

Implementing RIP

The Routing Information Protocol (RIP) is a distance vector Interior Gateway Protocol (IGP) designed to exchange information within an autonomous system (AS) of a small network.

This module describes the concepts and tasks to implement basic RIP routing. Cisco IOS XR software supports a standard implementation of RIP Version 2 (RIPv2) that supports backward compatibility with RIP Version 1 (RIPv1) as specified by RFC 2453.

Table 1: Feature History Table

Feature Name	Release Information	Feature Description
RIPv2	Release 7.4.1	This feature enables RIP as the IGP of your network. RIP broadcasts UDP data packets to exchange routing information in networks that are flat rather than hierarchical, reducing network complexity and network management time.

- Information About Implementing RIP , on page 1
- Routing Information Protocol Next Generation, on page 7
- How to Implement RIP, on page 10
- Configuration Examples for Implementing RIP, on page 13
- Additional References, on page 16

Information About Implementing RIP

While RIPv1 allows only contiguous blocks of hosts, subnets, or networks to be represented by a single route, RIPv2 allows Classless Inter-Domain Routing (CIDR). In addition, RIPv2 supports the following:

- Route summarization
- Variable-length subnet masks (VLSMs)
- Autonomous systems and the use of redistribution
- Multicast address 224.0.0.9 for RIP advertisements

The metric that RIP uses to rate the value of different routes is hop count. The hop count is the number of routers that can be traversed in a route. A directly connected network has a metric of zero; an unreachable

network has a metric of 16. This small range of metrics makes RIP an unsuitable routing protocol for large networks.

Routing information updates are advertised every 30 seconds by default, and new updates discovered from neighbor routers are stored in a routing table.

Only RIP Version 2 (RIP v2), as specified in RFC 2453, is supported on Cisco IOS XR software and, by default, the software only sends and receives RIP v2 packets. However, you can configure the software to send, or receive, or both, only Version 1 packets or only Version 2 packets or both version type packets per interface.

Here are some good reasons to use RIP:

- Compatible with diverse network devices
- Best for small networks, because there is very little overhead, in terms of bandwidth used, configuration, and management time
- Support for legacy host systems

Because of RIP's ease of use, it is implemented in networks worldwide.

Note VRF does not allow configuration of a group applied directly under router RIP. A group can be configured if it is applied globally or under VRF.

Split Horizon for RIP

Normally, routers that are connected to broadcast-type IP networks and that use distance-vector routing protocols employ the split horizon mechanism to reduce the possibility of routing loops. Split horizon blocks information about routes from being advertised by a router out of any interface from which that information originated. This behavior usually optimizes communications among multiple routers, particularly when links are broken.

If an interface is configured with secondary IP addresses and split horizon is enabled, updates might not be sourced by every secondary address. One routing update is sourced per network number unless split horizon is disabled.

Note The split horizon feature is enabled by default. In general, we recommend that you do not change the default state of split horizon unless you are certain that your operation requires the change in order to properly advertise routes.

Route Timers for RIP

RIP uses several timers that determine such variables as the frequency of routing updates, the length of time before a route becomes invalid, and other parameters. You can adjust these timers to tune routing protocol performance to better suit your internetwork needs, by making the following timer adjustments to:

- The rate (time in seconds between updates) at which routing updates are sent
- The interval of time (in seconds) after which a route is declared invalid

- The interval (in seconds) during which routing information regarding better paths is suppressed
- The amount of time (in seconds) that must pass before a route is removed from the RIP topology table
- The amount of time delay between RIP update packets

The first four timer adjustments are configurable by the **timers basic** command. The **output-delay** command changes the amount of time delay between RIP update packets. See Customizing RIP section for configuration details.

It also is possible to tune the IP routing support in the software to enable faster convergence of the various IP routing algorithms and quickly drop back to redundant routers, if necessary. The total result is to minimize disruptions to end users of the network in situations in which quick recovery is essential.

Route Redistribution for RIP

Redistribution is a feature that allows different routing domains, to exchange routing information. Networking devices that route between different routing domains are called boundary routers, and it is these devices that inject the routes from one routing protocol into another. Routers within a routing domain only have knowledge of routes internal to the domain unless route redistribution is implemented on the boundary routers.

When running RIP in your routing domain, you might find it necessary to use multiple routing protocols within your internetwork and redistribute routes between them. Some common reasons are:

- To advertise routes from other protocols into RIP, such as static, connected, OSPF, and BGP.
- To migrate from RIP to a new Interior Gateway Protocol (IGP) such as EIGRP.
- To retain routing protocol on some routers to support host systems, but upgrade routers for other department groups.
- To communicate among a mixed-router vendor environment. Basically, you might use a protocol specific to Cisco in one portion of your network and use RIP to communicate with devices other than Cisco devices.

Further, route redistribution gives a company the ability to run different routing protocols in work groups or areas in which each is particularly effective. By not restricting customers to using only a single routing protocol, Cisco IOS XR route redistribution is a powerful feature that minimizes cost, while maximizing technical advantage through diversity.

When it comes to implementing route redistribution in your internetwork, it can be very simple or very complex. An example of a simple one-way redistribution is to log into a router on which RIP is enabled and use the **redistribute static** command to advertise only the static connections to the backbone network to pass through the RIP network. For complex cases in which you must consider routing loops, incompatible routing information, and inconsistent convergence time, you must determine why these problems occur by examining how Cisco routers select the best path when more than one routing protocol is running administrative cost.

Default Administrative Distances for RIP

Administrative distance is used as a measure of the trustworthiness of the source of the IP routing information. When a dynamic routing protocol such as RIP is configured, and you want to use the redistribution feature to exchange routing information, it is important to know the default administrative distances for other route sources so that you can set the appropriate distance weight.

This table lists the Default Administrative Distances of Routing Protocols.

Table 2: Default Administrative Distances of Routing Protocols

Routing Protocols	Administrative Distance Value
Connected interface	0
Static route out an interface	0
Static route to next hop	1
EIGRP Summary Route	5
External BGP	20
Internal EIGRP	90
OSPF	110
IS-IS	115
RIP version 1 and 2	120
External EIGRP	170
Internal BGP	200
Unknown	255

An administrative distance is an integer from 0 to 255. In general, the higher the value, the lower the trust rating. An administrative distance of 255 means the routing information source cannot be trusted at all and should be ignored. Administrative distance values are subjective; there is no quantitative method for choosing them.

Default-Information Originate

The Default-Information Originate configuration allows the default route (0.0.0.0/0) to be advertised to peers in RIP. The receiving router accepts the default route, and then installs the route in the RIP database and the RIB table.

Starting from IOS XR Release 7.5.2, when a router is configured with the **default-information originate** command, the router ignores any received default routes from RIP peers. So, even if a default route is advertised to the router by a peer, the default route is not accepted or installed into the router's RIP database and the RIB table.

Example:

The following is a sample configuration:

```
router rip
 interface GigabitEthernet0/2/0/0
 !
 default-information originate
!
```

Routing Policy Options for RIP

Route policies comprise series of statements and expressions that are bracketed with the **route-policy** and **end-policy** keywords. Rather than a collection of individual commands (one for each line), the statements within a route policy have context relative to each other. Thus, instead of each line being an individual command, each policy or set is an independent configuration object that can be used, entered, and manipulated as a unit.

Each line of a policy configuration is a logical subunit. At least one new line must follow the **then** , **else** , and **end-policy** keywords. A new line must also follow the closing parenthesis of a parameter list and the name string in a reference to an AS path set, community set, extended community set, or prefix set. At least one new line must precede the definition of a route policy, AS path set, community set, extended community set, or prefix set. One or more new lines can follow an action statement. One or more new lines can follow a comma separator in a named AS path set, community set, extended community set, or prefix set. A new line must appear at the end of a logical unit of policy expression and may not appear anywhere else.

Authentication Using Keychain in RIP

Authentication using keychain in Cisco IOS XR Routing Information Protocol (RIP) provides mechanism to authenticate all RIP protocol traffic on RIP interface, based keychain authentication. This mechanism uses the Cisco IOS XR security keychain infrastructure to store and retrieve secret keys and use it to authenticate in-bound and out-going traffic on per-interface basis.

Keychain management is a common method of authentication to configure shared secrets on all entities that exchange secrets such as keys, before establishing trust with each other. Routing protocols and network management applications on Cisco IOS XR software often use authentication to enhance security while communicating with peers.

Note The Cisco IOS XR software system security component implements various system security features including keychain management. For detailed information on keychain management concepts, configuration tasks, examples, and commands used to configure keychain management, refer the Implementing Keychain Management module in the System Security Configuration Guide, and the Keychain Management Commands module in the System Security Command Reference.

Note The keychain by itself has no relevance; therefore, it must be used by an application that needs to communicate by using the keys (for authentication) with its peers. The keychain provides a secure mechanism to handle the keys and rollover based on the lifetime. The Cisco IOS XR keychain infrastructure takes care of the hit-less rollover of the secret keys in the keychain.

Once you have configured a keychain in the IOS XR keychain database and if the same has been configured on a particular RIP interface, it will be used for authenticating all incoming and outgoing RIP traffic on that interface. Unless an authentication keychain is configured on a RIP interface (on the default VRF or a non-default VRF), all RIP traffic will be assumed to be authentic and authentication mechanisms for in-bound RIP traffic and out-bound RIP traffic will be not be employed to secure it.

RIP employs two modes of authentication: keyed message digest mode and clear text mode. Use the **authentication keychain** *keychain-name* **mode** {**md5**|**text**} command to configure authentication using the keychain mechanism.

In cases where a keychain has been configured on RIP interface, but the keychain is actually not configured in the keychain database or keychain is not configured with MD5 cryptographic algorithm, all incoming RIP packets on the interface will be dropped. Outgoing packets will be sent without any authentication data.

In-bound RIP Traffic on an Interface

These are the verification criteria for all in-bound RIP packets on a RIP interface when the interface is configured with a keychain.

If ..	Then ..
The keychain configured on the RIP interface does not exist in the keychain database...	The packet is dropped. A RIP component-level debug message is be logged to provide the specific details of the authentication failure.
The keychain is not configured with a MD5 cryptographic algorithm...	The packet is dropped. A RIP component-level debug message is be logged to provide the specific details of the authentication failure.
The Address Family Identifier of the first (and only the first) entry in the message is not 0xFFFF, then authentication is not in use...	The packet will be dropped. A RIP component-level debug message is be logged to provide the specific details of the authentication failure.

If ..	Then ..
The MD5 digest in the ‘Authentication Data’ is found to be invalid...	The packet is dropped. A RIP component-level debug message is be logged to provide the specific details of the authentication failure.
Else, the packet is forwarded for the rest of the processing.	

Out-bound RIP Traffic on an Interface

These are the verification criteria for all out-bound RIP packets on a RIP interface when the interface is configured with a keychain.

If ..	Then ..
The MD5 digest in the ‘Authentication Data’ is found to be invalid...	The RIP packet passes authentication check at the remote/peer end, provided the remote router is also configured to authenticate the packets using the same keychain.
The keychain is configured with a MD5 cryptographic algorithm...	The RIP packet passes authentication check at the remote/peer end, provided the remote router is also configured to authenticate the packets using the same keychain.
Else, RIP packets fail authentication check.	

Routing Information Protocol Next Generation

Table 3: Feature History Table

Feature Name	Release Information	Feature Description
Routing Information Protocol New Generation (RIPng)	Release 7.5.2	This feature enables RIP enhancements for IPv6 that include support for IPv6 addresses and prefixes, and the use of the all-RIP-devices multicast group address FF02::9 as the destination address for RIP update messages. This feature functions the same and offers the same benefits as RIP in IPv4.

Based on RFC 2080, the Cisco software implementation of IPv6 RIP, each IPv6 RIP process maintains a local routing table, referred to as a Routing Information Database (RIB). The IPv6 RIP RIB contains a set of best-cost IPv6 RIP routes that the router learns from all its neighboring networking devices. If IPv6 RIP learns the same route from two different neighbors, but with different costs, it stores only the least cost route in the local RIB. The RIB also stores any expired routes that the RIP process is advertising to its neighbors running RIP. IPv6 RIP tries to insert every nonexpired route from its local RIB into the primary IPv6 RIB. If the router has learnt the same route from a different routing protocol with a better administrative distance than IPv6 RIP, the RIP route will not be added to the IPv6 RIB. However, the RIP route will still exist in the IPv6 RIP RIB.

RIPng feature supports the following functionalities:

- Route-filtering on IPv6 RIP interfaces and VRFs using route policies
- Default-Information origination and generation for RIP IPv6
- Poison-reverse and Split-Horizon on RIP IPv6
- Configuration of administrative distances

For more information on RIP, refer the Implementing RIP chapter.

Configuration Examples

```
Router(config)# router rip
Router(config-rip)# vrf vrf1
Router(config-rip-vrf))# address-family ipv6
Router(config-rip-vrf-af)# interface GigabitEthernet 0/0/0/0
```

Running Configuration

```
router rip
 vrf vrf1
 address-family ipv6
 interface GigabitEthernet 0/0/0/0
```

Verification

Verify RIP IPv6 VRF interface information.

```
Router# show rip ipv6 vrf vrf1 interface
Fri Dec  9 17:39:05.855 IST
```

```
GigabitEthernet0/0/0/0
Rip enabled?:               Yes
Out-of-memory state:        Normal
Accept Metric 0:           No
Interface state:            Up
IP address:                 10:10:10::2/64
Metric Cost:                0
Split horizon:              Enabled
Poison Reverse:             Disabled
Socket set options:
    Joined multicast group:    Yes
    LPTS filter set:           Yes

  Total packets received: 0
```

Verify RIP IPv6 process level information, VRF information, Routing Descriptor Block (RDB) information, and interface information.

```
Router# show rip ipv6 vrf vrf1 internal

Fri Dec  9 17:45:02.219 IST

RIP process level information
-----------------------------
Socket descriptor                       63
UDP connected:                         Yes
OOM state:                              1
OOM Severe state time:                  60 seconds
OOM Severer state timer running:       No
Number of routes alloced:               1
Number of paths alloced:                1
Garbage collection time:                300 seconds
RIB Batch buffer: Max buf length:       1434700
                  Max msg length:       524256
                  Max data length:      0
                  Msg length:           0
Number of RIB transit routes:           0
RIB RDB Q size:                         0

RIP VRF information
-------------------
Name:                                   vrf1
ID:                                     0x60000002
NSF:                                   No
NSF Lifetime:                           0
Distance:                               120
DistanceIP Q size:                      0
Active:                                Yes
Added to socket:                       Yes
PSL Send Q size:                        0
OOM Flags:                              0x0
Number of routes alloced:               1
Number of paths alloced:                1

RIP RDB information
-------------------
Table ID:                               0xe0800011
Q flags:                                0x0
Active:                                Yes
Handle:                                 0x2
Old Handle:                             0xffff
Connected Handle:                       0x0
BGP Handle:                             0xffff
Table Ready:                           Yes
```

```
Default Info originate:                No
RedInfo Q size:                         0
RIB Update Q size:                      0
RIB Delete pending:                    No
Triggered Update Q size:                0
Triggered Update batch count:           0
Garbage Q size:                         0
Converged:                             Yes
Convergence timer expired:             Yes
Convergence timer running:             No
Ageout timer running:                  Yes
Garbage timer running:                 Yes (231 seconds left)
OM Triggered Update timer running:     No
WorkQueue Request Q size:               0

RIP Interface information
-------------------------
Cumulative Peer Q size:                 0
```

Verify RIP IPv6 statistics.

```
Router# show rip ipv6 vrf vrf1 statistics
Fri Dec  9 17:45:12.223 IST

RIPv6 statistics:
Total messages sent:        1
Message send failures:      0
Regular updates sent:       0
Queries responsed to:       0
RIB updates:                0
Total packets received:     0
Discarded packets:          0
Discarded routes:           0
Packet received at standby: 0
Number of routes allocated: 1
Number of paths allocated:  1
Route malloc failures:      0
Path malloc failures:       0
```

Verify RIP IPv6 database.

```
Router# show rip ipv6 vrf vrf1 database
Fri Dec  9 17:44:14.868 IST

Routes held in RIP's topology database:
10:10:10::/64
    [0]    directly connected, GigabitEthernet0/0/0/0
RP/0/0/CPU0:PE2#show rip ipv6 vrf vrf1 interface
Fri Dec  9 17:44:20.547 IST

GigabitEthernet0/0/0/0
Rip enabled?:               Yes
Out-of-memory state:        Normal
Accept Metric 0:           No
Interface state:            Up
IP address:                 10:10:10::2/64
Metric Cost:                0
Split horizon:              Enabled
Poison Reverse:             Disabled
Socket set options:
    Joined multicast group:    Yes
    LPTS filter set:           Yes
```

```
  Total packets received: 0
```

Verify RIP IPv6 statistics information.

```
Router# show rip ipv6 vrf vrf1 statistics
Fri Dec  9 17:44:48.670 IST

RIPv6 statistics:
Total messages sent:        1
Message send failures:      0
Regular updates sent:       0
Queries responsed to:       0
RIB updates:                0
Total packets received:     0
Discarded packets:          0
Discarded routes:           0
Packet received at standby: 0
Number of routes allocated: 1
Number of paths allocated:  1
Route malloc failures:      0
Path malloc failures:       0
```

How to Implement RIP

This section contains instructions for the following tasks:

Note To save configuration changes, you must commit changes when the system prompts you.

Enabling RIP

This section enables RIP routing and establishes a RIP routing process.

Before you begin

Although you can configure RIP before you configure an IP address, no RIP routing occurs until at least one IP address is configured.

These commands enable the RIP routing process and designates a neighbor router for exchanging RIP protocol information.

```
Router# configure
Router(config)# router rip
Router(config-rip)# neighbor 172.160.1.2
```

This command configures RIP to send only Version 2 packets to the broadcast IP address rather than the RIP v2 multicast address (224.0.0.9). This command can be applied at the interface or global configuration level.

```
Router(config-rip)# broadcast-for-v2
```

These commands define the RIP routing protocol interface for accepting or sending packets that are RIP v1, RIP v2, or both RIP v1 and RIP v2.

```
Router(config-rip)# interface GigabitEthernet 0/1/0/0
Router(config-rip-if)# receive version 1 2
```

```
Router(config-rip-if)# send version 1 2
Router(config-rip-if)# commit
```

Customizing RIP

This section describes how to customize RIP for network timing and the acceptance of route entries.

```
Router# configure
Router(config)# router rip
```

(Optional) This command enables automatic route summarization of subnet routes into network-level routes. By default, auto-summary is disabled. If you have disconnected subnets, use the **no** keyword to disable automatic route summarization and permit the software to send subnet and host routing information across classful network boundaries.

```
Router(config-rip)# auto-summary
```

(Optional) This adjusts RIP network timers. You can view the current and default timer values from the **show rip** command output.

```
Router(config-rip)# timers basic 5 15 15 30
```

(Optional) This changes the interpacket delay for the RIP updates sent. Use this command if you have a high-end router sending at high speed to a low-speed router that might not be able to receive at that fast a rate.

```
Router(config-rip)# output-delay 10
```

(Optional) This command configures NSF on RIP routes after a RIP process shutdown or restart.

```
Router(config-rip)# nsf
Router(config-rip)# interface GigabitEthernet 0/1/0/0
```

(Optional) This allows the networking device to accept route entries received in update packets with a metric of zero (0). The received route entry is set to a metric of one (1).

```
Router(config-rip-if)# metric-zero-accept
```

(Optional) This command disables the split horizon mechanism. By default, split horizon is enabled. In general, we do not recommend changing the state of the default for the **split-horizon** command, unless you are certain that your application requires a change to properly advertise routes. If split horizon is disabled on a serial interface (and that interface is attached to a packet-switched network), you must disable split horizon for all networking devices in any relevant multicast groups on that network.

```
Router(config-rip-if)# split-horizon disable
```

Enables poison reverse processing of RIP router updates.

```
Router(config-rip-if)# poison-reverse
Router(config-rip-if)#  commit
```

Control Routing Information

This section describes how to control or prevent routing update exchange and propagation. Some reasons to control or prevent routing updates are:

- To slow or stop the update traffic on a WAN link—If you do not control update traffic on an on-demand WAN link, the link remains up constantly. By default, RIP routing updates occur every 30 seconds.
- To prevent routing loops—If you have redundant paths or are redistributing routes into another routing domain, you may want to filter the propagation of one of the paths.
- To filter network received in updates — If you do not want other routers from learning a particular device’s interpretation of one or more routes, you can suppress that information.

- To prevent other routers from processing routes dynamically— If you do not want to process routing updates entering the interface, you can suppress that information.
- To preserve bandwidth—You can ensure maximum bandwidth availability for data traffic by reducing unnecessary routing update traffic.

This configures RIP and specifies a RIP neighbor.

```
Router# configure
Router(config)# router rip
Router(config-rip)# neighbor 172.160.1.2
Router(config-rip)# interface GigabitEthernet 0/1/0/0
```

(Optional) This command suppresses the sending of RIP updates on an interface, but not to explicitly configured neighbors.

```
Router(config-rip-if)# passive-interface
Router(config-rip-if)# exit
```

(Optional) This applies a routing policy to updates advertised to or received from a RIP neighbor.

```
Router(config-rip)# interface GigabitEthernet 0/2/0/0
Router(config-rip-if)# route-policy out
Router(config-rip-if)# commit
```

Creating a Route Policy for RIP

This task defines a route policy and shows how to apply it to a RIP instance. A route policy starts with the **route-policy** command, policy statements are added to the route policy, and it ends with the **end-policy** command. A route policy can control routes that are sent, received, redistributed, and control origination of the default route. A route policy is useful only when it is applied to a routing protocol's routes.

```
Router# configure
Router(config)# route-policy IN-IPv4
```

This command sets the RIP metric attribute.

```
Router(config-rpl)# set rip metric 42
Router(config-rpl)# end-policy
Router(config-rpl)# commit
```

This task applies a routing policy to updates advertised to or received from an RIP neighbor.

```
Router# configure
Router(config)# router rip
Router(config-rip)# route-policy rp1 in
Router(config-rip)# commit
```

Configuring RIP Authentication Keychain

Before you begin

All keychains need to be configured in Cisco IOS XR keychain database using configuration commands described in *Implementing Keychain Management* module of *System Security Configuration Guide* before they can be applied to a RIP interface/VRF.

The **authentication keychain** *keychain-name* and **mode md5** configurations will accept the name of a keychain that has not been configured yet in the IOS XR keychain database or a keychain that has been configured in IOS XR keychain database without MD5 cryptographic algorithm. However, in both these cases, all incoming packets on the interface will be dropped and outgoing packets will be sent without authentication data.

Configuring RIP Authentication Keychain for IPv4 Interface on a Non-default VRF

Configures Keyed message digest (**md5**) or clear text (**text**) authentication for RIP, for the specified VRF and interface.

```
Router(config)# router rip
Router(config-rip)# vrf vrf_rip_auth
Router(config-rip-vrf)# interface POS 0/6/0/0
Router(config-rip-if)# authentication keychain key1 mode md5
OR
Router(config-rip-if)# authentication keychain key1 mode text
Router(config-rip-if)# commit
```

Configuring RIP Authentication Keychain for IPv4 Interface on Default VRF

Configures Keyed message digest (**md5**) or clear text (**text**) authentication for RIP, for the default VRF and the specified interface.

```
Router# configure
Router(config)# router rip
Router(config-rip)# interface POS 0/6/0/0
Router(config-rip-if)# authentication keychain key1 mode md5
OR
Router(config-rip-if)# authentication keychain key1 mode text
Router(config-rip-if)# commit
```

Configuration Examples for Implementing RIP

This section provides the following configuration examples:

Configuring a Basic RIP Configuration: Example

The following example shows two Gigabit Ethernet interfaces configured with RIP.

```
interface GigabitEthernet0/6/0/0
ipv4 address 172.16.0.1 255.255.255.0
!
interface GigabitEthernet0/6/0/2
ipv4 address 172.16.2.12 255.255.255.0
!
router rip
interface GigabitEthernet0/6/0/0
!
interface GigabitEthernet0/6/0/2
!
```

Configuring RIP on the Provider Edge: Example

The following example shows how to configure basic RIP on the PE with two VPN routing and forwarding (VRF) instances.

```
router rip
interface GigabitEthernet0/6/0/0
!
vrf vpn0
interface GigabitEthernet0/6/0/2
!
!
vrf vpn1
interface GigabitEthernet0/6/0/3
!
```

Adjusting RIP Timers for each VRF Instance: Example

The following example shows how to adjust RIP timers for each VPN routing and forwarding (VRF) instance.

For VRF instance vpn0, the **timers basic** command sets updates to be broadcast every 10 seconds. If a router is not heard from in 30 seconds, the route is declared unusable. Further information is suppressed for an additional 30 seconds. At the end of the flush period (45 seconds), the route is flushed from the routing table.

For VRF instance vpn1, timers are adjusted differently: 20, 60, 60, and 70 seconds.

The **output-delay** command changes the interpacket delay for RIP updates to 10 milliseconds on vpn1. The default is that interpacket delay is turned off.

```
router rip
interface GigabitEthernet0/6/0/0
!
vrf vpn0
interface GigabitEthernet0/6/0/2
!
timers basic 10 30 30 45
!
vrf vpn1
interface GigabitEthernet0/6/0/3
!
timers basic 20 60 60 70
output-delay 10
!
```

Configuring Redistribution for RIP: Example

The following example shows how to redistribute Border Gateway Protocol (BGP) and static routes into RIP.

The RIP metric used for redistributed routes is determined by the route policy. If a route policy is not configured or the route policy does not set RIP metric, the metric is determined based on the redistributed protocol. For VPNv4 routes redistributed by BGP, the RIP metric set at the remote PE router is used, if valid.

In all other cases (BGP, IS-IS, OSPF, EIGRP, connected, static), the metric set by the **default-metric** command is used. If a valid metric cannot be determined, then redistribution does not happen.

```
route-policy ripred set rip-metric 5
end-policy
!

router rip vrf vpn0
interface GigabitEthernet0/6/0/2
!
redistribute connected default-metric 3
!
vrf vpn1
interface GigabitEthernet0/6/0/3
!
redistribute bgp 100 route-policy ripred redistribute static
default-metric 3
```

Configuring Route Policies for RIP: Example

The following example shows how to configure inbound and outbound route policies that are used to control which route updates are received by a RIP interface or sent out from a RIP interface.

```
prefix-set pf1 10.1.0.0/24
end-set
!
prefix-set pf2 150.10.1.0/24
```

```
end-set
!
route-policy policy_in
if destination in pf1 then pass
endif end-policy
!
route-policy pass-all pass
end-policy
!
route-policy infil
if destination in pf2 then add rip-metric 2
pass endif
end-policy
!
router rip
interface GigabitEthernet0/6/0/0 route-policy policy_in in
!
interface GigabitEthernet0/6/0/2
!
route-policy infil in route-policy pass-all out
```

Configuring Passive Interfaces and Explicit Neighbors for RIP: Example

The following example shows how to configure passive interfaces and explicit neighbors. When an interface is passive, it only accepts routing updates. In other words, no updates are sent out of an interface except to neighbors configured explicitly.

```
router rip
interface GigabitEthernet0/6/0/0 passive-interface
!
interface GigabitEthernet0/6/0/2
!
neighbor 172.17.0.1
neighbor 172.18.0.5
!
```

Controlling RIP Routes: Example

The following example shows how to use the **distance** command to install RIP routes in the Routing Information Base (RIB). The **maximum-paths** command controls the number of maximum paths allowed per RIP route.

```
router rip
interface GigabitEthernet0/6/0/0 route-policy polin in
!
distance 110
maximum-paths 8
!
```

Configuring RIP Authentication Keychain: Example

This example shows how to apply an authentication keychain on a RIP default VRF interface:

```
router rip
interface POS0/6/0/0
authentication keychain key1 mode md5
!
end
```

This example shows how to apply an authentication keychain on a RIP non-default interface:

```
router rip
vrf rip_keychain_vrf interface POS0/6/0/0
authentication keychain key1 mode md5
```

```
!
end
```

Additional References

The following sections provide references related to implementing RIP.

Related Documents

Related Topic	Document Title
RIP commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Routing Command Reference Guide*
Site of Origin (SoO) support for RIP feature information	*Implementing MPLS Traffic Engineering on* module in the *MPLS Configuration Guide for Cisco NCS 5500 Series Routers*
Cisco IOS XR getting started documentation	
Information about user groups and task IDs	*Configuring AAA Services on* module in the *System Security Configuration Guide for Cisco NCS 5500 Series Routers*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
—	To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: https://mibs.cloudapps.cisco.com/ITDIT/MIBS/servlet/index

RFCs

RFCs	Title
RFC 2453	RIP Version 2

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport