IP Routing

THE CCNA EXAM TOPICS COVERED IN THIS CHAPTER INCLUDE THE FOLLOWING:

- ✓ **Describe how a network works**
  - Determine the path between two hosts across a network
- ✓ **Configure, verify, and troubleshoot basic router operation and routing on Cisco devices**
  - Describe basic routing concepts (including: packet forwarding, router lookup process)
  - Configure, verify, and troubleshoot RIPv2
  - Access and utilize the router to set basic parameters (including: CLI/SDM)
  - Connect, configure, and verify operation status of a device interface
  - Verify device configuration and network connectivity using ping, traceroute, telnet, SSH or other utilities
  - Perform and verify routing configuration tasks for a static or default route given specific routing requirements
  - Compare and contrast methods of routing and routing protocols
  - Configure, verify, and troubleshoot OSPF
  - Configure, verify, and troubleshoot EIGRP
  - Verify network connectivity (including: using ping, traceroute, and telnet or SSH)
  - Troubleshoot routing issues
  - Verify router hardware and software operation using SHOW & DEBUG commands
  - Implement basic router security

In this chapter, I'm going to discuss the IP routing process. This is an important subject to understand since it pertains to all routers and configurations that use IP. IP routing is the process of moving packets from one network to another network using routers. And as before, by routers I mean Cisco routers, of course!

But before you read this chapter, you must understand the difference between a routing protocol and a routed protocol. A *routing protocol* is used by routers to dynamically find all the networks in the internetwork and to ensure that all routers have the same routing table. Basically, a routing protocol determines the path of a packet through an internetwork. Examples of routing protocols are RIP, RIPv2, EIGRP, and OSPF.

Once all routers know about all networks, a *routed protocol* can be used to send user data (packets) through the established enterprise. Routed protocols are assigned to an interface and determine the method of packet delivery. Examples of routed protocols are IP and IPv6.

I'm pretty sure that I don't have to tell you that this is definitely important stuff to know. You most likely understand that from what I've said so far. IP routing is basically what Cisco routers do, and they do it very well. Again, this chapter is dealing with truly fundamental material—these are things you must know if you want to understand the objectives covered in this book!

In this chapter, I'm going to show you how to configure and verify IP routing with Cisco routers. I'll be covering the following:

- Routing basics
- The IP routing process
- Static routing
- Default routing
- Dynamic routing

In Chapter 7, "Enhanced IGRP (EIGRP) and Open Shortest Path First (OSPF)," I'll be moving into more advanced, dynamic routing with EIGRP and OSPF. But first, you've really got to nail down the basics of how packets actually move through an internetwork, so let's get started!

For up-to-the minute updates for this chapter, please see www.lammle.com and/or www.sybex.com.

Routing Basics

Once you create an internetwork by connecting your WANs and LANs to a router, you'll need to configure logical network addresses, such as IP addresses, to all hosts on the internetwork so that they can communicate across that internetwork.

The term *routing* is used for taking a packet from one device and sending it through the network to another device on a different network. Routers don't really care about hosts—they only care about networks and the best path to each network. The logical network address of the destination host is used to get packets to a network through a routed network, and then the hardware address of the host is used to deliver the packet from a router to the correct destination host.

If your network has no routers, then it should be apparent that you are not routing. Routers route traffic to all the networks in your internetwork. To be able to route packets, a router must know, at a minimum, the following:

- Destination address
- Neighbor routers from which it can learn about remote networks
- Possible routes to all remote networks
- The best route to each remote network
- How to maintain and verify routing information

The router learns about remote networks from neighbor routers or from an administrator. The router then builds a routing table (a map of the internetwork) that describes how to find the remote networks. If a network is directly connected, then the router already knows how to get to it.

If a network isn't directly connected to the router, the router must use one of two ways to learn how to get to the remote network: static routing, meaning that someone must hand-type all network locations into the routing table, or something called dynamic routing.

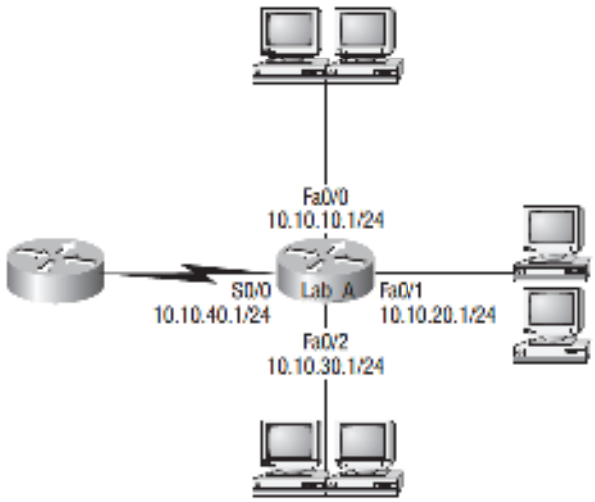
In *dynamic routing*, a protocol on one router communicates with the same protocol running on neighbor routers. The routers then update each other about all the networks they know about and place this information into the routing table. If a change occurs in the network, the dynamic routing protocols automatically inform all routers about the event. If *static routing* is used, the administrator is responsible for updating all changes by hand into all routers. Typically, in a large network, a combination of both dynamic and static routing is used.

Before we jump into the IP routing process, let's take a look at a simple example that demonstrates how a router uses the routing table to route packets out of an interface. We'll be going into a more detailed study of the process in the next section.

Figure 6.1 shows a simple two-router network. Lab_A has one serial interface and three LAN interfaces.

Looking at Figure 6.1, can you see which interface Lab_A will use to forward an IP datagram to a host with an IP address of 10.10.10.10?

FIGURE 6.1 A simple routing example



By using the command `show ip route`, we can see the routing table (map of the internetwork) that Lab_A uses to make forwarding decisions:

```
Lab_A#sh ip route
[output cut]
Gateway of last resort is not set
C       10.10.10.0/24 is directly connected, FastEthernet0/0
C       10.10.20.0/24 is directly connected, FastEthernet0/1
C       10.10.30.0/24 is directly connected, FastEthernet0/2
C       10.10.40.0/24 is directly connected, Serial 0/0
```

The C in the routing table output means that the networks listed are "directly connected," and until we add a routing protocol—something like RIP, EIGRP, etc.—to the routers in our internetwork (or use static routes), we'll have only directly connected networks in our routing table.

So let's get back to the original question: By looking at the figure and the output of the routing table, can you tell what IP will do with a received packet that has a destination IP address of 10.10.10.10? The router will packet-switch the packet to interface FastEthernet 0/0, and this interface will frame the packet and then send it out on the network segment.

Because we can, let's do another example: Based on the output of the next routing table, which interface will a packet with a destination address of 10.10.10.14 be forwarded from?

```
Lab_A#sh ip route
[output cut]
Gateway of last resort is not set
C       10.10.10.16/28 is directly connected, FastEthernet0/0
```

```
C       10.10.10.8/29 is directly connected, FastEthernet0/1
C       10.10.10.4/30 is directly connected, FastEthernet0/2
C       10.10.10.0/30 is directly connected, Serial 0/0
```

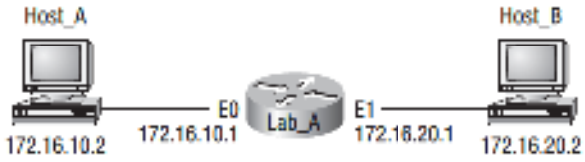
First, you can see that the network is subnetted and each interface has a different mask. And I have to tell you—you just can't answer this question if you can't subnet! 10.10.10.14 would be a host in the 10.10.10.8/29 subnet connected to the FastEthernet0/1 interface. Don't freak out if you don't get it. Just go back and reread Chapter 3 if you're struggling, and this should make perfect sense to you afterward.

For everyone who's ready to move on, let's get into this process in more detail.

The IP Routing Process

The IP routing process is fairly simple and doesn't change, regardless of the size of your network. For an example, we'll use Figure 6.2 to describe step-by-step what happens when Host_A wants to communicate with Host_B on a different network.

FIGURE 6.2 IP routing example using two hosts and one router



In this example, a user on Host_A pings Host_B's IP address. Routing doesn't get simpler than this, but it still involves a lot of steps. Let's work through them:

1. Internet Control Message Protocol (ICMP) creates an echo request payload (which is just the alphabet in the data field).
2. ICMP hands that payload to Internet Protocol (IP), which then creates a packet. At a minimum, this packet contains an IP source address, an IP destination address, and a Protocol field with 01h. (Remember that Cisco likes to use *0x* in front of hex characters, so this could look like 0x01.) All of that tells the receiving host whom it should hand the payload to when the destination is reached—in this example, ICMP.
3. Once the packet is created, IP determines whether the destination IP address is on the local network or a remote one.
4. Since IP determines that this is a remote request, the packet needs to be sent to the default gateway so the packet can be routed to the remote network. The Registry in Windows is parsed to find the configured default gateway.
5. The default gateway of host 172.16.10.2 (Host_A) is configured to 172.16.10.1. For this packet to be sent to the default gateway, the hardware address of the router's interface

Ethernet 0 (configured with the IP address of 172.16.10.1) must be known. Why? So the packet can be handed down to the Data Link layer, framed, and sent to the router's interface that's connected to the 172.16.10.0 network. Because hosts only communicate via hardware addresses on the local LAN, it's important to recognize that for Host_A to communicate to Host_B, it has to send packets to the Media Access Control (MAC) address of the default gateway on the local network.

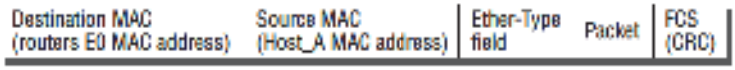
MAC addresses are always local on the LAN and never go through and past a router.

6. Next, the Address Resolution Protocol (ARP) cache of the host is checked to see if the IP address of the default gateway has already been resolved to a hardware address:
   - If it has, the packet is then free to be handed to the Data Link layer for framing. (The hardware destination address is also handed down with that packet.) To view the ARP cache on your host, use the following command:

```
C:\>arp -a
Interface: 172.16.10.2 --- 0x3
  Internet Address      Physical Address      Type
  172.16.10.1           00-15-05-06-31-b0     dynamic
```

   - If the hardware address isn't already in the ARP cache of the host, an ARP broadcast is sent out onto the local network to search for the hardware address of 172.16.10.1. The router responds to the request and provides the hardware address of Ethernet 0, and the host caches this address.
7. Once the packet and destination hardware address are handed to the Data Link layer, the LAN driver is used to provide media access via the type of LAN being used (in this example, Ethernet). A frame is then generated, encapsulating the packet with control information. Within that frame are the hardware destination and source addresses plus, in this case, an Ether-Type field that describes the Network layer protocol that handed the packet to the Data Link layer—in this instance, IP. At the end of the frame is something called a Frame Check Sequence (FCS) field that houses the result of the cyclic redundancy check (CRC). The frame would look something like what I've detailed in Figure 6.3. It contains Host_A's hardware (MAC) address and the destination hardware address of the default gateway. It does not include the remote host's MAC address—remember that!

FIGURE 6.3 Frame used from Host_A to the Lab_A router when Host_B is pinged



8. Once the frame is completed, it's handed down to the Physical layer to be put on the physical medium (in this example, twisted-pair wire) one bit at a time.
9. Every device in the collision domain receives these bits and builds the frame. They each run a CRC and check the answer in the FCS field. If the answers don't match, the frame is discarded.
   - If the CRC matches, then the hardware destination address is checked to see if it matches too (which, in this example, is the router's interface Ethernet 0).
   - If it's a match, then the Ether-Type field is checked to find the protocol used at the Network layer.
10. The packet is pulled from the frame, and what is left of the frame is discarded. The packet is handed to the protocol listed in the Ether-Type field—it's given to IP.
11. IP receives the packet and checks the IP destination address. Since the packet's destination address doesn't match any of the addresses configured on the receiving router itself, the router will look up the destination IP network address in its routing table.
12. The routing table must have an entry for the network 172.16.20.0 or the packet will be discarded immediately and an ICMP message will be sent back to the originating device with a destination network unreachable message.
13. If the router does find an entry for the destination network in its table, the packet is switched to the exit interface—in this example, interface Ethernet 1. The output below displays the Lab_A router's routing table. The C means "directly connected." No routing protocols are needed in this network since all networks (all two of them) are directly connected.

```
Lab_A>sh 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 interarea * - candidate default, U - per-user static
   route, o - ODR P - periodic downloaded static route

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 2 subnets
C       172.16.10.0 is directly connected, Ethernet0
C       172.16.20.0 is directly connected, Ethernet1
```

14. The router packet-switches the packet to the Ethernet 1 buffer.

15. The Ethernet 1 buffer needs to know the hardware address of the destination host and first checks the ARP cache.
    - If the hardware address of Host_B has already been resolved and is in the router's ARP cache, then the packet and the hardware address are handed down to the Data Link layer to be framed. Let's take a look at the ARP cache on the Lab_A router by using the `show ip arp` command:

      ```
      Lab_A#sh ip arp
      Protocol  Address      Age(min) Hardware Addr  Type   Interface
      Internet  172.16.20.1   -       00d0.58ad.05f4  ARPA   Ethernet0
      Internet  172.16.20.2   3       0030.9492.a5dd  ARPA   Ethernet0
      Internet  172.16.10.1   -       00d0.58ad.06aa  ARPA   Ethernet0
      Internet  172.16.10.2   12      0030.9492.a4ac  ARPA   Ethernet0
      ```

      The dash (-) means that this is the physical interface on the router. From the output above, we can see that the router knows the 172.16.10.2 (Host_A) and 172.16.20.2 (Host_B) hardware addresses. Cisco routers will keep an entry in the ARP table for 4 hours.
    - If the hardware address has not already been resolved, the router sends an ARP request out E1 looking for the hardware address of 172.16.20.2. Host_B responds with its hardware address, and the packet and destination hardware address are both sent to the Data Link layer for framing.
16. The Data Link layer creates a frame with the destination and source hardware address, Ether-Type field, and FCS field at the end. The frame is handed to the Physical layer to be sent out on the physical medium one bit at a time.
17. Host_B receives the frame and immediately runs a CRC. If the result matches what's in the FCS field, the hardware destination address is then checked. If the host finds a match, the Ether-Type field is then checked to determine the protocol that the packet should be handed to at the Network layer—IP in this example.
18. At the Network layer, IP receives the packet and checks the IP destination address. Since there's finally a match made, the Protocol field is checked to find out whom the payload should be given to.
19. The payload is handed to ICMP, which understands that this is an echo request. ICMP responds to this by immediately discarding the packet and generating a new payload as an echo reply.
20. A packet is then created including the source and destination addresses, Protocol field, and payload. The destination device is now Host_A.
21. IP then checks to see whether the destination IP address is a device on the local LAN or on a remote network. Since the destination device is on a remote network, the packet needs to be sent to the default gateway.
22. The default gateway IP address is found in the Registry of the Windows device, and the ARP cache is checked to see if the hardware address has already been resolved from an IP address.
23. Once the hardware address of the default gateway is found, the packet and destination hardware addresses are handed down to the Data Link layer for framing.

24. The Data Link layer frames the packet of information and includes the following in the header:
    - The destination and source hardware addresses
    - The Ether-Type field with 0x0800 (IP) in it
    - The FCS field with the CRC result in tow
25. The frame is now handed down to the Physical layer to be sent out over the network medium one bit at a time.
26. The router's Ethernet 1 interface receives the bits and builds a frame. The CRC is run, and the FCS field is checked to make sure the answers match.
27. Once the CRC is found to be okay, the hardware destination address is checked. Since the router's interface is a match, the packet is pulled from the frame and the Ether-Type field is checked to see what protocol at the Network layer the packet should be delivered to.
28. The protocol is determined to be IP, so it gets the packet. IP runs a CRC check on the IP header first and then checks the destination IP address.

IP does not run a complete CRC as the Data Link layer does—it only checks the header for errors.

Since the IP destination address doesn't match any of the router's interfaces, the routing table is checked to see whether it has a route to 172.16.10.0. If it doesn't have a route over to the destination network, the packet will be discarded immediately. (This is the source point of confusion for a lot of administrators—when a ping fails, most people think the packet never reached the destination host. But as we see here, that's not *always* the case. All it takes is for just one of the remote routers to be lacking a route back to the originating host's network and—*poof!*—the packet is dropped on the *return trip*, not on its way to the host.)

Just a quick note to mention that when (if) the packet is lost on the way back to the originating host, you will typically see a "request timed out" message because it is an unknown error. If the error occurs because of a known issue, such as if a route is not in the routing table on the way to the destination device, you will see a destination unreachable message. This should help you determine if the problem occurred on the way to the destination or on the way back.

29. In this case, the router does know how to get to network 172.16.10.0—the exit interface is Ethernet 0—so the packet is switched to interface Ethernet 0.
30. The router checks the ARP cache to determine whether the hardware address for 172.16.10.2 has already been resolved.

31. Since the hardware address to 172.16.10.2 is already cached from the originating trip to Host_B, the hardware address and packet are handed to the Data Link layer.
32. The Data Link layer builds a frame with the destination hardware address and source hardware address and then puts IP in the Ether-Type field. A CRC is run on the frame and the result is placed in the FCS field.
33. The frame is then handed to the Physical layer to be sent out onto the local network one bit at a time.
34. The destination host receives the frame, runs a CRC, checks the destination hardware address, and looks in the Ether-Type field to find out whom to hand the packet to.
35. IP is the designated receiver, and after the packet is handed to IP at the Network layer, it checks the protocol field for further direction. IP finds instructions to give the payload to ICMP, and ICMP determines the packet to be an ICMP echo reply.
36. ICMP acknowledges that it has received the reply by sending an exclamation point (!) to the user interface. ICMP then attempts to send four more echo requests to the destination host.

You've just experienced Todd's 36 easy steps to understanding IP routing. The key point to understand here is that if you had a much larger network, the process would be the *same*. In a really big internetwork, the packet just goes through more hops before it finds the destination host.

It's super-important to remember that when Host_A sends a packet to Host_B, the destination hardware address used is the default gateway's Ethernet interface. Why? Because frames can't be placed on remote networks—only local networks. So packets destined for remote networks must go through the default gateway.

Let's take a look at Host_A's ARP cache now:

```
C:\ >arp -a
Interface: 172.16.10.2 --- 0x3
  Internet Address      Physical Address      Type
  172.16.10.1           00-15-05-06-31-b0     dynamic
  172.16.20.1           00-15-05-06-31-b0     dynamic
```

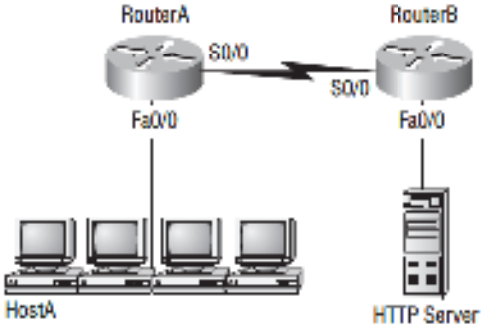
Did you notice that the hardware (MAC) address that Host_A uses to get to Host_B is the Lab_A E0 interface? Hardware addresses are *always* local, and they never pass a router's interface. Understanding this process is as important as air to you, so carve this into your memory!

Testing Your IP Routing Understanding

I really want to make sure you understand IP routing because it's super-important. So I'm going to use this section to test your understanding of the IP routing process by having you look at a couple of figures and answer some very basic IP routing questions.

Figure 6.4 shows a LAN connected to RouterA, which is, in turn, connected via a WAN link to RouterB. RouterB has a LAN connected with an HTTP server attached.

FIGURE 6.4 IP routing example 1



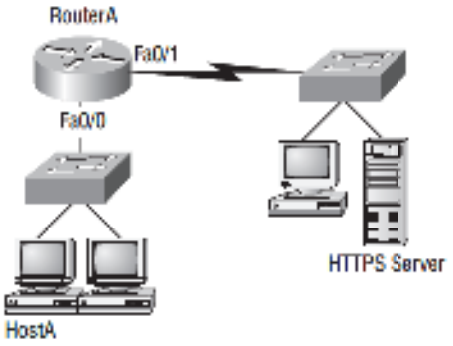
The critical information you need to glean from this figure is exactly how IP routing will occur in this example. Okay—we'll cheat a bit. I'll give you the answer, but then you should go back over the figure and see if you can answer example 2 without looking at my answers.

1. The destination address of a frame, from HostA, will be the MAC address of the F0/0 interface of the RouterA router.
2. The destination address of a packet will be the IP address of the network interface card (NIC) of the HTTP server.
3. The destination port number in the segment header will have a value of 80.

That example was a pretty simple one, and it was also very to the point. One thing to remember is that if multiple hosts are communicating to the server using HTTP, they must all use a different source port number. That is how the server keeps the data separated at the Transport layer.

Let's mix it up a little and add another internetworking device into the network and then see if you can find the answers. Figure 6.5 shows a network with only one router but two switches.

FIGURE 6.5 IP routing example 2



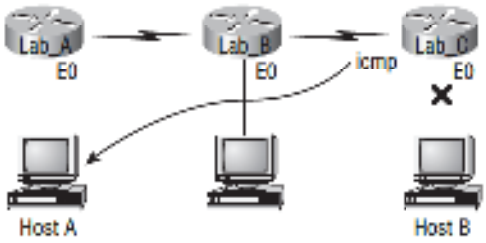
What you want to understand about the IP routing process here is what happens when HostA sends data to the HTTPS server:

1. The destination address of a frame, from HostA, will be the MAC address of the F0/0 interface of the RouterA router.
2. The destination address of a packet will be the IP address of the network interface card (NIC) of the HTTPS server.
3. The destination port number in the segment header will have a value of 443.

Notice that the switches weren't used as either a default gateway or another destination. That's because switches have nothing to do with routing. I wonder how many of you chose the switch as the default gateway (destination) MAC address for HostA? If you did, don't feel bad—just take another look with that fact in mind. It's very important to remember that the destination MAC address will always be the router's interface—if your packets are destined for outside the LAN, as they were in these last two examples.

Before we move into some of the more advanced aspects of IP routing, let's discuss ICMP in more detail, as well as how ICMP is used in an internetwork. Take a look at the network shown in Figure 6.6. Ask yourself what will happen if the LAN interface of Lab_C goes down.

FIGURE 6.6 ICMP error example



Lab_C will use ICMP to inform Host A that Host B can't be reached, and it will do this by sending an ICMP destination unreachable message. Lots of people think that the Lab_A router would be sending this message, but they would be wrong because the router that sends the message is the one with that interface that's down.

Let's look at another problem: Look at the output of a corporate router's routing table:

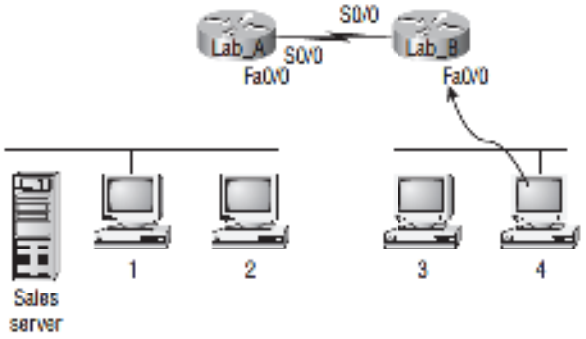
```
Corp#sh ip route
[output cut]
R    192.168.215.0 [120/2] via 192.168.20.2, 00:00:23, Serial0/0
R    192.168.115.0 [120/1] via 192.168.20.2, 00:00:23, Serial0/0
R    192.168.30.0 [120/1] via 192.168.20.2, 00:00:23, Serial0/0
C    192.168.20.0 is directly connected, Serial0/0
C    192.168.214.0 is directly connected, FastEthernet0/0
```

What do we see here? If I were to tell you that the corporate router received an IP packet with a source IP address of 192.168.214.20 and a destination address of 192.168.22.3, what do you think the Corp router will do with this packet?

If you said, "The packet came in on the FastEthernet 0/0 interface, but since the routing table doesn't show a route to network 192.168.22.0 (or a default route), the router will discard the packet and send an ICMP destination unreachable message back out interface FastEthernet 0/0," you're a genius! The reason it does this is because that's the source LAN where the packet originated from.

Now, let's check out another figure and talk about the frames and packets in detail. Really, we're not exactly chatting about anything new; I'm just making sure that you totally, completely, fully understand basic IP routing. That's because this book, and the exam objectives it's geared toward, are all about IP routing, which means you need to be all over this stuff! We'll use Figure 6.7 for the next few questions.

FIGURE 6.7 Basic IP routing using MAC and IP addresses



Referring to Figure 6.7, here's a list of all the questions you need the answers to emblazoned in your brain:

1. In order to begin communicating with the Sales server, Host 4 sends out an ARP request. How will the devices exhibited in the topology respond to this request?
2. Host 4 has received an ARP reply. Host 4 will now build a packet, then place this packet in the frame. What information will be placed in the header of the packet that leaves Host 4 if Host 4 is going to communicate to the Sales server?
3. At last, the Lab_A router has received the packet and will send it out Fa0/0 onto the LAN toward the server. What will the frame have in the header as the source and destination addresses?
4. Host 4 is displaying two web documents from the Sales server in two browser windows at the same time. How did the data find its way to the correct browser windows?

I probably should write the following in a teensy font and put them upside down in another part of the book so it would be really hard for you to cheat and peek, but since it's actually you who's going to lose out if you peek, here are your answers:

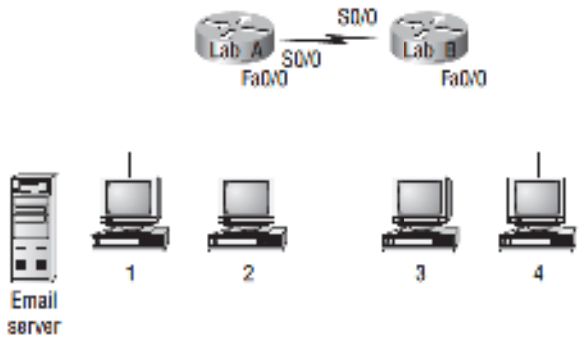
1. In order to begin communicating with the server, Host 4 sends out an ARP request. How will the devices exhibited in the topology respond to this request? Since MAC addresses must stay on the local network, the Lab_B router will respond with the MAC address of

the Fa0/0 interface and Host 4 will send all frames to the MAC address of the Lab_B Fa0/0 interface when sending packets to the Sales server.

2. Host 4 has received an ARP reply. Host 4 will now build a packet, then place this packet in the frame. What information will be placed in the header of the packet that leaves Host 4 if Host 4 is going to communicate to the Sales server? Since we're now talking about packets, not frames, the source address will be the IP address of Host 4 and the destination address will be the IP address of the Sales server.
3. Finally, the Lab_A router has received the packet and will send it out Fa0/0 onto the LAN toward the server. What will the frame have in the header as the source and destination addresses? The source MAC address will be the Lab_A router's Fa0/0 interface, and the destination MAC address will be the Sales server's MAC address. (All MAC addresses must be local on the LAN.)
4. Host 4 is displaying two web documents from the Sales server in two different browser windows at the same time. How did the data find its way to the correct browser windows? TCP port numbers are used to direct the data to the correct application window.

Great! But we're not quite done yet. I've got a few more questions for you before you actually get to configure routing in a real network. Ready? Figure 6.8 shows a basic network, and Host 4 needs to get email. Which address will be placed in the destination address field of the frame when it leaves Host 4?

FIGURE 6.8 Testing basic routing knowledge



The answer is that Host 4 will use the destination MAC address of the Fa0/0 interface of the Lab_B router—which I'm so sure you knew, right? Look at Figure 6.8 again: Host 4 needs to communicate to Host 1. Which OSI layer 3 source address will be placed in the packet header when it reaches Host 1?

Hopefully you know this: At layer 3, the source IP address will be Host 4 and the destination address in the packet will be the IP address of Host 1. Of course, the destination MAC address from Host 4 will always be the Fa0/0 address of the Lab_B router, right? And since we have more than one router, we'll need a routing protocol that communicates between both of them so that traffic can be forwarded in the right direction to reach the network in which Host 1 is attached.

Okay—one more question and you're on your way to being an IP routing genius! Again, using Figure 6.8., Host 4 is transferring a file to the email server connected to the Lab_A router. What would be the layer 2 destination address leaving Host 4? Yes, I've asked this question more than once. But not this one: What will be the source MAC address when the frame is received at the email server?

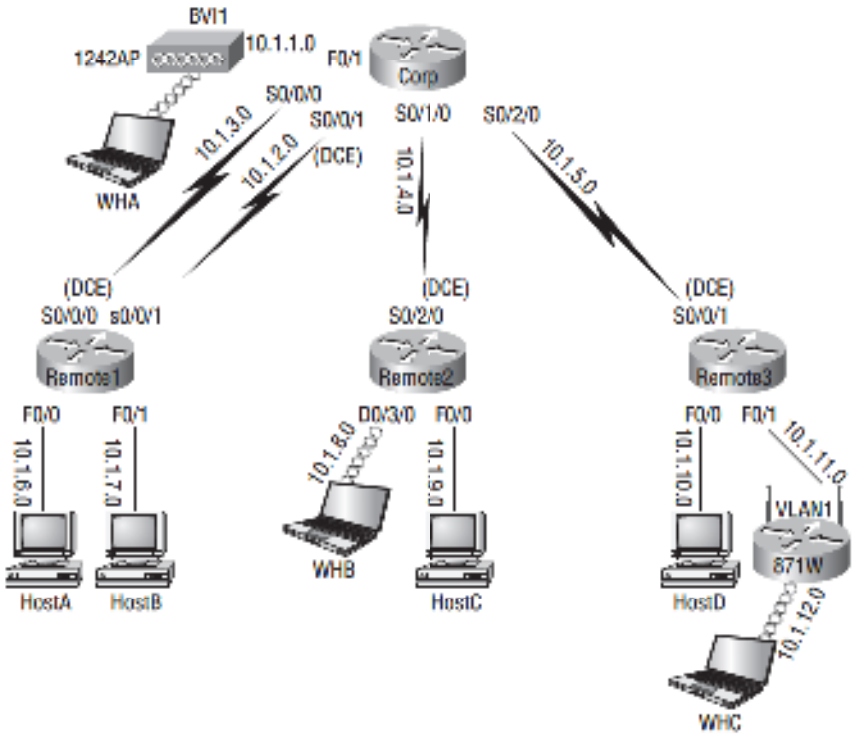
Hopefully, you answered that the layer 2 destination address leaving Host 4 will be the MAC address of the Fa0/0 interface of the Lab_B router and that the source layer 2 address that the email server will receive will be the Fa0/0 interface of the Lab_A router.

If you did, you're all set to get the skinny on how IP routing is handled in a larger network.

Configuring IP Routing

It's time to get serious and configure a real network! Figure 6.9 shows five routers: Corp, Remote1, Remote2, Remote3, and the 871W (which is a wireless router). Remember that, by default, these routers only know about networks that are directly connected to them. You also want to keep in mind that the 1242 shown in the figure is an access point—not a wireless router like the 871W. Think of the access point as more of a hub than a router.

FIGURE 6.9 Configuring IP routing



As you might guess, I've got quite a nice collection of routers for us to play with. The Corp router is a 2811 with a Wireless Controller module; something you'll get to see in Chapter 12. Remote routers 1 and 3 are 1841 ISR routers, and Remote2 is a 2801 with a wireless WIC card and a switch module. I'm simply going to call the group of remote routers R1, R2, and R3. (You can still perform most of the commands I use in this book with older routers, but you need at least a new 800 or 1800 series to run the SDM.)

The first step for this project is to correctly configure each router with an IP address on each interface. Table 6.1 shows the IP address scheme I'm going to use to configure the network. After we go over how the network is configured, I'll cover how to configure IP routing. Each network in the following table has a 24-bit subnet mask (255.255.255.0), which makes the interesting (subnet) octet the third one.

TABLE 6.1 Network Addressing for the IP Network

Router	Network Address	Interface	Address
CORP			
Corp	10.1.1.0	F0/1	10.1.1.1
Corp	10.1.2.0	S0/0/0	10.1.2.1
Corp	10.1.3.0	S0/0/1(DCE)	10.1.3.1
Corp	10.1.4.0	s0/1/0	10.1.4.1
Corp	10.1.5.0	s0/2/0	10.1.5.1
R1			
R1	10.1.2.0	S0/0/0 (DCE)	10.1.2.2
R1	10.1.3.0	S0/0/1	10.1.3.2
R1	10.1.6.0	F0/0	10.1.6.1
R1	10.1.7.0	F0/1	10.1.7.1
R2			
R2	10.1.4.0	S0/2/0 (DCE)	10.1.4.2
R2	10.1.8.0	D0/3/0	10.1.8.1
R2	10.1.9.0	F0/0	10.1.9.1

TABLE 6.1 Network Addressing for the IP Network *(continued)*

Router	Network Address	Interface	Address
R3			
R3	10.1.5.0	S0/0/0 (DCE)	10.1.5.2
R3	10.1.10.0	F0/0	10.1.10.1
R3	10.1.11.0	F0/1	10.1.11.1
871W			
871W	10.1.11.0	Vlan 1	10.1.11.2
871W	10.1.12.0	Dot11radio0	10.1.12.1
1242 AP			
1242 AP	10.1.1.0	BVI 1	10.1.1.2

The router configuration is really a pretty straightforward process since you just need to add IP addresses to your interfaces and then perform a `no shutdown` on those same interfaces. It gets a tad more complex later on, but for right now, let's configure the IP addresses in the network.

Corp Configuration

We need to configure five interfaces to configure the Corp router. And configuring the hostnames of each router will make identification much easier. While we're at it, why not set the interface descriptions, banner, and router passwords too? It's a really good idea to make a habit of configuring these commands on every router.

To get started, I performed an `erase startup-config` on the router and reloaded, so we'll start in setup mode. I choose `no` to entering setup mode, which will get us straight to the username prompt of the console. I'm going to configure all my routers this way except for R3, which I'll configure using the SDM just for fun. You can configure your routers either way.

Here's how I did all that:

```
        --- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]: n
[output cut]
Press RETURN to get started!
Router>en
```

```
Router#config t
Router(config)#hostname Corp
Corp(config)#enable secret todd
Corp(config)#interface fastEthernet 0/1
Corp(config-if)#ip address 10.1.1.1 255.255.255.0
Corp(config-if)#description Connection to 1242 AP
Corp(config-if)#no shutdown
Corp(config-if)#int s0/0/0
Corp(config-if)#ip address 10.1.2.1 255.255.255.0
Corp(config-if)#description 1st Connection to R1
Corp(config-if)#no shut
Corp(config-if)#int s0/0/1
Corp(config-if)#ip address 10.1.3.1 255.255.255.0
Corp(config-if)#description 2nd Connection to R1
Corp(config-if)#no shut
Corp(config-if)#int s0/1/0
Corp(config-if)#ip address 10.1.4.1 255.255.255.0
Corp(config-if)#description Connection to R2
Corp(config-if)#no shut
Corp(config-if)#int s0/2/0
Corp(config-if)#ip address 10.1.5.1 255.255.255.0
Corp(config-if)#description Connection to R3
Corp(config-if)#no shut
Corp(config-if)#line con 0
Corp(config-line)#password console
Corp(config-line)#login
Corp(config-line)#logging synchronous
Corp(config-line)#exec-timeout 0 0
Corp(config-line)#line aux 0
Corp(config-line)#password aux
Corp(config-line)#login
Corp(config-line)#exit
Corp(config)#line vty 0 ?
  <1-1180>  Last Line number
  <cr>
Corp(config)#line vty 0 1180
Corp(config-line)#password telnet
Corp(config-line)#login
Corp(config-line)#exit
Corp(config)#no ip domain-lookup
```

```
Corp(config)#banner motd # This is my Corp 2811 ISR Router #
Corp(config-if)#^Z
Corp#copy running-config startup-config
Destination filename [startup-config]?[enter]
Building configuration...
[OK]
Corp#
```

If you have a hard time understanding this configuration process, refer back to Chapter 4, "Cisco's Internetworking Operating System (IOS) and Security Device Manager (SDM)."

To view the IP routing tables created on a Cisco router, use the command `show ip route`. The command output is shown as follows:

```
Corp#sh 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, 1 subnets
C       10.1.1.0 is directly connected, FastEthernet0/1
Corp#
```

It's important to remember that only configured, directly connected networks are going to show up in the routing table. So why is it that I only see the FastEthernet0/1 interface in the routing table? No worries—that's just because you won't see the serial interfaces come up until the other side of the serial links is operational. As soon as we configure our R1, R2, and R3 routers, all those interfaces should pop right up.

But did you notice the C on the left side of the output of the routing table? When you see that there, it means that the network is directly connected. The codes for each type of connection are listed at the top of the `show ip route` command, along with their abbreviations.

In the interest of brevity, the codes will be cut in the rest of this chapter.

The Corp serial interface 0/0/1 is a DCE connection, which means that we need to add the `clock rate` command to the interface. Remember that you don't need to use the `clock rate` command in production. Even though this is very true, it's still imperative that you know how/when you can use it and that you understand it really well when studying for your CCNA exam!

We can see our clocking with the `show controllers` command:

```
Corp#sh controllers s0/0/1
Interface Serial0/0/1
Hardware is GT96K
DCE V.35, clock rate 2000000
```

One last thing before we get into configuring the Remote routers: Did you notice the clock rate is 2000000 under the s0/0/1 interface of the Corp router? That's important because if you think back to when we were configuring the Corp router, you'll recall that I didn't set the clock rate. The reason I didn't is because ISR routers will auto-detect a DCE-type cable and automatically configure the clock rate—a really sweet feature!

R1 Configuration

Now we're ready to configure the next router—R1. To make that happen correctly, keep in mind that we have four interfaces to deal with: serial 0/0/0, serial 0/0/1, FastEthernet 0/0, and FastEthernet 0/1. So let's make sure we don't forget to add the hostname, passwords, interface descriptions, and banner to the router configuration. As I did with the Corp router, I erased the configuration and reloaded.

Here's the configuration I used:

```
R1#erase start
% Incomplete command.
R1#erase startup-config
Erasing the nvram filesystem will remove all configuration files!
   Continue? [confirm][enter]
[OK]
Erase of nvram: complete
R1#reload
Proceed with reload? [confirm][enter]
[output cut]
%Error opening tftp://255.255.255.255/network-confg (Timed out)
%Error opening tftp://255.255.255.255/cisconet.cfg (Timed out)

         --- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]: n
```

Before we move on, I really want to discuss the above output with you. First, notice that the new 12.4 ISR routers will no longer take the command `erase start`. The router has only one command after `erase` that starts with s, as shown here:

```
Router#erase s?
startup-config
```

I know, you'd think that the IOS would continue to accept the command, but nope—sorry! The second thing I want to point out is that the output tells us the router is looking for a TFTP host to see if it can download a configuration. When that fails, it goes straight into setup mode. This gives you a great picture of the Cisco router default boot sequence we talked about in Chapter 5.

Okay, let's get back to configuring our router:

```
Press RETURN to get started!
Router>en
Router#config t
Router(config)#hostname R1
R1(config)#enable secret todd
R1(config)#int s0/0/0
R1(config-if)#ip address 10.1.2.2 255.255.255.0
R1(config-if)#Description 1st Connection to Corp Router
R1(config-if)#no shut
R1(config-if)#int s0/0/1
R1(config-if)#ip address 10.1.3.2 255.255.255.0
R1(config-if)#no shut
R1(config-if)#description 2nd connection to Corp Router
R1(config-if)#int f0/0
R1(config-if)#ip address 10.1.6.1 255.255.255.0
R1(config-if)#description Connection to HostA
R1(config-if)#no shut
R1(config-if)#int f0/1
R1(config-if)#ip address 10.1.7.1 255.255.255.0
R1(config-if)#description Connection to HostB
R1(config-if)#no shut
R1(config-if)#line con 0
R1(config-line)#password console
R1(config-line)#login
R1(config-line)#logging synchronous
R1(config-line)#exec-timeout 0 0
R1(config-line)#line aux 0
R1(config-line)#password aux
```

```
R1(config-line)#login
R1(config-line)#exit
R1(config)#line vty 0 ?
  <1-807>  Last Line number
  <cr>
R1(config)#line vty 0 807
R1(config-line)#password telnet
R1(config-line)#login
R1(config-line)#banner motd # This is my R1 ISR Router #
R1(config)#no ip domain-lookup
R1(config)#exit
R1#copy run start
Destination filename [startup-config]?[enter]
Building configuration...
[OK]
R1#
```

Let's take a look at our configuration of the interfaces.

```
R1#sh run | begin interface
interface FastEthernet0/0
 description Connection to HostA
 ip address 10.1.6.1 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet0/1
 description Connection to HostB
 ip address 10.1.7.1 255.255.255.0
 duplex auto
 speed auto
!
interface Serial0/0/0
 description 1st Connection to Corp Router
 ip address 10.1.2.2 255.255.255.0
!
interface Serial0/0/1
 description 2nd connection to Corp Router
 ip address 10.1.3.2 255.255.255.0
!
```

The `show ip route` command displays the following:

```
R1#show ip route
     10.0.0.0/24 is subnetted, 4 subnets
C       10.1.3.0 is directly connected, Serial0/0/1
C       10.1.2.0 is directly connected, Serial0/0/0
C       10.1.7.0 is directly connected, FastEthernet0/1
C       10.1.6.0 is directly connected, FastEthernet0/0
R1#
```

Notice that router R1 knows how to get to networks 10.1.3.0, 10.1.2.0, 10.1.7.0, and 10.1.6.0. We can now ping to the Corp router from R1:

```
R1#10.1.2.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.2.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
R1#
```

Now let's go back to the Corp router and look at the routing table:

```
Corp#sh ip route
[output cut]
     10.0.0.0/24 is subnetted, 4 subnets
C       10.1.3.0 is directly connected, Serial0/0/1
C       10.1.2.0 is directly connected, Serial0/0/0
C       10.1.1.0 is directly connected, FastEthernet0/1
Corp#
```

Since the serial links are up—remember, DCE is now detected automatically with ISR routers and the clock rate is automatically added to the interface configuration—we can now see all three. And once we configure R2 and R3, we'll see two more networks in the routing table of the Corp router. The Corp router can't see either the 10.1.6.0 or 10.1.7.0 networks because we don't have any routing configured yet.

R2 Configuration

To configure R2, we're going to do pretty much the same thing we did with the other two routers. There are three interfaces: serial 0/2/0, FastEthernet 0/0/0, and Dot11radio 0/3/0 to deal with, and again, we'll be sure to add the hostname, passwords, interface descriptions, and a banner to the router configuration:

```
Router>en
Router#config t
Router(config)#hostname R2
```

```
R2(config)#enable secret todd
R2(config)#int s0/2/0
R2(config-if)#ip address 10.1.4.2 255.255.255.0
R2(config-if)#description Connection to Corp ISR Router
R2(config-if)#no shut
R2(config-if)#int f0/0
R2(config-if)#ip address 10.1.9.1 255.255.255.0
R2(config-if)#description Connection to HostC
R2(config-if)#no shut
R2(config-if)#int dot11radio 0/3/0
R2(config-if)#ip address 10.1.8.1 255.255.255.0
R2(config-if)#description Admin WLAN
R2(config-if)#ssid ADMIN
R2(config-if-ssid)#guest-mode
R2(config-if-ssid)#authentication open
R2(config-if-ssid)#infrastructure-ssid
R2(config-if-ssid)#no shut
R2(config-if)#line con 0
R2(config-line)#password console
R2(config-line)#login
R2(config-line)#logging sync
R2(config-line)#exec-timeout 0 0
R2(config-line)#line aux 0
R2(config-line)#password aux
R2(config-line)#login
R2(config-line)#exit
R2(config)#line vty 0 ?
  <1-807>  Last Line number
  <cr>
R2(config)#line vty 0 807
R2(config-line)#password telnet
R2(config-line)#login
R2(config-line)#exit
R2(config)#banner motd # This is my R2 ISR Router #
R2(config)#no ip domain-lookup
R2(config)#^Z
R2#copy run start
Destination filename [startup-config]?[enter]
Building configuration...
[OK]
R2#
```

Nice—everything was pretty straightforward except for that wireless interface. It's true, the wireless interface is really just another interface on a router, and it looks just like that in the routing table as well. But, in order to bring up the wireless interface, more configurations are needed than for a simple FastEthernet interface. So check out the following output, and then I'll tell you about the special configuration needs for this wireless interface:

```
R2(config-if)#int dot11radio0/3/0
R2(config-if)#ip address 10.1.8.1 255.255.255.0
R2(config-if)#description Connection to Corp ISR Router
R2(config-if)#no shut
R2(config-if)#ssid ADMIN
R2(config-if-ssid)#guest-mode
R2(config-if-ssid)#authentication open
R2(config-if-ssid)#infrastructure-ssid
R2(config-if-ssid)#no shut
```

So, what we see here is that everything is pretty commonplace until we get to the SSID configuration. This is the Service Set Identifier that creates a wireless network that hosts can connect to. Unlike access points, the interface on the R2 router is actually a routed interface, which is the reason why the IP address is placed under the physical interface—typically the IP address would be placed under the management VLAN or Bridge-Group Virtual Interface (BVI).

That `guest-mode` line means that the interface will broadcast the SSID so wireless hosts will understand that they can connect to this interface. `Authentication open` means just that…no authentication. (Even so, you still have to type that command in at minimum to make the wireless interface work.) Last, the `infrastructure-ssid` indicates that this interface can be used to communicate to other access points, or other devices on the infrastructure—to the actual wired network itself.

But wait, we're not done yet—we still need to configure the DHCP pool for the wireless clients:

```
R2#config t
R2(config)#ip dhcp pool Admin
R2(dhcp-config)#network 10.1.8.0 255.255.255.0
R2(dhcp-config)#default-router 10.1.8.1
R2(dhcp-config)#exit
R2(config)#ip dhcp excluded-address 10.1.8.1
R2(config)#
```

Creating DHCP pools on a router is actually a pretty simple process. To do so, you just create the pool name, add the network/subnet and the default gateway, and exclude any addresses you don't want handed out (like the default gateway address). And you'd usually add a DNS server as well.

The output of the following `show ip route` command displays the directly connected networks of 10.1.9.0, 8.0, and 4.0, as you can see here:

```
R2#sh ip route
     10.0.0.0/24 is subnetted, 3 subnets
```

```
C       10.1.9.0 is directly connected, FastEthernet0/0
C       10.1.8.0 is directly connected, Dot11Radio0/3/0
C       10.1.4.0 is directly connected, Serial0/2/0
R2#
```

The Corp, R1, and R2 routers now have all their links up. But we still need to configure R3 (the 871W router) and the 1241 AP.

Wireless networks will be discussed in detail in Chapter 12, "Cisco's Wireless Technologies."

R3 Configuration

Just as I said, I'm going to use the SDM for the R3 router. My first step is to set an IP address on the F0/0 interface. I used a crossover cable to connect my PC directly to the f0/0 port.

Now since I want to set up the router with security, I've got to configure the router back to the factory defaults. I can do this via the CLI just as I showed you back in Chapter 4, but it's actually a whole lot easier to do this using SDM!

Using HTTP, I was able to access the R3 router, go to the Configure page, and choose Additional Tasks. Then, I just clicked on Configuration Management and Reset to Factory Default.

I clicked the Reset Router button in the bottom-right corner and then configured my PC using the directions shown on the screen in the above screen shot.

Again, using HTTPS, I connected back to SDM using the 10.10.10.1 address that was provided in the directions. SDM had me log in twice with the username *cisco* as well as a password of *cisco*. I then had to accept the certificate from the router, and I'm good to go with a secure connection.

The first thing the router had me do after SDM was loaded was change the username and password.

Then I needed to log in again using my new name and password.

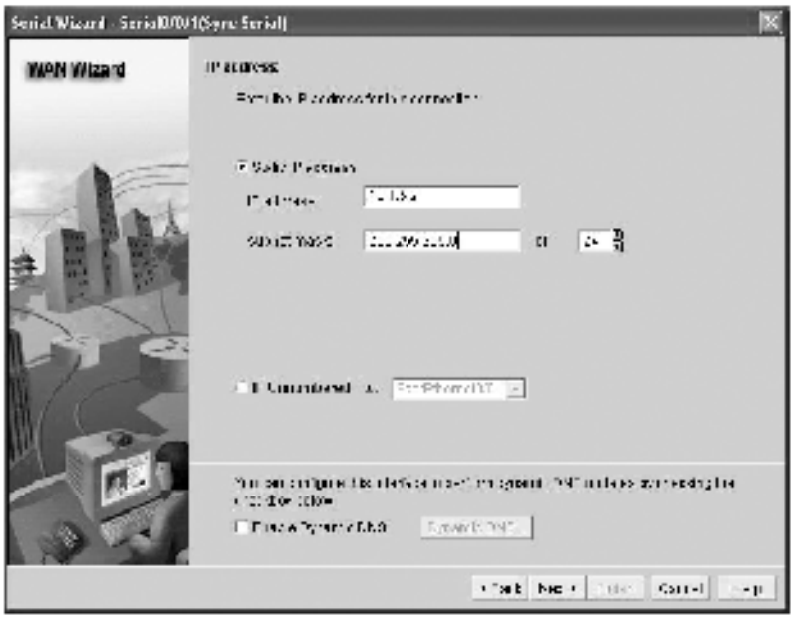
After that, I chose Configure and then Interfaces and Connections, which is in the upper-left corner, under Home. Clicking the Serial (PPP, HDLC or Frame Relay) button got me to where I could choose Create New Connection.

The Create New Connection button took me to the WAN Wizard.

I was then able to choose my interface; then I clicked Next.

I then chose High-Level Data Link Control and clicked Next. (I'll get into HDLC in Chapter 14.)

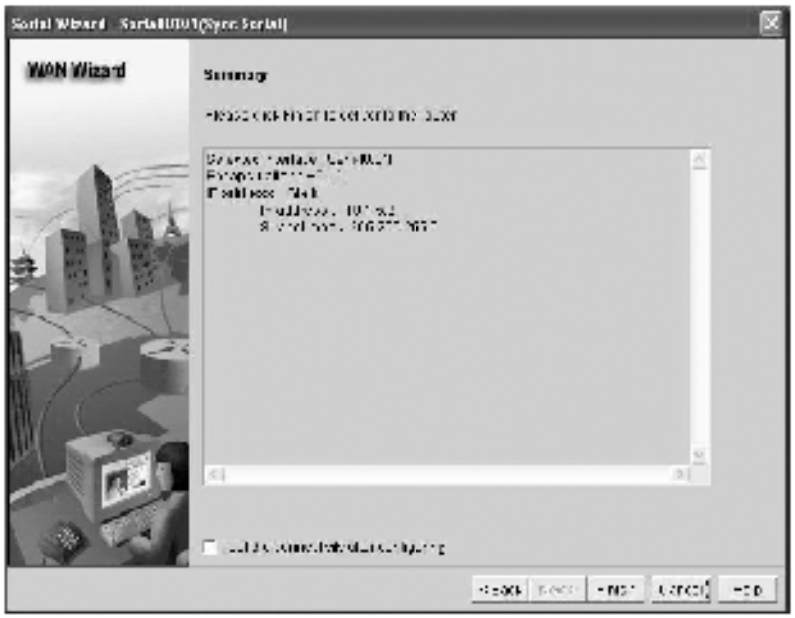
I was then able to add my IP address and mask.



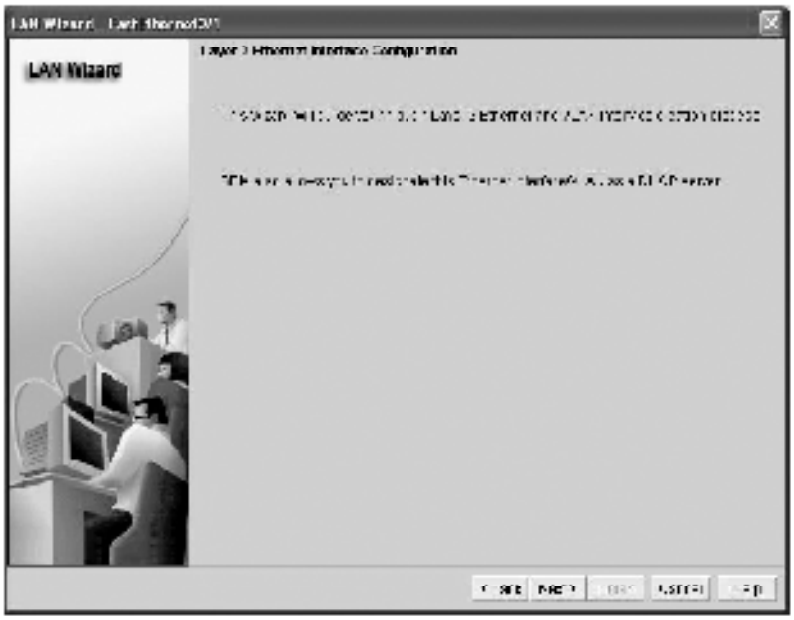
IP Unnumbered is truly an interesting configuration because it lets you set up a network connection without using an IP address. Instead, you "borrow" an IP address from another active interface. This comes in pretty handy if you happen to be a bit short on subnets!

Anyway, the next screen asked if I wanted to set up static routing and NAT. Again, this is something I'll get into more later on, so we're not going to configure it just yet.

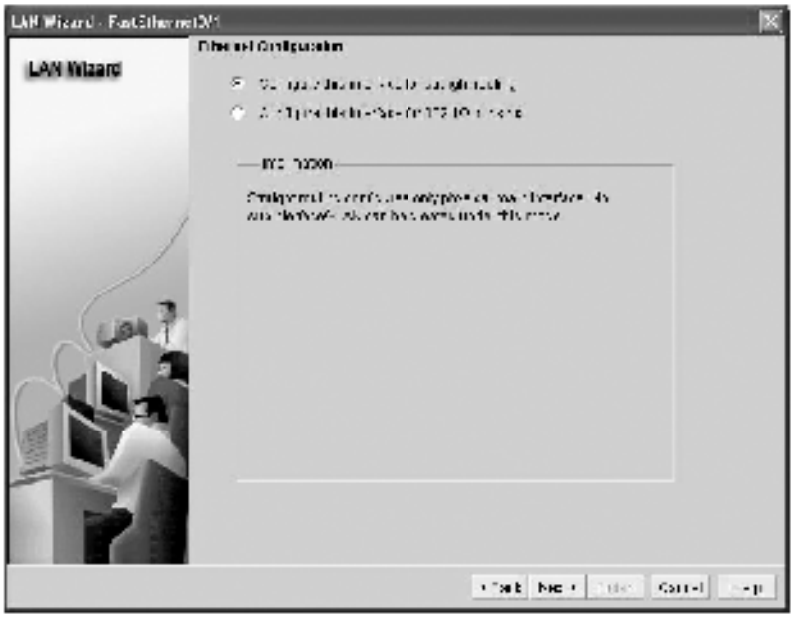
Moving on, I clicked Next and received a summary of my serial 0/0/1 configuration.



I clicked Finish, and the commands were uploaded to my R3 router. (I'm going to configure both the F0/0 and F0/1 interfaces the same way.)

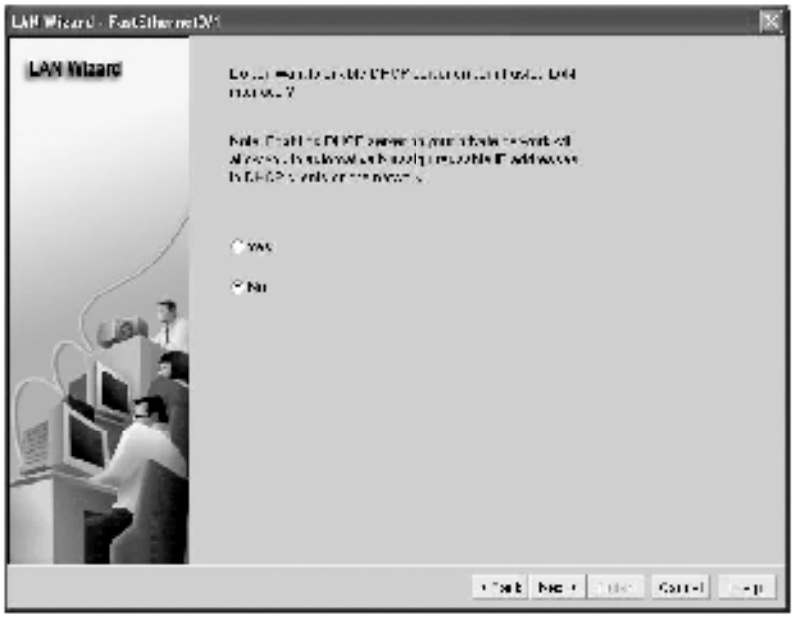


After choosing the FastEthernet 0/1 interface from the same location from where I started to configure the s0/0/1 interface, I chose Create New Connection and was taken to the LAN Wizard.

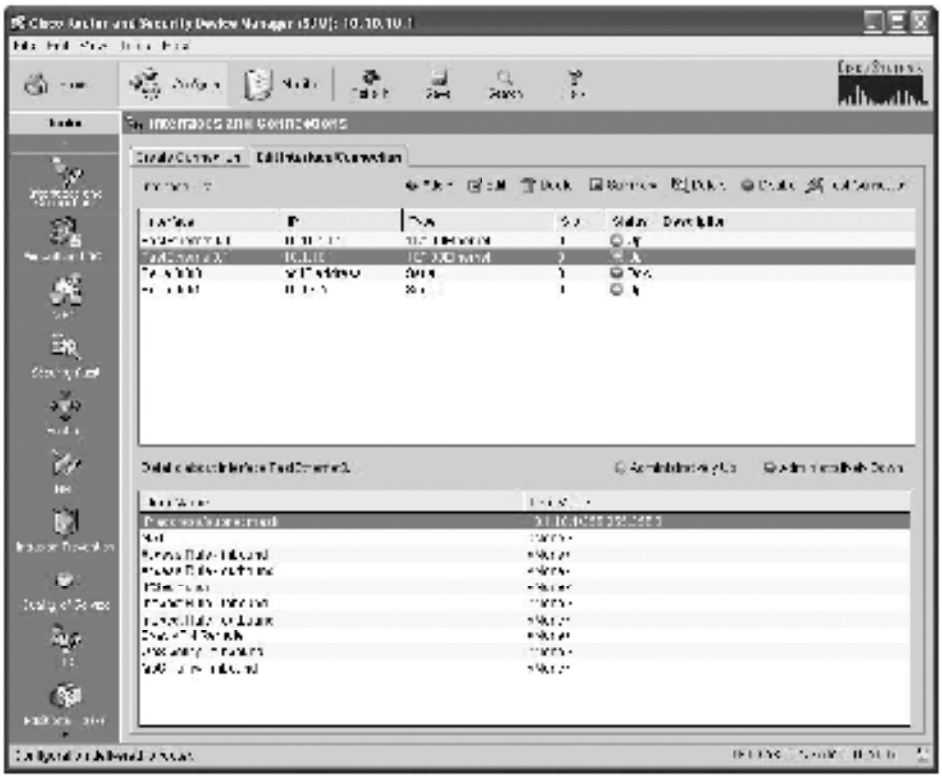


The LAN Wizard allows you to either choose straight routing (which is what we want to do here) or configure 802.1Q trunking, which I'll discuss in detail in Chapter 9, "Virtual LANs." I configured the IP address and mask and then clicked Next.

What's cool about the SDM at this point is that it would build a DHCP server for this LAN if I wanted it too. Man, this is easy.



Since I accidentally entered the wrong IP address for F0/1, the only way to change it now is to choose Configure and Edit Interface/Connection in the SDM, or use the CLI.



From here, I can double-click on the FastEthernet 0/1 interface and change the IP address.

After using the LAN Wizard to set up F0/0, I had to save the configuration and then reconfigure my PC into the right network and reconnect to SDM to verify my configuration.

We're good—R3 is now configured! Even though my console and VTY password get configured automatically when I set up the user todd, I still had to choose Configure and then Additional Tasks and then Router Properties to set the hostname and enable secret password.

871W Configuration

Although I can configure the 871 router with SDM, I'll configure it using the CLI. First, I need to erase the default configuration and reload it, as I did with the other routers (except R3).

```
Router>en
Router#config t
Router(config)#hostname 871W
871W(config)#int vlan 1
871W(config-if)#ip address 10.1.11.2 255.255.255.0
```

```
871W(config-if)#no shut
871W(config-if)#int dot11radio 0
871W(config-if)#ip address 10.1.12.1 255.255.255.0
871W(config-if)#no shut
871W(config-if)#ssid R3WLAN
871W(config-if-ssid)#guest-mode
871W(config-if-ssid)#authentication open
871W(config-if-ssid)#infrastructure-ssid
871W(config-if-ssid)#line con 0
871W(config-line)#password console
871W(config-line)#logging sync
871W(config-line)#exec-timeout 0 0
871W(config-line)#exit
871W(config)#line vty 0 ?
  <1-4>  Last Line number
  <cr>
871W(config)#line vty 0 4
871W(config-line)#password telnet
871W(config-line)#login
871W(config-line)#ip dhcp pool R3WLAN
871W(dhcp-config)#network 10.1.12.0 255.255.255.0
871W(dhcp-config)#default-router 10.1.12.1
871W(dhcp-config)#exit
871W(config)#ip dhcp excluded-address 10.1.12.1
871W(config)#exit
871W#copy run start
Destination filename [startup-config]?[enter]
Building configuration...
[OK]
871W#
```

The 871W has a four-port switch, which means that you've got to place the IP address under the management VLAN interface. You just can't get away with simply putting IP addresses on layer 2 switch interfaces.

To be totally honest, I think this was a faster configuration than using SDM. But I guess, in production, the SDM with HTTPS would really be a more secure way to administer the router. And as promised, I'll show you soon (in Chapter 12) why using SDM is the easier way to go when you want to set up wireless security.

Let's take a look at the routing table now:

```
871W#sh ip route
     10.0.0.0/24 is subnetted, 2 subnets
```

```
C       10.1.11.0 is directly connected, Vlan1
C       10.1.12.0 is directly connected, Dot11Radio0
```

We have both our networks showing directly connected. Let's configure our last device, and then we'll start configuring routing.

1242AP Configuration

Configuring the 1242AP is a bit different because it's an access point (again, think hub), not a router. I'll configure this device from the CLI, but you can use an HTTP interface as well. But you can't use SDM. The HTTP interface will be easier to use when we start adding security and when we get into some more complex configurations.

Check out the output:

```
ap>en
Password:
ap#config t
ap(config)#hostname 1242AP
1242AP(config)#enable secret todd
242AP(config)#int dot11Radio 0
1242AP(config-if)#description CORPWLAN
1242AP(config-if)#no shutdown
1242AP(config-if)#ssid CORPWLAN
1242AP(config-if-ssid)#guest-mode
1242AP(config-if-ssid)#authentication open
1242AP(config-if-ssid)#infrastructure-ssid
1242AP(config-if-ssid)#exit
1242AP(config-if)#exit
1242AP(config)#line con 0
1242AP(config-line)#password console
1242AP(config-line)#login
1242AP(config-line)#logging synchronous
1242AP(config-line)#exec-timeout 0 0
1242AP(config-line)#exit
1242AP(config)#line vty 0 ?
  <1-15>  Last Line number
  <cr>
1242AP(config)#line vty 0 15
1242AP(config-line)#password telnet
1242AP(config-line)#login
1242AP(config-line)#int bvi 1
1242AP(config-if)#ip address 10.1.1.2 255.255.255.0
1242AP(config-if)#no shut
1242AP(config-if)#exit
```

```
1242AP(config)#ip default-gateway 10.1.1.1
1242AP(config)#ip dhcp pool CORPWLAN
1242AP(dhcp-config)#network 10.1.1.0 255.255.255.0
1242AP(dhcp-config)#default-router 10.1.1.1
1242AP(dhcp-config)#exit
1242AP(config)#ip dhcp excluded-address 10.1.1.1
1242AP(config)#ip dhcp excluded-address 10.1.1.2
1242AP(config)#no ip domain-lookup
1242AP(config)#^Z
1242AP#copy run start
Destination filename [startup-config]?[enter]
Building configuration...
[OK]
1242AP#
```

Even though the SSID configuration is the same as it is for the R2 routed radio interface, notice there's no IP address under the Dot11radio 0 interface. Why? Because it's not a routed port, so the IP address is instead placed under the Bridge Virtual Interface (BVI). I also set a default gateway so this device can be managed from outside the LAN.

You need to know that just as with a switch, you don't need to add an IP address to the AP for it to function. I could just as easily have added the DHCP pool to the Corp router for the wireless LAN, not added an IP address or pool to the AP at all, and it still would have worked just the same.

Configuring IP Routing in Our Network

Our network is good to go—right? After all, it's been correctly configured with IP addressing, administrative functions, and even clocking (automatically on the ISR routers). But how does a router send packets to remote networks when the only way it can send them is by looking at the routing table to find out how to get to the remote networks? Our configured routers only have information about directly connected networks in each routing table. And what happens when a router receives a packet for a network that isn't listed in the routing table? It doesn't send a broadcast looking for the remote network—the router just discards it. Period.

So we're not exactly ready to rock after all. But no worries—there are several ways to configure the routing tables to include all the networks in our little internetwork so that packets will be forwarded. And what's best for one network isn't necessarily what's best for another. Understanding the different types of routing will really help you come up with the best solution for your specific environment and business requirements.

You'll learn about the following types of routing in the following sections:

- Static routing
- Default routing
- Dynamic routing

I'm going to start off by describing and implementing static routing on our network because if you can implement static routing *and* make it work, it means you have a solid understanding of the internetwork. So let's get started.

Static Routing

Static routing occurs when you manually add routes in each router's routing table. There are pros and cons to static routing, but that's true for all routing processes.

Static routing has the following benefits:

- There is no overhead on the router CPU, which means you could possibly buy a cheaper router than you would use if you were using dynamic routing.
- There is no bandwidth usage between routers, which means you could possibly save money on WAN links.
- It adds security because the administrator can choose to allow routing access to certain networks only.

Static routing has the following disadvantages:

- The administrator must really understand the internetwork and how each router is connected in order to configure routes correctly.
- If a network is added to the internetwork, the administrator has to add a route to it on all routers—by hand.
- It's not feasible in large networks because maintaining it would be a full-time job in itself.

Okay—that said, here's the command syntax you use to add a static route to a routing table:

```
ip route [destination_network] [mask] [next-hop_address or
  exitinterface] [administrative_distance] [permanent]
```

This list describes each command in the string:

`ip route` The command used to create the static route.

`destination_network` The network you're placing in the routing table.

`mask` The subnet mask being used on the network.

`next-hop_address` The address of the next-hop router that will receive the packet and forward it to the remote network. This is a router interface that's on a directly connected network. You must be able to ping the router interface before you add the route. If you type in the wrong next-hop address or the interface to that router is down, the static route will show up in the router's configuration but not in the routing table.

`exitinterface` Used in place of the next-hop address if you want, and shows up as a directly connected route.

`administrative_distance` By default, static routes have an administrative distance of 1 (or even 0 if you use an exit interface instead of a next-hop address). You can change the default value by adding an administrative weight at the end of the command. I'll talk a lot more about this subject later in the chapter when we get to the section on dynamic routing.

permanent If the interface is shut down or the router can't communicate to the next-hop router, the route will automatically be discarded from the routing table. Choosing the **permanent** option keeps the entry in the routing table no matter what happens.

Before we dive into configuring static routes, let's take a look at a sample static route and see what we can find out about it.

```
Router(config)#ip route 172.16.3.0 255.255.255.0 192.168.2.4
```

- The **ip route** command tells us simply that it is a static route.
- 172.16.3.0 is the remote network we want to send packets to.
- 255.255.255.0 is the mask of the remote network.
- 192.168.2.4 is the next hop, or router, we will send packets to.

However, suppose the static route looked like this:

```
Router(config)#ip route 172.16.3.0 255.255.255.0 192.168.2.4 150
```

The 150 at the end changes the default administrative distance (AD) of 1 to 150. No worries—I'll talk much more about AD when we get into dynamic routing. For now, just remember that the AD is the trustworthiness of a route, where 0 is best and 255 is worst.

One more example, then we'll start configuring:

```
Router(config)#ip route 172.16.3.0 255.255.255.0 s0/0/0
```

Instead of using a next-hop address, we can use an exit interface that will make the route show up as a directly connected network. Functionally, the next hop and exit interface work exactly the same. To help you understand how static routes work, I'll demonstrate the configuration on the internetwork shown previously in Figure 6.9.

Corp

Each routing table automatically includes directly connected networks. To be able to route to all networks within the internetwork, the routing table must include information that describes where these other networks are located and how to get to them.

The Corp router is connected to five networks. For the Corp router to be able to route to all networks, the following networks have to be configured into its routing table:

- 10.1.6.0
- 10.1.7.0
- 10.1.8.0
- 10.1.9.0
- 10.1.10.0
- 10.1.11.0
- 10.1.12.0

The following router output shows the static routes on the Corp router and the routing table after the configuration. For the Corp router to find the remote networks, I had to place an entry into the routing table describing the remote network, the remote mask, and where to send the packets. I am going to add a "150" at the end of each line to raise the administrative distance. (When we get to dynamic routing, you'll see why I did it this way.)

```
Corp(config)#ip route 10.1.6.0 255.255.255.0 10.1.2.2 150
Corp(config)#ip route 10.1.6.0 255.255.255.0 10.1.3.2 151
Corp(config)#ip route 10.1.7.0 255.255.255.0 10.1.3.2 150
Corp(config)#ip route 10.1.7.0 255.255.255.0 10.1.2.2 151
Corp(config)#ip route 10.1.8.0 255.255.255.0 10.1.4.2 150
Corp(config)#ip route 10.1.9.0 255.255.255.0 10.1.4.2 150
Corp(config)#ip route 10.1.10.0 255.255.255.0 10.1.5.2 150
Corp(config)#ip route 10.1.11.0 255.255.255.0 10.1.5.2 150
Corp(config)#ip route 10.1.12.0 255.255.255.0 10.1.5.2 150
Corp(config)#do show run | begin ip route
ip route 10.1.6.0 255.255.255.0 10.1.2.2 150
ip route 10.1.6.0 255.255.255.0 10.1.3.2 151
ip route 10.1.7.0 255.255.255.0 10.1.3.2 150
ip route 10.1.7.0 255.255.255.0 10.1.2.2 151
ip route 10.1.8.0 255.255.255.0 10.1.4.2 150
ip route 10.1.9.0 255.255.255.0 10.1.4.2 150
ip route 10.1.10.0 255.255.255.0 10.1.5.2 150
ip route 10.1.11.0 255.255.255.0 10.1.5.2 150
ip route 10.1.12.0 255.255.255.0 10.1.5.2 150
```

For networks 10.1.6.0 and 10.1.7.0, I put in both paths to each network, but I made one link a higher (151) AD. This will be a backup route in case the other link fails. If I made them both the same AD, we would end up with a routing loop. (Static routing can't handle multiple links to the same destination.) After the router is configured, you can type **show ip route** to see the static routes:

```
Corp(config)#do show ip route
     10.0.0.0/24 is subnetted, 12 subnets
S       10.1.11.0 [150/0] via 10.1.5.2
S       10.1.10.0 [150/0] via 10.1.5.2
S       10.1.9.0 [150/0] via 10.1.4.2
S       10.1.8.0 [150/0] via 10.1.4.2
S       10.1.12.0 [150/0] via 10.1.5.2
C       10.1.3.0 is directly connected, Serial0/0/1
C       10.1.2.0 is directly connected, Serial0/0/0
C       10.1.1.0 is directly connected, FastEthernet0/1
```

```
S       10.1.7.0 [150/0] via 10.1.3.2
S       10.1.6.0 [150/0] via 10.1.2.2
C       10.1.5.0 is directly connected, Serial0/2/0
C       10.1.4.0 is directly connected, Serial0/1/0
```

The Corp router is configured to route and know about all routes to all networks. I configured two routes to each remote network on R1, but the routing table will only show the route with the lower AD. The other link will show up in the routing table only if the link with that lower value it's currently using fails.

I want you to understand that if the routes don't appear in the routing table, it's because the router can't communicate with the next-hop address you've configured. You can use the `permanent` parameter to keep the route in the routing table even if the next-hop device can't be contacted.

The S in the preceding routing table entries means that the network is a static entry. The `[1/0]` is the administrative distance and metric (something we'll cover later) to the remote network. Here, the next-hop interface is 0, indicating that it's directly connected.

Okay—we're good. The Corp router now has all the information it needs to communicate with the other remote networks. But keep in mind that if the R1, R2, R3, and 871W routers aren't configured with all the same information, the packets will simply be discarded. We'll need to fix this by configuring static routes.

Don't stress about the 150/151 at the end of the static route configuration. I promise I will discuss the topic really soon in this chapter, not a later one! Be assured that you don't need to worry about it at this point.

R1

The R1 router is directly connected to the networks 10.1.2.0, 10.1.3.0, 10.1.6.0, and 10.1.7.0, so we've got to configure the following static routes on the R1 router:

- 10.1.1.0
- 10.1.4.0
- 10.1.5.0
- 10.1.8.0
- 10.1.9.0
- 10.1.10.0
- 10.1.11.0
- 10.1.12.0

Here's the configuration for the R1 router. Remember, we'll never create a static route to any network we're directly connected to, and we can use the next hop of either 10.1.2.1 or

10.1.3.1 since we have two links between the Corp and R1 routers. I'll change between next hops so all data doesn't go down one link. It really doesn't matter which link I use since I can't load-balance with static routing. We'll be able to load-balance when we use dynamic routing like RIP, EIGRP, and OSPF, but for now, the links will just provide a backup route to each network. Let's check out the output:

```
R1(config)#ip route 10.1.1.0 255.255.255.0 10.1.2.1 150
R1(config)#ip route 10.1.1.0 255.255.255.0 10.1.3.1 151
R1(config)#ip route 10.1.4.0 255.255.255.0 10.1.2.1 150
R1(config)#ip route 10.1.4.0 255.255.255.0 10.1.3.1 151
R1(config)#ip route 10.1.5.0 255.255.255.0 10.1.2.1 150
R1(config)#ip route 10.1.5.0 255.255.255.0 10.1.3.1 151
R1(config)#ip route 10.1.8.0 255.255.255.0 10.1.3.1 150
R1(config)#ip route 10.1.8.0 255.255.255.0 10.1.2.1 151
R1(config)#ip route 10.1.9.0 255.255.255.0 10.1.3.1 150
R1(config)#ip route 10.1.9.0 255.255.255.0 10.1.2.1 151
R1(config)#ip route 10.1.10.0 255.255.255.0 10.1.3.1 150
R1(config)#ip route 10.1.10.0 255.255.255.0 10.1.2.1 151
R1(config)#ip route 10.1.11.0 255.255.255.0 10.1.3.1 150
R1(config)#ip route 10.1.11.0 255.255.255.0 10.1.2.1 151
R1(config)#ip route 10.1.12.0 255.255.255.0 10.1.3.1 150
R1(config)#ip route 10.1.12.0 255.255.255.0 10.1.2.1 151
R1(config)#do show run | begin ip route
ip route 10.1.1.0 255.255.255.0 10.1.2.1 150
ip route 10.1.1.0 255.255.255.0 10.1.3.1 151
ip route 10.1.4.0 255.255.255.0 10.1.2.1 150
ip route 10.1.4.0 255.255.255.0 10.1.3.1 151
ip route 10.1.5.0 255.255.255.0 10.1.2.1 150
ip route 10.1.5.0 255.255.255.0 10.1.3.1 151
ip route 10.1.8.0 255.255.255.0 10.1.3.1 150
ip route 10.1.8.0 255.255.255.0 10.1.2.1 151
ip route 10.1.9.0 255.255.255.0 10.1.3.1 150
ip route 10.1.9.0 255.255.255.0 10.1.2.1 151
ip route 10.1.10.0 255.255.255.0 10.1.3.1 150
ip route 10.1.10.0 255.255.255.0 10.1.2.1 151
ip route 10.1.11.0 255.255.255.0 10.1.3.1 150
ip route 10.1.11.0 255.255.255.0 10.1.2.1 151
ip route 10.1.12.0 255.255.255.0 10.1.3.1 150
ip route 10.1.12.0 255.255.255.0 10.1.2.1 151
```

This was a pretty long configuration because I configured two paths to each network. By looking at the routing table, you can see that the R1 router now understands how to find each network:

```
R1(config)#do show ip route
     10.0.0.0/24 is subnetted, 12 subnets
S       10.1.11.0 [150/0] via 10.1.3.1
S       10.1.10.0 [150/0] via 10.1.3.1
S       10.1.9.0 [150/0] via 10.1.3.1
S       10.1.8.0 [150/0] via 10.1.3.1
S       10.1.12.0 [150/0] via 10.1.3.1
C       10.1.3.0 is directly connected, Serial0/0/1
C       10.1.2.0 is directly connected, Serial0/0/0
S       10.1.1.0 [150/0] via 10.1.2.1
C       10.1.7.0 is directly connected, FastEthernet0/1
C       10.1.6.0 is directly connected, FastEthernet0/0
S       10.1.5.0 [150/0] via 10.1.2.1
S       10.1.4.0 [150/0] via 10.1.2.1
```

The R1 router now has a complete routing table. As soon as the other routers in the internetwork have all the networks in their routing table, R1 will be able to communicate with all remote networks.

Remember, the route with the higher administrative distance will not show up in the routing table unless the route with the lower administrative distance goes away.

R2

The R2 router is directly connected to three networks 10.1.4.0, 10.1.8.0, and 10.1.9.0, so these routes need to be added:

- 10.1.1.0
- 10.1.2.0
- 10.1.3.0
- 10.1.5.0
- 10.1.6.0
- 10.1.7.0
- 10.1.10.0

- 10.1.11.0
- 10.1.12.0

Here's the configuration for the R2 router:

```
R2(config)#ip route 10.1.1.0 255.255.255.0 10.1.4.1 150
R2(config)#ip route 10.1.2.0 255.255.255.0 10.1.4.1 150
R2(config)#ip route 10.1.3.0 255.255.255.0 10.1.4.1 150
R2(config)#ip route 10.1.5.0 255.255.255.0 10.1.4.1 150
R2(config)#ip route 10.1.6.0 255.255.255.0 10.1.4.1 150
R2(config)#ip route 10.1.7.0 255.255.255.0 10.1.4.1 150
R2(config)#ip route 10.1.10.0 255.255.255.0 10.1.4.1 150
R2(config)#ip route 10.1.11.0 255.255.255.0 10.1.4.1 150
R2(config)#ip route 10.1.12.0 255.255.255.0 10.1.4.1 150
R2(config)#do show run | begin ip route
ip route 10.1.1.0 255.255.255.0 10.1.4.1 150
ip route 10.1.2.0 255.255.255.0 10.1.4.1 150
ip route 10.1.3.0 255.255.255.0 10.1.4.1 150
ip route 10.1.5.0 255.255.255.0 10.1.4.1 150
ip route 10.1.6.0 255.255.255.0 10.1.4.1 150
ip route 10.1.7.0 255.255.255.0 10.1.4.1 150
ip route 10.1.10.0 255.255.255.0 10.1.4.1 150
ip route 10.1.11.0 255.255.255.0 10.1.4.1 150
ip route 10.1.12.0 255.255.255.0 10.1.4.1 150
```

The following output shows the routing table on the R2 router:

```
R2(config)#do show ip route
     10.0.0.0/24 is subnetted, 12 subnets
S       10.1.11.0 [150/0] via 10.1.4.1
S       10.1.10.0 [150/0] via 10.1.4.1
C       10.1.9.0 is directly connected, FastEthernet0/0
C       10.1.8.0 is directly connected, Dot11Radio0/3/0
S       10.1.12.0 [150/0] via 10.1.4.1
S       10.1.3.0 [150/0] via 10.1.4.1
S       10.1.2.0 [150/0] via 10.1.4.1
S       10.1.1.0 [150/0] via 10.1.4.1
S       10.1.7.0 [150/0] via 10.1.4.1
S       10.1.6.0 [150/0] via 10.1.4.1
S       10.1.5.0 [150/0] via 10.1.4.1
C       10.1.4.0 is directly connected, Serial0/2/0
```

R2 now shows all 12 networks in the internetwork, so it too can now communicate with all routers and networks (that are configured so far).

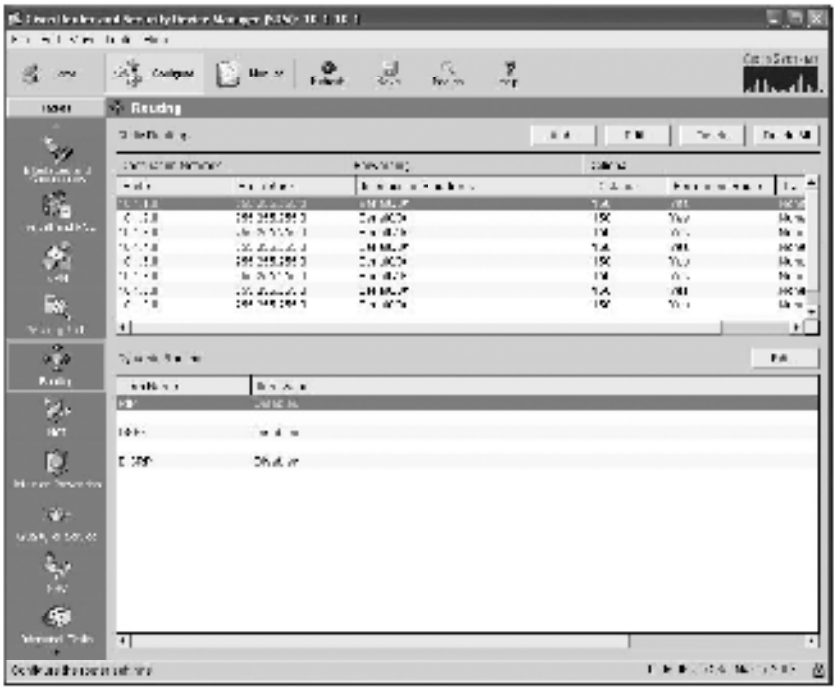
R3

The R3 router is directly connected to networks 10.1.5.0, 10.1.10.0, and 10.1.11.0, but we need to add these routes:

- 10.1.1.0
- 10.1.2.0
- 10.1.3.0
- 10.1.4.0
- 10.1.6.0
- 10.1.7.0
- 10.1.8.0
- 10.1.9.0
- 10.1.12.0

As before, I'm going to use SDM to configure the static routing for the R3 router. The configuration is pretty simple, and I can use either the next-hop address or the exit interface. Since I like to type as little as possible, I'm going with the exit interface because it only takes a mouse click.

After all our routes are configured, we can see them in the routing screen.



From this screen, it is easy to edit the static routes.

Let's take a look at the configuration and the routing table uploaded to the router from SDM:

```
R3#show run | begin ip route
ip route 10.1.1.0 255.255.255.0 Serial0/0/1 150 permanent
ip route 10.1.2.0 255.255.255.0 Serial0/0/1 150 permanent
ip route 10.1.3.0 255.255.255.0 Serial0/0/1 150 permanent
ip route 10.1.4.0 255.255.255.0 Serial0/0/1 150 permanent
ip route 10.1.6.0 255.255.255.0 Serial0/0/1 150 permanent
ip route 10.1.7.0 255.255.255.0 Serial0/0/1 150 permanent
ip route 10.1.8.0 255.255.255.0 Serial0/0/1 150 permanent
ip route 10.1.9.0 255.255.255.0 Serial0/0/1 150 permanent
ip route 10.1.12.0 255.255.255.0 FastEthernet0/1 150 permanent
R3#show ip route
     10.0.0.0/24 is subnetted, 12 subnets
C       10.1.11.0 is directly connected, FastEthernet0/1
C       10.1.10.0 is directly connected, FastEthernet0/0
S       10.1.9.0 is directly connected, Serial0/0/1
S       10.1.8.0 is directly connected, Serial0/0/1
```

```
S       10.1.12.0 is directly connected, FastEthernet0/1
S       10.1.3.0 is directly connected, Serial0/0/1
S       10.1.2.0 is directly connected, Serial0/0/1
S       10.1.1.0 is directly connected, Serial0/0/1
S       10.1.7.0 is directly connected, Serial0/0/1
S       10.1.6.0 is directly connected, Serial0/0/1
C       10.1.5.0 is directly connected, Serial0/0/1
S       10.1.4.0 is directly connected, Serial0/0/1
R3#
```

Looking at the `show ip route` command output, you can see that the static routes are listed as directly connected. Strange? Not really, because I used the exit interface instead of the next-hop address, and functionally, there's no difference. We really don't need the `permanent` command because all that will do is ensure that the route stays in the routing table even if the link to that route goes down. I configured the `permanent` command only because it was easy to do with SDM (just another mouse click). We're almost there—just one more router to go: the 871W.

871W

Now for this router, I'm going to configure something called default routing since the 871W is configured as a stub. A stub indicates that the wireless network in this design has only one way out to reach all other networks. I'll show you the configuration, verify the network in the next section, then I'll discuss default routing in detail. Here's the configuration:

```
871W(config)#ip route 0.0.0.0 0.0.0.0 10.1.11.1
871W(config)#ip classless
871W(config)#do show ip route
     10.0.0.0/24 is subnetted, 2 subnets
C       10.1.11.0 is directly connected, Vlan1
C       10.1.12.0 is directly connected, Dot11Radio0
S*   0.0.0.0/0 [1/0] via 10.1.11.1
871W(config)#
```

This seems a lot easier, doesn't it? And it is, but there's a catch—you can't do things like this on all routers, only on stub networks. I could've used default routing in routers R1 and R2 as well, and I didn't add the 150 to this default route even though I easily could have. I didn't do that because it's really simple to just remove the route when we get to dynamic routing later.

So we're there—we've done it! All the routers have the correct routing table, so all routers and hosts should be able to communicate without a hitch—for now. But if you add even one more network or another router to the internetwork, you'll have to update each and every router's routing tables by hand—yikes! This isn't a problem at all if you've got a small network, but it's obviously extremely time-consuming if you're dealing with a large internetwork!

Verifying Your Configuration

We're not done yet—once all the routers' routing tables are configured, they need to be verified. The best way to do this, besides using the `show ip route` command, is with the Ping program. I'll start by pinging from the 1242AP to the 871W router.

Here's the output:

```
871W#ping 10.1.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

From router 871W, a ping to HostA, B, C, and D will also test for good IP connectivity. Here's the router output:

```
871W#ping 10.1.6.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.6.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/6/12 ms
871W#ping 10.1.7.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.7.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms
871W#ping 10.1.9.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.9.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms
871W#ping 10.1.10.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.10.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5)
```

Also, we can trace from the 871W router to see the hops the packet takes to get to HostA:

```
871W#trace 10.1.6.2
Type escape sequence to abort.
Tracing the route to 10.1.6.2
  1 10.1.11.1 0 msec 0 msec 0 msec
  2 10.1.5.1 4 msec 0 msec 4 msec
```

```
3 10.1.2.2 0 msec 0 msec 4 msec
4 10.1.6.2 4 msec 4 msec *
```

Since we can communicate from end to end and to each host without a problem, our static route configuration has been successful!

Default Routing

We use *default routing* to send packets with a remote destination network not in the routing table to the next-hop router. You should only use default routing on stub networks—those with only one exit path out of the network.

In the internetworking example used in the previous section, the only routers that are considered to be in a stub network are R1, R2, and the 871W. If you tried to put a default route on router R3, packets wouldn't be forwarded to the correct networks because they have more than one interface routing to other routers. You can easily create loops with default routing, so be careful!

To configure a default route, you use wildcards in the network address and mask locations of a static route (as I demonstrated in the 871W configuration). In fact, you can just think of a default route as a static route that uses wildcards instead of network and mask information.

By using a default route, you can just create one static route entry instead. This sure is easier then typing in all those routes!

```
871W(config)#ip route 0.0.0.0 0.0.0.0 10.1.11.1
871W(config)#ip classless
871W(config)#do show ip route
  Gateway of last resort is 10.1.11.1 to network 0.0.0.0
 10.0.0.0/24 is subnetted, 2 subnets
C       10.1.11.0 is directly connected, Vlan1
C       10.1.12.0 is directly connected, Dot11Radio0
S*   0.0.0.0/0 [1/0] via 10.1.11.1
871W(config)#
```

If you look at the routing table, you'll see only the two directly connected networks plus an S*, which indicates that this entry is a candidate for a default route. I could have completed the default route command another way:

```
871W(config)#ip route 0.0.0.0 0.0.0.0 vlan1
```

What this is telling us is that if you don't have an entry for a network in the routing table, just forward it out VLAN1 (which will send it out FastEthernet0/0). You can choose the IP address of the next-hop router or the exit interface—either way, it will work the same. Remember, I used this exit interface configuration with the R3 static route configs.

Notice also in the routing table that the gateway of last resort is now set. Even so, there's one more command you must be aware of when using default routes: the `ip classless` command.

All Cisco routers are classful routers, meaning they expect a default subnet mask on each interface of the router. When a router receives a packet for a destination subnet that's not in the routing table, it will drop the packet by default. If you're using default routing, you must use the `ip classless` command because it is possible that no remote subnets will be in the routing table.

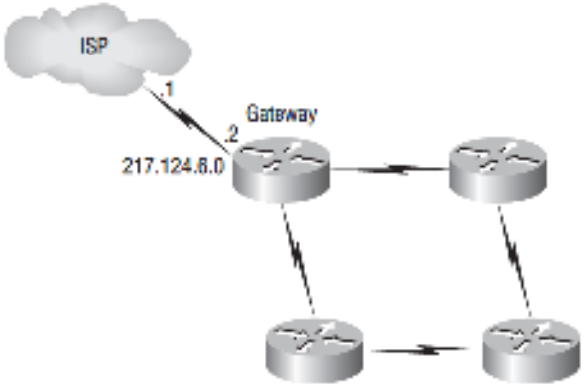
Since I have version 12.4 of the IOS on my routers, the `ip classless` command is on by default. If you're using default routing and this command isn't in your configuration, you will need to add it if you have subnetted networks on your routers. The command is shown here:

```
871W(config)#ip classless
```

Notice that it's a global configuration mode command. The interesting part of the `ip classless` command is that without it, default routing sometimes works but sometimes doesn't. To be on the safe side, you should always turn on the `ip classless` command when you use default routing.

There's another command you can use to configure a gateway of last resort—the `ip default-network` command. Figure 6.10 shows a network that needs to have a gateway of last resort statement configured.

FIGURE 6.10 Configuring a gateway of last resort



Here are three commands (all providing the same solution) for adding a gateway of last resort on the gateway router to the ISP.

```
Gateway(config)#ip route 0.0.0.0 0.0.0.0 217.124.6.1

Gateway(config)#ip route 0.0.0.0 0.0.0.0 s0/0

Gateway(config)#ip default-network 217.124.6.0
```

As I said before, all three of these commands would accomplish the goal of setting the gateway of last resort, but there are some small differences between them. First, the exit interface

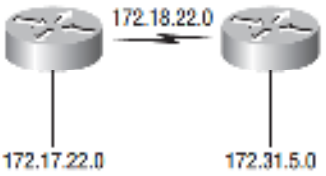
solution would be used over the other two solutions because it has an AD of 0. Also, the `ip default-network` command would advertise the default network when you configure an IGP (like RIP) on the router. This is so other routers in your internetwork will receive this route as a default route automatically.

But what happens if you misconfigured a default route? Let's take a look at the output of a `show ip route` command and compare that to the network in Figure 6.11 and see if you can find a problem:

```
Router#sh ip route
[output cut]
Gateway of last resort is 172.19.22.2 to network 0.0.0.0

C       172.17.22.0 is directly connected, FastEthernet0/0
C       172.18.22.0  is directly connected, Serial0/0
S*      0.0.0.0/0 [1/0] via 172.19.22.2
```

FIGURE 6.11 Misconfigured default route



Find anything? You can see by looking at the figure and the directly connected routes in the routing table that the WAN link is on network 172.18.22.0 and that the default route is forwarding all packets to the 172.19.22.0 network. This is just bad—it will never work, so the problem is a misconfigured static (default) route.

One last thing before moving on to dynamic routing. If you have the routing table output as shown in the following lines, what happens if the router receives a packet from 10.1.6.100 destined for host 10.1.8.5?

```
Corp#sh ip route
[output cut]
Gateway of last resort is 10.1.5.5 to network 0.0.0.0

R     10.1.3.0 [120/1] via 10.1.2.2, 00:00:00, Serial 0/0
C     10.1.2.0  is directly connected, Serial0/0
C     10.1.5.0  is directly connected, Serial0/1
C     10.1.6.0  is directly connected, Fastethernet0/0
R*    0.0.0.0/0 [120/0] via 10.1.5.5, 00:00:00 Serial 0/1
```

This is a tad different than what I've shown you up until now because the default route is listed as R*, which means it's a RIP-injected route. This is because someone configured the

`ip default-network` command on a remote router as well as configuring RIP, causing RIP to advertise this route through the internetwork as a default route. Since the destination address is 10.1.8.5 and there is no route to network 10.1.8.0, the router would use the default route and send the packet out serial 0/1.

Dynamic Routing

Dynamic routing is when protocols are used to find networks and update routing tables on routers. True, this is easier than using static or default routing, but it'll cost you in terms of router CPU processes and bandwidth on the network links. A routing protocol defines the set of rules used by a router when it communicates routing information between neighbor routers.

The routing protocol I'm going to talk about in this chapter is Routing Information Protocol (RIP) versions 1 and 2, with a bit of Interior Gateway Routing Protocol (IGRP) thrown in.

Two types of routing protocols are used in internetworks: interior gateway protocols (IGPs) and exterior gateway protocols (EGPs). IGPs are used to exchange routing information with routers in the same autonomous system (AS). An AS is a collection of networks under a common administrative domain, which basically means that all routers sharing the same routing table information are in the same AS. EGPs are used to communicate between ASes. An example of an EGP is Border Gateway Protocol (BGP), which is beyond the scope of this book.

Since routing protocols are so essential to dynamic routing, I'm going to give you the basic information you need to know about them next. Later on in this chapter, we'll focus on configuration.

Routing Protocol Basics

There are some important things you should know about routing protocols before getting deeper into RIP. Specifically, you need to understand administrative distances, the three different kinds of routing protocols, and routing loops. We will look at each of these in more detail in the following sections.

Administrative Distances

The *administrative distance (AD)* is used to rate the trustworthiness of routing information received on a router from a neighbor router. An administrative distance is an integer from 0 to 255, where 0 is the most trusted and 255 means no traffic will be passed via this route.

If a router receives two updates listing the same remote network, the first thing the router checks is the AD. If one of the advertised routes has a lower AD than the other, then the route with the lowest AD will be placed in the routing table.

If both advertised routes to the same network have the same AD, then routing protocol metrics (such as *hop count* or bandwidth of the lines) will be used to find the best path to the remote network. The advertised route with the lowest metric will be placed in the routing table. But if both advertised routes have the same AD as well as the same metrics, then the routing protocol will load-balance to the remote network (which means that it sends packets down each link).

Table 6.2 shows the default administrative distances that a Cisco router uses to decide which route to take to a remote network.

TABLE 6.2 Default Administrative Distances

Route Source	Default AD
Connected interface	0
Static route	1
EIGRP	90
IGRP	100
OSPF	110
RIP	120
External EIGRP	170
Unknown	255 (this route will never be used)

If a network is directly connected, the router will always use the interface connected to the network. If you configure a static route, the router will then believe that route over any other learned routes. You can change the administrative distance of static routes, but by default, they have an AD of 1. In our static route configuration, the AD of each route is set at 150 or 151. This lets us configure routing protocols without having to remove the static routes. They'll be used as backup routes in case the routing protocol experiences a failure of some type.

For example, if you have a static route, a RIP-advertised route, and an IGRP-advertised route listing the same network, then by default, the router will always use the static route unless you change the AD of the static route—which we did.

Routing Protocols

There are three classes of routing protocols:

Distance vector The *distance-vector protocols* find the best path to a remote network by judging distance. Each time a packet goes through a router, that's called a *hop*. The route with the least number of hops to the network is determined to be the best route. The vector indicates the direction to the remote network. Both RIP and IGRP are distance-vector routing protocols. They send the entire routing table to directly connected neighbors.

Link state In *link-state protocols*, also called *shortest-path-first protocols*, the routers each create three separate tables. One of these tables keeps track of directly attached

neighbors, one determines the topology of the entire internetwork, and one is used as the routing table. Link-state routers know more about the internetwork than any distance-vector routing protocol. OSPF is an IP routing protocol that is completely link state. Link-state protocols send updates containing the state of their own links to all other routers on the network.

Hybrid *Hybrid protocols* use aspects of both distance vector and link state—for example, EIGRP.

There's no set way of configuring routing protocols for use with every business. This is something you really have to do on a case-by-case basis. If you understand how the different routing protocols work, you can make good, solid decisions that truly meet the individual needs of any business.

Distance-Vector Routing Protocols

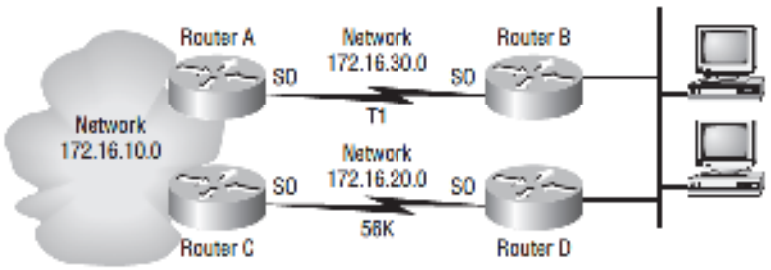
The distance-vector routing algorithm passes complete routing table contents to neighboring routers, which then combine the received routing table entries with their own routing tables to complete the router's routing table. This is called routing by rumor, because a router receiving an update from a neighbor router believes the information about remote networks without actually finding out for itself.

It's possible to have a network that has multiple links to the same remote network, and if that's the case, the administrative distance of each received update is checked first. If the AD is the same, the protocol will have to use other metrics to determine the best path to use to that remote network.

RIP uses only hop count to determine the best path to a network. If RIP finds more than one link with the same hop count to the same remote network, it will automatically perform a round-robin load balancing. RIP can perform load balancing for up to six equal-cost links (four by default).

However, a problem with this type of routing metric arises when the two links to a remote network are different bandwidths but the same hop count. Figure 6.12, for example, shows two links to remote network 172.16.10.0.

FIGURE 6.12 Pinhole congestion

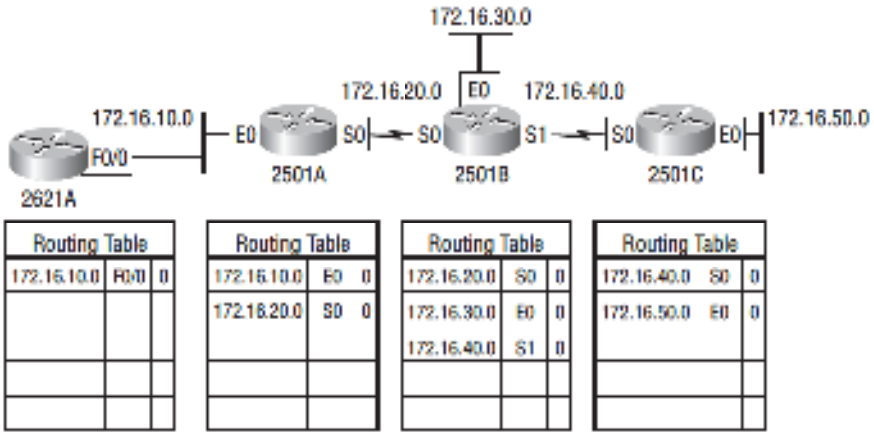


Since network 172.16.30.0 is a T1 link with a bandwidth of 1.544Mbps and network 172.16.20.0 is a 56K link, you'd want the router to choose the T1 over the 56K link, right? But because hop count is the only metric used with RIP routing, the two links would be seen as being of equal cost. This little snag is called *pinhole congestion*.

It's important to understand what a distance-vector routing protocol does when it starts up. In Figure 6.13, the four routers start off with only their directly connected networks in their routing tables. After a distance-vector routing protocol is started on each router, the routing tables are updated with all route information gathered from neighbor routers.

As shown in Figure 6.13, each router has only the directly connected networks in each routing table. Each router sends its complete routing table out to each active interface. The routing table of each router includes the network number, exit interface, and hop count to the network.

FIGURE 6.13 The internetwork with distance-vector routing



In Figure 6.14, the routing tables are complete because they include information about all the networks in the internetwork. They are considered *converged*. When the routers are converging, it is possible that no data will be passed. That's why fast convergence time is a serious plus. In fact, that's one of the problems with RIP—its slow convergence time.

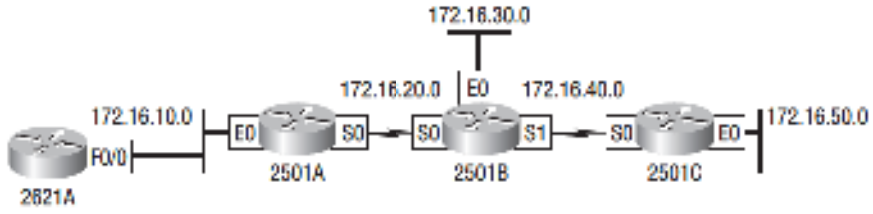
The routing table in each router keeps information regarding the remote network number, the interface to which the router will send packets to reach that network, and the hop count or metric to the network.

Routing Loops

Distance-vector routing protocols keep track of any changes to the internetwork by broadcasting periodic routing updates out all active interfaces. This broadcast includes the complete routing table. This works just fine, but it's expensive in terms of CPU process and link bandwidth. And if a network outage happens, real problems can occur. Plus, the slow convergence of distance-vector routing protocols can result in inconsistent routing tables and routing loops.

Routing loops can occur because every router isn't updated simultaneously, or even close to it. Here's an example—let's say that the interface to Network 5 in Figure 6.15 fails. All routers know about Network 5 from RouterE. RouterA, in its tables, has a path to Network 5 through RouterB.

FIGURE 6.14 Converged routing tables



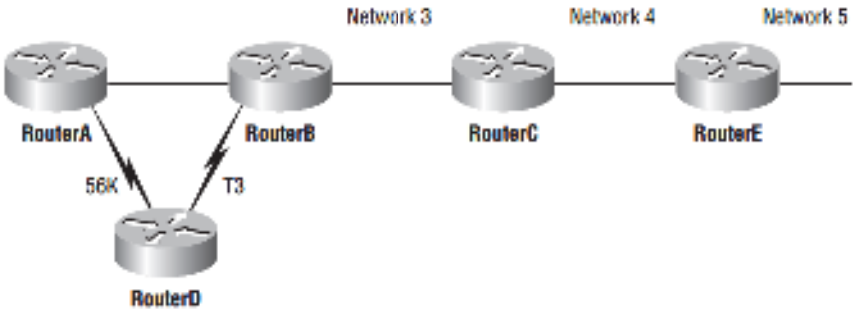
Routing Table		
172.16.10.0	FO/0	0
172.16.20.0	FO/0	1
172.16.30.0	FO/0	2
172.16.40.0	FO/0	2
172.16.50.0	FO/0	3

Routing Table		
172.16.10.0	E0	0
172.16.20.0	S0	0
172.16.30.0	S0	1
172.16.40.0	S0	1
172.16.50.0	S0	2

Routing Table		
172.16.20.0	S0	0
172.16.30.0	E0	0
172.16.40.0	S1	0
172.16.10.0	S0	1
172.16.50.0	S1	1

Routing Table		
172.16.40.0	S0	0
172.16.50.0	E0	0
172.16.10.0	S0	2
172.16.20.0	S0	1
172.16.30.0	S0	1

FIGURE 6.15 Routing loop example



When Network 5 fails, RouterE tells RouterC. This causes RouterC to stop routing to Network 5 through RouterE. But routers A, B, and D don't know about Network 5 yet, so they keep sending out update information. RouterC will eventually send out its update and cause B to stop routing to Network 5, but routers A and D are still not updated. To them, it appears that Network 5 is still available through RouterB with a metric of 3.

The problem occurs when RouterA sends out its regular 30-second "Hello, I'm still here—these are the links I know about" message, which includes the ability to reach Network 5, and now routers B and D receive the wonderful news that Network 5 can be reached from RouterA,

so routers B and D then send out the information that Network 5 is available. Any packet destined for Network 5 will go to RouterA, to RouterB, and then back to RouterA. This is a routing loop—how do you stop it?

Maximum Hop Count

The routing loop problem just described is called *counting to infinity*, and it's caused by gossip (broadcasts) and wrong information being communicated and propagated throughout the internetwork. Without some form of intervention, the hop count increases indefinitely each time a packet passes through a router.

One way of solving this problem is to define a *maximum hop count*. RIP permits a hop count of up to 15, so anything that requires 16 hops is deemed unreachable. In other words, after a loop of 15 hops, Network 5 will be considered down. Thus, the maximum hop count will control how long it takes for a routing table entry to become invalid or questionable.

Split Horizon

Another solution to the routing loop problem is called *split horizon*. This reduces incorrect routing information and routing overhead in a distance-vector network by enforcing the rule that routing information cannot be sent back in the direction from which it was received.

In other words, the routing protocol differentiates which interface a network route was learned on, and once this is determined, it won't advertise the route back out that same interface. This would have prevented RouterA from sending the updated information it received from RouterB back to RouterB.

Route Poisoning

Another way to avoid problems caused by inconsistent updates and stop network loops is *route poisoning*. For example, when Network 5 goes down, RouterF initiates route poisoning by advertising Network 5 as 16, or unreachable (sometimes referred to as *infinite*).

This poisoning of the route to Network 5 keeps RouterC from being susceptible to incorrect updates about the route to Network 5. When RouterC receives a route poisoning from RouterF, it sends an update, called a *poison reverse*, back to RouterF. This ensures that all routes on the segment have received the poisoned route information.

Holddowns

A *holddown* prevents regular update messages from reinstating a route that is going up and down (called *flapping*). Typically, this happens on a serial link that's losing connectivity and then coming back up. If there wasn't a way to stabilize this, the network would never converge and that one flapping interface could bring the entire network down!

Holddowns prevent routes from changing too rapidly by allowing time for either the downed route to come back up or the network to stabilize somewhat before changing to the next best route. These also tell routers to restrict, for a specific time period, changes that might affect recently removed routes. This prevents inoperative routes from being prematurely restored to other routers' tables.

Routing Information Protocol (RIP)

Routing Information Protocol (RIP) is a true distance-vector routing protocol. RIP sends the complete routing table out to all active interfaces every 30 seconds. RIP only uses hop count to determine the best way to a remote network, but it has a maximum allowable hop count of 15 by default, meaning that 16 is deemed unreachable. RIP works well in small networks, but it's inefficient on large networks with slow WAN links or on networks with a large number of routers installed.

RIP version 1 uses only *classful routing*, which means that all devices in the network must use the same subnet mask. This is because RIP version 1 doesn't send updates with subnet mask information in tow. RIP version 2 provides something called *prefix routing* and does send subnet mask information with the route updates. This is called *classless routing*.

In the following sections, we will discuss the RIP timers and then RIP configuration.

RIP Timers

RIP uses four different kinds of timers to regulate its performance:

Route update timer Sets the interval (typically 30 seconds) between periodic routing updates in which the router sends a complete copy of its routing table out to all neighbors.

Route invalid timer Determines the length of time that must elapse (180 seconds) before a router determines that a route has become invalid. It will come to this conclusion if it hasn't heard any updates about a particular route for that period. When that happens, the router will send out updates to all its neighbors letting them know that the route is invalid.

Holddown timer This sets the amount of time during which routing information is suppressed. Routes will enter into the holddown state when an update packet is received that indicated the route is unreachable. This continues either until an update packet is received with a better metric or until the holddown timer expires. The default is 180 seconds.

Route flush timer Sets the time between a route becoming invalid and its removal from the routing table (240 seconds). Before it's removed from the table, the router notifies its neighbors of that route's impending demise. The value of the route invalid timer must be less than that of the route flush timer. This gives the router enough time to tell its neighbors about the invalid route before the local routing table is updated.

Configuring RIP Routing

To configure RIP routing, just turn on the protocol with the `router rip` command and tell the RIP routing protocol which networks to advertise. That's it. Let's configure our five-router internetwork (Figure 6.9) with RIP routing.

Corp

RIP has an administrative distance of 120. Static routes have an administrative distance of 1 by default, and since we currently have static routes configured, the routing tables won't be

propagated with RIP information. However, because I added the 150/151 to the end of each static route, we're good to go.

You can add the RIP routing protocol by using the `router rip` command and the `network` command. The `network` command tells the routing protocol which classful network to advertise.

Look at the Corp router configuration and see how easy this is:

```
Corp#config t
Corp(config)#router rip
Corp(config-router)#network 10.0.0.0
```

That's it. Two or three commands and you're done—sure makes your job a lot easier than when using static routes, doesn't it? However, keep in mind the extra router CPU process and bandwidth that you're consuming.

Notice I didn't type in subnets, only the classful network address (all subnet bits and host bits off!). It is the job of the routing protocol to find the subnets and populate the routing tables. Since we have no router buddies running RIP, we won't see any RIP routes in the routing table yet.

Remember that RIP uses the classful address when configuring the network address. Because of this, all subnet masks must be the same on all devices in the network (this is called classful routing). To clarify this, let's say you're using a Class B network address of 172.16.0.0/24 with subnets 172.16.10.0, 172.16.20.0, and 172.16.30.0. You would only type in the classful network address of 172.16.0.0 and let RIP find the subnets and place them in the routing table.

R1

Let's configure our R1 router :

```
R1#config t
R1(config)#router rip
R1(config-router)#network 10.0.0.0
R1(config-router)#do show ip route
     10.0.0.0/24 is subnetted, 12 subnets
S       10.1.11.0 [150/0] via 10.1.3.1
S       10.1.10.0 [150/0] via 10.1.3.1
S       10.1.9.0 [150/0] via 10.1.3.1
S       10.1.8.0 [150/0] via 10.1.3.1
S       10.1.12.0 [150/0] via 10.1.3.1
C       10.1.3.0 is directly connected, Serial0/0/1
C       10.1.2.0 is directly connected, Serial0/0/0
R       10.1.1.0 [120/1] via 10.1.3.1, 00:00:04, Serial0/0/1
```

```
                [120/1] via 10.1.2.1, 00:00:04, Serial0/0/0
C       10.1.7.0 is directly connected, FastEthernet0/1
C       10.1.6.0 is directly connected, FastEthernet0/0
R       10.1.5.0 [120/1] via 10.1.3.1, 00:00:04, Serial0/0/1
                [120/1] via 10.1.2.1, 00:00:04, Serial0/0/0
R       10.1.4.0 [120/1] via 10.1.3.1, 00:00:09, Serial0/0/1
                [120/1] via 10.1.2.1, 00:00:09, Serial0/0/0
R1(config-router)#
```

That was pretty straightforward. Let's talk about this routing table. Since we have one RIP buddy out there that we are exchanging routing tables with, we can see the RIP networks coming from the Corp router. (All the other routes still show up as static.) RIP also found both connections to the Corp router and will load-balance between them.

R2

Let's configure our R2 router with RIP:

```
R2#config t
R2(config)#router rip
R2(config-router)#network 10.0.0.0
R2(config-router)#do show ip route
     10.0.0.0/24 is subnetted, 12 subnets
S       10.1.11.0 [150/0] via 10.1.4.1
S       10.1.10.0 [150/0] via 10.1.4.1
C       10.1.9.0 is directly connected, FastEthernet0/0
C       10.1.8.0 is directly connected, Dot11Radio0/3/0
S       10.1.12.0 [150/0] via 10.1.4.1
R       10.1.3.0 [120/1] via 10.1.4.1, 00:00:03, Serial0/2/0
R       10.1.2.0 [120/1] via 10.1.4.1, 00:00:03, Serial0/2/0
R       10.1.1.0 [120/1] via 10.1.4.1, 00:00:03, Serial0/2/0
R       10.1.7.0 [120/2] via 10.1.4.1, 00:00:03, Serial0/2/0
R       10.1.6.0 [120/2] via 10.1.4.1, 00:00:03, Serial0/2/0
R       10.1.5.0 [120/1] via 10.1.4.1, 00:00:03, Serial0/2/0
```

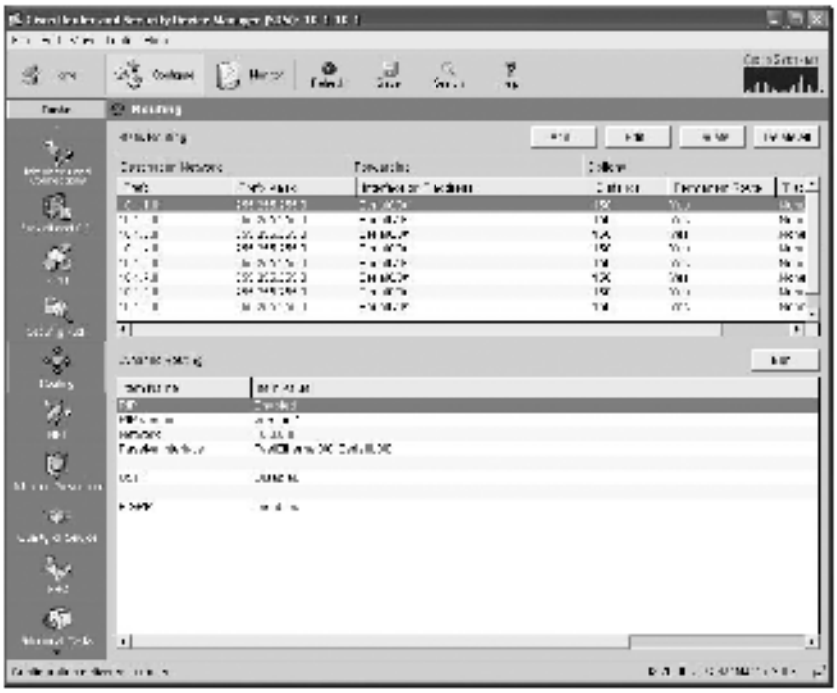
The routing table is growing Rs as we add RIP buddies! We can still see that all routes are in the routing table; some are still static routes. Two more routers to go.

R3

Let's configure our R3 router with RIP—as usual with R3, we'll use the SDM.

From the routing screen, I clicked the Edit button to the right of Dynamic Routing. I then was able to configure RIP and the network number and then clicked on the interfaces I didn't want RIP to be broadcast out. The interfaces that RIP will broadcast out will be unchecked.

These are called passive interfaces and we'll talk about it more in a minute. No reason to broadcast RIP out an interface where no routers will be.



From the SDM screen, we can see that we're done with R3.

871W

Here is the last router's RIP configuration:

```
871W#config t
871W(config)#no ip route 0.0.0.0 0.0.0.0 10.1.11.1
871W(config)#router rip
871W(config-router)#network 10.0.0.0
871W(config-router)#do sh ip route
     10.0.0.0/24 is subnetted, 12 subnets
C       10.1.11.0 is directly connected, Vlan1
R       10.1.10.0 [120/1] via 10.1.11.1, 00:00:23, Vlan1
R       10.1.9.0 [120/3] via 10.1.11.1, 00:00:23, Vlan1
R       10.1.8.0 [120/3] via 10.1.11.1, 00:00:23, Vlan1
C       10.1.12.0 is directly connected, Dot11Radio0
R       10.1.3.0 [120/2] via 10.1.11.1, 00:00:23, Vlan1
R       10.1.2.0 [120/2] via 10.1.11.1, 00:00:23, Vlan1
R       10.1.1.0 [120/2] via 10.1.11.1, 00:00:23, Vlan1
R       10.1.7.0 [120/3] via 10.1.11.1, 00:00:24, Vlan1
R       10.1.6.0 [120/3] via 10.1.11.1, 00:00:24, Vlan1
R       10.1.5.0 [120/1] via 10.1.11.1, 00:00:24, Vlan1
R       10.1.4.0 [120/2] via 10.1.11.1, 00:00:24, Vlan1
871W#
```

Finally, all routes showing in the routing table are RIP injected routes.

It's important to remember administrative distances and why we needed to either remove the static routes before we added RIP routing or set them higher than 120 as we did.

By default, directly connected routes have an administrative distance of 0, static routes have an administrative distance of 1, and RIP has an administrative distance of 120. I call RIP the "gossip protocol" because it reminds me of junior high school, where if you hear a rumor (advertised route), it just has to be true without exception. And that pretty much sums up how RIP behaves on an internetwork—rumor mill as protocol!

Verifying the RIP Routing Tables

Each routing table should now have all directly connected routes as well as RIP-injected routes received from neighboring routers.

This output shows us the contents of the Corp routing table:

```
     10.0.0.0/24 is subnetted, 12 subnets
R       10.1.11.0 [120/1] via 10.1.5.2, 00:00:28, Serial0/2/0
R       10.1.10.0 [120/1] via 10.1.5.2, 00:00:28, Serial0/2/0
```

```
R       10.1.9.0 [120/1] via 10.1.4.2, 00:00:26, Serial0/1/0
R       10.1.8.0 [120/1] via 10.1.4.2, 00:00:26, Serial0/1/0
R       10.1.12.0 [120/2] via 10.1.5.2, 00:00:28, Serial0/2/0
C       10.1.3.0 is directly connected, Serial0/0/1
C       10.1.2.0 is directly connected, Serial0/0/0
C       10.1.1.0 is directly connected, FastEthernet0/1
R       10.1.7.0 [120/1] via 10.1.3.2, 00:00:07, Serial0/0/1
                 [120/1] via 10.1.2.2, 00:00:10, Serial0/0/0
R       10.1.6.0 [120/1] via 10.1.3.2, 00:00:07, Serial0/0/1
                 [120/1] via 10.1.2.2, 00:00:10, Serial0/0/0
C       10.1.5.0 is directly connected, Serial0/2/0
C       10.1.4.0 is directly connected, Serial0/1/0
```

This output shows us that the routing table has the same entries that it had when we were using static routes—except for that R. The R means that the networks were added dynamically using the RIP routing protocol. The [120/1] is the administrative distance of the route (120) along with the number of hops to that remote network (1). From the Corp router, all networks are one hop away except network 10.1.12.0, which is two hops away.

So while yes, it's true that RIP has worked in our little internetwork, it's not the solution for every enterprise. That's because this technique has a maximum hop count of only 15 (16 is deemed unreachable). Plus, it performs full routing-table updates every 30 seconds, which would bring a larger internetwork to a painful crawl pretty quick!

There's one more thing I want to show you about RIP routing tables and the parameters used to advertise remote networks. Notice, as an example, that the following routing table shows [120/15] in the 10.1.3.0 network metric. This means that the administrative distance is 120, the default for RIP, but the hop count is 15. Remember that each time a router sends out an update to a neighbor router, it increments the hop count by one for each route.

```
R3#sh ip route
     10.0.0.0/24 is subnetted, 12 subnets
C       10.1.11.0 is directly connected, FastEthernet0/1
C       10.1.10.0 is directly connected, FastEthernet0/0
R       10.1.9.0 [120/2] via 10.1.5.1, 00:00:15, Serial0/0/1
R       10.1.8.0 [120/2] via 10.1.5.1, 00:00:15, Serial0/0/1
R       10.1.12.0 [120/1] via 10.1.11.2, 00:00:00, FastEthernet0/1
R       10.1.3.0 [120/15] via 10.1.5.1, 00:00:15, Serial0/0/1
R       10.1.2.0 [120/1] via 10.1.5.1, 00:00:15, Serial0/0/1
R       10.1.1.0 [120/1] via 10.1.5.1, 00:00:15, Serial0/0/1
R       10.1.7.0 [120/2] via 10.1.5.1, 00:00:15, Serial0/0/1
R       10.1.6.0 [120/2] via 10.1.5.1, 00:00:15, Serial0/0/1
C       10.1.5.0 is directly connected, Serial0/0/1
R       10.1.4.0 [120/1] via 10.1.5.1, 00:00:15, Serial0/0/1
R3#
```

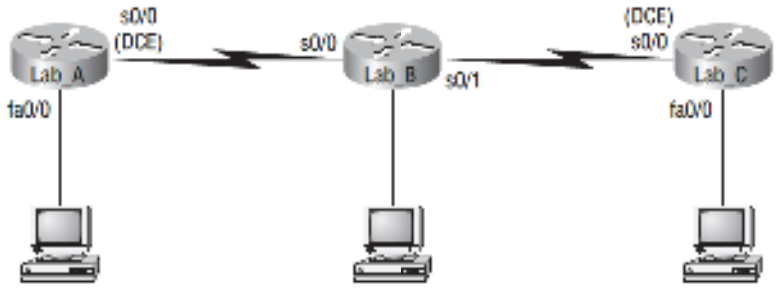
So this [120/15] is really bad because the next router that receives the table from router R3 will just discard the route to network 10.1.3.0 since the hop count would then be 16, which is invalid.

If a router receives a routing update that contains a higher-cost path to a network that's already in its routing table, the update will be ignored.

Configuring RIP Routing Example 2

Before we move onto learning more about RIP configurations, let's take a look at Figure 6.16. In this example, we first will find and implement our subnets and then add the RIP configuration to the router.

FIGURE 6.16 RIP routing example 2



For this configuration, we are going to consider that the Lab_B and Lab_C routers are already configured and we just need to configure the Lab_A router. We will use the network ID of 192.168.164.0/28. The s0/0 interface of Lab_A will use the last available IP address in the eighth subnet and the fa0/0 will use the last available IP address in the second subnet. Do not consider the zero subnet valid.

Before we start, you do know that /28 is a 255.255.255.240 mask, right? And that we have a block size of 16 in the fourth octet? It is very important that you know this, and if you need another review of Chapters 2 and 3, that's okay! Reviewing subnetting will never hurt you.

Since we have a block size of 16, our subnets are 16 (remember we are not starting at zero for this example), 32, 48, 64, 80, 96, 112, 128, 144, etc. The eighth subnet (which we will use for the s0/0 interface) is subnet 128. The valid host range for the 128 subnet is 129 through 142, and 143 is the broadcast address of the 128 subnet. The second subnet (which we will use for the fa0/0 interface) is the 32 subnet. The valid hosts are 33 through 46, and 47 is the broadcast address of the 32 subnet.

So, here is what our configuration on the Lab_A router will look like:

```
Lab_A(config)#interface s0/0
Lab_A(config-if)#ip address 192.168.164.142 255.255.255.240
Lab_A(config-if)#no shutdown
```

```
Lab_A(config-if)#interface fa0/0
Lab_A(config-if)#ip address 192.168.164.46 255.255.255.240
Lab_A(config-if)#no shutdown
Lab_A(config-if)#router rip
Lab_A(config-router)#network 192.168.164.0
Lab_A(config-router)#^Z
Lab_A#
```

Finding the subnets and configuring the last valid host should be pretty straightforward. If not, head back to Chapter 3. However, what I really want you to notice is that although we added two subnets to the Lab_A router, we only had one network statement under RIP. Sometimes it is hard to remember that you configure only the classful network statement, which means you turn all host bits off.

This was the real purpose of this second RIP configuration example—to remind you of classful network addressing. And it never hurts to practice subnetting, right?

Holding Down RIP Propagations

You probably don't want your RIP network advertised everywhere on your LAN and WAN. There's not a whole lot to be gained by advertising your RIP network to the Internet, now, is there?

There's a few different ways to stop unwanted RIP updates from propagating across your LANs and WANs, and the easiest one is through the `passive-interface` command that I showed you during the R3 configuration. This command prevents RIP update broadcasts from being sent out a specified interface, yet that same interface can still receive RIP updates.

Here's an example of how to configure a `passive-interface` on a router using the CLI:

```
Lab_A#config t
Lab_A(config)#router rip
Lab_A(config-router)#network 192.168.10.0
Lab_A(config-router)#passive-interface serial 0/0
```

This command will stop RIP updates from being propagated out serial interface 0/0, but serial interface 0/0 can still receive RIP updates. This is easily done within the SDM configuration as well, as I demonstrated with the R3 router.

RIP Version 2 (RIPv2)

Let's spend a couple of minutes discussing RIPv2 before we move into the distance-vector, Cisco-proprietary routing protocol IGRP.

RIP version 2 is mostly the same as RIP version 1. Both RIPv1 and RIPv2 are distance-vector protocols, which means that each router running RIP sends its complete routing tables out all active interfaces at periodic time intervals. Also, the timers and loop-avoidance schemes are the same in both RIP versions (i.e., holddown timers and split horizon rule). Both RIPv1 and RIPv2 are configured as classful addressing (but RIPv2 is considered classless because subnet information is sent with each route update), and both have the same administrative distance (120).

Real World Scenario

Should We Really Use RIP in an Internetwork?

You have been hired as a consultant to install a couple of Cisco routers into a growing network. They have a couple of old Unix routers that they want to keep in the network. These routers do not support any routing protocol except RIP. I guess this means you just have to run RIP on the entire network.

Well, yes and no. You can run RIP on a router connecting that old network, but you certainly don't need to run RIP throughout the whole internetwork!

You can do what is called *redistribution*, which is basically translating from one type of routing protocol to another. This means that you can support those old routers using RIP but use Enhanced IGRP, for example, on the rest of your network.

This will stop RIP routes from being sent all over the internetwork and eating up all that precious bandwidth.

But there are some important differences that make RIPv2 more scalable than RIPv1. And I've got to add a word of advice here before we move on; I'm definitely not advocating using RIP of either version in your network. But since RIP is an open standard, you can use RIP with any brand of router. You can also use OSPF (discussed in Chapter 7) since OSPF is an open standard as well. RIP just requires too much bandwidth, making it pretty intensive to use in your network. Why go there when you have other, more elegant options?

Table 6.3 discusses the differences between RIPv1 and RIPv2.

TABLE 6.3 RIPv1 vs. RIPv2

RIPv1	RIPv2
Distance vector	Distance vector
Maximum hop count of 15	Maximum hop count of 15
Classful	Classless
Broadcast based	Uses multicast 224.0.0.9
No support for VLSM	Supports VLSM networks
No authentication	Allows for MD5 authentication
No support for discontiguous networks	Supports discontiguous networks

RIPv2, unlike RIPv1, is a classless routing protocol (even though it is configured as classful, like RIPv1), which means that it sends subnet mask information along with the route updates. By sending the subnet mask information with the updates, RIPv2 can support Variable Length Subnet Masks (VLSMs) as well as the summarization of network boundaries. In addition, RIPv2 can support discontiguous networking, which I'll go over more in Chapter 7.

Configuring RIPv2 is pretty straightforward. Here's an example:

```
Lab_C(config)#router rip
Lab_C(config-router)#network 192.168.40.0
Lab_C(config-router)#network 192.168.50.0
Lab_C(config-router)#version 2
```

That's it; just add the command **version 2** under the **(config-router)#** prompt and you are now running RIPv2.

RIPv2 is classless and works in VLSM and discontiguous networks.

Interior Gateway Routing Protocol (IGRP)

Interior Gateway Routing Protocol (IGRP) is a Cisco-proprietary distance-vector routing protocol. This means that to use IGRP in your network, all your routers must be Cisco routers. Cisco created this routing protocol to overcome the problems associated with RIP.

IGRP has a maximum hop count of 255 with the default being 100 (same as EIGRP). This is helpful in larger networks and solves the problem of 15 hops being the maximum possible in a RIP network.

IGRP also uses a different metric than RIP. IGRP uses bandwidth and delay of the line by default as a metric for determining the best route to an internetwork. This is called a *composite metric*. Reliability, load, and maximum transmission unit (MTU) can also be used, although they are not used by default.

The main difference between RIP and IGRP configuration is that when you configure IGRP, you supply the autonomous system number. All routers must use the same number in order to share routing table information.

Table 6.4 shows a list of IGRP characteristics that you won't find in RIP.

TABLE 6.4 IGRP vs. RIP

IGRP	RIP
Can be used in large internetworks	Works best in smaller networks
Uses an autonomous system number for activation	Does not use autonomous system numbers
Gives a full route table update every 90 seconds	Gives a full route table update every 30 seconds
Has an administrative distance of 100	Has an administrative distance of 120
Uses bandwidth and delay of the line as metric (lowest composite metric), with a maximum hop count of 255	Uses only hop count to determine the best path to a remote network, with 15 hops being the maximum

Why is this the end of the IGRP section? Because watch what happens when I try to configure IGRP on my router:

```
R3#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#router igrp 10
                   ^
% Invalid input detected at '^' marker.
R3(config)#
```

There's your reason—Cisco no longer supports IGRP. Why should it? All you have to do is put an *E* in front of *IGRP* and you're running a much, much better routing protocol. We'll get to EIGRP in the next chapter, but first, let's go through some verification commands for RIP.

Verifying Your Configurations

It's important to verify your configurations once you've completed them, or at least once you *think* you've completed them. The following list includes the commands you can use to verify the routed and routing protocols configured on your Cisco routers:

- `show ip route`
- `show ip protocols`
- `debug ip rip`

The first command was covered in the previous section—I'll go over the others in the sections that follow.

The *show ip protocols* Command

The `show ip protocols` command shows you the routing protocols that are configured on your router. Looking at the following output, you can see that RIP is running on the router and the timers that RIP uses:

```
R3#sh 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
  Sending updates every 30 seconds, next due in 24 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Redistributing: rip
  Default version control: send version 1, receive version 1
    Interface             Send  Recv  Triggered RIP  Key-chain
    FastEthernet0/1       1     1
    Serial0/0/1           1     1
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    10.0.0.0
  Passive Interface(s):
    FastEthernet0/0
    Serial0/0/0
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.1.11.2            120      00:00:10
    10.1.5.1             120      00:00:22
  Distance: (default is 120)
```

Notice in this output that RIP is sending updates every 30 seconds, which is the default. The timers used in distance vector are also shown.

Notice further down that RIP is routing for directly connected interfaces f0/1 and s0/0/1. The version is listed to the right of the interfaces—RIPv1.

F0/0 and s0/0/0 are listed as passive interfaces (they will not send RIP information out). The neighbors it found are 10.1.11.2 and 10.1.5.1. The last entry is the default AD for RIP (120).

Troubleshooting with the *show ip protocols* Command

Let's use a sample router and use the `show ip protocols` command to see what we can determine about routing by looking at this output from a router on another network:

```
Router#sh ip protocols
Routing Protocol is "rip"
```

```
Sending updates every 30 seconds, next due in 6 seconds
Invalid after 180 seconds, hold down 180, flushed after240
Outgoing update filter list for all interfaces is
Incoming update filter list for all interfaces is
Redistributing: rip
Default version control: send version 1, receive any version
  Interface         Send  Recv   Key-chain
  Serial0/0          1     1 2
  Serial0/1          1     1 2
Routing for Networks:
  10.0.0.0
Routing Information Sources:
  Gateway         Distance      Last Update
  10.168.11.14         120      00:00:21
Distance: (default is 120)
```

Let's also look at the `show ip interface brief` command from the same router and see what we find out:

```
Router#sh ip interface brief
Interface        IP-Address      OK?   Method Status
FastEthernet0/0  192.168.18.1    YES   manual  up
Serial0/0        10.168.11.17    YES   manual  up
FastEthernet0/1  unassigned      YES   NRAM    Administatively down
Serial0/1        192.168.11.21   YES   manual  up
```

Under the `show ip protocols` output, you can see that we're using RIP routing for network 10.0.0.0, which means our configuration would look like this:

```
Router(config)#router rip
Router(config-router)#network 10.0.0.0
```

Also, only serial 0/0 and serial 0/1 are participating in the RIP network. And last, our neighbor router is 10.168.11.14.

From the output of the `show ip interface brief` command, you can see that only serial 0/0 is in the 10.0.0.0 network. This means that the router will only send and receive routing updates with the 10.0.0.0 network and not advertise the 192.168.0.0 networks out any interface.

The *debug ip rip* Command

The `debug ip rip` command sends routing updates as they are sent and received on the router to the console session. If you are telnetted into the router, you'll need to use the `terminal monitor` command to be able to receive the output from the `debug` commands.

We can see in this output that RIP is both sending and receiving (the metric is the hop count):

```
R3#debug ip rip
RIP protocol debugging is on
R3#terminal monitor
*Mar 17 19:08:34.371: RIP: sending v1 update to 255.255.255.255 via
   Serial0/0/1 (10.1.5.2)
*Mar 17 19:08:34.371: RIP: build update entries
*Mar 17 19:08:34.371:   subnet 10.1.10.0 metric 1
*Mar 17 19:08:34.371:   subnet 10.1.11.0 metric 1
*Mar 17 19:08:34.371:   subnet 10.1.12.0 metric 2
*Mar 17 19:08:40.107: RIP: received v1 update from 10.1.5.1 on
   Serial0/0/1
*Mar 17 19:08:40.107:      10.1.1.0 in 1 hops
*Mar 17 19:08:40.107:      10.1.2.0 in 1 hops
*Mar 17 19:08:40.107:      10.1.3.0 in 1 hops
*Mar 17 19:08:40.107:      10.1.4.0 in 1 hops
*Mar 17 19:08:40.107:      10.1.6.0 in 2 hops
*Mar 17 19:08:40.107:      10.1.7.0 in 2 hops
*Mar 17 19:08:40.107:      10.1.8.0 in 2 hops
*Mar 17 19:08:40.107:      10.1.9.0 in 2 hops
*Mar 17 19:08:47.535: RIP: sending v1 update to 255.255.255.255 via
   FastEthernet0/1 (10.1.11.1)
*Mar 17 19:08:47.535: RIP: build update entries
*Mar 17 19:08:47.535:   subnet 10.1.1.0 metric 2
*Mar 17 19:08:47.535:   subnet 10.1.2.0 metric 2
*Mar 17 19:08:47.535:   subnet 10.1.3.0 metric 2
*Mar 17 19:08:47.535:   subnet 10.1.4.0 metric 2
*Mar 17 19:08:47.535:   subnet 10.1.5.0 metric 1
*Mar 17 19:08:47.535:   subnet 10.1.6.0 metric 3
*Mar 17 19:08:47.535:   subnet 10.1.7.0 metric 3
*Mar 17 19:08:47.535:   subnet 10.1.8.0 metric 3
*Mar 17 19:08:47.535:   subnet 10.1.9.0 metric 3
*Mar 17 19:08:47.535:   subnet 10.1.10.0 metric 1
*Mar 17 19:08:49.331: RIP: received v1 update from 10.1.11.2 on
   FastEthernet0/1
*Mar 17 19:08:49.331:      10.1.12.0 in 1 hops
R3#undeug all
*Mar 17 19:08:47.535:   subnet 10.1.10.0 metric 1
*Mar 17 19:08:49.331: RIP: received v1 update from 10.1.11.2 on
   FastEthernet0/1
```

Let's talk about the parts I highlighted. First, RIP is sending v1 packet to 255.255.255.255—an "all-hands" broadcast—out interface serial0/0/1 via 10.1.5.2. This is where RIPv2 will come in handy. Why? Because RIPv2 doesn't send broadcasts; it used the multicast 224.0.0.9. So even though the RIP packets could be transmitted onto a network with no routers, all hosts would just ignore them, making RIPv2 a bit of an improvement over RIPv1. On our R3, we are using the `passive-interface` so we are not sending broadcasts out to a LAN with no routers connected.

Okay—now check out the fact that it's sending advertisements for all networks except 10.1.11.0 and 10.1.12.0 out FastEthernet0/1, yet the last advertisement out serial0/0/1 is only advertising networks 10.1.10.0, 10.1.11.0, and 10.1.12.0. Why? If you answered split horizon rules, you nailed it! Our R3 router will not advertise all those networks received from the Corp router back to the Corp router.

If the metric of a route shows 16, this is a route poison, and the route being advertised is unreachable.

Troubleshooting with the *debug ip rip* Command

Now let's use the `debug ip rip` command to both discover a problem and figure out how RIP was configured on a router from a different sample network:

```
07:12:58: RIP: sending v1 update to 255.255.255.255 via
  FastEthernet0/0 (172.16.1.1)
07:12:58:  network 10.0.0.0, metric 1
07:12:58:  network 192.168.1.0, metric 2
07:12:58: RIP: sending v1 update to 255.255.255.255 via
  Serial0/0 (10.0.8.1)
07:12:58:  network 172.16.0.0, metric 1
07:12:58: RIP: Received v1 update from 10.0.15.2 n Serial0/0
07:12:58: 192.168.1.0 in one hop
07:12:58: 192.168.168.0 in 16 hops (inaccessible)
```

You can see from the updates that we're sending out information about networks 10.0.0.0, 192.168.1.0, and 172.16.0.0. But both the 10.0.0.0 network and the 172.16.0.0 network are being advertised with a hop count (metric) of 1, meaning that these networks are directly connected. The 192.168.1.0 is being advertised as a metric of 2, which means that it is not directly connected.

For this to be happening, our configuration would have to look like this:

```
Router(config)#router rip
Router(config-router)#network 10.0.0.0
Router(config-router)#network 172.16.0.0
```

And there's something else you can find out by looking at this: There are at least two routers participating in the RIP network because we're sending out two interfaces but only receiving RIP updates on one interface. Also, notice that the network 192.168.168.0 is being advertised as 16 hops away. RIP has a maximum hop count of 15, so 16 is considered unreachable, making this network inaccessible. So what will happen if you try to ping to a host on network 192.168.168.0? You just will not be successful, that's what! But if you try any pings to network 10.0.0.0, you should be successful.

I have one more output I want to show you—see if you can find the problem. Both a **debug ip rip** and a **show ip route** output are shown from our sample router:

```
07:12:56: RIP: received v1 update from 172.16.100.2 on Serial0/0
07:12:56:      172.16.10.0 in 1 hops
07:12:56:      172.16.20.0 in 1 hops
07:12:56:      172.16.30.0 in 1 hops

Router#sh ip route
[output cut]
Gateway of last resort is not set

   172.16.0.0/24 is subnetted, 8 subnets
C  172.16.150.0 is directly connected, FastEthernet0/0
C  172.16.220.0 is directly connected, Loopback2
R  172.16.210.0 is directly connected, Loopback1
R  172.16.200.0 is directly connected, Loopback0
R  172.16.30.0 [120/2] via 172.16.100.2, 00:00:04, Serial0/0
S  172.16.20.0 [120/2] via 172.16.150.15
R  172.16.10.0 [120/2] via 172.16.100.2, 00:00:04, Serial0/0
R  172.16.100.0 [120/2] is directly connected, Serial0/0
```

Looking at the two outputs, can you tell why users can't access 172.16.20.0?

The debug output shows that network 172.16.20.0 is one hop away and being received on serial0/0 from 172.16.100.2. By checking out the **show ip route** output, you can see that packets with a destination of 172.16.20.0 are being sent to 172.16.150.15 because of a static route. The output also shows that 172.16.150.0 is directly connected to FastEthernet 0/0 and network 172.16.20.0 is out serial 0/0.

Enabling RIPv2 on Our Internetwork

Before we move on to Chapter 7 and configure EIGRP and OSPF, I want to enable RIPv2 on our routers. It'll only take a second. Here are my configurations:

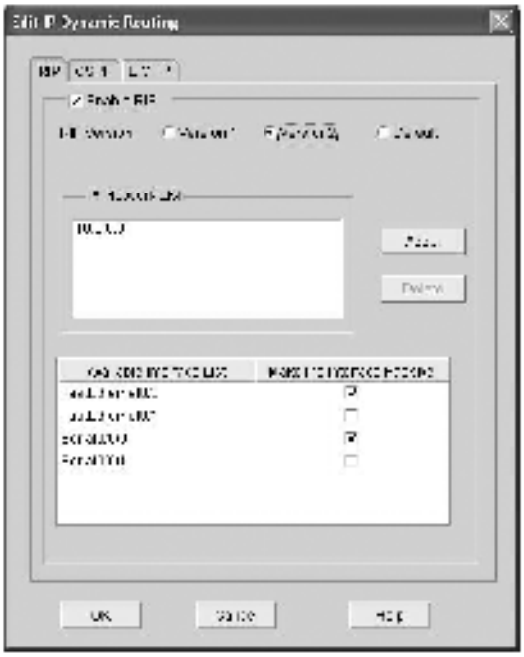
```
Corp#config t
Corp(config)#router rip
```

```
Corp(config-router)#version 2
Corp(config-router)#^Z

R1#config t
R1(config)#router rip
R1(config-router)#version 2
R1(config-router)#^Z

R2#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#router rip
R2(config-router)#version 2
R2(config-router)#^Z
```

For the R3 router, I just clicked the Version2 button and clicked OK. Done.




```
871W#config t
871W#(config)#router rip
871W#(config-router)#version 2
871W#(config-router)#^Z
```

This was probably the easiest configuration we have done in the book so far. Let's see if we can find a difference in our routing tables. Here's the R3 router's routing table now:

```
     10.0.0.0/24 is subnetted, 12 subnets
C       10.1.11.0 is directly connected, FastEthernet0/1
C       10.1.10.0 is directly connected, FastEthernet0/0
R       10.1.9.0 [120/2] via 10.1.5.1, 00:00:23, Serial0/0/1
R       10.1.8.0 [120/2] via 10.1.5.1, 00:00:23, Serial0/0/1
R       10.1.12.0 [120/1] via 10.1.11.2, 00:00:18, FastEthernet0/1
R       10.1.3.0 [120/1] via 10.1.5.1, 00:00:23, Serial0/0/1
R       10.1.2.0 [120/1] via 10.1.5.1, 00:00:23, Serial0/0/1
R       10.1.1.0 [120/1] via 10.1.5.1, 00:00:23, Serial0/0/1
R       10.1.7.0 [120/2] via 10.1.5.1, 00:00:23, Serial0/0/1
R       10.1.6.0 [120/2] via 10.1.5.1, 00:00:23, Serial0/0/1
C       10.1.5.0 is directly connected, Serial0/0/1
R       10.1.4.0 [120/1] via 10.1.5.1, 00:00:23, Serial0/0/1
R3#
```

Well—looks the same to me. I'm going to turn on debugging and see if that shows us anything new:

```
^Mar 17 19:34:00.123: RIP: sending v2 update to 224.0.0.9 via
   Serial0/0/1 (10.1.5.2)
^Mar 17 19:34:00.123: RIP: build update entries
*Mar 17 19:34:00.123:   10.1.10.0/24 via 0.0.0.0, metric 1, tag 0
^Mar 17 19:34:00.123:   10.1.11.0/24 via 0.0.0.0, metric 1, tag 0
*Mar 17 19:34:00.123:   10.1.12.0/24 via 0.0.0.0, metric 2, tag 0col
^Mar 17 19:34:03.795: RIP: received v2 update from 10.1.5.1 on
   Serial0/0/1
[output cut]
```

Bingo! Look at that! The networks are still being advertised every 30 seconds, but they're now sending the advertisements as v2 and as a multicast address of 224.0.0.9. Let's take a look at the `show ip protocols` output:

```
R3#sh 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
  Sending updates every 30 seconds, next due in 27 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Redistributing: rip
  Default version control: send version 2, receive version 2
```

```
  Interface              Send  Recv  Triggered RIP  Key-chain
  FastEthernet0/1        2     2
  Serial0/0/1            2     2
Automatic network summarization is not in effect
Maximum path: 4
Routing for Networks:
  10.0.0.0
Passive Interface(s):
  FastEthernet0/0
  Serial0/0/0
Routing Information Sources:
  Gateway         Distance      Last Update
  10.1.11.2             120      00:00:00
  10.1.5.1              120      00:00:02
Distance: (default is 120)
```

We are now sending and receiving RIPv2. Nice when things work out well, huh? You're ready now to move on to the next chapter!

Summary

This chapter covered IP routing in detail. It's extremely important that you really understand the basics we covered in this chapter because everything that's done on a Cisco router typically will have some type of IP routing configured and running.

You learned in this chapter how IP routing uses frames to transport packets between routers and to the destination host. From there, we configured static routing on our routers and discussed the administrative distance used by IP to determine the best route to a destination network. If you have a stub network, you can configure default routing, which sets the gateway of last resort on a router.

We then discussed dynamic routing in detail, specifically RIP and how it works on an internetwork (not well). We finished by verifying RIP and then adding RIPv2 to our little internetwork.

In the next chapter, we'll continue on with dynamic routing by discussing EIGRP and OSPF.

Exam Essentials

Understand the basic IP routing process. You need to remember that the frame changes at each hop but that the packet is never changed or manipulated in any way until it reaches the destination device.

Understand that MAC addresses are always local. A MAC (hardware) address will only be used on a local LAN. It will never pass a router's interface.

Understand that a frame carries a packet to only two places. A frame uses MAC (hardware) addresses to send a packet on a LAN. The frame will take the packet to either a host on the LAN or a router's interface if the packet is destined for a remote network

Understand how to configure RIP routing. To configure RIP routing, first you must be in global configuration mode and then you type the command **`router rip`**. Then you add all directly connected networks, making sure to use the classful address.

Remember how to verify RIP routing. The **`show ip route`** command will provide you with the contents of the routing table. An R on the left side of the table indicates a RIP-found route. The **`debug ip rip`** command will show you RIP updates being sent and received on your router. If you see a route with a metric of 16, that route is considered down.

Remember the differences between RIPv1 and RIPv2. RIPv1 sends broadcasts every 30 seconds and has an AD of 120. RIPv2 sends multicasts (224.0.0.9) every 30 seconds and also has an AD of 120. RIPv2 sends subnet mask information with the route updates, which allows it to support classless networks and discontiguous networks. RIPv2 also supports authentication between routers and RIPv1 does not.

Written Lab 6

Write the answers to the following questions:

1. Create a static route to network 172.16.10.0/24 with a next-hop gateway of 172.16.20.1 and an administrative distance of 150.
2. From the SDM you have just enabled RIP and the passive-interface box for your serial interface is unchecked. What does this mean?
3. What command will you type to create a default route to 172.16.40.1?
4. If you are using default routing, what command must also be used?
5. You would use a default route on which type of network?
6. To see the routing table on your router, what command will you use?
7. When creating a static or default route, you don't have to use the next-hop IP address; you can use the _____.
8. True/False: To reach a destination host, you must know the MAC address of the remote host.
9. True/False: To reach a destination host, you must know the IP address of the remote host.
10. If you have a DCE serial interface, what command must you enter for that interface to work?

11. Write the commands used to turn RIP routing on in a router and advertise network 10.0.0.0.
12. Write the commands to stop a router from propagating RIP information out serial 1.
13. What works with triggered updates to help stop routing loops in distance-vector networks?
14. What stops routing loops in distance-vector networks by sending out a maximum hop count as soon as a link fails?
15. What stops routing loops in distance-vector networks by not resending information learned on an interface out that same interface?
16. What command is used to send RIP routing updates as they are sent and received on the router to the console session?

(The answers to Written Lab 6 can be found following the answers to the review questions for this chapter.)

Hands-on Labs

In the following hands-on labs, you will configure a network with three routers.

The hands-on labs in this section is included for use with real Cisco routers. If you are using software from RouterSim or Sybex, please use the hands-on labs found in those programs.

This chapter includes:
Lab 6.1: Creating Static Routes
Lab 6.2: Configuring RIP Routing
Figure 6.17 will be used to configure all routers.

FIGURE 6.17 Hands-on lab internetwork

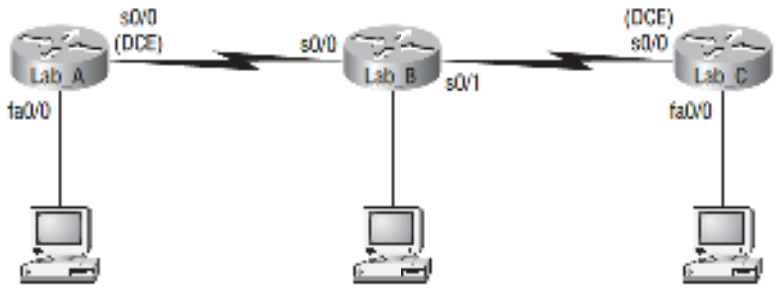


Table 6.5 shows our IP addresses for each router (each interface uses a /24 mask).

TABLE 6.5 Our IP Addresses

Router	Interface	IP Address
Lab_A	F0/0	172.16.10.1
Lab_A	S0/0	172.16.20.1
Lab_B	S0/0	172.16.20.2
Lab_B	S0/1	172.16.30.1
Lab_C	S0/0	172.16.30.2
Lab_C	Fa0/0	172.16.40.1

These labs were written without using the LAN interface on the Lab_B router. You can choose to add that LAN into the labs if necessary.

Hands-on Lab 6.1: Creating Static Routes

In this lab, you will create a static route in all three routers so that the routers see all networks. Verify with the Ping program when complete.

1. The Lab_A router is connected to two networks, 172.16.10.0 and 172.16.20.0. You need to add routes to networks 172.16.30.0 and 172.16.40.0.

```
Lab_A#config t
Lab_A(config)#ip route 172.16.30.0 255.255.255.0
  172.16.20.2
Lab_A(config)#ip route 172.16.40.0 255.255.255.0
  172.16.20.2
```

2. Save the current configuration for the Lab_A router by going to the privileged mode, typing **copy run start**, and pressing Enter.
3. On the Lab_B router, you have direct connections to networks 172.16.20.0 and 172.16.30.0. You need to add routes to networks 172.16.10.0 and 172.16.40.0.

```
Lab_B#config t
Lab_B(config)#ip route 172.16.10.0 255.255.255.0
  172.16.20.1
Lab_B(config)#ip route 172.16.40.0 255.255.255.0
  172.16.30.2
```

4. Save the current configuration for router Lab_B by going to the enabled mode, typing **copy run start**, and pressing Enter.
5. On Router Lab_C, create a static route to see networks 172.16.10.0 and 172.16.20.0, which are not directly connected. Create static routes so that Router Lab_C can see all networks, as shown here:

```
Lab_C#config t
Lab_C(config)#ip route 172.16.10.0 255.255.255.0
  172.16.30.1
Lab_C(config)#ip route 172.16.20.0 255.255.255.0
  172.16.30.1
```

6. Save the current configuration for Router 2501B by going to the enable mode, typing **copy run start**, and pressing Enter.
7. Check your routing tables to make sure all four networks show up.
8. Now ping from each router to your hosts and from each router to each router. If it is set up correctly, it will work.

Hands-on Lab 6.2: Configuring RIP Routing

In this lab, we will use the dynamic routing protocol RIP instead of static routing.

1. Remove any static routes or default routes configured on your routers by using the **no ip route** command. For example, here is how you would remove the static routes on the Lab_A router:

```
Lab_A#config t
Lab_A(config)#no ip route 172.16.30.0 255.255.255.0
  172.16.20.2
Lab_A(config)#no ip route 172.16.40.0 255.255.255.0
  172.16.20.2
```

   Do the same thing for routers Lab_B and Lab_C. Verify that only your directly connected networks are in the routing tables.
2. After your static and default routes are clear, go into configuration mode on Router Lab_A by typing **config t**.
3. Tell your router to use RIP routing by typing **router rip** and pressing Enter, as shown here:

```
config t
router rip
```

4. Add the network number you want to advertise by typing **network 172.16.0.0** and pressing Enter.
5. Press Ctrl+Z to get out of configuration mode.

6. Go to routers Lab_B and Lab_C and type the same commands, as shown here:

```
Config t
Router rip
network 172.16.0.0
```

7. Verify that RIP is running at each router by typing the following commands at each router:

```
show ip protocols
show ip route
show running-config or show run
```

8. Save your configurations by typing **copy run start** or **copy running-config startup-config** and pressing Enter at each router.
9. Verify the network by pinging all remote networks and hosts.

Review Questions

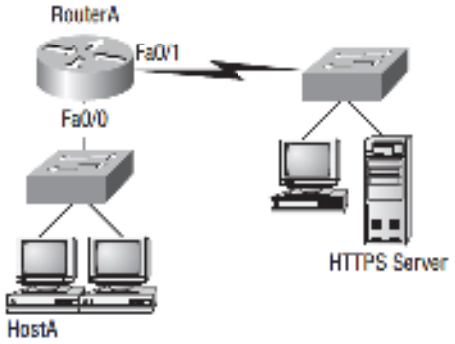
The following questions are designed to test your understanding of this chapter's material. For more information on how to get additional questions, please see this book's Introduction.

1. Network 206.143.5.0 was assigned to the Acme Company to connect to its ISP. The administrator of Acme would like to configure one router with the commands to access the Internet. Which commands could be configured on the Gateway router to allow Internet access to the entire network? (Choose two.)
   - A. `Gateway(config)#ip route 0.0.0.0 0.0.0.0 206.143.5.2`
   - B. `Gateway(config)#router rip`
     `Gateway(config-router)#network 206.143.5.0`
   - C. `Gateway(config)#router rip`
     `Gateway(config-router)#network 206.143.5.0 default`
   - D. `Gateway(config)#ip route 206.143.5.0 255.255.255.0 default`
   - E. `Gateway(config)#ip default-network 206.143.5.0`

2. What command is used to stop RIP routing updates from exiting out an interface but still allow the interface to receive RIP route updates?
   - A. `Router(config-if)#no routing`
   - B. `Router(config-if)#passive-interface`
   - C. `Router(config-router)#passive-interface s0`
   - D. `Router(config-router)#no routing updates`

3. Which of the following statements are true regarding the command `ip route 172.16.4.0 255.255.255.0 192.168.4.2`? (Choose two.)
   - A. The command is used to establish a static route.
   - B. The default administrative distance is used.
   - C. The command is used to configure the default route.
   - D. The subnet mask for the source address is 255.255.255.0.
   - E. The command is used to establish a stub network.

4. What destination addresses will be used by Host_A to send data to the HTTPS server as shown in the following network? (Choose two.)



   A. The IP address of the switch
   B. The MAC address of the remote switch
   C. The IP address of the HTTPS server
   D. The MAC address of the HTTPS server
   E. The IP address of RouterA's Fa0/0 interface
   F. The MAC address of RouterA's Fa0/0 interface

5. Which of the following is true regarding the following output? (Choose two.)

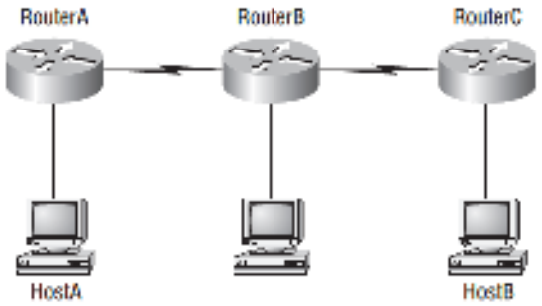
```
04:06:16: RIP: received v1 update from 192.168.40.2 on Serial0/1
04:06:16:      192.168.50.0 in 16 hops (inaccessible)
04:06:40: RIP: sending v1 update to 255.255.255.255 via
   FastEthernet0/0 (192.168.30.1)
04:06:40: RIP: build update entries
04:06:40:       network 192.168.20.0 metric 1
04:06:40:       network 192.168.40.0 metric 1
04:06:40:       network 192.168.50.0 metric 16
04:06:40: RIP: sending v1 update to 255.255.255.255 via Serial0/1
   (192.168.40.1)
```

   A. There are three interfaces on the router participating in this update.
   B. A ping to 192.168.50.1 will be successful.
   C. There are at least two routers exchanging information.
   D. A ping to 192.168.40.2 will be successful.

6. What is split horizon?

   A. Information about a route should not be sent back in the direction from which the original update came.
   B. It splits the traffic when you have a large bus (horizon) physical network.
   C. It holds the regular updates from broadcasting to a downed link.
   D. It prevents regular update messages from reinstating a route that has gone down.

7. Which of the following would be true if HostA is trying to communicate to HostB and interface F0/0 of RouterC goes down? (Choose two.)



   A. RouterC will use an ICMP to inform HostA that HostB cannot be reached.
   B. RouterC will use ICMP to inform RouterB that HostB cannot be reached.
   C. RouterC will use ICMP to inform HostA, RouterA, and RouterB that HostB cannot be reached.
   D. RouterC will send a destination unreachable message type.
   E. RouterC will send a router selection message type.
   F. RouterC will send a source quench message type.

8. Which statement is true regarding classless routing protocols? (Choose two.)
   A. The use of discontiguous networks is not allowed.
   B. The use of variable length subnet masks is permitted.
   C. RIPv1 is a classless routing protocol.
   D. IGRP supports classless routing within the same autonomous system.
   E. RIPv2 supports classless routing.

9. Which two of the following are true regarding the distance vector and link-state routing protocols?
   A. Link state sends its complete routing table out all active interfaces on periodic time intervals.
   B. Distance vector sends its complete routing table out all active interfaces on periodic time intervals.
   C. Link state sends updates containing the state of its own links to all routers in the internetwork.
   D. Distance vector sends updates containing the state of its own links to all routers in the internetwork.

10. Which command displays RIP routing updates?
   A. `show ip route`
   B. `debug ip rip`
   C. `show protocols`
   D. `debug ip route`

11. What does RIPv2 use to prevent routing loops? (Choose two.)
    A. CIDR
    B. Split horizon
    C. Authentication
    D. Classless masking
    E. Holddown timers

12. A network administrator views the output from the **show ip route** command. A network that is advertised by both RIP and IGRP appears in the routing table flagged as an IGRP route. Why is the RIP route to this network not used in the routing table?
    A. IGRP has a faster update timer.
    B. IGRP has a lower administrative distance.
    C. RIP has a higher metric value for that route.
    D. The IGRP route has fewer hops.
    E. The RIP path has a routing loop.

13. You type **debug ip rip** on your router console and see that 172.16.10.0 is being advertised to you with a metric of 16. What does this mean?
    A. The route is 16 hops away.
    B. The route has a delay of 16 microseconds.
    C. The route is inaccessible.
    D. The route is queued at 16 messages a second.

14. IGRP uses which of the following as default parameters for finding the best path to a remote network? (Choose two.)
    A. Hop count
    B. MTU
    C. Cumulative interface delay
    D. STP
    E. Path bandwidth value

15. The Corporate router receives an IP packet with a source IP address of 192.168.214.20 and a destination address of 192.168.22.3. Looking at the output from the Corporate router, what will the router do with this packet?

```
Corp#sh ip route
[output cut]
R    192.168.215.0 [120/2] via 192.168.20.2, 00:00:23, Serial0/0
R    192.168.115.0 [120/1] via 192.168.20.2, 00:00:23, Serial0/0
R    192.168.30.0 [120/1] via 192.168.20.2, 00:00:23, Serial0/0
C    192.168.20.0 is directly connected, Serial0/0
C    192.168.214.0 is directly connected, FastEthernet0/0
```

A. The packet will be discarded.

B. The packet will be routed out the S0/0 interface.

C. The router will broadcast looking for the destination.

D. The packet will be routed out the Fa0/0 interface.

16. If your routing table has a static, a RIP, and an IGRP route to the same network, which route will be used to route packets by default?

A. Any available route

B. RIP route

C. Static route

D. IGRP route

E. They will all load-balance.

17. You have the following routing table. Which of the following networks will not be placed in the neighbor routing table?

```
R    192.168.30.0/24 [120/1] via 192.168.40.1, 00:00:12, Serial0
C    192.168.40.0/24 is directly connected, Serial0
     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.30.0 is directly connected, Loopback0
R    192.168.20.0/24 [120/1] via 192.168.40.1, 00:00:12, Serial0
R    10.0.0.0/8 [120/15] via 192.168.40.1, 00:00:07, Serial0
C    192.168.50.0/24 is directly connected, Ethernet0
```

A. 172.16.30.0

B. 192.168.30.0

C. 10.0.0.0

D. All of them will be placed in the neighbor routing table.

18. Two connected routers are configured with RIP routing. What will be the result when a router receives a routing update that contains a higher-cost path to a network already in its routing table?

A. The updated information will be added to the existing routing table.

B. The update will be ignored and no further action will occur.

C. The updated information will replace the existing routing table entry.

D. The existing routing table entry will be deleted from the routing table and all routers will exchange routing updates to reach convergence.

19. What is route poisoning?

A. It sends back the protocol received from a router as a poison pill, which stops the regular updates.

B. It is information received from a router that can't be sent back to the originating router.

C. It prevents regular update messages from reinstating a route that has just come up.

D. It describes when a router sets the metric for a downed link to infinity.

20. Which of the following is true regarding RIPv2?

A. It has a lower administrative distance than RIPv1.

B. It converges faster than RIPv1.

C. It has the same timers as RIPv1.

D. It is harder to configure than RIPv1.

Answers to Review Questions

1. A, G. There are actually three different ways to configure the same default route, but only two are shown in the answer. First, you can set a default route with the 0.0.0.0 0.0.0.0 mask and then specify the next hop, as in answer A. Or you can use 0.0.0.0 0.0.0.0 and use the exit interface instead of the next hop. Finally, you can use answer G with the `ip default-network` command.

2. C. The `(config-router)#passive-interface` command stops updates from being sent out an interface, but route updates are still received.

3. A, B. Although answer D almost seems right, it is not; the mask is the mask used on the remote network, not the source network. Since there is no number at the end of the static route, it is using the default administrative distance of 1.

4. C, F. The switches are not used as either a default gateway or other destination. Switches have nothing to do with routing. It is very important to remember that the destination MAC address will always be the router's interface. The destination address of a frame, from HostA, will be the MAC address of the F0/0 interface of RouterA. The destination address of a packet will be the IP address of the network interface card (NIC) of the HTTPS server. The destination port number in the segment header will have a value of 443 (HTTPS).

5. C, D. The route to 192.168.50.0 is unreachable and only interfaces s0/1 and FastEthernet 0/0 are participating in the RIP update. Since a route update was received, at least two routers are participating in the RIP routing process. Since a route update for network 192.168.40.0 is being sent out f0/0 and a route was received from 192.168.40.2, we can assume a ping to that address will be successful.

6. A. A split horizon will not advertise a route back to the same router it learned the route from.

7. A, D. RouterC will use ICMP to inform HostA that HostB cannot be reached. It will perform this by sending a destination unreachable ICMP message type.

8. B, F. Classful routing means that all hosts in the internetwork use the same mask. Classless routing means that you can use Variable Length Subnet Masks (VLSMs) and can also support discontiguous networking.

9. B, C. The distance-vector routing protocol sends its complete routing table out all active interfaces at periodic time intervals. Link-state routing protocols send updates containing the state of its own links to all routers in the internetwork.

10. B. `Debug ip rip` is used to show the Internet Protocol (IP) Routing Information Protocol (RIP) updates being sent and received on the router.

11. B, F. RIPv2 uses the same timers and loop-avoidance schemes as RIPv1. Split horizon is used to stop an update from being sent out the same interface it was received on. Holddown timers allow time for a network to become stable in the case of a flapping link.

12. B. RIP has an administrative distance (AD) of 120, while IGRP has an administrative distance of 100, so the router will discard any route with a higher AD than 100.

13. C. You cannot have 16 hops on a RIP network by default. If you receive a route advertised with a metric of 16, this means it is inaccessible.

14. C, E. IGRP uses bandwidth and delay of the line, by default, to determine the best path to a remote network. Delay of the line can sometimes be called the cumulative interface delay.

15. A. Since the routing table shows no route to the 192.168.22.0 network, the router will discard the packet and send an ICMP destination unreachable message out interface FastEthernet 0/0, which is the source LAN where the packet originated from.

16. C. Static routes have an administrative distance of 1 by default. Unless you change this, a static route will always be used over any other found route. IGRP has an administrative distance of 100, and RIP has an administrative distance of 120, by default.

17. C. The network 10.0.0.0 cannot be placed in the next router's routing table because it already is at 15 hops. One more hop would make the route 16 hops, and that is not valid in RIP networking.

18. B. When a routing update is received by a router, the router first checks the administrative distance (AD) and always chooses the route with the lowest AD. However, if two routes are received and they both have the same AD, then the router will choose the one route with the lowest metrics, or in RIP's case, hop count.

19. D. Another way to avoid problems caused by inconsistent updates and to stop network loops is route poisoning. When a network goes down, the distance-vector routing protocol initiates route poisoning by advertising the network with a metric of 16, or unreachable (sometimes referred to as *infinite*).

20. C. RIPv2 is pretty much just like RIPv1. It has the same administrative distance and timers and is configured just like RIPv1.

Answers to Written Lab 6

1. `ip route 172.16.10.0 255.255.255.0 172.16.20.1 150`
2. If the box next to an interface is unchecked, this means that passive-interface will not be used and RIP will be sent and received on that interface.
3. `ip route 0.0.0.0 0.0.0.0 172.16.40.1`
4. `Router(config)#ip classless`
5. Stub network
6. `Router#show ip route`
7. Exit interface
8. False. The MAC address would be the router interface, not the remote host.
9. True
10. `Router(config-if)#clock rate speed`
11. `Router rip, network 10.0.0.0`
12. `Router rip, passive-interface s1`
13. Holddown timers
14. Route poisoning
15. Split horizon
16. `debug ip rip`