This book was originally developed by Juniper Networks Inc. in conjunction with Sybex Inc. It is being offered in electronic format because the original book (ISBN: 0-7821-4071-8) is now out of print. Every effort has been made to remove the original publisher's name and references to the original bound book and its accompanying CD. The original paper book may still be available in used book stores or by contacting, John Wiley & Sons, Publishers. www.wiley.com.

Copyright © 2003-6 by Juniper Networks Inc. All rights reserved.

This publication may be used in assisting students to prepare for a Juniper JNCIA exam but Juniper Networks Inc. cannot warrant that use of this publication will ensure passing the relevant exam.
The Routing Information Protocol (RIP)

JNCIA EXAM OBJECTIVES COVERED IN THIS CHAPTER:

✓ Describe the basic characteristics of the RIP protocol
✓ Describe the enhancements included in the RIPv2 specification
✓ Define the message types used in RIP
✓ List the steps required to advertise routes into a RIP network
In this chapter, you’ll learn about the most basic of IP routing protocols: the Routing Information Protocol (RIP). We begin with an overview of the protocol—its features and its limitations. Next, we cover the protocol’s theory of operation, examine how neighbors are acquired, describe the two RIP packet types, and show you how RIPv2 solves some of the limitations of RIPv1. Finally, we look at how you can configure the JUNOS software to use RIP in the network.

In today’s environment, fewer and fewer sites are running RIP for reasons that we discuss in this chapter. First, however, you need to have a basic understanding of how the protocol works.

Overview of RIP

RIP is a dynamic routing protocol that uses the shortest path computation algorithm known as the Bellman-Ford algorithm. RIP routers exchange routing information with their neighbors in the form of routing updates. These updates contain the length of the path to the destination (distance), as well as the address of the next router along the path (vector). RIP is also considered to be an Interior Gateway Protocol (IGP) because it is meant for intra-autonomous system routing. It is best used for homogenous networks (having similar interface speeds) of small to moderate size.

RIP has its roots in the protocols developed by Xerox at its Palo Alto Research Center (PARC). In the mid-1970s, Xerox developed the PARC Universal Protocol (PUP) for use on its experimental 3Mbps network that eventually became Ethernet. PUP was routed by another protocol called the Gateway Information Protocol (GWINFO). PUP evolved into the Xerox Network System (XNS) protocol suite, and GWINFO became the XNS Routing Information Protocol (XNS RIP). XNS RIP provided the foundation for other vendors’ protocols, such as Novell’s IPX RIP, AppleTalk’s Routing Table Maintenance Protocol (RTMP), and, of course, IP RIP.

The evolving protocol gained large popularity when it was refined and shipped as the protocol accompanying the Berkeley Software Distribution (BSD) release 4.2 of Unix in 1982. More recent versions of the Unix operating system implement the protocol as either routed or gated. RIP’s popularity led to its eventual standardization in 1988 by the Internet Engineering Task Force (IETF).

Since RIP was one of the first distance-vector protocols in the Internet, it will be helpful to examine its features in greater detail. Let’s do that now.
RIP Standards

The use of RIP in IP networks is specified and standardized by three Requests for Comment (RFCs):

- RFC 1058, “Routing Information Protocol”
- RFC 2082, “RIP-2 MD5 Authentication”
- RFC 2453, “RIP Version 2”

RFC 1058 is the basic RFC of IP RIP (RIPv1). It specifies the limitations of the protocol, the operation of distance-vector algorithms, and features designed to stabilize the protocol. In addition, it describes the specifics of message formats, timers, and message processing. Compatibility issues, too, are discussed. Of course, this original RIP specification is over 10 years old. Two newer RFCs provide updates to the protocol’s specifications.

RFC 2082 is the first RFC 2082, which describes the use of an MD5 hash algorithm as an authentication mechanism. MD5 does provide a good security feature to protect your network, but it’s only available with RIPv2.

If you want to bother with only one RFC about RIP, RFC 2453 is the one to read. This second update is the latest document to describe the extensions to the protocol. In addition to the RIPv2 enhancements, RFC 2453 contains virtually all the information from the original RIP specification, RFC 1058.

To access a particular RFC, go to http://www.rfc-editor.org and select the RFC Search link. You are presented with a number of criteria upon which to search. Simply enter your search criteria to find the RFC you’re looking for.

RIP Features

RIP was designed to be a very “lightweight” protocol. That means that it can be operated on very basic machines (Unix hosts, for example) as well as sophisticated Internet routers. This portability between systems is due in large part to the features of the protocol:

- RIP routers use UDP (User Datagram Protocol) port 520 to send messages to their neighbors. UDP is sometimes referred to as “send and pray” because there is no reliable delivery mechanism inherent in UDP. In addition, the RIP specification does not provide its own mechanism for reliable delivery.
- RIP routers exchange packets with their neighbors; the packets contain network addresses and an associated metric, or hop count, to reach those addresses.
The metric (hop count) is a measure of the distance from the source of the update to the destination network. Each hop in the path is assigned a value, which is typically 1. Most implementations (including the JUNOS software) let you alter this value.

RIPv1 sends messages to the broadcast address (255.255.255.255) on broadcast-capable networks such as Ethernet. This means the RIP messages are sent to all devices on the subnet—including the RIP routers and all other hosts on the subnet, many of which do not need the information.

RIP sends routing update messages at regular intervals (30 seconds) and when warranted by network topology changes.

RIP uses an infinity metric to prevent routing loops, as described in the following section.

**Infinity Metric**

To prevent routing loops, the RIP protocol depends on a function known as “counting to infinity.” A maximum metric (the infinity metric) is defined within the protocol, and all routes with a larger metric are deemed unusable. For RIP, the maximum hop count is defined as 15. If a router receives a RIP update with a metric value over 15 (that is, 16 or greater) after it is incremented, the router must throw the update away and the destination is considered unreachable.

For a closer look at the infinity metric, see Figure 5.1. Here, the routers are not implementing additional route-loop prevention measures such as split horizon and triggered updates. They are relying only on the infinity metric to stop a loop.

**FIGURE 5.1** Counting to infinity: initial route advertisements

Merlot advertises the 10.1.1.0 /24 subnet to Cabernet with a metric of 1, which is in turn advertised to Riesling with a metric of 2. At this point, all three of the routers have converged on a common view of the network.

The route to the 10.1.1.0 /24 network disappears from Merlot’s routing table, but before it can update its neighbors, it receives an update from Cabernet, which advertises the 10.1.1.0 /24 subnet with a metric of 2. Merlot now believes that it can reach the “lost” 10.1.1.0 /24 subnet through Cabernet and installs this route in its local routing table with a metric of 3.

Merlot now advertises its entire routing table to its neighbors again and informs Cabernet that it can reach the 10.1.1.0 /24 subnet with a metric of 3. From the perspective of Cabernet, Merlot is the current next hop for that subnet and Merlot is now advertising a larger metric of 3. Cabernet then updates its routing table to reflect this new metric. It will then advertise this new metric of 4 to all neighbors in its next update. This current situation can be seen in Figure 5.2.
RIP Operations

Now that we have some basic terminology under our belts, we can begin looking at how RIP operates in a networking environment. We examine how a router sends and receives routing updates and look at some stability features built into RIP. A short discussion of various distance-vector timers follows. Finally, we explore some limitations of RIP and explain why many users choose not to run RIP in a large-scale network.

Input Processing

RIP routers can receive two types of messages from their neighbors: Request messages and Response messages. Each of these message types performs a specific function.

Request Messages

RIP routers receive Request messages from their neighbors. The purpose of a Request message is to ask for all or some part of the local router’s current routing table. An optimum time to observe a Request message is when a router first boots up or its routing process is restarted. In
this situation, the router will send a Request message to all its neighbors in order to populate its routing table as quickly as possible.

RIP routers process incoming Request messages entry by entry. Often, a Request message has a single entry in it with a metric of 16 and an address family identifier field that contains all zeros. This message translates into “Send me your entire routing table.” A Request message, however, may contain one or more specific route entries. In this case, the local router consults its routing table for each of the destinations listed. If the destination is found, information about the route is sent by unicast back to the requesting router in a Response message. If the destination is not found, the metric field for the route is set to infinity (16) and is sent by unicast back to the requesting router.

Response Messages

A RIP router receives Response messages for one of three different reasons:

- In response to a Request message generated by the local router
- A regular (unsolicited) Response message sent by a neighbor
- A triggered update Response message sent by a neighbor

Regardless of the reason, all Response messages are processed in the same way. The first step performed by the local router after receiving a Response message is to validate the message by checking certain fields. For example, is the message from a valid RIP neighbor? Is the source address of the update on a directly connected network? If the answer to each of these questions is yes, then the message is accepted. Some validity checks cause the router to reject the updates. For instance, is the packet from one of the local router’s own interfaces? It would not be good to process one of our own Response messages!

Next, the listed route entries are processed one by one. Each entry is checked for validity. Is the destination address of the update valid? Is the metric between 1 and 16?

Once an individual route entry is validated, the metric for that route is updated by adding to it the cost of the interface on which the message was received. The interface cost is usually 1. If the final metric is now larger than infinity, then the infinity metric of 16 becomes the metric.

The local router then consults its routing table to determine if there is already an explicit route to that destination. If no entry exists, then the received route is added to the table, except when the metric is currently set to infinity.

If the route does currently exist, the next-hop address of the route is checked. If the next hop of the received route is equal to the address in the routing table, the local router reinitializes the Timeout timer and compares the metrics of the two routes.

If the metric values are the same or the metric of the received route is higher, the existing route remains in the table and the router stops processing this route entry. Should the metric of the received route be lower than the current metric, then the router installs the new copy of the route with the lower metric and sends a triggered update. When the metric of the received route is the infinity metric, the local router starts the process to delete the route from the table.
Output Processing

RIP routers can both generate and send Request and Response messages. The use of Request messages for output is identical to that described in the preceding section on input processing. A router may send a Response message to a neighbor in response to the following triggers:

- Processing of a Request message
- Expiration of the Update timer every 30 seconds (we explain Update timers in the section “Timers” later in this chapter)
- Resulting from a network topology change

Prior to sending the Response message, each route in the current routing table is examined. If the route should be included in the update due to local administrative controls, then the destination address and metric of the route are added to the message. Because of limitations in the size of a RIP packet, the Response message can contain no more than 25 entries, or routes. When there are more than 25 routes to be sent, the router must send multiple packets.

Stability Features

Modern RIP implementations include several features that add stability to the protocol. The features are all directed at minimizing the network convergence time (the time when all routers in the network have the same view). These are important features of the protocol.

Split Horizon

If Router A sends a route update to Router B that describes all of the entries in Router A’s routing table, is it necessary for Router B to send that same information back to Router A? The answer is of course not; it is a waste of bandwidth. The concept of split horizon fixes this “problem.” When the Update timer expires and a Response message is generated, split horizon prevents the local router from including any routes learned from a neighbor on the interface from which the message is being sent out. This technique also minimizes the time it can take to “count to infinity” because it prevents a router from advertising reachability to a down link through a neighbor that is actually further away and does not yet know the link is down.

The concept of split horizon is one simple method to solve the routing loop that we formed earlier in the chapter. In Figure 5.3, we have the same three routers—Merlot, Cabernet, and Riesling—but this time the routers are using split horizon to help prevent routing loops. The result is that Cabernet will never advertise the 10.1.1.0 /24 subnet back to Merlot. When the 10.1.1.0 /24 subnet disappears from Merlot’s routing table, Merlot will not receive any updates for this route and will then be able to remove it from its routing table without forming a loop.

Split Horizon with Poisoned Reverse

The split horizon function can have additional functionality when the concept of poisoned reverse is added. In this case, instead of never advertising a route back to the neighbor it was learned from, the router advertises it with an infinity metric. This “poisoning” of the route
The Routing Information Protocol (RIP) provides robustness to the protocol because each router now has more information to use to make a routing decision. In the case of our routers in Figure 5.3, when 10.1.1.0 /24 is lost from its routing table, Merlot will know that Cabernet is not a viable alternate route since Cabernet will be actively advertising the 10.1.1.0 /24 subnet to Merlot with a metric of 16.

The drawback of this process adds unnecessary protocol traffic to the network links. As such, the JUNOS software does not utilize this RIP feature.

**FIGURE 5.3** Split horizon in action

**Triggered Updates**

If a RIP router were to wait for a regular update interval to occur before notifying any neighbors of topology changes, the wait could be as long as 30 seconds. Clearly, this is not an optimum solution to topology-based notifications. The concept of triggered updates relaxes this timer “problem.” In this case, a topology change for the local router will be announced immediately to all active neighbors. Each downstream RIP router will in turn send an almost immediate update to its neighbors using triggered updates. We say “almost immediate” because each router implements a short timer with triggered updates, which prevents the router from immediately sending multiple triggered updates individually. When a triggered update is received, the router waits for a short period of time, between 1 and 5 seconds, to see if other triggered updates arrive. If this occurs, the local router will bundle the updates and send them together to all neighbors. This reduces the computational burden of receiving and processing Response messages while helping speed the convergence of the RIP network.
Hold-Downs

The concept of a hold-down is not specifically mentioned in the RFCs for RIP. They are, however, widely used in routing implementations, including the JUNOS software. This real-world application is derived from one advantage of a hold-down—it prevents the propagation of bad routing information throughout the network. The Hold-Down timer is used when an update is received by the local router that contains a route from a peer with a higher metric (other than infinity) than the one in the current routing table. In addition, the current active route is using this peer as a next hop. In this case, a Hold-Down timer is started and the new information (with a higher metric) is not sent to the rest of the router’s neighbors until the timer expires. In a sense, a hold-down has the opposite effect of a triggered update in that it slows down convergence in an attempt to ensure that only “good” routing knowledge is propagated.

Timers

RIP uses a number of timers in its operation, among them the Update timer, the Hold-Down timer, the Timeout timer, and the Garbage Collection timer.

A RIP router uses an Update timer to advertise its complete routing table (less split horizon) to all its neighbors. The JUNOS software uses 30 seconds as the default Update timer. Each timer cycle on the router is offset by a small random amount as it runs due to problems experienced in the past.

Randomizing Timers

Administrators of some networking environments have found that routers on a broadcast network tend to become synchronized in their updates. This has meant that all the RIP routers would process their Response messages simultaneously and attempt to transmit them all to the network. Of course, basic Ethernet theory tells us that only one packet at a time can be transmitted to the network. The multiple transmission attempts led to excessive collisions on the network, occurring every 30 seconds. Since collisions are something to avoid in a broadcast environment, vendors have randomized the timers within their software. This randomizing of timers occurs with virtually all JUNOS software timers, so we always mention the “intended” timer value without qualifying the statement by mentioning the timer randomization. What does this mean to you? If you are “sniffing” traffic on a network, don’t be surprised when the actual time between packets varies on a regular basis.

We discussed the functionality of the Hold-Down timer in great detail in the previous section. Recall that it is the time a router should wait before advertising a route with a higher metric. The time value used for the Hold-Down timer within the JUNOS software is 180 seconds and is not configurable.
A RIP router uses both the Timeout and the Garbage Collection timers within the current routing table. The **Timeout timer** is used to ensure that the copy of the route is valid and usable. When the route is first installed in the table, this timer is initialized to 120 seconds, its maximum value. The timer value is updated when a Response message from a neighbor is processed and the route is maintained in the table. In this case, the timer is reset to 120. Since the Update timer runs every 30 seconds, the Timeout timer’s maximum value of 120 seconds means that four RIP updates can be missed before the local router declares the route to be unusable. At this point, the metric is changed to infinity (16) and the route is advertised to all RIP neighbors.

Locally, the route is not removed from the routing table yet, and the Garbage Collection timer is initialized to a value of 0. The **Garbage Collection timer** runs to a maximum non-configurable value of 180 seconds, at which time the route will be removed from the routing table. These two timers allow the local router to have plenty of time to ensure that all downstream neighbors also know the route is down and unusable.

### Real World Scenario

#### RIP Timers in Action

While it is nice to define the various RIP timers and their values, it often doesn’t make sense until you are talking about a real route. Let’s explore how these timers interact with the 10.0.0.0 /8 route on a Juniper Networks router.

In our example, Router A sends the 10.0.0.0 /8 route with a metric of 5 to Router B in a RIP Response message. Router A sends this route every 30 seconds as defined by the Update timer. When Router B receives the route, it sets the Timeout timer to 120 seconds and begins to decrement it to 0. The next message from Router A (about 30 seconds later) prompts Router B to reset the Timeout timer to 120 seconds. Should Router A stop sending RIP messages, Router B waits until four update cycles have been missed, at which point the Timeout timer is at 0. Router B starts both the Hold-Down and Garbage Collection timers (both 180 seconds long), sets the metric for 10.0.0.0 /8 to 16, and places the route into **holddown** in the routing table. As the timers reach their maximum value of 180 seconds, the route is removed from the local routing table.

During the period of time in which a RIP route is in **holddown**, other RIP neighbors might advertise the same 10.0.0.0 /8 route to Router B. There are three variations on how Router B receives these routes and on its response to those messages. First, Router B receives the 10.0.0.0 /8 route with a lower metric than 5 from a neighbor on any operational interface. Router B immediately installs the new version of the route, sets the Timeout timer to 120 seconds, and advertises the new metric to all downstream neighbors.
Second, Router B receives the 10.0.0.0 /8 route with a higher metric than 5 on a different interface than the route currently in hold-down. Router B waits for the Garbage Collection and Hold-Down timers to reach 180 seconds before using this new route advertisement. As the timers expire, the old version of the route is deleted and the next advertisement of the 10.0.0.0 /8 route from the new neighbor is installed in the routing table. The Timeout timer is set to 120 seconds and the route is advertised downstream.

Finally, the 10.0.0.0 /8 route is received by Router B with a higher metric than 5 on the same interface that the hold-down route was received on. Router B immediately begins using this new version of the route. The Timeout timer is set to 120 seconds and the route is advertised to all downstream neighbors.

Limitations

So far, so good. We’ve looked at the RIP standards and some real-world enhancements in the form of timers and stability features. But why do many administrators still refuse to use RIP on a large scale? The answer lies in the various limitations of the protocol. Let’s examine a few of them in some detail.

**Scalability**  RIP does not scale well for large networking environments. One issue is the maximum hop count used (discussed next). Another issue is the use of the 255.255.255.255 broadcast address for Response message updates in RIP version 1. On broadcast networks, this is quite disruptive to other IP (non-RIP) hosts.

**Small hop count limit**  Sixteen hops is the defined infinity metric that denotes an unreachable or unusable subnet. This value limits the size or “diameter” of the networks that can be built using RIP.

**Slow convergence**  Although triggered updates can help advertise new information into RIP, the timers can have the opposite effect. When a route needs to be removed from the protocol, the timer values for the Hold-Down, Timeout, and Garbage Collection timers can mean that a topology change at one end of the network may not be known at the other end of the network for several minutes.

**Suboptimal routing**  Since RIP routers utilize only the hop count as the metric, some suboptimal routing may occur. This occurs because hop count does not allow for dissimilar bandwidths, fewer delays, or less congestion on other alternate paths to a destination. When these alternate paths are available, RIP will always pick the one with the smallest hop count regardless of the interface speeds of the other path.

**Nonhierarchical design**  As the size of the RIP routing domain grows larger and approaches the maximum diameter of 15 routers, there is no mechanism to divide the domain into smaller, more manageable subdomains.
The Routing Information Protocol (RIP)

Classful routing protocol  RIPv1 is designed as a classful routing protocol and it does not transmit a subnet mask in the messages. The subnet masks must be assumed by one of two methods—either the subnet mask of the receiving interface address is used, or the classful network mask (that is, Class A, B, or C) is used.

Noncontiguous networks that are not visible  If a RIPv1 network has been segmented so that different subnets of the same classful network (Class B) are separated by a second classful network (Class A), the two separated sections of the Class B network will not have detailed visibility of each other. This occurs due to the automatic summarization described above.

The JUNOS software implementation does not summarize routes based on classful boundaries. Received routes are assigned a 32-bit host route mask prior to being placed in the routing table.

Variable-length subnet masks  RIPv1 routers do not support the ability to variably subnet the network using different mask lengths. This greatly limits the ability to conserve IP addressing space within a network.

Weak security  RIPv1 has no authentication mechanism available to verify that received Response messages have been sent by a “trusted” source.

Packet Types

Earlier we stated that RIP uses two packet types to send information to its neighbors:

Request  Sent to a neighbor to request that the neighbor send all or part of the neighbor’s routing table. Requests are usually seen only at router startup or if the routing protocol is restarted.

Response  Sent as a reply to a neighbor’s request packet or an unsolicited regular routing update. Most RIP packets are unsolicited Responses sent out at the 30-second Update timer interval.

Both the Request and the Response messages use the same packet format for all communications. So, when discussing packet types and field usage, the only real difference is whether a RIPv1 router or a RIPv2 router sends the packet.

The maximum size of an IP RIP packet is limited to 512 bytes. This means that an individual Request or Response message can contain no more than 25 entries. If a RIP router requires more space (in other words, it has more than 25 routes to send), then multiple packets must be used.
Version 1 Packet Format

RIPv1 packets are encapsulated in an UDP datagram; they contain a RIP specific header and up to 25 route entries. The format of the packet is shown in Figure 5.4.

**Figure 5.4** RIPv1 packet format

First, let’s analyze the parts of the RIP header.

- The Command section comprises 1 octet. It indicates whether the packet is a Request or Response message, as follows:

  1. Request
  2. Response
  3. Traceon (obsolete)
  4. Traceoff (obsolete)
  5. Reserved

- The Version section (1 octet) is set to 1, for Version 1.
- The next section is unused and is set to 0 (2 octets).

Now let’s look at the components of the route entry portion of the packet. Each route entry consists of the following fields:

**Address Family Identifier (2 octets)** This field is used to identify routing information for various protocols. It is set to 2 for IP.

**Unused (2 octets)** This field is set to 0.

**IP Address (4 octets)** This is the IP address of the destination network.

**Unused (4 octets)** This field is set to 0.
Unused (4 octets)  This field is set to 0.

Metric (4 octets)  This field displays the number of hops traversed. The possible values are between 1 and 16, with a value of 16 representing an unreachable network.

Version 2 Packet Format

RIPv2 packets, like those in RIPv1, are encapsulated in a UDP datagram. They contain a RIPv-specific header and up to 25 route entries. The format of the packet is shown in Figure 5.5.

**FIGURE 5.5** RIPv2 packet format

The RIPv2 header format is unchanged from RIPv1, except that the version number is now set to 2.

Each route entry in the RIPv2 packet consists of the following fields:

**Address Family Identifier (2 octets)**  Used to identify routing information for various protocols. Set to 2 for IP. If the Address Family Identifier (AFI) for the first entry of the packet is 0xFFFF, this field indicates that the entry contains authentication information.

**Route Tag (2 octets)**  Used to administratively “mark” a route. This can indicate whether the route is internal or external, or can be used in conjunction with routing policies.

**IP Address (4 octets)**  Contains the IP address of the destination network.

**Subnet Mask (4 octets)**  Contains the subnet mask of the destination network. Can also be set to 0 if no subnet mask was specified.

**Next Hop (4 octets)**  Contains the IP address of the next hop. Packets destined for the destination network should be sent to this address to be forwarded.

**Metric (4 octets)**  Provides the number of hops traversed. Its value must be between 1 and 16, with a value of 16 indicating an unreachable network.
RIPv2 Extensions

Version 2 of the RIP specification addresses some of the shortcomings of RIPv1. Since both versions use the same packet types and formats, RIPv2 is backward compatible with RIPv1. Notice that the previously unused fields in a RIPv1 packet are now used in a RIPv2 packet. If a RIPv1 router receives a RIPv2 packet, it just ignores those now populated fields. This allows version 1 and version 2 RIP routers to interoperate with each other.

Some of the enhancements that are included in the RIPv2 specification are:

**VLSM support**  By default, all RIPv2 Response updates include the subnet mask. This allows v2 routers to support variable-length subnet mask (VLSM) routing and provides for a classless network routing environment.

**Multicast announcements**  RIPv2 sends all Request and Response messages to a multicast address (224.0.0.9) instead of the 255.255.255.255 broadcast address. This provides for better scalability since only RIP-speaking routers (or hosts) need to process the packets.

**Authentication**  RIPv2 supports authentication by means of a password. This allows a RIP router to accept Response messages only from a “trusted” source. Although RFC 2453 specifies the use of a plain-text password only, the JUNOS software also supports the use of MD5 hashes, as defined in RFC 2082.

If you reexamine the message format of a RIPv2 packet, you will notice that authentication is communicated to neighboring routers through the use of a route entry. The AFI of the route entry is set to 0xFFFF, and the plain-text authentication password is encoded in the remainder of the fields. Therefore, the use of authentication means that any single update packet can contain a maximum of only 24 routes.

The use of MD5 authentication adds a 20-byte trailer to the length of the RIP message for encoding additional authentication data. Also, an additional route entry is used by the algorithm, reducing the total number of RIP routes to 23 in a single Response message.

**Route tag**  RIPv2 supports a 16-bit field called a route tag. This field was originally included to indicate whether the route was derived internally or externally to the RIP network. This field can also be used for other purposes, including administrative routing policy control.

**Next hop address**  RIPv2 allows the sending router to advertise the immediate next hop address for a route entry. Similar to an ICMP redirect message, this field is helpful in a broadcast environment to avoid an extra forwarding hop when the advertising RIP router is not the immediate next hop for the route.
JUNOS software Configuration

Now that we’ve covered the theory of how RIP should work, let’s get to the “good stuff”—actually configuring the protocol within the JUNOS software.

By default, the RIP protocol is disabled on all Juniper Networks platforms. For a RIP router to exchange routes with other routers, you must first add RIP under the [edit protocols] section of the configuration hierarchy. Then, both RIP groups and RIP neighbors must be configured because RIP updates received from routers that have not been configured as RIP neighbors are ignored. Likewise, RIP routes are sent only to routers that have been configured as RIP neighbors. At this point, a RIP router will be able to receive all RIP routes advertised to it but will not advertise routes to any neighbors. This is caused by the default JUNOS software implementation of the RIP protocol. To have your RIP router advertise routes to any configured neighbor, you must write and apply a routing policy to your RIP configuration.

Many engineers are surprised by the defaults for RIP within the JUNOS software. Keep in mind that Juniper Networks routers were originally designed to operate in the core of the Internet. Most Internet providers are not using RIP as their IGP within their networks. These networks, however, might need to interface with a set of hosts (server farm) that is using RIP for connectivity. The JUNOS software was designed to receive those RIP routes and place them into the local routing table. From there, those routes could be redistributed into the core IGP within the provider’s internal network.

Minimum RIP Configuration

A minimum RIP configuration must include at least the rip, group, and neighbor statements. All other RIP configuration statements are optional. (The term neighbor is a misnomer, by the way, in that you don’t include the address of a neighbor but the name of the logical interface that connects to a neighbor or neighbors.) You include one neighbor statement for each logical interface on which you want to receive routes. The router imports all routes by default from this neighbor (that is, interface) but will not advertise any routes unless you write and apply a routing policy. Because of this, one might argue that the minimum configuration should include a policy and its relevant export statements. We explore this “enhanced” minimum configuration in the “Applying Export Policy” section later in this chapter.

Figure 5.6 shows the basic network map of the routers in the ACME Corporation network. The administrators at ACME have decided to use RIP to advertise routes in their network.
The following configuration for Cabernet allows it to receive RIP routes from both Riesling and Shiraz:

```
[edit protocols]
user@Cabernet# show rip {
    group neighbor-routers {
        neighbor fe-0/0/0.0;
        neighbor fe-0/0/1.0;
    }
}
```

Once this configuration has been committed, we can view the operational status of RIP by using the `show rip neighbor` command. The results look like this:

```
user@Cabernet> show rip neighbor

Source      Destination     Send   Receive   In
Neighbor          State  Address         Address         Mode   Mode     Met
--------          -----  -------         -----------     ----   -------  ---
fe-0/0/0.0           Up  172.16.1.2      224.0.0.9       mcast  both       1
fe-0/0/1.0           Up  172.16.2.1      224.0.0.9       mcast  both       1
```

At this point, however, none of the routers in the network have received any RIP routes. We can view all RIP routes in the routing table by using the `show route protocol rip` command, as follows:

```
user@Cabernet> show route protocol rip

inet.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
user@Cabernet>
```
This lack of routing information is due to the JUNOS software default for RIP route advertisements—which is to not advertise any routes. We need to configure some routing policies, so let’s look at that next.

**Real World Scenario**

**Using RIP in the Real World**

Based on the discussion in the previous section, you might be wondering what good it does you to have a router receive but not advertise RIP routes. The answer lies in what a Juniper Networks router uses RIP for—to inject routing information into the network core.

Keep in mind that the primary market for a Juniper Networks router is the core of a large network. For the reasons outlined in this chapter, RIP is not a suitable protocol for operating within the core of this environment. However, because RIP is a relatively straightforward protocol to implement, some network-capable devices communicate only with RIP—for example, a Remote Access Server (RAS) used for terminating modem dial calls.

A RAS device generates a 32-bit host route for each terminated call. Since each host route represents a user’s address on the network, this reachability information needs to be advertised to the network core. Most large networks place all of their RAS devices on a single broadcast or switched infrastructure and allow each device to advertise, via RIP, which user addresses are currently connected. A router is also configured for RIP on the broadcast media to listen for these routing advertisements. The router can then redistribute the routes directly into the protocol operating in the network core. Another alternative is to configure some aggregate routes (the exact number depends on the number of RAS devices) on the router. The received RIP routes activate the aggregate route in the routing table, which is then redistributed into the network core.

Either of these options works very well with the JUNOS software RIP defaults. The RIP routes are received on the local router and are not required to be advertised to any other RIP-enabled devices in the network.

**Applying Export Policy**

By default, RIP does not export any routes from the local routing table to its neighbors. This includes the directly connected interfaces running the RIP protocol. To have RIP export routes, we must configure a routing policy within the `[edit policy-options]` portion of the configuration hierarchy. We then apply this policy to RIP at the group level.

You can define and apply more than one export policy to RIP. If no routes match any configured policies, the local router uses the RIP default policy, which does not export any routes to its neighbors. Export policies are applied after all RIP configuration options, such as `metric-out`, have been applied.
Let's say Cabernet would like to advertise the directly connected RIP interfaces to its neighbors. In addition, it would like to advertise transit RIP routes between Riesling and Shiraz. We have created two routing policies to accomplish these goals. We then apply these policies to RIP at the group level, as shown here:

```
[edit policy-options]
user@Cabernet# show
policy-statement connected-routes {
  term advertise-routes {
    from protocol direct;
    then accept;
  }
}
policy-statement transit-rip-routes {
  term advertise-routes {
    from protocol rip;
    then accept;
  }
}

[edit protocols]
user@Cabernet# show
rip {
  group neighbor-routers {
    export [connected-routes transit-rip-routes];
    neighbor fe-0/0/0.0;
    neighbor fe-0/0/1.0;
  }
}
```

Riesling now has knowledge of the RIP routes from Cabernet, including the routes from Shiraz:

```
user@Riesling> show route protocol rip
inet.0: 27 destinations, 27 routes (27 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

  172.16.2.0/24 *[RIP/100] 00:07:25, metric 2
    > to 172.16.1.2 via fe-0/0/0.0
  192.168.8.1/32 *[RIP/100] 00:07:25, metric 2
    > to 172.16.1.2 via fe-0/0/0.0
```
Applying Import Policy

The JUNOS software allows you to filter routes being imported by the local router from its neighbors. You can use import policies to reject unwanted routes or to alter the metric on routes received from certain neighbors. To accomplish these goals, you create a routing policy, which you then apply to the RIP configuration. If you specify more than one policy, they are evaluated in order (first to last) and the first matching policy is applied to the route. If no match is found, the local router imports all usable RIP routes from all neighbors.

In addition to its current configuration, Cabernet would like to reject all routes within the 192.168.0.0 /16 subnet from Riesling. In the following code snippet, we’ve created a policy within [edit policy-options] and applied it to only the neighbor statement for Riesling:

```
[edit policy-options]
user@Cabernet# show
policy-statement filter-Riesling {
   term filter-routes {
       from {
           protocol rip;
           route-filter 192.168.0.0/16 orlonger;
       }
       then reject;
    }
}

[edit protocols]
user@Cabernet# show
rip {
   group neighbor-routers {
       export [connected-routes transit-rip-routes];
       neighbor fe-0/0/0.0 {
           import filter-Riesling;
       }
       neighbor fe-0/0/1.0;
    }
}
```
Modifying the Incoming Metric

All routes received within a Response message from a configured neighbor will, by default, be added to the routing table if the metric is below infinity (16). Recall that the routing table metric is the sum of the advertised metric and the local metric cost. The JUNOS software default for the local metric cost is 1. You can modify this default cost on either a neighbor or global basis by using the `metric-in` command.

Take care when using the `metric-in` command. Raising the metric over the default of 1 can place a limit on the distance the route can propagate. Remember that if the new metric ever becomes 16, the route will be considered unreachable.

Let's say Riesling would like all RIP routes received from Cabernet to have the metric increased by 5. So, we configure the `metric-in` command within the `neighbor` statement relating to Cabernet:

```
[edit protocols]
user@Riesling# show
rip {
    group neighbor-routers {
        export [connected-routes];
        neighbor fe-0/0/0.0 {
            metric-in 5;
        }
    }
}
```

In a previous output, the 172.16.2.0 /24 route was displayed with a metric of 2 in Riesling’s routing table. After we apply the `metric-in` command, that metric increases to 6:

```
user@Riesling> show route protocol rip
inet.0: 27 destinations, 27 routes (27 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

  172.16.2.0/24  *[RIP/100] 00:08:25, metric 6
                > to 172.16.1.2 via fe-0/0/0.0
  192.168.8.1/32 *[RIP/100] 00:08:25, metric 6
                > to 172.16.1.2 via fe-0/0/0.0
  192.168.24.1/32 *[RIP/100] 00:01:02, metric 7
                > to 172.16.1.2 via fe-0/0/0.0
```
Chapter 5 • The Routing Information Protocol (RIP)

Modifying the Outgoing Metric

After you’ve configured the appropriate policies, RIP advertises routes to neighbors configured with the `neighbor` statement. You can increase the metric of locally-generated routes on a per-group basis by using the `metric-out` command. These routes, your loopback address, and other directly connected subnets are advertised with a metric equal to the value configured using `metric-out`.

Suppose Shiraz would like all local routes advertised to Cabernet to have their current metric set to 3. Here, we’ve configured the `metric-out` command within the group level for Cabernet:

```
[edit protocols]
user@Shiraz# show rip {
    group neighbor-routers {
        metric-out 3;
        export [connected-routes transit-rip-routes];
        neighbor fe-0/0/1.0;
    }
}
```

We can see this change by issuing the `show route` command on Riesling. Within the ACME network, only the loopback address of 192.168.24.1/32 is being advertised by Shiraz into the network. This is the only route affected in the Riesling routing table.

```
user@Riesling> show route protocol rip
inet.0: 27 destinations, 27 routes (27 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

  172.16.2.0/24      *[RIP/100] 00:19:33, metric 6
      > to 172.16.1.2 via fe-0/0/0.0

  192.168.8.1/32     *[RIP/100] 00:19:33, metric 6
      > to 172.16.1.2 via fe-0/0/0.0

  192.168.24.1/32    *[RIP/100] 00:06:04, metric 9
      > to 172.16.1.2 via fe-0/0/0.0
```

Configuring Authentication

By default, authentication between RIP neighbors is disabled within the JUNOS software. You can configure it globally for all peers or on a peer-by-peer basis within the `neighbor` configuration hierarchy. Authentication can be quite useful in a networking environment since RIP will accept all Response messages received on a configured interface, a practice that could be potentially dangerous.
To configure authentication, you need to use both the `authentication-type` and the `authentication-key` commands. `authentication-key` is the password to use for authentication, and `authentication-type` is the method you use to authenticate your neighbor. The JUNOS software supports two authentication methods:

**Simple authentication**  Uses a plain-text password that is included in the transmitted packet. The receiving router uses its configured password to verify the packet. This is not a secure method of authentication because any protocol analyzer could capture a packet and view the plain-text password. It will, however, prevent the receipt of RIP messages from “untrusted” sources.

**MD5 authentication**  Sends the result of a one-way hashing algorithm in the transmitted packet. The password (plain-text or encrypted) never appears “on the wire.” The receiving router uses its authentication key (password) and the same algorithm to calculate its one-way hash value and compares it with the one in the packet. If they are identical, the packet is authenticated and the update will be processed. If the values differ, the router discards the packet.

In this example, all routers in our network would like to enable MD5 authentication globally for all neighbors with a configured password of `test`:

```plaintext
[edit protocols]
user@Cabernet# show
rip {
    authentication-type md5;
    authentication-key ' $9$09-4OhrW87Vs4xN'; # SECRET-DATA
    group neighbor-routers {
        export [connected-routes transit-rip-routes];
        neighbor fe-0/0/0.0;
        neighbor fe-0/0/1.0;
    }
}
```

Within the JUNOS software, all passwords are encrypted within the configuration files. This helps to enhance security since it prevents capture of a password through the casual viewing of a configuration file.

### Controlling Route Preference

The JUNOS software default for the preference of RIP routes within the routing table is 100. The routing table uses the preference values to select the best route when multiple protocols are advertising the same destination prefix. The local router selects the route with the lowest preference value and installs it into the forwarding table. The preference value of an individual route is within the range of 0 to 4294967295 \((2^{32} - 1)\).
To modify the default RIP preference value, include the `preference` command at the group level of the RIP configuration. In the following example, Shiraz would like its local RIP routes to have a preference of 90. This configuration will alter the routes only on Shiraz because the preference value is local to each router and is not transmitted within a RIP Response message. It looks like this:

```
[edit protocols]
user@Shiraz# show
rip {
  group neighbor-routers {
    preference 90;
    export [connected-routes transit-rip-routes];
    neighbor fe-0/0/1.0;
  }
}
user@Shiraz> show route protocol rip
inet.0: 29 destinations, 30 routes (29 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

172.16.1.0/24      *[RIP/90] 00:21:48, metric 2
> to 172.16.2.1 via fe-0/0/1.0
192.168.8.1/32     *[RIP/90] 00:21:48, metric 2
> to 172.16.2.1 via fe-0/0/1.0
192.168.16.1/32    *[RIP/90] 00:21:48, metric 3
> to 172.16.2.1 via fe-0/0/1.0
```

### Configuring Update Messages

By default, all RIP routers will advertise RIPv2 messages via multicast to all configured neighbors. In addition, all routers are able to receive both RIPv1 and RIPv2 messages. You can alter the inbound defaults by using the `receive` `receive-options` command. The `send` `send-options` command alters the outbound defaults. Both of these commands are configured at either the global or the neighbor level.

The `receive-options` values are:

- `both`  Accept RIPv1 and v2 packets.
- `none`   Do not receive RIP packets.
- `version-1`  Accept only RIPv1 packets.
- `version-2`  Accept only RIPv2 packets.
The send-options values are:

- **broadcast**: Broadcast RIPv2 packets (RIPv1 compatible).
- **multicast**: Multicast RIPv2 packets.
- **version-1**: Broadcast RIPv1 packets.
- **none**: Do not send RIP updates.

Our ACME corporation network has been extended to communicate with a partner network that is represented by the Merlot router. This is shown in Figure 5.7.

**FIGURE 5.7** ACME Corporation and partner

Merlot is able to communicate using the RIPv1 specification only, so Shiraz has added both the send and receive options within its neighbor statement for Merlot:

```
[edit protocols]
user@Shiraz# show
rip {  
    group neighbor-routers {  
        export [connected-routes transit-rip-routes];  
        neighbor fe-0/0/1.0;  
        neighbor fe-0/0/2.0 {  
            send version-1;  
            receive version-1;  
        }  
    }  
}
```
Shiraz now communicates with Merlot using RIPv1 and with Cabernet using RIPv2. Additionally, Shiraz translates the routing updates from either neighbor into the appropriate RIP version.

**Configuring the Number of Route Entries in an Update Message**

You can increase the default size of the RIP Response messages to include more than 25 route entries in each Update message. The maximum number of route entries you can advertise is 255 in a single message. You can change the default by using the `message-size` command at either the global or the neighbor level.

For example, suppose Riesling would like to advertise a maximum of 100 route entries in a single Response message to all configured neighbors:

```
[edit protocols]
user@Riesling# show
rip {
    message-size 100;
    group neighbor-routers {
        export [connected-routes];
        neighbor fe-0/0/0.0 {
            metric-in 5;
        }
    }
}
```

**Accepting Packets Whose Reserved Fields Are Nonzero**

Recall that the Request and Response messages for both RIPv1 and RIPv2 were identical. The difference between them was in the use of the message fields. RIPv1 viewed many fields as reserved, while the RIPv2 specification used those same fields for subnet mask, next hop, and so forth.

An RIPv1 router expects the reserved fields to contain zeroes and will discard any received packets with nonzero values. In addition, a RIPv2 router will discard any received packet with a nonzero value in a field that must be set to zero. The JUNOS software allows you to alter this behavior so that the RIP process can receive packets that are being sent in violation of the RFC 1058 and RFC 2453 specifications. You can do this with the `no-check-zero` command.

Suppose Shiraz is connecting to a partner network on interface so-1/0/0.0 to a RIP router that may not adhere to the correct standards. To guarantee that all possible RIP packets will be received, Shiraz configures `no-check-zero` at the global RIP level:

```
[edit protocols]
```
Exam Essentials

user@Shiraz# show
rip {
   no-check-zero;
   group neighbor-routers {
      export [connected-routes transit-rip-routes];
      neighbor fe-0/0/1.0;
      neighbor so-1/0/0.0;
   }
}

Summary

This chapter discussed the operational aspects of the RIP routing protocol. We covered issues related to timers, routing loops, and protocol limitations. We also examined the protocol enhancements specified in RIPv2, including authentication, multicast updates, route tagging, specific next hops, and support for variable-length subnet masks.

After investigating how these features were designed to work, we showed you how to configure the protocol on a Juniper Networks router. We examined a basic configuration for exchanging routes in the network and included examples that showed how to activate and use configuration knobs within the JUNOS software.

Exam Essentials

Be able to describe the characteristics of a distance-vector protocol. Each router has knowledge only of its immediate next-hop routers (the vector) and what routes are available through those routers. Each advertised route contains a metric (hop count) detailing the cost (distance) to reach the destination network.

Be able to identify the infinity metric. The largest usable metric allowed in a RIP network is 15. A metric of 16 is considered unreachable and is defined as the infinity metric.

Know the stability features of RIP. The operation of split horizon, triggered updates, and a Hold-Down timer is critical to avoiding routing loops in a distance-vector protocol.

Understand the timer values used in RIP. A RIP router will advertise its entire routing table to all configured neighbors on a regular cycle. This regular update is controlled by the Update timer, which runs every 30 seconds.
Be familiar with the protocol enhancements of RIPv2. The version 2 specification for RIP contains several enhancements that allow RIP to operate in today’s networking environment. Included in the update is support for VLSM, authentication, route tagging, next hop address, and multicast updates.

Know how to advertise RIP routes within the JUNOS software. The default operation of RIP within the JUNOS software is to receive routes but not to advertise routes. Route advertisement using RIP can be accomplished only with the use of a routing policy configured within [edit policy-options].

### Key Terms

Before you take the exam, be certain you are familiar with the following terms:

- Bellman-Ford algorithm
- Request messages
- distance-vector
- Response messages
- Garbage Collection timer
- split horizon
- Hold-Down timer
- Timeout timer
- infinity metric
- triggered updates
- interface cost
- UDP (User Datagram Protocol)
- poisoned reverse
- Update timer
Review Questions

1. What is the defined infinity metric for RIP?
   A. 14
   B. 15
   C. 16
   D. 17

2. What distance-vector concept keeps a RIP router from advertising routes to the next-hop neighbor for those routes?
   A. Poisoned reverse
   B. Split horizon
   C. Count to infinity
   D. Hold-down

3. Which RIP function will speed convergence times in a network?
   A. Triggered updates
   B. Hold-down
   C. Split horizon
   D. Count to infinity

4. RIP updates are sent using ______.
   A. TCP port 250
   B. TCP port 520
   C. UDP port 250
   D. UDP port 520

5. Which two timers are started when the local router misses four update messages?
   A. Update
   B. Hold-Down
   C. Timeout
   D. Garbage Collection

6. RIPv2 updates are sent to a destination address of ______.
   A. 224.0.0.5
   B. 224.0.0.6
   C. 224.0.0.9
   D. 224.0.0.10
7. Which RIPv2 enhancement takes advantage of the subnet mask field in Update messages?
   A. Authentication
   B. Multicast updates
   C. Route tag
   D. Variable-length subnet masks

8. RIPv2 allows a network administrator to assign an arbitrary value to a RIP route. What function allows this?
   A. Authentication
   B. Route Tag field
   C. Next-hop address
   D. Multicast updates

9. RIPv2 is considered to be backward compatible with RIPv1. How is this accomplished?
   A. Authentication support
   B. Identical message formats
   C. Multicast updates
   D. RIPv2 is not compatible.

10. When you’re using authentication, the password data is encoded within which message field?
    A. Option
    B. Version
    C. Route Tab Entry
    D. Route Entry

11. When would a RIP router send a Request message?
    A. When the routing process terminates
    B. When the routing process activates
    C. At the end of the Update timer
    D. At the end of the Hold-Down timer

12. When would a RIP router send a Response message?
    A. When the routing process terminates
    B. When the routing process activates
    C. At the end of the Update timer
    D. At the end of the Hold-Down timer
13. Which field allows RIPv2 to support VLSM?
   A. Next Hop
   B. Subnet Mask
   C. Metric
   D. Route Tag

14. What is the maximum number of route entries in a RIPv1 update?
   A. 23
   B. 24
   C. 25
   D. 26

15. What JUNOS software command allows a router to receive RIP messages?
   A. metric-in
   B. metric-out
   C. neighbor
   D. export

16. By default, the JUNOS software implementation of RIP will not advertise routes to neighbors. What will allow the router to advertise routes?
   A. Applying an import policy
   B. Applying an export policy
   C. Configuring neighbors
   D. Configuring send options

17. What configuration option will allow a RIP router to increase the metric of routes in a Response message from a neighbor?
   A. no-check-zero
   B. message-size
   C. metric-in
   D. metric-out

18. What two methods of authentication does the JUNOS software support?
   A. Plain-text password
   B. Authentication header
   C. MD5 hash algorithm
   D. Encapsulating payload
19. What two configuration options should be configured to allow a RIP router to communicate with a neighbor using only RIPv1?
   A. message-size
   B. send
   C. neighbor
   D. receive

20. What command should be used to advertise more than 25 routes in a Response message?
   A. no-check-zero
   B. neighbor
   C. message-size
   D. send broadcast
Answers to Review Questions

1. C. Since the maximum hop count permitted in the RIP protocol is 15, the defined infinity metric is 1 larger than that—16.

2. B. Split horizon is a loop-prevention mechanism that prevents a RIP from advertising routes to a neighbor if that neighbor is the active next hop for the route.

3. A. Instead of waiting for the expiration of the Update timer before advertising any routes to neighbors, a triggered update will let a router send a Response message for a new or recently “lost” route.

4. D. There is no reliable acknowledgment mechanism for RIP. As such, it uses the User Datagram Protocol (UDP) for all messages. Those messages are sent to port 520.

5. B, D. When a RIP router misses four Response messages, both the Hold-Down and Garbage Collection timers are started. These timers allow all downstream routers in the network to know the route is no longer usable.

6. C. RIPv2 messages are sent to the multicast address of 224.0.0.9. Other routing protocols, such as OSPF (224.0.0.5, 224.0.0.6) and EIGRP (224.0.0.10), use the other multicast addresses listed.

7. D. Since the subnet mask is explicitly advertised in all RIPv2 messages, each router does not have to “assume” a mask locally. This provides support for VLSM in a RIP network.

8. B. Like a BGP community, the Route Tag field in a RIP message allows a network administrator to assign an arbitrary value to a route. This provides a method for “tagging” a route for the purposes of routing policy.

9. B. Both RIPv1 and RIPv2 use the same message formats. This allows the protocols to interoperate and provides for backward compatibility.

10. D. RIPv2 authentication data is passed to all neighbors in a Route Entry field. The AFI of the entry is set to 0xFFFF, with the remaining fields containing the configured password.

11. B. When the routing process activates, a RIP router will send a Request message to all neighbors asking for their entire routing table.

12. C. Response messages can be sent for a variety of reasons, but often they are a result of the Update timer expiring.

13. B. The Subnet Mask field in a RIPv2 update message provides support for VLSM.

14. C. Since RIPv1 does not support authentication, at most 25 route entries can be advertised in a Response message.

15. C. A RIP router will not receive or send messages until the neighbor command is configured. Recall that the interface must be used as the variable for this command.

16. B. Until a routing policy is configured within policy-options and applied within the RIP hierarchy, a RIP router will only receive routes; no routes will be advertised prior to this occurring.
17. C. By default, a RIP router will increase by 1 the metric of all received routes. To increase this value, use the `metric-in` command, which you can configure at either the global or the neighbor level.

18. A, C. While the RIP RFCs specify support only for a plain-text password, the JUNOS software also allows a network administrator to use a MD5 hash algorithm. This provides better security in a network since only messages from “trusted” neighbors will be accepted.

19. B, D. By default, a Juniper Networks router will send RIPv2 messages. It will receive both version 1 and version 2 messages. To alter this default, use both the `send` and `receive` commands. The options for these commands will allow the router to send version 1 messages as well as receive version 1 messages.

20. C. To increase the number of route entries in an update message, use the `message-size` command. This will allow up to 255 entries to be sent in a single message.