CHAPTER 14

Routing Protocol Concepts and Configuration

The United States Postal Service routes a huge number of letters and packages each day. To do so, the postal sorting machines run fast, sorting lots of letters. Then the letters are placed in the correct container and onto the correct truck or plane to reach the final destination. However, if no one programs the letter-sorting machines to know where letters to each ZIP code should be sent, the sorter cannot do its job. Similarly, Cisco routers can route many packets, but if the router does not know any routes—routes that tell the router where to send the packets—the router cannot do its job.

This chapter introduces the basic concepts of how routers fill their routing tables with routes. Routers learn routes by being directly connected to local subnets, by being statically configured with information about routes, and by using dynamic routing protocols.

As you might guess by now, to fully appreciate the nuances of how routing protocols work, you need a thorough understanding of routing—the process of forwarding packets—as well as subnetting. So, this chapter includes a few additional comments on routing and subnetting, to link the ideas from Chapter 5, "Fundamentals of IP Addressing and Routing," Chapter 12, "IP Addressing and Subnetting," and Chapter 13, "Operating Cisco Routers," together so you can better understand dynamic routing protocols.

"Do I Know This Already?" Quiz

The "Do I Know This Already?" quiz allows you to assess if you should read the entire chapter. If you miss no more than one of these ten self-assessment questions, you might want to move ahead to the "Exam Preparation Tasks" section. Table 14-1 lists the major headings in this chapter and the "Do I Know This Already?" quiz questions covering the material in those headings so you can assess your knowledge of these specific areas. The answers to the "Do I Know This Already?" quiz appear in Appendix A.

Table 14-1 *"Do I Know This Already?" Foundation Topics Section-to-Question Mapping*

Foundation Topics Section	Questions
Connected and Static Routes	1, 2
Routing Protocol Overview	3–6
Configuring and Verifying RIP-2	7–10

1. Which of the following must be true for a static route to be installed in a router's IP routing table?
   a. The outgoing interface associated with the route must be in an "up and up" state.
   b. The router must receive a routing update from a neighboring router.
   c. The **ip route** command must be added to the configuration.
   d. The outgoing interface's **ip address** command must use the **special** keyword.

2. Which of the following commands correctly configures a static route?
   a. **ip route 10.1.3.0 255.255.255.0 10.1.130.253**
   b. **ip route 10.1.3.0 serial 0**
   c. **ip route 10.1.3.0 /24 10.1.130.253**
   d. **ip route 10.1.3.0 /24 serial 0**

3. Which of the following routing protocols are considered to use distance vector logic?
   a. RIP
   b. IGRP
   c. EIGRP
   d. OSPF

4. Which of the following routing protocols are considered to use link-state logic?
   a. RIP
   b. RIP-2
   c. IGRP
   d. EIGRP
   e. OSPF
   f. Integrated IS-IS

5. Which of the following routing protocols support VLSM?
   a. RIP
   b. RIP-2
   c. IGRP
   d. EIGRP
   e. OSPF
   f. Integrated IS-IS

6. Which of the following routing protocols are considered to be capable of converging quickly?

   a. RIP
   b. RIP-2
   c. IGRP
   d. EIGRP
   e. OSPF
   f. Integrated IS-IS

7. Router1 has interfaces with addresses 9.1.1.1 and 10.1.1.1. Router2, connected to Router1 over a serial link, has interfaces with addresses 10.1.1.2 and 11.1.1.2. Which of the following commands would be part of a complete RIP Version 2 configuration on Router2, with which Router2 advertises out all interfaces, and about all routes?

   a. **router rip**
   b. **router rip 3**
   c. **network 9.0.0.0**
   d. **version 2**
   e. **network 10.0.0.0**
   f. **network 10.1.1.1**
   g. **network 10.1.1.2**
   h. **network 11.0.0.0**
   i. **network 11.1.1.2**

8. Which of the following **network** commands, following a **router rip** command, would cause RIP to send updates out two interfaces whose IP addresses are 10.1.2.1 and 10.1.1.1, mask 255.255.255.0?

   a. **network 10.0.0.0**
   b. **network 10.1.1.0 10.1.2.0**
   c. **network 10.1.1.1. 10.1.2.1**
   d. **network 10.1.0.0 255.255.0.0**
   e. **network 10**
   f. You cannot do this with only one **network** command.

9. What command(s) list(s) information identifying the neighboring routers that are sending routing information to a particular router?

   a. **show ip**
   b. **show ip protocol**
   c. **show ip routing-protocols**
   d. **show ip route**
   e. **show ip route neighbor**
   f. **show ip route received**

10. Review the snippet from a **show ip route** command on a router:

```
R       10.1.2.0 [120/1] via 10.1.128.252, 00:00:13, Serial0/0/1
```

   Which of the following statements are true regarding this output?

   a. The administrative distance is 1.
   b. The administrative distance is 120.
   c. The metric is 1.
   d. The metric is not listed.
   e. The router added this route to the routing table 13 seconds ago.
   f. The router must wait 13 seconds before advertising this route again.

Foundation Topics

Connected and Static Routes

Routers need to have routes in their IP routing tables for the packet forwarding process (routing) to work. Two of the most basic means by which a router adds routes to its routing table are by learning about the subnets connected to its interfaces, and by configuring a route by using a global configuration command (called a static route). This section explains both, with the remainder of the chapter focusing on the third method of learning routes—dynamic routing protocols.

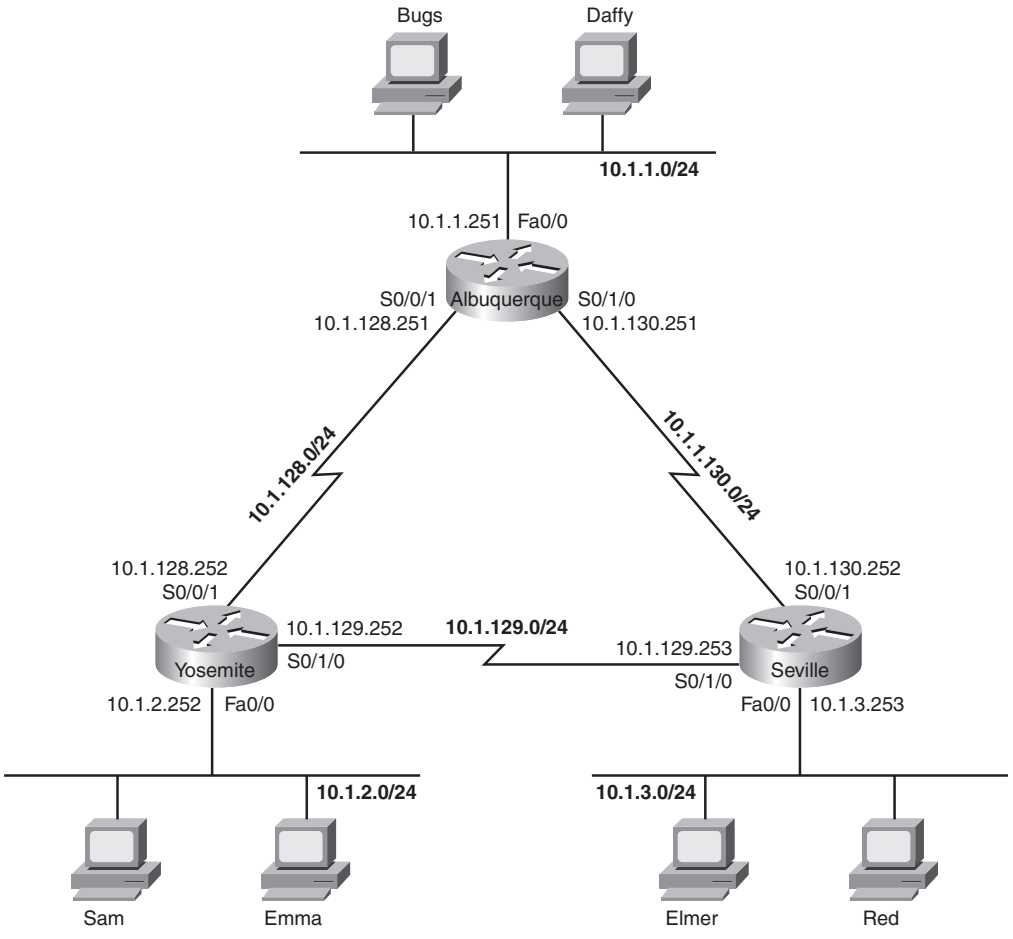
Connected Routes

A router adds routes to its routing table for the subnets connected to each of the router's interfaces. For this to occur, the router must have an IP address and mask configured on the interface (statically with the **ip address** command or dynamically using Dynamic Host Configuration Protocol [DHCP]) and both interface status codes must be "up." The concept is simple: if a router has an interface in a subnet, the router has a way to forward packets into that subnet, so the router needs a route in its routing table.

Figure 14-1 illustrates a sample internetwork that will be used in Example 14-1 to show some connected routes and some related **show** commands. Figure 14-1 shows an internetwork with six subnets, with each of the three routers having three interfaces in use. Each of the LANs in this figure could consist of one switch, one hub, or lots of switches and/or hubs together—but for the purposes of this chapter, the size of the LAN does not matter. Once the interfaces have been configured as shown in the figure, and once each interface is up and working, each of the routers should have three connected routes in their routing tables.

Example 14-1 shows the connected routes on Albuquerque after its interfaces have been configured with the addresses shown in Figure 14-1. The example includes several comments, with more detailed comments following the example.

Figure 14-1 *Sample Internetwork Used Throughout Chapter 14*



Example 14-1 *Albuquerque Connected Routes*

```
! The following command just lists the IP address configuration on Albuquerque.
! The output has been edited to show only the three interfaces used in Figure
! 14-1.
!
Albuquerque#show running-config
interface FastEthernet0/0
 ip address 10.1.1.251 255.255.255.0
!
interface Serial 0/0/1
 ip address 10.1.128.251 255.255.255.0
!
interface Serial 0/1/0
 ip address 10.1.130.251 255.255.255.0
```

Example 14-1 *Albuquerque Connected Routes (Continued)*

```
! Lines omitted for brevity
! The next command lists the interfaces, and confirms that Albuquerque's three
! interfaces shown in Figure 14-1 are in an "up and up" status.
!
Albuquerque#show ip interface brief
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0            10.1.1.251      YES manual up                    up
FastEthernet0/1            unassigned      YES manual administratively down down
Serial0/0/0                unassigned      YES NVRAM  administratively down down
Serial0/0/1                10.1.128.251    YES NVRAM  up                    up
Serial0/1/0                10.1.130.251    YES NVRAM  up                    up
Serial0/1/1                unassigned      YES NVRAM  administratively down down
!
! The next command lists the routes known by Albuquerque - all connected routes
!
Albuquerque#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route
Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 3 subnets
C       10.1.1.0 is directly connected, FastEthernet0/0
C       10.1.130.0 is directly connected, Serial0/1/0
C       10.1.128.0 is directly connected, Serial0/0/1
!
! The next command changes the mask format used by the show ip route command
!
Albuquerque#terminal ip netmask-format decimal
Albuquerque#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0 255.255.255.0 is subnetted, 3 subnets
C       10.1.1.0 is directly connected, FastEthernet0/0
C       10.1.130.0 is directly connected, Serial0/1/0
C       10.1.128.0 is directly connected, Serial0/0/1
```

To begin, the **show ip interface brief** command in Example 14-1 confirms that Albuquerque's three interfaces meet the requirements to have their connected subnets added to the routing table. Note that all three interfaces are in an "up and up" state and have an IP address configured.

The output of the **show ip route** command confirms that Albuquerque indeed added a route to all three subnets to its routing table. The output begins with a single-letter code legend, with "C" meaning "connected." The individual routes begin with a code letter on the far left—in this case, all three routes have the letter C. Also, note that the output lists the mask in prefix notation by default. Additionally, in cases when one mask is used throughout a single classful network—in other words, static-length subnet masking (SLSM) is used—the **show ip route** command output lists the mask on a heading line above the subnets of that classful network. For example, the lines with 10.1.1.0, 10.1.128.0, and 10.1.130.0 do not show the mask, but the line just above those three lines does list classful network 10.0.0.0 and the mask, as highlighted in the example.

Finally, you can change the format of the display of the subnet mask in **show** commands, for the duration of your login session to the router, using the **terminal ip netmask-format decimal** EXEC command, as shown at the end of Example 14-1.

> **NOTE** To be well prepared for the exams, you should look at all items in the output of the **show ip interface brief** and **show ip route** commands in each example in this chapter. Example 14-6, later in this chapter, provides more detailed comments about the **show ip route** command's output.

Static Routes

Although the connected routes on each router are important, routers typically need other routes to forward packets to all subnets in an internetwork. For example, Albuquerque can successfully ping the IP addresses on the other end of each serial link, or IP addresses on its local connected LAN subnet (10.1.1.0/24). However, a ping of an IP address in a subnet besides the three connected subnets will fail, as demonstrated in Example 14-2. Note that this example assumes that Albuquerque still only knows the three connected routes shown in Example 14-1.

Example 14-2 *Albuquerque Pings—Works to Connected Subnets Only*

```
! This first ping is a ping of Yosemite's S0/0/1 interface
Albuquerque#ping 10.1.128.252
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.128.252, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/8 ms
```

Example 14-2 *Albuquerque Pings—Works to Connected Subnets Only (Continued)*

```
! This next ping is a ping of Yosemite's Fa0/0 interface
Albuquerque#ping 10.1.2.252
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.2.252, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

The **ping** command sends an ICMP echo request packet to the stated destination address. The TCP/IP software at the destination then replies to the ping echo request packet with a similar packet, called an *ICMP echo reply*. The **ping** command sends the first packet and waits on the response. If a response is received, the command displays a "!". If no response is received within the default timeout of 2 seconds, the **ping** command displays a ".". The Cisco IOS software **ping** command sends five of these packets by default.

In Example 14-2, the **ping 10.1.128.252** command works (showing all !'s), because Albuquerque's route to 10.1.128.0/24 matches the destination address of 10.1.128.252. However, the **ping** to 10.1.2.252 does not work, because Albuquerque does not have a route for the subnet in which 10.1.2.252 resides, subnet 10.1.2.0/24. As a result, Albuquerque cannot even send the five ping packets, so the output lists five periods.

The simple and typical solution to this problem is to configure a routing protocol on all three routers. However, you can configure static routes instead. Example 14-3 shows two **ip route** global configuration commands on Albuquerque, which add static routes for the two LAN subnets connected to Yosemite and Seville. The addition of the first of the two **ip route** commands makes the failed ping from Example 14-2 work.

Example 14-3 *Static Routes Added to Albuquerque*


```
Albuquerque#configure terminal
Albuquerque(config)#ip route  10.1.2.0 255.255.255.0  10.1.128.252
Albuquerque(config)#ip route  10.1.3.0 255.255.255.0  10.1.130.253
Albuquerque#show ip route static
     10.0.0.0/24 is subnetted, 5 subnets
S       10.1.3.0 [1/0] via 10.1.130.253
S       10.1.2.0 [1/0] via 10.1.128.252
```

The **ip route** global configuration command supplies the subnet number, mask, and the next-hop IP address. One **ip route** command defines a route to 10.1.2.0 (mask 255.255.255.0), which is located off Yosemite, so the next-hop IP address as configured on Albuquerque is 10.1.128.252, which is Yosemite's Serial0/0/1 IP address. Similarly, Albuquerque's route to 10.1.3.0/24, the subnet off Seville, points to Seville's Serial0/0/1 IP address, 10.1.130.253. Note that the next-hop IP address should be an IP address in

a directly connected subnet. Now Albuquerque knows how to forward routes to both subnets.

Whereas you can see all routes using the **show ip route** command, the **show ip route static** command lists only statically configured IP routes. The "S" in the first column means that these two routes were statically configured. Also, to actually be added to the IP routing table, the **ip route** command must be configured, and the outgoing interface implied by the next-hop router IP address must be in an "up and up" state. For example, the next-hop address on the first **ip route** command is 10.1.128.252, which is in the subnet connected to Albuquerque's S0/0/1 interface. If Albuquerque's S0/0/1 interface is not currently in an "up and up" state, this static route would not be listed in the IP routing table.

The **ip route** command allows a slightly different syntax on point-to-point serial links. For such links, you can configure the outgoing interface instead of the next-hop IP address. For instance, you could have configured **ip route 10.1.2.0 255.255.255.0 serial0/0/1** for the first route in Example 14-3.

Unfortunately, adding these two static routes to Albuquerque does not solve all the network's routing problems—you would also need to configure static routes on the other two routers as well. Currently, the static routes help Albuquerque deliver packets to these two remote LAN subnets, but the other two routers do not have enough routing information to forward packets back toward Albuquerque's LAN subnet (10.1.1.0/24). For instance, PC Bugs cannot ping PC Sam in this network yet. The problem is that although Albuquerque has a route to subnet 10.1.2.0, where Sam resides, Yosemite does not have a route to 10.1.1.0, where Bugs resides. The ping request packet goes from Bugs to Sam correctly, but Sam's ping response packet cannot be routed by the Yosemite router back through Albuquerque to Bugs, so the ping fails.

Extended ping Command

In real life, you might not be able to find a user, like Bugs, to ask to test your network by pinging, and it might be impractical to physically travel to some other site just to type a few **ping** commands on some end-user PCs. A better alternative might be to telnet to a router connected to that user's subnet, and use the IOS **ping** command to try similar tests. However, to make the **ping** command on the router more closely resemble a **ping** issued by the end user requires the extended **ping** command.

The extended IOS **ping** command, available from privileged EXEC mode, allows the CLI user to change many options for what the **ping** command does, including the source IP address used for the ICMP echo requests sent by the command. To see the significance of this option, Example 14-4 shows Albuquerque with the working standard **ping 10.1.2.252** command, but with an extended **ping** command that works similarly to a **ping** from Bugs

to Sam—a **ping** that fails in this case, because router Yosemite cannot send the ICMP echo reply back to Albuquerque.

Example 14-4 *Albuquerque: Working Ping After Adding Default Routes, Plus Failing Extended* ***ping***

```
Albuquerque#show ip route static
     10.0.0.0/24 is subnetted, 5 subnets
S       10.1.3.0 [1/0] via 10.1.130.253
S       10.1.2.0 [1/0] via 10.1.128.252
Albuquerque#ping 10.1.2.252

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.2.252, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/8 ms

Albuquerque#ping
Protocol [ip]:
Target IP address: 10.1.2.252
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 10.1.1.251
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.2.252, timeout is 2 seconds:
. . . . .
Success rate is 0 percent (0/5)
```

The simple (standard) **ping 10.1.2.252** command works for one obvious reason and one not-so-obvious reason. First, Albuquerque can forward a packet to subnet 10.1.2.0 because of the static route. The return packet, sent by Yosemite, is sent to address 10.1.128.251—Albuquerque's Serial0/0/1 IP address. Why? Well, the following points are true about the **ping** command on a Cisco router:

- The Cisco **ping** command uses, by default, the output interface's IP address as the packet's source address, unless otherwise specified in an extended **ping**. The first ping in Example 14-4 uses a source of 10.1.128.251, because Albuquerque's route used to send the packet to 10.1.2.252 refers to interface Serial0/0/1 as the outgoing interface—and Albuquerque's S0/0/1 interface IP address is 10.1.128.251.

- Ping response packets reverse the IP addresses used in the original ping request. So, in this example, Albuquerque used 10.1.128.251 as the source IP address of the original packet, so Yosemite uses 10.1.128.251 as the destination of the ping response packet—and Yosemite has a connected route to reach subnet 10.1.128.0/24, which includes address 10.1.128.251.

When you troubleshoot this internetwork, you can use the extended **ping** command to act like you issued a **ping** from a computer on that subnet, without having to call a user and ask to enter a **ping** command for you on the PC. The extended version of the **ping** command can be used to refine the problem's underlying cause by changing several details of what the **ping** command sends in its request. In real networks, when a **ping** from a router works, but a **ping** from a host does not, the extended ping could help you re-create the problem without needing to work with the end user on the phone.

For example, in Example 14-4, the extended **ping** command on Albuquerque uses a source IP address of 10.1.1.251 (Albuquerque's Fa0/0 interface IP address), destined to 10.1.2.252 (Yosemite's Fa0/0 IP address). According to the command output, no ping response was received by Albuquerque. Normally, Albuquerque's **ping** would be sourced from the IP address of the outgoing interface. With the use of the extended ping source address option, the source IP address of the echo packet is set to Albuquerque's Fa0/0 IP address, 10.1.1.251. Because the ICMP echo generated by the extended ping is sourced from an address in subnet 10.1.1.0, the packet looks more like a packet from an end user in that subnet. Yosemite builds a reply, with destination 10.1.1.251—but Yosemite does not have a route to subnet 10.1.1.0/24. So, Yosemite cannot send the ping reply packet back to Albuquerque, causing the ping to fail.

The solution in this case is pretty simple: either add a static route on Yosemite for subnet 10.1.1.0/24, or enable a routing protocol on all three routers.

Default Routes

As part of the routing (forwarding) process, a router compares each packet's destination IP address to the router's routing table. If the router does not match any routes, the router discards the packet, and makes no attempt to recover from the loss.

A *default route* is a route that is considered to match all destination IP addresses. With a default route, when a packet's destination IP address does not match any other routes, the router uses the default route for forwarding the packet.

Default routes work best when only one path exists to a part of the network. For example, in Figure 14-2, R1 is a branch office router with a single serial link connecting it to the rest of the enterprise network. There may be hundreds of subnets located outside R1's

branch office. The engineer has three main options for helping R1 know routes to reach all the rest of the subnets:

- Configure hundreds of static routes on R1—but all of those routes would use S0/1 as R1's outgoing interface, with next-hop IP address 172.16.3.2 (R2).
- Enable a routing protocol on the routers to learn the routes.
- Add a default route to R1 with outgoing interface S0/1.

Figure 14-2 *Sample Network in Which a Default Route Is Useful*

R1 Routing Table

Subnet	Outgoing Interface
Subnet 1	**S0/1**
Subnet 2	**S0/1**
Subnet 3	**S0/1**
•	**S0/1**
•	**S0/1**
•	**S0/1**

By coding a special static route called a default route, R1 can have a single route that forwards all packets out its S0/1 interface toward R2. The **ip route** command lists a special subnet and mask value, each 0.0.0.0, which means "match all packets." Example 14-5 shows the default static route on R1, pointing to R2 (172.16.3.2) as the next-hop router.

Example 14-5 *R1 Static Default Route Configuration and Routing Table*

```
R1(config)#ip route  0.0.0.0  0.0.0.0  172.16.3.2
R1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

                                                                    continues
```

Example 14-5 *R1 Static Default Route Configuration and Routing Table (Continued)*

```
Gateway of last resort is 172.16.3.2 to network 0.0.0.0

     172.16.0.0/24 is subnetted, 3 subnets
C       172.16.1.0 is directly connected, FastEthernet0/0
C       172.16.3.0 is directly connected, Serial0/1
S*      0.0.0.0/0 [1/0] via 172.16.3.2
```

The **show ip route** command shows a couple of interesting facts about this special default route. The output lists a code of "S" just like other static routes, but with an * as well. The * means that the route might be used as the default route, meaning it will be used for packets that do not match any other routes in the routing table. Without a default route, a router discards packets that do not match the routing table. With a default route, the router forwards packets that do not match any other routes, as in the case in this example.

NOTE Chapter 4, "IP Routing," in the *CCNA ICND2 Official Exam Certification Guide*, explains default routes in more detail.

You could use static routes, including static default routes, on all routers in an internetwork. However, most enterprises use a dynamic routing protocol to learn all the routes. The next section covers some additional concepts and terminology for routing protocols, with the remainder of the chapter focusing on how to configure RIP-2.

Routing Protocol Overview

IP routing protocols have one primary goal: to fill the IP routing table with the current best routes it can find. The goal is simple, but the process and options can be complicated.

Routing protocols help routers learn routes by having each router advertise the routes it knows. Each router begins by knowing only connected routes. Then, each router sends messages, defined by the routing protocol, that list the routes. When a router hears a routing update message from another router, the router hearing the update learns about the subnets and adds routes to its routing table. If all the routers participate, all the routers can learn about all subnets in an internetwork.

When learning routes, routing protocols must also prevent loops from occurring. A loop occurs when a packet keeps coming back to the same router due to errors in the routes in the collective routers' routing tables. These loops can occur with routing protocols, unless the routing protocol makes an effort to avoid the loops.

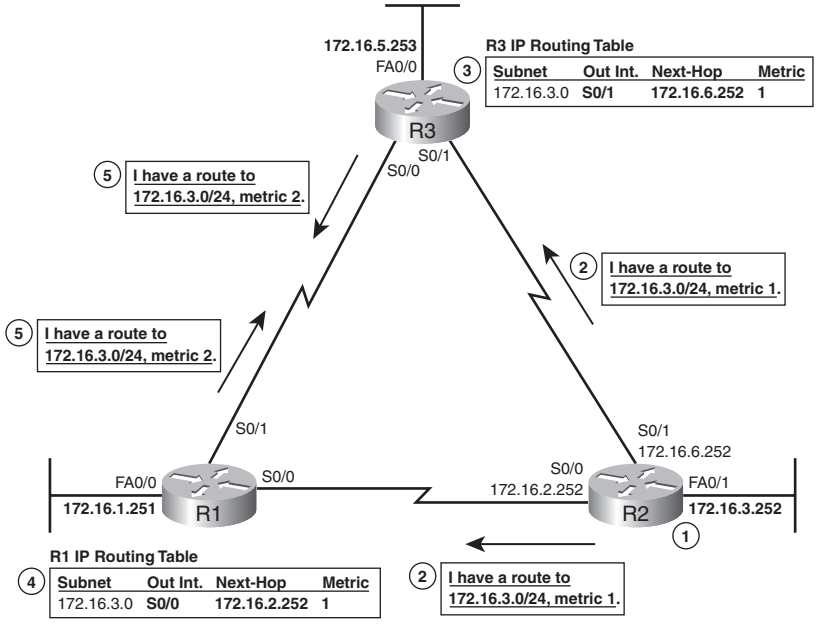
This section starts by explaining how RIP-2 works in a little more detail than was covered in Chapter 5. Following that, the various IP routing protocols are compared.

RIP-2 Basic Concepts

Routers using RIP-2 advertise a small amount of simple information about each subnet to their neighbors. Their neighbors in turn advertise the information to their neighbors, and so on, until all routers have learned the information. In fact, it works a lot like how rumors spread in a neighborhood, school, or company. You might be out in the yard, stop to talk to your next-door neighbor, and tell your neighbor the latest gossip. Then, that neighbor sees his other next-door neighbor, and tells them the same bit of gossip—and so on, until everyone in the neighborhood knows the latest gossip. Distance vector protocols work the same way, but hopefully, unlike rumors in a real neighborhood, the rumor has not changed by the time everyone has heard about it.

For example, consider what occurs in Figure 14-3. The figure shows RIP-2 advertising a subnet number, mask (shown in prefix notation), and metric to its neighbors.

Figure 14-3 *Example of How RIP-2 Advertises Routes*



For the sake of keeping the figure less cluttered, Figure 14-3 only shows how the routers advertise and learn routes for subnet 172.16.3.0/24, even though the routers do advertise about other routes as well. Following the steps in the figure:

1. Router R2 learns a connected route for subnet 172.16.3.0/24.
2. R2 sends a *routing update* to its neighbors, listing a subnet (172.16.3.0), mask (/24), and a distance, or metric (1 in this case).

3. R3 hears the routing update, and adds a route to its routing table for subnet 172.16.3.0/24, referring to R2 as the next-hop router.
4. Around the same time, R1 also hears the routing update sent directly to R1 by R2. R1 then adds a route to its routing table for subnet 172.16.3.0/24, referring to R2 as the next-hop router.
5. R1 and R3 then send a routing update to each other, for subnet 172.16.3.0/24, metric 2.

By the end of this process, both R1 and R3 have heard of two possible routes to reach subnet 172.16.3.0/24—one with metric 1, and one with metric 2. Each router uses its respective lower-metric (metric 1) routes to reach 172.16.3.0.

Interestingly, distance vector protocols such as RIP-2 repeat this process continually on a periodic basis. For example, RIP routers send periodic routing updates about every 30 seconds by default. As long as the routers continue to hear the same routes, with the same metrics, the routers' routing tables do not need to change. However, when something changes, the next routing update will change or simply not occur due to some failure, so the routers will react and converge to use the then-best working routes.

Now that you have seen the basics of one routing protocol, the next section explains a wide variety of features of different routing protocols for the sake of comparison.

Comparing and Contrasting IP Routing Protocols

IP's long history and continued popularity has driven the need for several different competing routing protocols over time. So, it is helpful to make comparisons between the different IP routing protocols to see their relative strengths and weaknesses. This section describes several technical points on which the routing protocols can be compared. Then, this chapter examines RIP-2 in more detail; the *CCNA ICND2 Official Exam Certification Guide* explains OSPF and EIGRP in more detail.

One of the first points of comparison is whether the protocol is defined in RFCs, making it a public standard, or whether it is Cisco proprietary. Another very important consideration is whether the routing protocol supports variable-length subnet masking (VLSM). Although the details of VLSM are not covered in this book, but instead are covered in the *CCNA ICND2 Official Exam Certification Guide*, VLSM support is an important consideration today. This section introduces several different terms and concepts used to compare the various IP routing protocols, with Table 14-4 at the end of this section summarizing the comparison points for many of the IP routing protocols.

Interior and Exterior Routing Protocols

IP routing protocols fall into one of two major categories:

- **Interior Gateway Protocol (IGP):** A routing protocol that was designed and intended for use inside a single autonomous system
- **Exterior Gateway Protocol (EGP):** A routing protocol that was designed and intended for use between different autonomous systems

> **NOTE** The terms IGP and EGP include the word *gateway* because routers used to be called gateways.

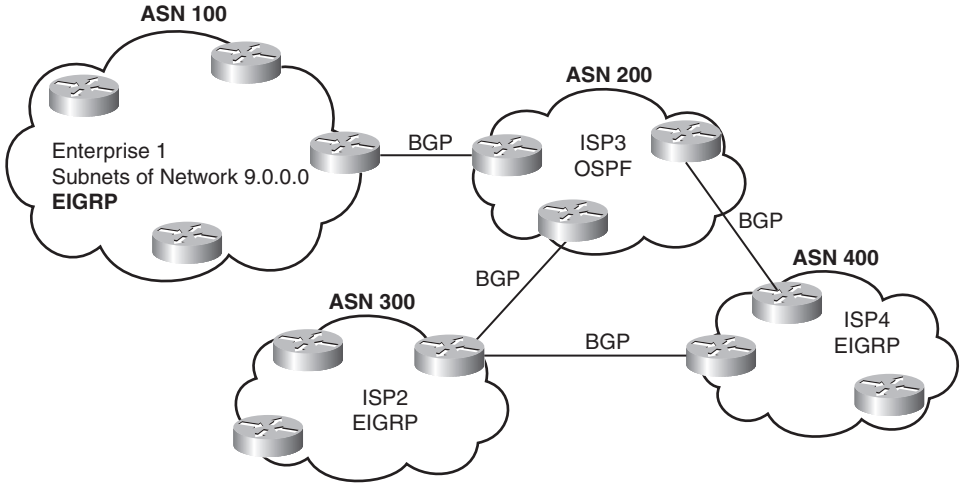
These definitions use another new term: *autonomous system*. An autonomous system is an internetwork under the administrative control of a single organization. For instance, an internetwork created and paid for by a single company is probably a single autonomous system, and an internetwork created by a single school system is probably a single autonomous system. Other examples include large divisions of a state or national government, where different government agencies may be able to build their own separate internetworks.

Some routing protocols work best inside a single autonomous system, by design, so these routing protocols are called IGPs. Conversely, only one routing protocol, *Border Gateway Protocol (BGP)*, is used today to exchange routes between routers in different autonomous systems, so it is called an EGP.

Each autonomous system can be assigned a number, called (unsurprisingly) an *autonomous system number (ASN)*. Like public IP addresses, the Internet Corporation for Assigned Network Numbers (ICANN) controls the worldwide rights to assign ASNs, delegating that authority to other organizations around the planet, typically to the same organizations that assign public IP addresses. By assigning each autonomous organization an ASN, BGP can ensure that packets do not loop around the global Internet by making sure that packets do not pass through the same autonomous system twice.

Figure 14-4 shows a small view into the worldwide Internet. Two companies and three ISPs use IGPs (OSPF and EIGRP) inside their own networks, with BGP being used between the ASNs.

Figure 14-4 *Comparing Locations for Using IGPs and EGPs*



Routing Protocol Types/Algorithms

Each IGP can be classified as using a particular class, or type, of underlying logic. Table 14-2 lists the three options, noting which IGPs use which class of algorithm.

Key Topic

Table 14-2 *Routing Protocol Classes/Algorithms and Protocols that Use Them*

Class/Algorithm	IGPs
Distance vector	RIP-1, RIP-2, IGRP
Link-state	OSPF, Integrated IS-IS
Balanced hybrid (also called advanced distance vector)	EIGRP

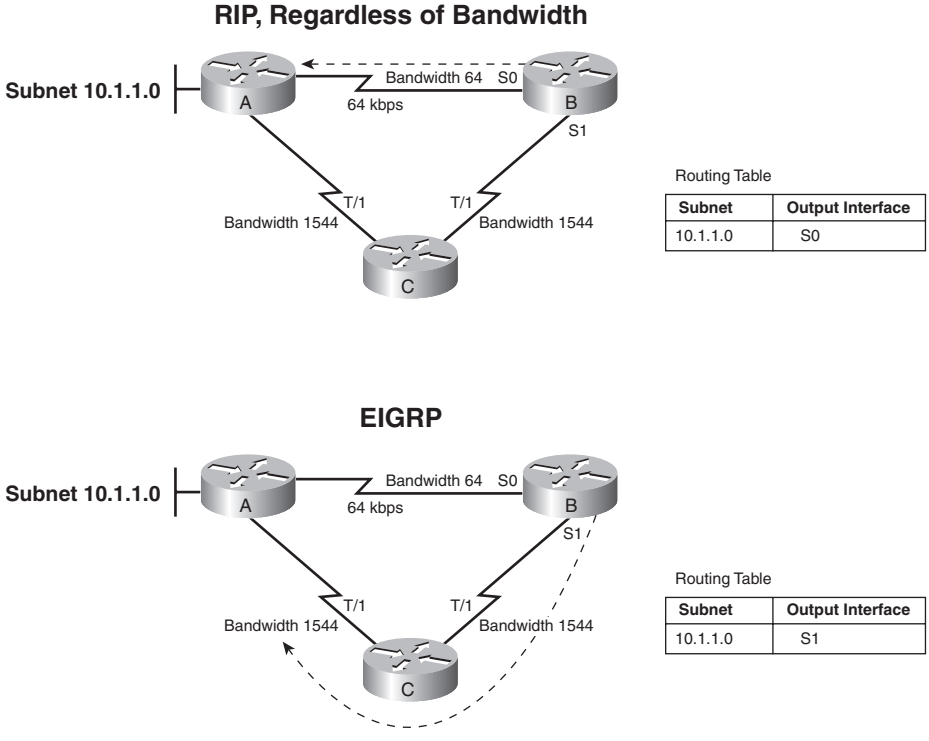
The *CCNA ICND2 Official Exam Certification Guide* covers the theory behind each of these classes of routing protocols. However, because the only IGP this book covers to any level of detail is RIP-2, most of the conceptual materials in this chapter actually show how distance vector protocols work.

Metrics

Routing protocols must have some way to decide which route is best when a router learns of more than one route to reach a subnet. To that end, each routing protocol defines a *metric* that gives an objective numeric value to the "goodness" of each route. The lower the metric, the better the route. For example, earlier, in Figure 14-3, R1 learned a metric 1 route for subnet 172.16.3.0/24 from R2, and a metric 2 route for that same subnet from R3, so R1 chose the lower-metric (1) route through R2.

Some metrics work better than others. To see why, consider Figure 14-5. The figure shows two analyses of the same basic internetwork, focusing on router B's choice of a route to reach subnet 10.1.1.0, which is on the LAN on the left side of router A. In this case, the link between A and B is only a 64-kbps link, whereas the other two links are T1s, running at 1.544 Mbps each. The top part of the figure shows router B's choice of route when using RIP (Version 1 or Version 2), whereas the bottom part of the figure shows router B's choice when the internetwork uses EIGRP.

Figure 14-5 *Comparing the Effect of the RIP and EIGRP Metrics*



RIP uses a metric called hop count, which measures the number of routers (hops) between a router and a subnet. With RIP, router B would learn two routes to reach subnet 10.1.1.0: a one-hop route through router A, and a two-hop route first through router C and then to router A. So, router B, using RIP, would add a route for subnet 10.1.1.0 pointing to router A as the next-hop IP address (represented as the dashed line in Figure 14-5).

EIGRP, on the other hand, uses a metric that (by default) considers both the interface bandwidth and interface delay settings as input into a mathematical formula to calculate the metric. If routers A, B, and C were configured with correct **bandwidth** interface

subcommands, as listed in Figure 14-5, EIGRP would add a route for subnet 10.1.1.0 to its routing table, but with router C as the next-hop router, again shown with a dashed line.

> **NOTE** For a review of the **bandwidth** command, refer to the section "Bandwidth and Clock Rate on Serial Interfaces" in Chapter 13, "Operating Cisco Routers."

Autosummarization and Manual Summarization

Routers generally perform routing (forwarding) more quickly with smaller routing tables, and less quickly with larger routing tables. Route summarization helps shorten the routing table while retaining all the needed routes in the network.

Two general types of route summarization can be done, with varying support for these two types depending on the routing protocol. The two types, both of which are introduced in this section, are called *autosummarization* and *manual summarization*. Manual summarization gives the network engineer a great deal of control and flexibility, allowing the engineer to choose what summary routes to advertise, instead of just being able to summarize with a classful network. As a result, support for manual summarization is the more useful feature as compared to autosummarization.

Chapter 5 in the *CCNA ICND2 Official Exam Certification Guide* explains both autosummarization and manual summarization in great detail.

Classless and Classful Routing Protocols

Some routing protocols must consider the Class A, B, or C network number that a subnet resides in when performing some of its tasks. Other routing protocols can ignore Class A, B, and C rules altogether. Routing protocols that must consider class rules are called *classful routing protocols*; those that do not need to consider class rules are called *classless routing protocols*.

Classless routing protocols and classful routing protocols are identified by the same three criteria, as summarized in Table 14-3.

Key Topic

Table 14-3 *Comparing Classless and Classful Routing Protocols*

Feature	Classless	Classful
Supports VLSM	Yes	No
Sends subnet mask in routing updates	Yes	No
Supports manual route summarization	Yes	No

Convergence

The term *convergence* refers to the overall process that occurs with routing protocols when something changes in a network topology. When a link comes up or fails, or when a router fails or is first turned on, the possible routes in the internetwork change. The processes used by routing protocols to recognize the changes, to figure out the now-best routes to each subnet, and to change all the routers' routing tables, is called convergence.

Some routing protocols converge more quickly than others. As you might imagine, the capability to converge quickly is important, because in some cases, until convergence completes, users might not be able to send their packets to particular subnets. (Table 14-4 in the next section summarizes the relative convergence speed of various IP routing protocols, along with other information.)

Miscellaneous Comparison Points

Two other minor comparison points between the various IGPs are interesting as well. First, the original routing protocol standards defined that routing updates should be sent to the IP all-local-hosts broadcast address of 255.255.255.255. After those original routing protocols were defined, IP multicast emerged, which allowed newer routing protocols to send routing updates only to other interested routers by using various IP multicast IP addresses.

The earlier IGPs did not include any authentication features. As time went on, it became obvious that attackers could form a denial-of-service (DoS) attack by causing problems with routing protocols. For example, an attacker could connect a router to a network and advertise lots of lower-metric routes for many subnets, causing the packets to be routed to the wrong place—and possibly copied by the attacker. The later-defined IGPs typically support some type of authentication, hoping to mitigate the exposure to these types of DoS attacks.

Summary of Interior Routing Protocols

For convenient comparison and study, Table 14-4 summarizes the most important features of interior routing protocols. Note that the most important routing protocol for the ICND1 exam is RIP, specifically RIP-2. The ICND2 and CCNA exams include more detailed coverage of RIP-2 theory, as well as the theory, configuration, and troubleshooting of OSPF and EIGRP.

Key Topic

Table 14-4 *Interior IP Routing Protocols Compared*

Feature	RIP-1	RIP-2	EIGRP	OSPF	IS-IS
Classless	No	Yes	Yes	Yes	Yes
Supports VLSM	No	Yes	Yes	Yes	Yes
Sends mask in update	No	Yes	Yes	Yes	Yes
Distance vector	Yes	Yes	No[1]	No	No
Link-state	No	No	No[1]	Yes	Yes
Supports autosummarization	No	Yes	Yes	No	No
Supports manual summarization	No	Yes	Yes	Yes	Yes
Proprietary	No	No	Yes	No	No
Routing updates sent to a multicast IP address	No	Yes	Yes	Yes	N/A
Supports authentication	No	Yes	Yes	Yes	Yes
Convergence	Slow	Slow	Very fast	Fast	Fast

1. EIGRP is often described as a balanced hybrid routing protocol, instead of link-state or distance vector. Some documents refer to EIGRP as an advanced distance vector protocol.

NOTE For reference, IGRP has the same characteristics as RIP-1 in Table 14-4, with the exception that IGRP is proprietary and RIP-1 is not.

Configuring and Verifying RIP-2

RIP-2 configuration is actually somewhat simple as compared to the concepts related to routing protocols. The configuration process uses three required commands, with only one command, the **network** command, requiring any real thought. You should also know the more-popular **show** commands for helping you analyze and troubleshoot routing protocols.

RIP-2 Configuration

The RIP-2 configuration process takes only the following three required steps, with the possibility that the third step might need to be repeated:

Key Topic

Step 1 Use the **router rip** configuration command to move into RIP configuration mode.

Step 2 Use the **version 2** RIP subcommand to tell the router to use RIP Version 2 exclusively.

Step 3 Use one or more **network** *net-number* RIP subcommands to enable RIP on the correct interfaces.

Step 4 (Optional) As needed, disable RIP on an interface using the **passive-interface** *type number* RIP subcommand.

Of the required first three steps, only the third step—the RIP **network** command—requires much thought. Each RIP **network** command enables RIP on a set of interfaces. The RIP **network** command only uses a classful network number as its one parameter. For any of the router's interface IP addresses in that entire classful network, the router does the following three things:

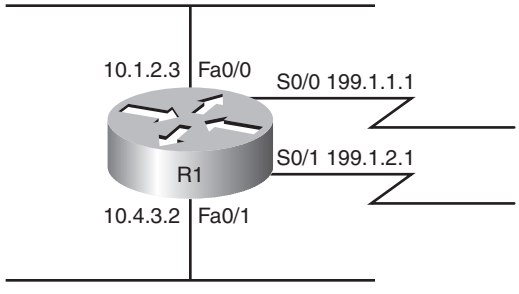
- The router multicasts routing updates to a reserved IP multicast IP address, 224.0.0.9.
- The router listens for incoming updates on that same interface.
- The router advertises about the subnet connected to the interface.

Key Topic

Sample RIP Configuration

Keeping these facts in mind, now consider how to configure RIP on a single router. Examine Figure 14-6 for a moment and try to apply the first three configuration steps to this router and anticipate the configuration required on the router to enable RIP on all interfaces.

Figure 14-6 *RIP-2 Configuration: Sample Router with Four Interfaces*



The first two configuration commands are easy, **router rip**, followed by **version 2**, with no parameters to choose. Then you need to pick which **network** commands need to be configured at Step 3. To match interface S0/0, you have to figure out that address 199.1.1.1 is in Class C IP network 199.1.1.0, meaning you need a **network 199.1.1.0** RIP subcommand. Similarly, to match interface S0/1, you need a **network 199.1.2.0** command, because IP address 199.1.2.1 is in Class C network 199.1.2.0. Finally, both of the LAN interfaces have an IP address in Class A network 10.0.0.0, so a single **network 10.0.0.0** command matches both interfaces. Example 14-6 shows the entire configuration process, with all five configuration commands.

Example 14-6 *Sample Router Configuration with RIP Enabled*

```
R1#configure terminal
R1(config)#router rip
R1(config-router)#version 2
R1(config-router)#network 199.1.1.0
R1(config-router)#network 199.1.2.0
R1(config-router)#network 10.0.0.0
```

With this configuration, R1 starts using RIP—sending RIP updates, listening for incoming RIP updates, and advertising about the connected subnet—on each of its four interfaces. However, imagine that for some reason you wanted to enable RIP on R1's Fa0/0 interface, but did not want to enable RIP on Fa0/1's interface. Both interfaces are in network 10.0.0.0, so both are matched by the **network 10.0.0.0** command.

RIP configuration does not provide a way to enable RIP on only some of the interfaces in a single Class A, B, or C network. So, if you needed to enable RIP only on R1's Fa0/0 interface, and not on the Fa0/1 interface, you would actually need to use the **network 10.0.0.0** command to enable RIP on both interfaces, and then disable the sending of RIP updates on Fa0/1 using the **passive-interface** *type number* RIP subcommand. For example, to enable RIP on all interfaces of router R1 in Figure 14-6, except for Fa0/1, you could use the same configuration in Example 14-6, but then also add the **passive-interface fa0/1** subcommand while in RIP configuration mode. This command tells R1 to stop sending RIP updates out its Fa0/1 interface, disabling one of the main functions of RIP.

NOTE The **passive-interface** command only stops the sending of RIP updates on the interface. Other features outside the scope of this book could be used to disable the processing of received updates and the advertisement of the connected subnet.

One final note on the **network** command: IOS will actually accept a parameter besides a classful network number on the command, and IOS does not supply an error message, either. However, IOS, knowing that the parameter must be a classful network number, changes the command. For example, if you typed **network 10.1.2.3** in RIP configuration mode, IOS would accept the command, with no error messages. However, when you look at the configuration, you would see a **network 10.0.0.0** command, and the **network 10.1.2.3** command that you had typed would not be there. The **network 10.0.0.0** command would indeed match all interfaces in network 10.0.0.0.

RIP-2 Verification

IOS includes three primary **show** commands that are helpful to confirm how well RIP-2 is working. Table 14-5 lists the commands and their main purpose.

Table 14-5 *RIP Operational Commands*

Command	Purpose
show ip interface brief	Lists one line per router interface, including the IP address and interface status; an interface must have an IP address, and be in an "up and up" status, before RIP begins to work on the interface.
show ip route [rip]	Lists the routing table, including RIP-learned routes, and optionally just RIP-learned routes.
show ip protocols	Lists information about the RIP configuration, plus the IP addresses of neighboring RIP routers from which the local router has learned routes.

To better understand these commands, this section uses the internetwork shown in Figure 14-1. First, consider the RIP-2 configuration required on each of the three routers. All three interfaces on all three routers are in classful network 10.0.0.0. So each router needs only one **network** command, **network 10.0.0.0**, to match all three of its interfaces. The configuration needs to be the same on all three routers, as follows:

```
router rip
 version 2
 network 10.0.0.0
```

Now, to focus on the **show** commands, Example 14-7 lists a couple of variations of the **show ip route** command, with some explanations in the example, and some following the example. Following that, Example 14-8 focuses on the **show ip protocols** command. Note that Example 14-1, earlier in this chapter, shows the output from the **show ip interfaces brief** command on the Albuquerque router, so it is not repeated here.

Example 14-7 *The* **show ip route** *Command*

```
Albuquerque#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route
Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 6 subnets
R       10.1.3.0 [120/1] via 10.1.130.253, 00:00:16, Serial0/1/0
R       10.1.2.0 [120/1] via 10.1.128.252, 00:00:09, Serial0/0/1
C       10.1.1.0 is directly connected, FastEthernet0/0
```

continues

Example 14-7 *The* **show ip route** *Command (Continued)*

```
C       10.1.130.0 is directly connected, Serial0/1/0

R       10.1.129.0 [120/1] via 10.1.130.253, 00:00:16, Serial0/1/0
                   [120/1] via 10.1.128.252, 00:00:09, Serial0/0/1
C       10.1.128.0 is directly connected, Serial0/0/1
!
! The next command lists just the RIP routes, so no code legend is listed
!
Albuquerque#show ip route rip
     10.0.0.0/24 is subnetted, 6 subnets
R       10.1.3.0 [120/1] via 10.1.130.253, 00:00:20, Serial0/1/0
R       10.1.2.0 [120/1] via 10.1.128.252, 00:00:13, Serial0/0/1
R       10.1.129.0 [120/1] via 10.1.130.253, 00:00:20, Serial0/1/0
                   [120/1] via 10.1.128.252, 00:00:13, Serial0/0/1
!
! The next command lists the route matched by this router for packets going to the
! listed IP address 10.1.2.1.
!
Albuquerque#show ip route 10.1.2.1
Routing entry for 10.1.2.0/24
  Known via "rip", distance 120, metric 1
  Redistributing via rip
  Last update from 10.1.128.252 on Serial0/0/1, 00:00:18 ago
  Routing Descriptor Blocks:
  * 10.1.128.252, from 10.1.128.252, 00:00:18 ago, via Serial0/0/1
      Route metric is 1, traffic share count is 1
!
! The same command again, but for an address that does not have a matching route in
! the routing table.
Albuquerque#show ip route 10.1.7.1
% Subnet not in table
Albuquerque#
```

Interpreting the Output of the show ip route Command

Example 14-7 shows the **show ip route** command, which lists all IP routes, the **show ip route rip** command, which lists only RIP-learned routes, and the **show ip route** *address* command, which lists details about the route matched for packets sent to the listed IP address. Focusing on the **show ip route** command, note that the legend lists "R," which means that a route has been learned by RIP, and that three of the routes list an R beside them. Next, examine the details in the route for subnet 10.1.3.0/24, highlighted in the example. The important details are as follows:

- The subnet number is listed, with the mask in the heading line above.
- The next-hop router's IP address, 10.1.130.253, which is Seville's S0/0/1 IP address.

- Albuquerque's S0/1/0 interface is the outgoing interface.
- The length of time since Albuquerque last heard about this route in a periodic RIP update, 16 seconds ago in this case.
- The RIP metric for this route (1 in this case), listed as the second number in the square brackets. For example, between Albuquerque and subnet 10.1.3.0/24, one other router (Seville) exists.
- The administrative distance of the route (120 in this case; the first number in brackets).

Take the time now to review the other two RIP routes, noting the values for these various items in those routes. As you can see, the **show ip route rip** command output lists the routes in the exact same format, the difference being that only RIP-learned routes are shown, and the legend is not displayed at the top of the command output. The **show ip route** *address* command lists more detailed output about the route that matches the destination IP address listed in the command, with the command output supplying more detailed information about the route.

Administrative Distance

When an internetwork has redundant links, and uses a single routing protocol, each router may learn multiple routes to reach a particular subnet. As stated earlier in this chapter, the routing protocol then uses a metric to choose the best route, and the router adds that route to its routing table.

In some cases, internetworks use multiple IP routing protocols. In such cases, a router might learn of multiple routes to a particular subnet using different routing protocols. In these cases, the metric does not help the router choose which route is best, because each routing protocol uses a metric unique to that routing protocol. For example, RIP uses the hop count as the metric, but EIGRP uses a math formula with bandwidth and delay as inputs. A route with RIP metric 1 might need to be compared to an EIGRP route, to the same subnet, but with metric 4,132,768. (Yes, EIGRP metrics tend to be large numbers.) Because the numbers have different meanings, there is no real value in comparing the metrics.

The router still needs to choose the best route, so IOS solves this problem by assigning a numeric value to each routing protocol. IOS then chooses the route whose routing protocol has the lower number. This number is called the *administrative distance (AD)*. For example, EIGRP defaults to use an AD of 90, and RIP defaults to use the value of 120, as seen in the routes in Example 14-7. So, an EIGRP route to a subnet would be chosen instead of a competing RIP route. Table 14-6 lists the AD values for the most common sources of routing information.

Key Topic

Table 14-6 *IOS Defaults for Administrative Distance*

Route Source	Administrative Distance
Connected routes	0
Static routes	1
EIGRP	90
IGRP	100
OSPF	110
IS-IS	115
RIP (V1 and V2)	120
Unknown or unbelievable	255

While this may be a brief tangent away from RIP and routing protocols, now that this chapter has explained administrative distance, the concept behind a particular type of static route, called a *backup static route*, can be explained. Static routes have a default AD that is better than all routing protocols, so if a router has a static route defined for a subnet, and the routing protocol learns a route to the same subnet, the static route will be added to the routing table. However, in some cases, the static route is intended to be used only if the routing protocol fails to learn a route. In these cases, an individual static route can be configured with an AD higher than the routing protocol, making the routing protocol more believable.

For example, the **ip route 10.1.1.0 255.255.255.0 10.2.2.2 150** command sets this static route's AD to 150, which is higher than all the default AD settings in Table 14-6. If RIP-2 learned a route to 10.1.1.0/24 on this same router, the router would place the RIP-learned route into the routing table, assuming a default AD of 120, which is better than the static route's AD in this case.

The show ip protocols Command

The final command for examining RIP operations is the **show ip protocols** command. This command identifies some of the details of RIP operation. Example 14-8 lists the output of this command, again on Albuquerque. Due to the variety of information in the command output, the example includes many comments inside the example.

Example 14-8 *The* **show ip protocols** *Command*


```
Albuquerque#show ip protocols
Routing Protocol is "rip"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
!
! The next line identifies the time interval for periodic routing updates, and when this
! router will send its next update.
  Sending updates every 30 seconds, next due in 22 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Redistributing: rip
!
! The next few lines result from the version 2 command being configured
  Default version control: send version 2, receive version 2
    Interface             Send  Recv  Triggered RIP  Key-chain
    FastEthernet0/0       2     2
    Serial0/0/1           2     2
    Serial0/1/0           2     2
  Automatic network summarization is in effect
  Maximum path: 4
!
! The next two lines reflect the fact that this router has a single network command,
! namely network 10.0.0.0. If other network commands were configured, these networks
! would also be listed.
  Routing for Networks:
    10.0.0.0
!
! The next section lists the IP addresses of neighboring routers from which Albuquerque
! has received routing updates, and the time since this router last heard from the
! neighbors. Note 10.1.130.253 is Seville, and 10.1.128.252 is Yosemite.
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.1.130.253         120      00:00:25
    10.1.128.252         120      00:00:20
  Distance: (default is 120)
```

Of particular importance for real-life troubleshooting and for the exam, focus on both the version information and the routing information sources. If you forget to configure the **version 2** command on one router, that router will send only RIP-1 updates by default, and the column labeled "Send" would list a 1 instead of a 2. The other routers, only listening for Version 2 updates, could not learn routes from this router.

Also, a quick way to find out if the local router is hearing RIP updates from the correct routers is to look at the list of routing information sources listed at the end of the **show ip protocols** command. For example, given the internetwork in Figure 14-1, you should expect Albuquerque to receive updates from two other routers (Yosemite and Seville). The end of Example 14-8 shows exactly that, with Albuquerque having heard from both routers in the last 30 seconds. If only one router was listed in this command's output, you could figure out which one Albuquerque was hearing from, and then investigate the problem with the missing router.

Examining RIP Messages with debug

The best way to understand whether RIP is doing its job is to use the **debug ip rip** command. This command enables a debug option that tells the router to generate log messages each time the router sends and receives a RIP update. These messages include information about every subnet listed in those advertisements as well, and the meaning of the messages is relatively straightforward.

Example 14-9 shows the output generated by the **debug ip rip** command on the Albuquerque router, based on Figure 14-1. Note that to see these messages, the user needs to be connected to the console of the router, or use the **terminal monitor** privileged mode EXEC command if using Telnet or SSH to connect to the router. The notes inside the example describe some of the meaning of the messages, in five different groups. The first three groups of messages describe Albuquerque's updates sent on each of its three RIP-enabled interfaces; the fourth group includes messages generated when Albuquerque receives an update from Seville; and the last group describes the update received from Yosemite.

Example 14-9 *Example RIP Debug Output*

```
Albuquerque#debug ip rip
RIP protocol debugging is on
Albuquerque#

! Update sent by Albuquerque out Fa0/0:
! The next two messages tell you that the local router is sending a version 2 update
! on Fa0/0, to the 224.0.0.9 multicast IP address. Following that, 5 lines list the
! 5 subnets listed in the advertisement.
*Jun  9 14:35:08.855: RIP: sending v2 update to 224.0.0.9 via FastEthernet0/0 (10.1.1.251)
*Jun  9 14:35:08.855: RIP: build update entries
*Jun  9 14:35:08.855:   10.1.2.0/24 via 0.0.0.0, metric 2, tag 0
*Jun  9 14:35:08.855:   10.1.3.0/24 via 0.0.0.0, metric 2, tag 0
*Jun  9 14:35:08.855:   10.1.128.0/24 via 0.0.0.0, metric 1, tag 0
*Jun  9 14:35:08.855:   10.1.129.0/24 via 0.0.0.0, metric 2, tag 0
*Jun  9 14:35:08.855:   10.1.130.0/24 via 0.0.0.0, metric 1, tag 0
```

Example 14-9 *Example RIP Debug Output (Continued)*

```
! The next 5 debug messages state that this local router is sending an update on its
! S0/1/0 interface, listing 3 subnets/masks
*Jun  9 14:35:10.351: RIP: sending v2 update to 224.0.0.9 via Serial0/1/0 (10.1.130.251)
*Jun  9 14:35:10.351: RIP: build update entries
*Jun  9 14:35:10.351:   10.1.1.0/24 via 0.0.0.0, metric 1, tag 0
*Jun  9 14:35:10.351:   10.1.2.0/24 via 0.0.0.0, metric 2, tag 0
*Jun  9 14:35:10.351:   10.1.128.0/24 via 0.0.0.0, metric 1, tag 0

! The next 5 debug messages state that this local router is sending an update on its
! S0/0/1 interface, listing 3 subnets/masks
*Jun  9 14:35:12.443: RIP: sending v2 update to 224.0.0.9 via Serial0/0/1 (10.1.128.251)
*Jun  9 14:35:12.443: RIP: build update entries
*Jun  9 14:35:12.443:   10.1.1.0/24 via 0.0.0.0, metric 1, tag 0
*Jun  9 14:35:12.443:   10.1.3.0/24 via 0.0.0.0, metric 2, tag 0
*Jun  9 14:35:12.443:   10.1.130.0/24 via 0.0.0.0, metric 1, tag 0

! The next 4 messages are about a RIP version 2 (v2) update received by Albuquerque
! from Seville (S0/1/0), listing three subnets. Note the mask is listed as /24.
*Jun  9 14:35:13.819: RIP: received v2 update from 10.1.130.253 on Serial0/1/0
*Jun  9 14:35:13.819:      10.1.2.0/24 via 0.0.0.0 in 2 hops
*Jun  9 14:35:13.819:      10.1.3.0/24 via 0.0.0.0 in 1 hops
*Jun  9 14:35:13.819:      10.1.129.0/24 via 0.0.0.0 in 1 hops

! The next 4 messages are about a RIP version 2 (v2) update received by Albuquerque
! from Yosemite (S0/0/1), listing three subnets. Note the mask is listed as /24.
*Jun  9 14:35:16.911: RIP: received v2 update from 10.1.128.252 on Serial0/0/1
*Jun  9 14:35:16.915:      10.1.2.0/24 via 0.0.0.0 in 1 hops
*Jun  9 14:35:16.915:      10.1.3.0/24 via 0.0.0.0 in 2 hops
*Jun  9 14:35:16.915:      10.1.129.0/24 via 0.0.0.0 in 1 hops

Albuquerque#undebug all
All possible debugging has been turned off
Albuquerque#show process
CPU utilization for five seconds: 0%/0%; one minute: 0%; five minutes: 0%
 PID QTy       PC Runtime (ms)    Invoked   uSecs    Stacks TTY Process
   1 Cwe 601B2AE8            0          1       0 5608/6000   0 Chunk Manager
```

First, if you take a broader look at the five sets of messages, it helps reinforce the expected updates that Albuquerque should both send and receive. The messages state that Albuquerque is sending updates on Fa0/0, S0/0/1, and S0/1/0, on which RIP should be enabled. Additionally, other messages state that the router received updates on interface S0/1/0, which is the link connected to Seville, and S0/0/1, which is the link connected to Yosemite.

Most of the details in the messages can be easily guessed. Some messages mention "v2," for RIP Version 2, and the fact that the messages are being sent to multicast IP address 224.0.0.9. (RIP-1 sends updates to the 255.255.255.255 broadcast address.) The majority of the messages in the example describe the routing information listed in each update, specifically the subnet and prefix length (mask), and the metric.

A close examination of the number of subnets in each routing update shows that the routers do not advertise all routes in the updates. In Figure 14-1, six subnets exist. However, the updates in the example have either three or five subnets listed. The reason has to do with the theory behind RIP, specifically a feature called split horizon. This loop-avoidance feature, which is described in Chapter 8 of the ICND2 book, limits which subnets are advertised in each update to help avoid some forwarding loops.

NOTE Chapter 8, "Routing Protocol Theory," in the *CCNA ICND2 Official Exam Certification Guide* covers split horizon in greater detail.

Finally, a few comments about the **debug** command itself can be helpful. First, before using the **debug** command, it is helpful to look at the router's CPU utilization with the **show process** command, as shown at the end of Example 14-9. This command lists the router's CPU utilization as a rolling average over three short time periods. On routers with a higher CPU utilization, generally above 30 to 40 percent, be very cautious when enabling debug options, as this may drive the CPU to the point of impacting packet forwarding. Also, you might have noticed the time stamps on the debug messages; to make the router generate time stamps, you need to configure the **service timestamps** global configuration command.

Exam Preparation Tasks

Review All the Key Topics

Review the most important topics in the chapter, noted with the key topics icon in the outer margin of the page. Table 14-7 lists a reference of these key topics and the page numbers on which each is found.

Table 14-7 *Key Topics for Chapter 14*

Key Topic Element	Description	Page Number
Example 14-3	Shows how to configure static routes	443
Definitions	IGP and EGP	451
Table 14-2	List of IGP algorithms and the IGPs that use them	452
Table 14-3	Comparison points for classless and classful routing protocols	454
Table 14-4	Summary of comparison points for IGPs	456
List	RIP-2 configuration checklist	456-457
List	The three things that occur on an interface matched by a RIP network command	457
Table 14-6	List of routing protocols and other routing sources and their default administrative distance settings	462
Example 14-8	Lists the **show ip protocol** command and how it can be used to troubleshoot RIP problems	463

Complete the Tables and Lists from Memory

Print a copy of Appendix H, "Memory Tables" (found on the CD-ROM), or at least the section for this chapter, and complete the tables and lists from memory. Appendix I, "Memory Tables Answer Key," also on the CD-ROM, includes completed tables and lists to check your work.

Definitions of Key Terms

Define the following key terms from this chapter and check your answers in the glossary.

administrative distance, autonomous system, backup static route, balanced hybrid, classful routing protocol, classless routing protocol, convergence, default route, distance vector, Exterior Gateway Protocol (EGP), Interior Gateway Protocol (IGP), link state, metric, routing update, variable-length subnet masking (VLSM)

Command References

Although you should not necessarily memorize the information in the tables in this section, this section does include a reference for the configuration commands (Table 14-8) and EXEC commands (Table 14-9) covered in this chapter. Practically speaking, you should memorize the commands as a side effect of reading the chapter and doing all the activities in this exam preparation section. To check to see how well you have memorized the commands, cover the left side of the table with a piece of paper, read the descriptions in the right side, and see if you remember the command.

Table 14-8 *Chapter 14 Configuration Command Reference*

Command	Description
router rip	Global command that moves the user into RIP configuration mode.
network *network-number*	RIP subcommand that lists a classful network number, enabling RIP on all of that router's interfaces in that classful network.
version {**1** | **2**}	RIP subcommand that sets the RIP version.
passive-interface [**default**] {*interface-type interface-number*}	RIP subcommand that tells RIP to no longer advertise RIP updates on the listed interface.
ip address *ip-address mask*	Interface subcommand that sets the router's interface IP address and mask.
ip route *prefix mask* {*ip-address* | *interface-type interface-number*}	Global command that defines a static route.
service timestamps	Global command that tells the router to put a timestamp on log messages, including debug messages.

Table 14-9 *Chapter 14 EXEC Command Reference*

Command	Purpose
show ip interface brief	Lists one line per router interface, including the IP address and interface status; an interface must have an IP address, and be in an "up and up" status, before RIP begins to work on the interface.
show ip route [**rip** | **static** | **connected**]	Lists the routing table, including RIP-learned routes, and optionally just RIP-learned routes.
show ip route *ip-address*	Lists details about the route the router would match for a packet sent to the listed IP address.
show ip protocols	Lists information about the RIP configuration, plus the IP addresses of neighboring RIP routers from which the local router has learned routes.
show process	Lists information about the various processes running in IOS, and most importantly, overall CPU utilization statistics.
terminal ip netmask-format decimal	For the length of the user's session, causes IOS to display mask information in dotted-decimal format instead of prefix format.
debug ip rip	Tells the router to generate detailed messages for each sent and received RIP update.

This chapter covers the following subjects:

IP Troubleshooting Tips and Tools: This section suggests some tips for how to approach host routing issues, routing related to routers, and IP addressing problems, including how to use several additional tools not covered elsewhere in this book.

A Troubleshooting Scenario: This section shows a three-part scenario, with tasks for each part that can be performed before seeing the answers.