsl *adslx* pvc *pvc-id* pppoa *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The PPPoA unit number. This number must be unique across all PPPoA interfaces. In addition, only one PPPoA instance can be configured on a PVC. PPPoA units range from 0 to 15 and the resulting interfaces are named pppoa0 to pppoa15.
ADSL PPPoE	adsl *adslx* pvc *pvc-id* pppoe *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.

Interface Type	Syntax	Parameters
Bridge	bridge *brx*	*brx* The name of a Bridge group. The range is **br0** through **br999**.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet PPPoE	ethernet *ethx* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.
Ethernet Vif PPPoE	ethernet *ethx* vif *vlan-id* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Loopback	loopback *lo*	*lo* The name of the loopback interface.
Multilink	multilink *mlx* vif *1*	*mlx* The identifier of the multilink bundle. You can create up to two multilink bundles. Supported values are **ml0** (“em ell zero”) through **ml23** (“em ell twenty-three”). *1* The identifier of the virtual interface. Currently, only one vif is supported for multilink interfaces, and the identifier must be 1. The vif must already have been defined.
OpenVPN	openvpn *vtunx*	*vtunx* The identifier for the OpenVPN interface. This may be **vtun0** to **vtun*x***, where ***x*** is a non-negative integer.
Pseudo-Ethernet	pseudo-ethernet *pethx*	*pethx* The name of a pseudo-Ethernet interface. The range is **peth0** through **peth999**.
Serial Cisco HDLC	serial *wanx* cisco-hdlc vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for Cisco HDLC interfaces, and the identifier must be 1. The vif must already have been defined.

Interface Type	Syntax	Parameters
Serial Frame Relay	serial *wanx* frame-relay vif *dlci*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *dlci* The identifier of the virtual interface. For Frame Relay interfaces, this is the DLCI number for the interface. the range is 16 to 991. The vif must already have been defined.
Serial PPP	serial *wanx* ppp vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for point-to-point interfaces, and the identifier must be 1. The vif must already have been defined.
Tunnel	tunnel *tunx*	*tunx* An identifier for the tunnel interface you are defining. The range is **tun0** to **tun23**.
Wireless	wireless *wlanx*	*wlanx* The identifier for the wireless interface you are using. This may be **wlan0** to **wlan999**.
Wireless Modem	wirelessmodem *wlmx*	*wlmx* The identifier for the wirelessmodem interface you are using. This may be **wlm0** to **wlm999**.

Use the **set** form of this command to enable RIP on an interface.

Use the **delete** form of this command to remove all RIP configuration and disable RIP on the interface.

Use the **show** form of this command to display RIP configuration.

interfaces <interface> ip rip authentication

Specify RIP authentication for the interface.

Syntax

set interfaces *interface* **ip rip authentication** [**md5** *md5-key* **password** *md5-password* | **plaintext-password** *password*]

delete interfaces *interface* **ip rip authentication** [**md5** *md5-key* **password** | **plaintext-password**]

show interfaces *interface* **ip rip authentication** [**md5** *md5-key* **password** | **plaintext-password**]

Command Mode

Configuration mode.

Configuration Statement

```
interfaces interface {
    ip {
        rip {
            authentication {
                md5 md5-key {
                    password md5-password
                }
                plaintext-password password
            }
        }
    }
}
```

Parameters

interface	Mandatory. The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in the Usage Guidelines below.
md5-key	Optional. The authentication key ID. This must be the same on both the sending and receiving systems. The range is 1 to 255.
md5-password	Optional. The password to use in MD5 authentication. This must be the same on both the sending and receiving systems.

password	Optional. The password to use in simple (plain-text) authentication. This must be the same on both the sending and receiving systems.

Default

None.

Usage Guidelines

Use this command to specify the authentication method to be used for RIP on an interface. This authentication is independent of the authentication configured for the RIP area.

In plain text authentication, passwords are sent through the network in plain text. In MD5 authentication, the system uses the Message Digest 5 (MD5) algorithm to compute a hash value from the contents of the RIP packet and the password. The hash value and the MD5 key are included in the transmitted packet, and the receiving system (configured with the same password) calculates its own hash function, which must match.

The authentication parameters must be the same for all routers that are to establish two-way communication within a network. If two routers do not agree on these parameters, they will not consider establish adjacencies, and will disregard one another's communications.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
ADSL Bridged Ethernet	adsl *adslx* pvc *pvc-id* bridged-ethernet	*adslx* The name of a Bridged Ethernet- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
ADSL Classical IPOA	adsl *adslx* pvc *pvc-id* classical-ipoa	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.

Interface Type	Syntax	Parameters
ADSL PPPoA	adsl *adslx* pvc *pvc-id* pppoa *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The PPPoA unit number. This number must be unique across all PPPoA interfaces. In addition, only one PPPoA instance can be configured on a PVC. PPPoA units range from 0 to 15 and the resulting interfaces are named pppoa0 to pppoa15.
ADSL PPPoE	adsl *adslx* pvc *pvc-id* pppoe *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.
Bridge	bridge *brx*	*brx* The name of a Bridge group. The range is **br0** through **br999**.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet PPPoE	ethernet *ethx* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.

Interface Type	Syntax	Parameters
Ethernet Vif PPPoE	ethernet *ethx* vif *vlan-id* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Loopback	loopback *lo*	*lo* The name of the loopback interface.
Multilink	multilink *mlx* vif *1*	*mlx* The identifier of the multilink bundle. You can create up to two multilink bundles. Supported values are **ml0** (“em ell zero”) through **ml23** (“em ell twenty-three”). *1* The identifier of the virtual interface. Currently, only one vif is supported for multilink interfaces, and the identifier must be 1. The vif must already have been defined.
OpenVPN	openvpn *vtunx*	*vtunx* The identifier for the OpenVPN interface. This may be **vtun0** to **vtun*x***, where ***x*** is a non-negative integer.
Pseudo-Ethernet	pseudo-ethernet *pethx*	*pethx* The name of a pseudo-Ethernet interface. The range is **peth0** through **peth999**.
Serial Cisco HDLC	serial *wanx* cisco-hdlc vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for Cisco HDLC interfaces, and the identifier must be 1. The vif must already have been defined.
Serial Frame Relay	serial *wanx* frame-relay vif *dlci*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *dlci* The identifier of the virtual interface. For Frame Relay interfaces, this is the DLCI number for the interface. the range is 16 to 991. The vif must already have been defined.
Serial PPP	serial *wanx* ppp vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for point-to-point interfaces, and the identifier must be 1. The vif must already have been defined.
Tunnel	tunnel *tunx*	*tunx* An identifier for the tunnel interface you are defining. The range is **tun0** to **tun23**.
Wireless	wireless *wlanx*	*wlanx* The identifier for the wireless interface you are using. This may be **wlan0** to **wlan999**.

Interface Type	Syntax	Parameters
Wireless Modem	wirelessmodem *wlmx*	*wlmx* The identifier for the wirelessmodem interface you are using. This may be **wlm0** to **wlm999**.

Use the **set** form of this command to set RIP authentication for an interface.

Use the **delete** form of this command to remove RIP interface authentication configuration information.

Use the **show** form of this command to display RIP interface authentication configuration information.

interfaces <interface> ip rip split-horizon

Configures split-horizon in RIP updates coming from this interface.

Syntax

set interfaces *interface* **ip rip split-horizon** [**disable** | **poison-reverse**]

delete interfaces *interface* **ip rip split-horizon** [**disable** | **poison-reverse**]

show interfaces *interface* **ip rip split-horizon**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces interface {
    ip {
        rip {
            split-horizon {
                disable
                poison-reverse
            }
        }
    }
}
```

Parameters

interface	Mandatory. The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in the Usage Guidelines below.
disable	Disables split-horizon on the interface.
poison-reverse	Enables poison-reverse on the interface.

Default

Split-horizon is enabled.

Usage Guidelines

Use this command to disable split-horizon or enable split-horizon poison-reverse on an interface running RIP.

Split-horizon is a stability feature that reduces the possibility of network loops, particularly in the case where links become disconnected. It stops an interface from including in its network updates any routes that it learned from that interface. Split horizon is effective at preventing loops between routers that are directly connected to one another, and speeds convergence when network conditions change and is the default setting in RIP.

Poison reverse is a variation of split horizon. When an interface with poison reverse enabled detects that a link is down, it increases the metric for that route to 16, and propagates that information in its next update. Since 15 is the largest number of hops considered reachable on a RIP network, increasing the metric to 16 renders the route unreachable as far as downstream RIP routers are concerned. This is called "poisoning" the route. Poison reverse can be useful for propagating information about bad routes to routers that are downstream but not immediate neighbors, where split horizon is ineffective.

When this option is enabled, the router includes the route in announcements to the neighbor from which it was learned. When this option is disabled, the router omits the route in announcements to the neighbor from which it was learned.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
ADSL Bridged Ethernet	adsl *adslx* pvc *pvc-id* bridged-ethernet	*adslx* The name of a Bridged Ethernet- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
ADSL Classical IPOA	adsl *adslx* pvc *pvc-id* classical-ipoa	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.

Interface Type	Syntax	Parameters
ADSL PPPoA	adsl *adslx* pvc *pvc-id* pppoa *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The PPPoA unit number. This number must be unique across all PPPoA interfaces. In addition, only one PPPoA instance can be configured on a PVC. PPPoA units range from 0 to 15 and the resulting interfaces are named pppoa0 to pppoa15.
ADSL PPPoE	adsl *adslx* pvc *pvc-id* pppoe *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.
Bridge	bridge *brx*	*brx* The name of a Bridge group. The range is **br0** through **br999**.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet PPPoE	ethernet *ethx* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.

Interface Type	Syntax	Parameters
Ethernet Vif PPPoE	ethernet *ethx* vif *vlan-id* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Loopback	loopback *lo*	*lo* The name of the loopback interface.
Multilink	multilink *mlx* vif *1*	*mlx* The identifier of the multilink bundle. You can create up to two multilink bundles. Supported values are **ml0** ("em ell zero") through **ml23** ("em ell twenty-three"). *1* The identifier of the virtual interface. Currently, only one vif is supported for multilink interfaces, and the identifier must be 1. The vif must already have been defined.
OpenVPN	openvpn *vtunx*	*vtunx* The identifier for the OpenVPN interface. This may be **vtun0** to **vtun*x***, where ***x*** is a non-negative integer.
Pseudo-Ethernet	pseudo-ethernet *pethx*	*pethx* The name of a pseudo-Ethernet interface. The range is **peth0** through **peth999**.
Serial Cisco HDLC	serial *wanx* cisco-hdlc vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for Cisco HDLC interfaces, and the identifier must be 1. The vif must already have been defined.
Serial Frame Relay	serial *wanx* frame-relay vif *dlci*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *dlci* The identifier of the virtual interface. For Frame Relay interfaces, this is the DLCI number for the interface. the range is 16 to 991. The vif must already have been defined.
Serial PPP	serial *wanx* ppp vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for point-to-point interfaces, and the identifier must be 1. The vif must already have been defined.
Tunnel	tunnel *tunx*	*tunx* An identifier for the tunnel interface you are defining. The range is **tun0** to **tun23**.
Wireless	wireless *wlanx*	*wlanx* The identifier for the wireless interface you are using. This may be **wlan0** to **wlan999**.

Interface Type	Syntax	Parameters
Wireless Modem	wirelessmodem *wlmx*	*wlmx* The identifier for the wirelessmodem interface you are using. This may be **wlm0** to **wlm999**.

Use the **set** form of this command to configure split-horizon and split-horizon poison-reverse on an interface running RIP.

Use the **delete** form of this command to restore the default configuration.

Use the **show** form of this command to display split-horizon configuration.

Glossary of Acronyms

ACL	access control list
ADSL	Asymmetric Digital Subscriber Line
API	Application Programming Interface
AS	autonomous system
ARP	Address Resolution Protocol
BGP	Border Gateway Protocol
BIOS	Basic Input Output System
BPDU	Bridge Protocol Data Unit
CA	certificate authority
CCMP	AES in counter mode with CBC-MAC
CHAP	Challenge Handshake Authentication Protocol
CLI	command-line interface
DDNS	dynamic DNS
DHCP	Dynamic Host Configuration Protocol
DHCPv6	Dynamic Host Configuration Protocol version 6
DLCI	data-link connection identifier
DMI	desktop management interface
DMZ	demilitarized zone
DN	distinguished name
DNS	Domain Name System

DSCP	Differentiated Services Code Point
DSL	Digital Subscriber Line
eBGP	external BGP
EGP	Exterior Gateway Protocol
ECMP	equal-cost multipath
ESP	Encapsulating Security Payload
FIB	Forwarding Information Base
FTP	File Transfer Protocol
GRE	Generic Routing Encapsulation
HDLC	High-Level Data Link Control
I/O	Input/Ouput
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers
IGP	Interior Gateway Protocol
IPS	Intrusion Protection System
IKE	Internet Key Exchange
IP	Internet Protocol
IPOA	IP over ATM
IPsec	IP security
IPv4	IP Version 4
IPv6	IP Version 6
ISP	Internet Service Provider
L2TP	Layer 2 Tunneling Protocol
LACP	Link Aggregation Control Protocol
LAN	local area network

LDAP	Lightweight Directory Access Protocol
LLDP	Link Layer Discovery Protocol
MAC	medium access control
MIB	Management Information Base
MLPPP	multilink PPP
MRRU	maximum received reconstructed unit
MTU	maximum transmission unit
NAT	Network Address Translation
ND	Neighbor Discovery
NIC	network interface card
NTP	Network Time Protocol
OSPF	Open Shortest Path First
OSPFv2	OSPF Version 2
OSPFv3	OSPF Version 3
PAM	Pluggable Authentication Module
PAP	Password Authentication Protocol
PAT	Port Address Translation
PCI	peripheral component interconnect
PKI	Public Key Infrastructure
PPP	Point-to-Point Protocol
PPPoA	PPP over ATM
PPPoE	PPP over Ethernet
PPTP	Point-to-Point Tunneling Protocol
PVC	permanent virtual circuit
QoS	quality of service
RADIUS	Remote Authentication Dial-In User Service

RIB	Routing Information Base
RIP	Routing Information Protocol
RIPng	RIP next generation
Rx	receive
SLAAC	Stateless Address Auto-Configuration
SNMP	Simple Network Management Protocol
SMTP	Simple Mail Transfer Protocol
SONET	Synchronous Optical Network
SSH	Secure Shell
SSID	Service Set Identifier
STP	Spanning Tree Protocol
TACACS+	Terminal Access Controller Access Control System Plus
TCP	Transmission Control Protocol
TKIP	Temporal Key Integrity Protocol
ToS	Type of Service
Tx	transmit
UDP	User Datagram Protocol
vif	virtual interface
VLAN	virtual LAN
VPN	Virtual Private Network
VRRP	Virtual Router Redundancy Protocol
WAN	wide area network
WAP	wireless access point
WPA	Wired Protected Access