Chapter 11
Routing Information Protocol (RIP)

11.1 Overview

One of the most widely used interior gateway protocols is the Routing Information Protocol (RIP). RIP is an implementation of a distance-vector, or Bellman-Ford, algorithm. RIP classifies routers as active and passive (silent). Active routers advertise their routes (reachability information) to others; passive routers listen and update their routes based on advertisements, but do not advertise. Typically, routers run RIP in active mode, while hosts use passive mode.

A router running RIP in active mode sends updates at set intervals. Each update contains paired values, where each pair consists of an IP network address and an integer distance to that network. RIP uses a hop count metric to measure the distance to a destination. In the RIP metric, a router advertises directly connected networks at a metric of 1 by default. Networks that are reachable through one other gateway are 2 hops, etc. Thus, the number of hops or hop count along a path from a given source to a given destination refers to the number of gateways that a datagram would encounter along that path. Using hop counts to calculate shortest paths does not always produce optimal results. For example, a path with a hop count 3 that crosses three Ethernets may be substantially faster than a path with a hop count 2 that crosses two slow-speed serial lines. To compensate for differences in technology, many routers advertise artificially high hop counts for slow links.

RIP dynamically builds on information received through RIP updates. When started up, RIP issues a request for routing information and then listens for responses to the request. If a system configured to supply RIP hears the request, it responds with a response packet based on information in its routing database. The response packet contains destination network addresses and the routing metric for each destination.

When a RIP response packet is received, the routing daemon takes the information and rebuilds the routing database, adding new routes and “better” (lower metric) routes to destinations already listed in the database. RIP also deletes routes from the database if the next router to that destination reports that the route contains more than 15 hops, or if the route is deleted. All routes through a gateway are deleted if no updates are received from that gateway for a specified time period. In general, routing updates are issued every 30 seconds. In many implementations, if a gateway is not heard from for 180 seconds, all routes from that gateway are deleted from the routing database. This 180-second interval also applies to deletion of specific routes.

RIP version 2 (more commonly known as RIP II) adds additional capabilities to RIP. Some of these capabilities are compatible with RIP I and some are not. To avoid supplying information to RIP I routes that could be misinterpreted, RIP II can use only non-compatible features when its packets are multicast. On interfaces that are not capable of IP multicast, RIP-I-compatible packets are used that do not contain potentially confusing information.

Some of the most notable RIP II enhancements are:

- Next hop
- Network mask
- Authentication
- RIP tag field

These features in RIP I and II are contrasted in the following paragraphs.

Next hop

With RIP II, a router can advertise a next hop other than itself. Next hop is useful when advertising a static route to a dumb router that does not run RIP, because it avoids having packets that are passed through the dumb router from having to cross a network twice. Because RIP I routers will ignore next hop information in RIP II packets, packets might cross a network twice, which is exactly what happens with RIP I. Next hop information is provided in RIP-I-compatible RIP II packets.

Network mask

RIP I assumes that all subnetworks of a given network are classful (Class A,B,C). RIP I uses this assumption to calculate the network masks for all routes received. This assumption prevents subnets with classless netmasks from being included in RIP packets. RIP II adds the ability to specify the network mask with each network in a packet. Because RIP I routers will ignore the network mask in RIP II packets, their calculation of the network mask will quite possibly be wrong. For this reason, RIP-I-compatible RIP II packets must not contain networks that would be misinterpreted. These networks must be provided only in native RIP II packets that are multicast.

RIP I derives the network mask of received networks and hosts from the network mask of the interface via which the packet was received. If a received network or host is on the same natural network as the interface over which it was received, and that network is subnetted (the specified mask is more or less specific than the natural netmask), the interface’s subnet mask is applied to the destination. If bits outside the mask are set, it is assumed to be a host; otherwise, it is assumed to be a subnet. On point-to-point interfaces, the netmask is applied to the remote address. The netmask on these interfaces is ignored if it matches the natural network of the remote address, or is all ones. Unlike previous releases, the zero subnet (a subnetwork that matches the natural network of the interface, but has a more specific, or longer, network mask) is advertised. If this is not desirable, a route filter may be used to reject it.

Authentication

RIP II packets may contain one of two types of authentication strings that may be used to verify the validity of the supplied routing data. Authentication may be used in RIP-I-compatible RIP II packets, but be aware that RIP I routers will ignore these packets (unless **nocheckzero** is selected). The first method is a simple password in which an authentication key of up to 16 characters is included in the packet. If this key does not match what is expected, the packet will be discarded. This method provides very little security because it is possible to learn the authentication key by watching RIP packets.

The second method uses the MD5 algorithm to create a crypto-checksum of a RIP packet and an authentication key of up to 16 characters. The transmitted packet does not contain the authentication key itself; instead, it contains a crypto-checksum, called the “digest”. The receiving router will perform a calculation using the correct authentication key and discard the packet if the digest does not match. In addition, a sequence number is maintained to prevent the replay of older packets. This method provides a much stronger assurance that routing data originated from a router with a valid authentication key.

Two authentication methods can be specified per interface. Packets are always sent using the primary method, but received packets are checked with both the primary and secondary methods before being discarded. In addition, a separate authentication key is used for non-router queries.

RIP tag field

RIP tags are supported by this implementation.

11.2 RIP Syntax

```
rip ( on | off ) [ {
    broadcast | nobroadcast ;
    ignorehostroutes ;
    expire-time expire_time ;
    update-time update_time ;
    nocheckzero ;
    preference preference ;
    defaultmetric metric ;
    query authentication none ;
    query authentication simple password ;
    query authentication md5 password ;
    query authentication md5 key password id number [ {
        [ start-accept YYYY/MM/DD HH:MM ] ;
        [ stop-accept YYYY/MM/DD HH:MM ] ;
        [ start-generate YYYY/MM/DD HH:MM ] ;
        [ stop-generate YYYY/MM/DD HH:MM ] ;
    } ; ]
    interface interface_list
        [ noripin | ripin ]
        [ noripout | ripout ]
        [ metricin metric ]
        [ metricout metric ]
        [ version 1 | ( version 2 [ multicast | broadcast ] ) ]
        [ secondary ] authentication none ;
        [ secondary ] authentication simple password ;
        [ secondary ] authentication md5 password ;
        [ secondary ] authentication md5 key password id number [ {
            [ start-accept YYYY/MM/DD HH:MM ] ;
            [ stop-accept YYYY/MM/DD HH:MM ] ;
            [ start-generate YYYY/MM/DD HH:MM ] ;
            [ stop-generate YYYY/MM/DD HH:MM ] ;
         } ; ]
    ;
    trustedgateways gateway_list ;
```

```
        sourcegateways gateway_list ;
        traceoptions trace_options ;
    } ] ;
```

More detailed descriptions of these commands can be found on page 57 of the *Command Reference Guide*. See “Chapter 32 Route Exportation” on page 145 for information about exporting and RIP.

11.3 RIP Sample Configurations

11.3.1 RIP version 1 with Broadcast

This configuration broadcasts RIP updates and listens for RIP updates on a single interface. Note that more than one interface must be enabled and IP forwarding must be enabled in order to broadcast RIP updates by default. **`broadcast`** forces broadcast of RIP updates.

```
rip yes {
        traceoptions all ;
        broadcast ;
        interface fxp0 ripin ripout ;
        interface fxp1 noripin noripout ;
        interface fxp2 noripin noripout ;
};
```

11.3.2 RIP version 2 with Broadcast and Simple Authentication

This configuration broadcasts RIP updates and listens for RIP updates on a single interface. Version 2 and simple authentication are used.

```
rip yes {
        traceoptions all ;
        broadcast ;
        interface fxp0 ripin ripout version 2 authentication simple
"foo" ;
        interface 10.3.25.25 noripin noripout ;
};
```

11.3.3 RIP version 2 with Multicast and Simple Authentication

This configuration enables version 2 multicast on two interfaces. Note that multicast is the default if version 2 is specified.

```
rip yes {
        traceoptions all ;
        interface fxp0 version 2 authentication simple "foo" ;
        interface fxp1 version 2 authentication simple "bar" ;
};
```

11.3.4 RIP version 2 with Broadcast and MD5 Authentication

This configuration enables a single interface for version 2 RIP with an MD5 authentication key. By default, the key has an infinite lifetime.

```
rip yes {
        traceoptions all;
        broadcast ;
        interface fxp0 version 2 authentication md5 key “foo” id 20;
};
```

11.3.5 RIP version 2 with Source and Trusted Gateways

This configuration uses **`sourcegateways`** and **`trustedgateways`** to enable GateD to announce RIP to a single gateway and receive RIP from a single gateway.

```
rip yes {
        traceoptions all ;
        nobroadcast ;
        sourcegateways 10.131.10.12 ;
        trustedgateways 10.131.10.12 ;
        interface 10.131.10.16 version 2 authentication simple "foo";
};
```