Chapter 18

Routing Information Protocol (RIP)

Introduction

Routing Information Protocol (RIP) is a simple distance vector routing protocol. It enables the switch to learn routes to other networks. The following RFCs describe RIP:

- RFC 1058, *Routing Information Protocol*
- RFC 1723, *RIP Version 2—Carrying Additional Information*
- RFC 1582, *Extensions to RIP to Support Demand Circuits*

Route selection

RIP determines the number of hops between the destination and the switch, where one hop is one link. This hop count is referred to as the RIP metric. Given a choice of routes, RIP uses the route with the lowest metric, and therefore the route that takes the lowest number of hops. If multiple routes have the same metric, RIP chooses the first route it finds.

RIP is limited to routes of 15 hops or less. If a network is more than 15 hops away, RIP does not put its route into the switch's routing table.

RIP suits star topologies very well. It is less suited to a meshed (multiply connected) network, because in meshed networks it learns multiple copies of routes, with different metrics.

Neighbours

To maintain its table of RIP routes, the switch periodically receives broadcasts of routing information from neighbouring routers, called RIP neighbours. Similarly, the switch periodically broadcasts its routing information to its neighbours. The switch removes routes from the table if the neighbouring routers do not keep them up to date (refresh them).

Each switch interface's RIP neighbours must be in the same subnet as the interface. For security reasons, the switch only accepts RIP broadcasts from addresses in its subnet.

RIPv2

RFC 1723 describes RIP version 2, which enables RIP updates to contain subnet masks and next hop information. The ability to carry subnet masks means different subnets within the same network can use different sized subnet masks.

RIP on demand

RFC 1582 extends RIP so that you can use it over dial-on-demand connections, which are activated when there is traffic to send. The two ends of the link exchange route information when their routing tables change. The routes do not age out, so RIP on demand does not need to exchange frequent messages to keep routes alive.

Configuring RIP

RIP automatically exchanges routing information with each neighbour when you specify the interface over which it accesses that neighbour. To start the process, use the command:

```
add ip rip interface=interface [other-options...]
```

To remove RIP neighbours, use the command:

```
delete ip rip interface=interface [other-options...]
```

If you delete all RIP neighbours, this also disables RIP broadcasts.

To modify the settings of a RIP neighbour, use the command:

```
set ip rip interface=interface [other-options...]
```

When you configure RIP over a dial-on-demand connection, use the command:

```
add ip rip interface=interface demand=yes [other-options...]
```

You can assign specific RIP metrics to statically-defined routes, by using one of the commands:

```
add ip route=ipadd interface=interface nexthop=ipadd
   metric1=1..16 [other-options...]
```

```
set ip route=ipadd interface=interface nexthop=ipadd
   metric1=1..16 [other-options...]
```

To display the neighbours to which the switch is sending RIP broadcasts, use the command:

```
show ip rip [other-options...]
```

RIP propagates interface routes as long as their status is up at a physical level. For VLANs, this means that RIP propagates the VLAN's interface route when at least one port in the VLAN is active. You can check which interfaces active with one of the following commands:

```
show ip interface
```

```
show ip route [other-options...]
```

In both of the above commands, a hash symbol (#) next to the interface name indicates that the interface is down.

Timers

The operation of RIP is controlled by four global timers. To set the value of these timers, use the command:

```
set ip riptimer [flush=1..4294967295]
   [holddown=1..4294967295] [invalid=1..4294967295]
   [update=1..4294967295]
```

If you change a timer, existing routes are not affected, only new routes.

To display current values of the RIP timers, use the command:

```
show ip riptimer
```

Redistributing Routes into RIP

You can redistribute statically-configured routes as RIP routes.

Statically-Configured Routes

By default, RIP imports statically-configured routes into the RIP routing table and advertises them to RIP neighbours. To avoid advertising statically-configured routes over an interface, use one of the commands:

```
add ip rip interface=interface staticexport=no
    [other-options...]

set ip rip interface=interface staticexport=no
    [other-options...]
```

To start advertising statically-configured routes again, use the command:

```
set ip rip interface=interface staticexport=yes
    [other-options...]
```

Configuration Example

The following example shows how to configure RIP on a LAN.

The scenario in this example is (Figure 18-1 on page 18-5):

- Switch A receives OSPF routes from a Metropolitan Area Network. RIP transfers these routes to routers and switches on the LAN.
- Switch B is on the same LAN as switch A.
- An ADSL modem connects a remote office to switch B, through switch port 2 on vlan2. The remote office is on a different subnet from switch A. RIP enables routing between the remote office and the LAN.

The configuration is shown in:

- Figure 18-1 on page 18-5—a diagram of the scenario
- Figure 18-2 on page 18-5—the commands to configure Switch A
- Figure 18-3 on page 18-5—the commands to configure Switch B

Figure 18-1: Example configuration for RIP

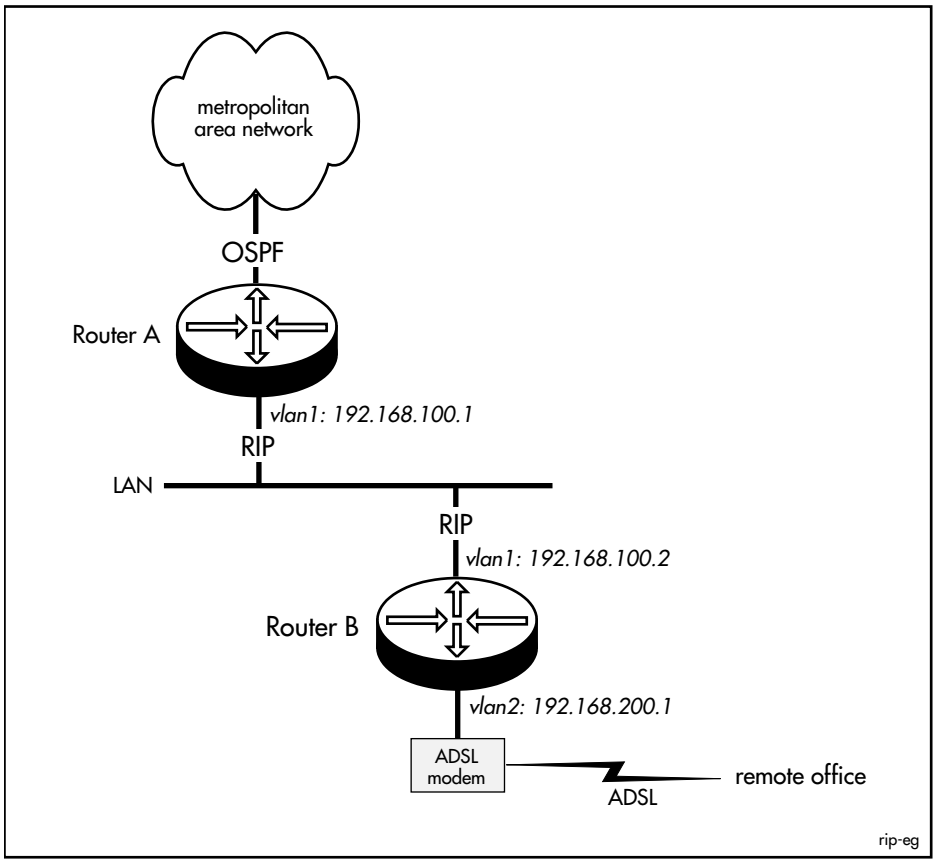

Figure 18-2: Example script for configuring RIP on Switch A

```
# Configuring RIP on Switch A

# enable IP routing
enable ip

# give vlan1 an IP address
add ip int=vlan1 ip=192.168.100.1 mask=255.255.255.0

# configure RIP over vlan1
add ip rip int=vlan1 send=rip2 receive=rip2
```

Figure 18-3: Example script for configuring RIP on Switch B

```
# Configuring RIP on Switch B

# enable IP routing
enable ip

# give vlan1 an IP address
add ip int=vlan1 ip=192.168.100.2 mask=255.255.255.0

# create vlan2, assign a port to it, and give it an IP address
create vlan=vlan2 vid=2
add vlan=2 port=2
add ip int=vlan2 ip=192.168.200.1 mask=255.255.255.0

# configure RIP over vlan1 and vlan2
add ip rip int=vlan1 send=rip2 receive=rip2
add ip rip int=vlan2 send=rip2 receive=rip2
```

Command Reference

This section describes the commands available on the switch to configure and manage RIP.

The shortest valid command is denoted by capital letters in the Syntax section. See "Conventions" on page xxxviii of About this Software Reference in the front of this manual for details of the conventions used to describe command syntax. See Appendix A, Messages for a complete list of error messages and their meanings.

add ip rip interface

Syntax

```
ADD IP RIP INTerface=interface [IP=ipadd] [NEXThop=ipadd]
   [SENd={NOne|RIP1|RIP2|COmpatible}] [RECeive={NOne|RIP1|
   RIP2|BOth}] [DEMand={False|NO|OFF|ON|True|YES}]
   [AUth={NOne|PASSword|MD5}] [PASSword=password]
   [STATicexport={YES|NO}]
```

where:

- *interface* is an interface name formed by concatenating a Layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number from 0 to 15. If a logical interface is not specified, 0 is assumed.
- *ipadd* is an IP address in dotted decimal notation.
- *password* is a string 1 to 16 characters long. Valid characters are any printable character.

Description

This command configures an interface for RIP. The switch exchanges routing information with one or more neighbours via that interface.

The **interface** parameter specifies an existing interface on which to send or receive RIP packets. Valid interfaces are:

- VLAN (such as vlan1, vlan1-1)

To see a list of currently-available interfaces, use the **show interface** command on page 12-36 of Chapter 12, Interfaces.

The **ip** parameter specifies the IP address of the RIP neighbour. The following table shows the difference between specifying an address and not specifying an address.

If you...	The switch accepts...	And sends RIP updates to...
specify an address by using **ip**	RIP packets from that address on this interface	that address.
do not specify **ip**	all RIP packets on this interface	the RIP multicast address 224.0.0.9 if the **send** parameter is **rip2** or **compatible**. the broadcast address if the **send** parameter is **rip1**.

The **nexthop** parameter specifies the IP address of the best next hop to the destination. In most configurations, the switch is the best next hop when it is the originator of the RIP update. In that case, specify **nexthop=0.0.0.0**. However, in some configurations, the best route to the destination is out another router. In that case, specify the IP address of that router as the next hop. This stops RIP neighbours from unnecessarily sending packets via the switch. See Appendix A of RFC 1723 for an example in which part of a network uses RIP and part uses another routing protocol such as OSPF. The **nexthop** parameter is only valid when you specify **ip** and when **send** is **rip2** or **compatible**. The default next hop is **0.0.0.0**.

The **send** parameter specifies the version of RIP packet to send. If you specify **none**, then the switch does not send RIP packets. If you specify **rip1**, then the switch sends RIPv1 packets. If you specify **rip2**, then the switch sends RIPv2 packets. If you specify **compatible**, then the switch sends RIPv2 packets that are compatible with routers that only receive RIPv1 packets. It achieves compatibility by excluding routes that such routers treat as host routes. The default is **rip1**.

The **receive** parameter specifies the version of RIP packets to receive. If you specify **none**, then the switch does not accept RIP packets from the specified IP address on the specified interface. If you specify **rip1**, then the switch accepts RIPv1 packets. If you specify **rip2**, then the switch accepts RIPv2 packets. If you specify **both**, then the switch accepts both RIPv1 and RIPv2 packets but only keeps routes that conform to RIPv1. Routes conform to RIPv1 if they are classful—for example, the network 172.16.x.x conforms as long as it uses a Class B mask of 255.255.0.0. The switch discards non-conforming routes. The default is **both**.

The **demand** parameter specifies whether the switch uses RIP demand procedures. This means that it:

- only sends RIP updates when it has new routing information
- does not time routes out

Specify **yes** if the connection to the neighbour is a dial-on-demand connection. Configure both ends of the connection to have the same setting. The default is **no**.

The **authentication** parameter specifies how the switch authenticates RIP packets. If you specify **none**, the switch does not authenticate RIP packets. If you specify **password**, the switch uses a plaintext password authenticate RIP packets. If you specify **md5**, the switch uses an encrypted password. You must specify **none** unless the switch uses RIPv2. The default is **none**.

The **password** parameter specifies the password that the switch uses if **authentication** is **password** or **md5**. This parameter is required when authentication is used. The password can be up to 63 characters long, but the switch only uses the first 16 characters.

The **staticexport** parameter specifies whether the switch propagates static routing information from this interface. If you specify **yes**, the switch includes static routes in routing exports. If you specify **no**, the switch omits them. The default is **yes**.

Examples To broadcast RIPv1 on an Ethernet interface (vlan1), use the command:

```
add ip rip int=vlan1
```

To receive RIPv2 packets from only one host (172.16.248.33) on an Ethernet interface (vlan1), and broadcast RIPv1 packets on the interface, use the commands:

```
add ip rip int=vlan1 ip=172.16.248.33 rec=rip2 sen=no
add ip rip int=vlan1 rec=no
```

Related Commands

delete ip rip interface
set ip rip interface
show ip in Chapter 13, Internet Protocol (IP)
show ip rip

add ip trusted

Syntax

```
ADD IP TRusted=ipadd
```

where *ipadd* is an IP address in dotted decimal notation

Description

This command defines a *trusted router*. A trusted router is a source of RIP broadcasts that you trust to provide up-to-date, valid routing information. If you define trusted routers, the switch only accepts routing information and includes it in the routing table if it comes from a trusted router. A maximum of 32 trusted routers can be defined.

If you do not define trusted routers, the switch accepts routing information from any source, unless you have applied a route filter. See "Routing Information Protocol (RIP)" on page 20-10 of Chapter 20, Filtering IP Routes.

The **trusted** parameter specifies the IP address of a device from which RIP information is accepted. Adding one or more trusted routers automatically enables the trusted router option.

Examples

To specify the device with an IP address of 172.16.8.33 as a trusted source of RIP information, use:

```
add ip tr=172.16.8.33
```

Related Commands

delete ip trusted
show ip trusted

delete ip rip interface

Syntax `DELete IP RIP INTerface=`*`interface`* `[IP=`*`ipadd`*`]`

where:

- *interface* is an interface name formed by concatenating a Layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number from 0 to 15. If a logical interface is not specified, 0 is assumed.
- *ipadd* is an IP address in dotted decimal notation.

Description This command deletes a RIP neighbour. The switch stops exchanging routing information with the RIP neighbour.

The **interface** parameter specifies the interface via which the switch receives RIP packets from the RIP neighbour. Valid interfaces are:

- VLAN (such as vlan1, vlan1-1)

To see a list of currently available interfaces, use the **show interface** command on page 12-36 of Chapter 12, Interfaces, or the **show ip interface** command on page 13-154 of Chapter 13, Internet Protocol (IP).

The **ip** parameter specifies the IP address of the neighbour to delete.

Examples To delete a neighbour that is broadcasting RIP on an Ethernet interface (vlan1), use the command:

```
del ip rip int=vlan1
```

Related Commands
add ip rip interface
set ip rip interface
show ip in Chapter 13, Internet Protocol (IP)
show ip rip

delete ip trusted

Syntax `DELete IP TRusted=`*`ipadd`*

where *ipadd* is an IP address in dotted decimal notation

Description This command deletes an entry from the trusted router table.

The **trusted** parameter specifies the IP address of a host from which RIP information is no longer accepted. Deleting all trusted routers automatically disables the trusted router option.

Examples To delete the host with an IP address of 172.16.8.33 as a trusted source of RIP information, use:

```
del ip tr=172.16.8.33
```

Related Commands
add ip trusted
show ip trusted

set ip rip interface

Syntax

```
SET IP RIP INTerface=interface [IP=ipadd]
   [NEWipaddress=newipadd] [NEXThop=ipadd] [SENd={NOne|
   RIP1|RIP2|COmpatible}] [RECeive={NOne|RIP1|RIP2|BOth}]
   [DEMand={False|NO|OFF|ON|True|YES}] [AUth={NOne|
   PASSword|MD5}] [PASSword=password] [STATicexport={YES|
   NO}]
```

where:

- *interface* is an interface name formed by concatenating a Layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number from 0 to 15. If a logical interface is not specified, 0 is assumed.
- *ipadd* is an IP address in dotted decimal notation that identifies a specific RIP interface.
- *newipadd* is an IP address in dotted decimal notation that you want this RIP interface to use. A value of 0.0.0.0 means you wish to remove the existing IP address from this RIP interface.
- *password* is a string 1 to 16 characters long. Valid characters are any printable character.

Description

This command sets attributes of the RIP neighbour. The IP address and the interface identify which RIP neighbour to change.

The **interface** parameter specifies an existing interface that the RIP neighbour is on. Valid interfaces are:

- VLAN (such as vlan1, vlan1-1)

To see a list of currently-available interfaces, use the **show interface** command on page 12-36 of Chapter 12, Interfaces, or the **show ip interface** command on page 13-154 of Chapter 13, Internet Protocol (IP).

Use the **ip** parameter to specify the IP address of the RIP neighbour you wish to set new values for. This parameter is mandatory if the neighbour was created using an IP address, as it is needed to correctly identify the neighbour.

Use the **newipaddress** parameter to change the currently defined IP address for the RIP neighbour to either a new address or no address, or to add an IP address if one was not previously set.

The following table shows the different outcomes that are possible:.

If you have...	And you...	The switch accepts...	And sends RIP updates to...
1) An RIP interface with an IP address already defined	Use **newipaddress** to set a new IP address.	RIP packets from that address on this interface	The new address.
	Use **newipaddress** to set the IP address to nothing (0.0.0.0)	all RIP packets on this interface	the RIP multicast address 224.0.0.9 if the **send** parameter is **rip2** or **compatible**. the broadcast address if the **send** parameter is **rip1**.
2) An RIP interface without a current IP address	Use **newipaddress** to set an IP address.	RIP packets from that address on this interface.	The new address.

The **nexthop** parameter specifies the IP address of the best next hop to the destination. In most configurations, the switch is the best next hop when it is the originator of the RIP update. In that case, specify **nexthop=0.0.0.0**. However, in some configurations, the best route to the destination is out another router. In that case, specify the IP address of that router as the next hop. This stops RIP neighbours from unnecessarily sending packets via the switch. See Appendix A of RFC 1723 for an example in which part of a network uses RIP and part uses another routing protocol such as OSPF. The **nexthop** parameter is only valid when you specify **ip** and when **send** is **rip2** or **compatible**. The default next hop is **0.0.0.0**.

The **send** parameter specifies the version of RIP packet to send. If you specify **none**, then the switch does not send RIP packets. If you specify **rip1**, then the switch sends RIPv1 packets. If you specify **rip2**, then the switch sends RIPv2 packets. If you specify **compatible**, then the switch sends RIPv2 packets that are compatible with routers that only receive RIPv1 packets. It achieves compatibility by excluding routes that such routers treat as host routes. The default is **rip1**.

The **receive** parameter specifies the version of RIP packets to receive. If you specify **none**, then the switch does not accept RIP packets from the specified IP address on the specified interface. If you specify **rip1**, then the switch accepts RIPv1 packets. If you specify **rip2**, then the switch accepts RIPv2 packets. If you specify **both**, then the switch accepts both RIPv1 and RIPv2 packets but only keeps routes that conform to RIPv1. Routes conform to RIPv1 if they are classful—for example, the network 172.16.x.x conforms as long as it uses a Class B mask of 255.255.0.0. The switch discards non-conforming routes. The default is **both**.

The **demand** parameter specifies whether the switch uses RIP demand procedures. This means that it:

- only sends RIP updates when it has new routing information
- does not time routes out

Specify **yes** if the connection to the neighbour is a dial-on-demand connection. Configure both ends of the connection to have the same setting. The default is **no**.

The **authentication** parameter specifies how the switch authenticates RIP packets. If you specify **none**, the switch does not authenticate RIP packets. If you specify **password**, the switch uses a plaintext password authenticate RIP packets. If you specify **md5**, the switch uses an encrypted password. You must specify **none** unless the switch uses RIPv2. The default is **none**.

The **password** parameter specifies the password that the switch uses if **authentication** is **password** or **md5**. This parameter is required when authentication is used. The password can be up to 63 characters long, but the switch only uses the first 16 characters.

The **staticexport** parameter specifies whether the switch propagates static routing information from this interface. If you specify **yes**, the switch includes static routes in routing exports. If you specify **no**, the switch omits them. The default is **yes**.

Examples To change the password for an RIP neighbour using authentication, use the command:

```
set ip rip int=vlan2 ip=172.16.248.33 pass=supersecret
```

To change an RIP neighbour so that it uses RIPv2 instead of RIPv1, use the command:

```
set ip rip int=vlan2 ip=172.16.248.33 sen=rip2 rec=rip2
```

Related Commands

add ip rip interface
delete ip rip interface
set ip riptimer
show ip rip

set ip riptimer

Syntax

```
SET IP RIPTimer [FLush=1..4294967295]
   [HOlddown=1..4294967295] [INvalid=1..4294967295]
   [UPdate=1..4294967295]
```

Description

This command sets the values of the global RIP timers in seconds. This command does not change **flush**, **holddown**, or **invalid** time intervals for existing IP RIP routes. Existing routes continue to be invalidated by time intervals previously set.

The **update** parameter sets the time between RIP updates for all interfaces not using RIP on demand. The default is 30 seconds.

The **invalid** parameter sets the time after which the switch deems a route to be invalid because no update has been received. The default is 180 seconds.

The **holddown** parameter sets the time after a route has become invalid during which the switch ignores updates for the route that would normally make the route valid again. The default is 120 seconds.

The **flush** parameter sets the time for when the route is last updated until it is flushed from the route table. This time must equal or exceed the sum of the **invalid** and **holddown** times. The default is 300 seconds.

After a valid update, the **flush** and **invalid** timers are restarted. When the **invalid** timer expires, the route is invalidated and the **holddown** timer started. The **flush** timer continues to run. When the **holddown** timer expires, valid updates for the route result in the switch being reinstated. When the **flush** timer expires, the route is deleted from the route table.

Examples

To force RIP routes to be invalidated and flushed as soon as a single update is missed, use the command:

```
set ip ript in=35 ho=0 fl=35
```

Related Commands

set ip rip interface
show ip rip
show ip riptimer

show ip rip

Syntax `SHow IP RIP [INTerface=`*`interface`*`] [IP=`*`ipadd`*`]`

where:

- *interface* is an interface name formed by concatenating a Layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number in the range. If a logical interface is not specified, 0 is assumed.
- *ipadd* is an IP address in dotted decimal notation.

Description This command displays information about the RIP configuration for IP. The **interface** and **ip** parameters can be used to restrict the display to RIP neighbours on specific interfaces or with specific IP addresses. Valid interfaces are:

- VLAN (such as vlan1, vlan1-1)

Examples To show the RIP configuration for the vlan1 interface, use the command:

```
sh ip rip int=vlan1
```

Related Commands

add ip rip interface
delete ip rip interface
set ip rip interface
show ip in Chapter 13, Internet Protocol (IP)

show ip rip counter

Syntax

```
SHow IP RIP COUnter[={Detail|Summary}]
   [INTerface=interface] [IP=ipadd]
```

where:

- *interface* is an interface name formed by concatenating a Layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number from 0 to 15. If a logical interface is not specified, 0 is assumed.
- *ipadd* is an IP address in dotted decimal notation.

Description

This command displays counters for RIP (Figure 18-4, Table 18-1).

The **counter** parameter specifies whether to display summary or detailed information. If **detail** is specified, counters for each RIP neighbour and total counts for all RIP neighbours are displayed. Otherwise, the total counts for all RIP neighbours are displayed.

The **interface** and **ip** parameters restrict the display to RIP neighbours on specific interfaces or with specific IP addresses. Valid interfaces are:

- VLAN (such as vlan1, vlan1-1)

Figure 18-4: Example output from the **show ip rip counter=detail** command

```
IP RIP Counters:
Interface: eth0
  Input:                          Output:
    inResponses ...... 2568         outResponses ..... 2567
    inTrigRequests ...... 0         outTrigRequests ..... 0
    inTrigResponses ..... 0         outTrigResponses .... 0
    inTrigAcks .......... 0         outTrigAcks ......... 0
    inDiscards .......... 0

IP RIP Counter Summary:
  Input:                        Output:
    inTrigResponses .... 0        outTrigResponses ..... 0
    inTrigAcks ......... 0        outTrigAcks .......... 0
    inDiscards ......... 0
```

Table 18-1: Parameters in output of the **show ip rip counter** command

Parameter	Meaning
Interface	Interface of the RIP neighbour. When multihoming is enabled (two or more logical interfaces have been assigned to a single Layer 2 interface), all interface names include a hyphen and the logical interface number.
IP Address	IP address of the RIP neighbour.
inResponses	Number of response packets received.
inTrigRequests	Number of triggered request packets received.
inTrigResponses	Number of triggered response packets received.
inTrigAcks	Number of triggered acknowledge packets received.

Table 18-1: Parameters in output of the **show ip rip counter** command

Parameter	Meaning
inDiscards	Number of packets discarded. Packets may be discarded due to authentication failure, packets received when receive is disabled, or mismatched sequence number of a triggered acknowledgement.
outResponses	Number of response packets transmitted.
outTrigRequests	Number of triggered request packets transmitted.
outTrigResponses	Number of triggered response packets transmitted.
outTrigAcks	Number of triggered acknowledge packets transmitted.

Related Commands **show ip counter** in Chapter 13, Internet Protocol (IP)
show ip rip

show ip riptimer

Syntax `SHow IP RIPTimer`

Description This command displays the current settings of the global RIP timers (Figure 18-5, Table 18-2).

Figure 18-5: Example output from the **show ip riptimer** command

```
IP RIP timers
Timer name    Default     Current
-----------------------------------
Update        30          5
Invalid       180         15
Holddown      120         60
Flush         300         75
-----------------------------------
```

Table 18-2: Parameters in the output of the **show ip riptimer** command

Parameter	Meaning
Timer name	Timer name.
Default	Default in seconds for the timer.
Current	Current value in seconds for the timer.
Update	Time in seconds between RIP updates for all interfaces not using RIP on demand.
Invalid	Time in seconds after which the switch deems a route to be invalid when no update has been received for the route.
Holddown	Time in seconds after a route has become invalid during which the switch ignores updates for the route that would normally make the route valid again.
Flush	Time in seconds from the last update of a route until the route is flushed from the route table.

Examples To display the current settings of the global RIP timers, use the command:

```
sh ip ript
```

Related Commands **set ip riptimer**

show ip trusted

Syntax `SHow IP TRusted`

Description This command displays the contents of the trusted router table and the state of the enable flag (Figure 18-6). The trusted router table ensures that the switch's routing table is updated only by *trusted* sources of routing information. Other routers are not filtered but their routing information is not used until they are added to the table.

Figure 18-6: Example output from the **show ip trusted** command

```
   Host address
------------------
   172.16.8.33
------------------
```

Related Commands **add ip trusted**
delete ip trusted