CHAPTER 2

Deploying Interior Routing Protocols

A highly effective network must move data through optimal paths and find new paths when outages occur. It should also be expandable and flexible enough to accommodate demands from a growing number of users and geographical locations—not to mention increasing demands for service stability.

Routing is the network service that moves data through your organization. It governs how easily you can grow your network and how stable your service will be. Routing is the circulatory system of your network. When it works, information flows transparently and efficiently—even in very complex networks with many users. However, when routing doesn't work, the flow stops, applications cease working, and a large user population is usually affected.

This chapter covers routing fundamentals and basic configuration for the most widely deployed routing protocols. The main topics of this chapter are

- A Brief Review of Internetworking
- Deploying RIP
- Deploying IGRP
- Deploying Enhanced IGRP
- Deploying OSPF

As you can see from the preceding list, this chapter covers configuration of the most common interior routing protocols: Routing Information Protocol (RIP), Interior Gateway Routing Protocol (IGRP), Enhanced IGRP (EIGRP), and Open Shortest Path First (OSPF). The objective of this chapter is to provide a *baseline* for the routing services covered in Chapter 3, "Managing Routing Protocols." If you are already familiar with configuring the routing protocols found in this chapter, you might want to skip ahead to Chapter 3.

A Brief Review of Internetworking

Internetworking is the practice of connecting multiple individual networks so they function as a single large network (called an *internetwork* or *internet*). The public Internet (spelled with a capital *I*) is an example of an internetwork: It is a collection of many diverse networks, yet it functions as one large network.

Ideally, from any point on the Internet, you can reach any other point on the Internet (assuming the destination you are trying to reach is open to you). Such extensive connectivity is a powerful feature of the Internet. Before internetworking, networks were islands of connectivity: They typically had a local reach, they were locally administered, and they served a specific purpose. Imagine connecting to a network in New York just to exchange e-mail with your friends who live there, or connecting to another network in California to do file transfers with an office in Los Angeles. Although it sounds ridiculous, that was networking before *inter*networking.

Internetworking is made possible by a service called *routing*. Routing is the process of finding a path through an internetwork to a destination (see Figure 2-1).

Figure 2-1 *Routing Is the Process of Finding a Path to a Destination*

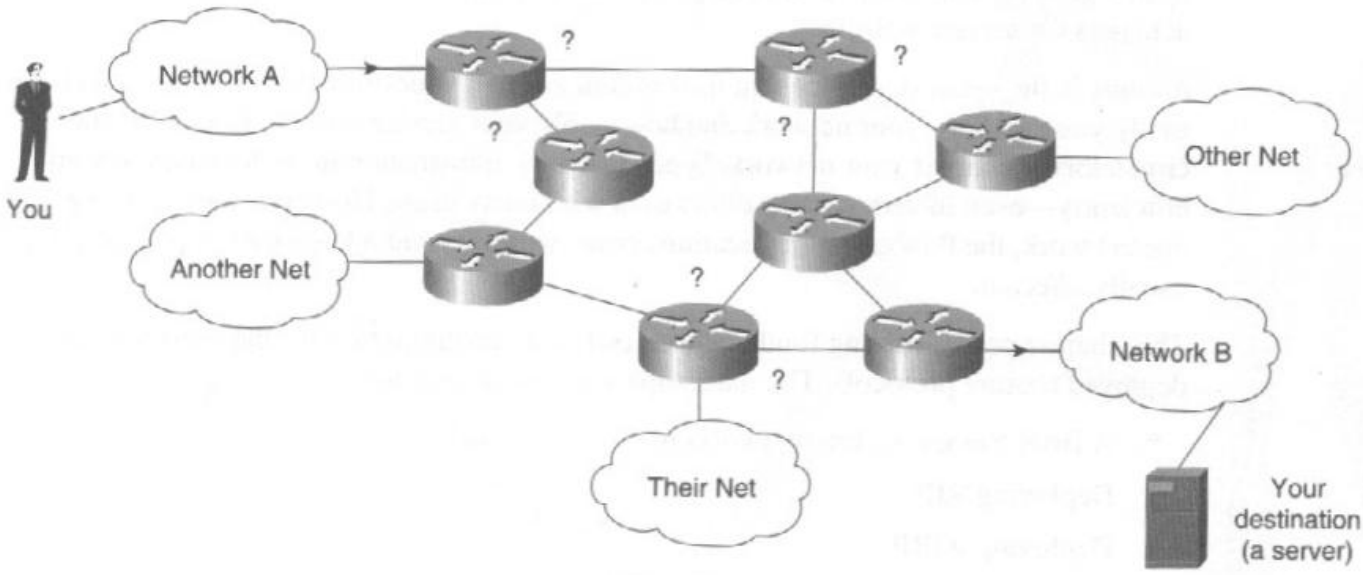


Figure 2-1 depicts multiple individual networks intertwined with routers to form one cohesive internetwork. Now, suppose you are in Network A and you need to send something to a server in Network B. How does your data get there and what path will it take? The **?** symbols in the figure represent some of the possible paths or forks in the road your data can follow. Ultimately, decisions must be made on how your data will weave its way to the destination. The routing service within the internetwork makes these decisions and determines the best path to the destination. With a routing service in place, you don't have to worry about the details of the internetwork itself; instead, you can focus on your application and the reason you need to talk to the server in Network B.

NOTE A classic application example is the World Wide Web and Web browsing. When you click a link in your Web browser, you don't want to hassle with how your request gets routed through the maze of the Internet; you simply want the Web page you requested to pop up on your screen.

Routing is like mapping a road trip from your hometown to a city far away in another part of the country. What is the shortest path? Is the shortest path necessarily the fastest path? If a highway is closed, what is a good alternate path? In the case of the road trip, your brain performs a routing task, calculating a path based on data gathered from maps, directions from people, and highway advisories. On an internetwork, algorithms programmed in routers (and similar networking devices) perform the routing task. Routers calculate paths based on their software configuration and network "directions and advisories" exchanged with other routers through a routing protocol.

A *routing protocol* is a language for routers. It is what routers use to exchange information about the topology and health of the internetwork. Based on routing information gathered from other routers, a router can calculate a suitable path to a destination. You can think of a routing protocol as a management or system protocol—overhead traffic (extremely vital overhead traffic) that routers use to keep each other informed. Routers can then ensure that data flows the right way through the internetwork.

NOTE There is a distinction between a routing protocol and a *routed* protocol. A routing protocol is a management protocol used by routers that carries information about the topology and status of the network. A routed protocol is used by hosts (client and servers) and carries data for user applications. TCP/IP, Novell IPX, AppleTalk, and DECnet are examples of routed protocols. The word *routed* means the protocol supports internetworking: It supports interconnecting multiple networks with routers. In contrast, *non-routed* protocols (such as NetBIOS and DEC LAT) were designed to support only one network, typically a LAN, and do not natively support internetworking.

There are two major classes of routing protocols: interior routing protocols and exterior routing protocols. Interior routing protocols (also called interior gateway protocols or IGPs) are used within an *autonomous system*: an internetwork typically under the control of one organization (a company, university, or ISP, for example). Exterior routing protocols (also called exterior gateway protocols or EGPs) are used to interconnect autonomous systems. That is, they are typically used to connect an organization to an ISP, an organization to another organization, or an ISP to another ISP.

NOTE As mentioned earlier, this book focuses on interior routing protocols (RIP, IGRP, OSPF, and EIGRP). At the time of this writing, Border Gateway Protocol (BGP) is the single most predominant exterior routing protocol in use. See the Bibliography for a BGP resource.

It is not the intention of this book to cover routing theory in depth nor to contrast in detail one routing protocol to another—plenty of good books have done this (see the Bibliography for some of them). Where appropriate, the following sections provide pointers to documents on Cisco's Web site that offer background on selected routing concepts.

If you are unfamiliar with basic routing concepts (metrics, next hop, convergence, distance vector versus link-state protocols, and so on), you can find a brief routing tutorial at http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/55171.htm. This tutorial, however, is not as thorough as the books listed in the Bibliography. If this URL has changed, search the Cisco Web site for the keywords *routing basics*.

This chapter also assumes that you know how to perform basic configuration tasks such as activating router interfaces and assigning IP addresses to them. See Appendix E, "A Crash Course in Cisco IOS," for a tutorial on how to perform these and other common tasks.

Deploying RIP

RIP is one of the oldest and simplest routing protocols. The following is a list of some key points for RIP:

- RIP is a simple, distance vector routing protocol. A RIP router periodically (roughly every 30 seconds) sends the contents of its routing table to neighboring routers. This periodic activity is common to distance vector protocols. Link-state protocols on the other hand, typically send small advertisements everywhere, and only when network changes occur. Link-state advertisements contain the status about a router's directly connected links (networks) rather than the router's entire routing table.
- RIP is a classful routing protocol and does not support VLSM. The exception is RIP version 2, which supports VLSM but is not deployed as widely as RIP, OSPF or EIGRP. See Chapter 1, "Managing Your IP Address Space," for more information on VLSM.
- As a consequence of not supporting VLSM, RIP requires subnets to be contiguous. That is, subnets of a major net must not be separated from each other by a different, intermediary major net.
- RIP uses a hop count for its metric. With RIP, the maximum distance any network can be is 15 hops—this is called the *network diameter*. A destination more than 15 hops (15 routers) away is considered unreachable.
- RIP converges slowly compared to routing protocols such as OSPF and EIGRP. This means users are more likely to experience temporary outages when network changes occur. Slower convergence is a typical trait of distance vector protocols.
- RIP internetworks are simple, but flat: They generally cannot be organized into hierarchies like internetworks built with OSPF and EIGRP. RIP is generally unaware of autonomous systems and address summarization.
- RIP is easy to implement, and compatibility of RIP among diverse devices is good.

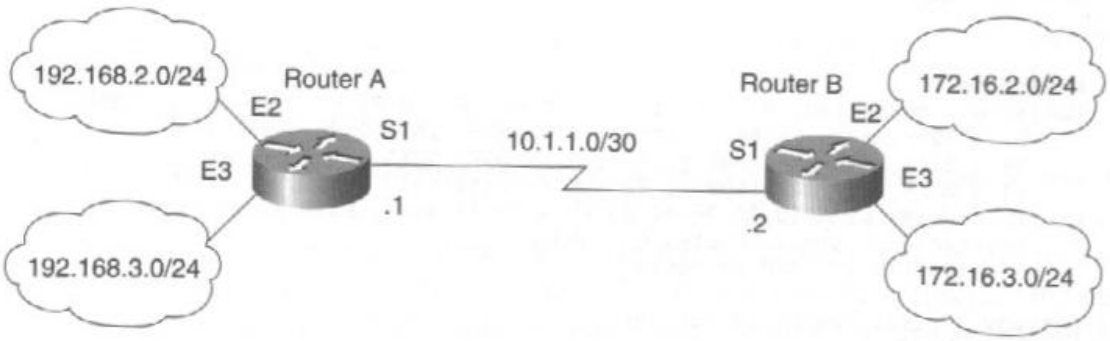
- Because of its limitations and simplicity, RIP does not scale well in large internetworks.
- The RIP standard is defined in RFC 1058.

The following sections describe RIP configuration, beginning with a brief coverage of directly connected networks and their significance to a router.

Directly Connected Networks

When you connect networks to a router and assign IP addresses to each of the router's interfaces, the router immediately knows some basic routing information: It knows how to route to its directly connected networks. Before jumping to RIP configuration, consider the network topology in Figure 2-2 initially without RIP (and then later with RIP).

Figure 2-2 *An Example for RIP Configuration*



In Figure 2-2, Router A is connected to major nets 192.168.2.0 and 192.168.3.0. Each of these has a /24 mask, and for class C networks this mask means the major net is not subnetted (there is no subnet field). Router A is also attached to major net 10.0.0.0, a class A network that is subnetted with a /30 mask and joins Router A to Router B (10.1.1.0/30 is a subnet of major net 10.0.0.0). Router B also connects to two subnets, 172.16.2.0 and 172.16.3.0. These are subnets from major net 172.16.0.0 (a class B subnetted with a /24 mask).

Without RIP, Router A has no idea that 172.16.0.0 exists, because it knows only its directly attached networks. The routing table displayed with **show ip route** validates this:

```
RTA#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, * - candidate default
    U - per-user static route, o - ODR
    T - traffic engineered route

Gateway of last resort is not set

   10.0.0.0/30 is subnetted, 1 subnet
```

continues

```
C    10.1.1.0 is directly connected, Serial1
C  192.168.2.0/24 is directly connected, Ethernet2
C  192.168.3.0/24 is directly connected, Ethernet3
```

In the preceding output, Router A has routes (paths) to three destinations: 10.1.1.0/30, 192.168.2.0/24, and 192.168.3.0/24. These routes are highlighted in boldface. From Figure 2-2, you can verify that these three routes are Router A's directly connected networks. This means Router A can route packets that flow among these three networks.

Missing from the preceding output is a route to the 172.16.0.0 subnets connected to Router B. Router A does not know how to reach 172.16.0.0 because RIP has not been configured yet. As it stands now, Router A is not able to forward any packets destined for major net 172.16.0.0.

The letter at the beginning of a line in **show ip route** is a code and tells you how the route was learned. In the case of Router A's three routes, code **C** indicates the routes are known because they are directly connected to the router. The key at the top of the **show ip route** output provides the meaning of other codes. You will see code **R** (for RIP) shortly.

The following shows the output of **show ip route** for Router B (also not yet configured with RIP):

```
RTB#sh ip ro
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, * - candidate default
    U - per-user static route, o - ODR
    T - traffic engineered route

Gateway of last resort is not set

   172.16.0.0/24 is subnetted, 2 subnets
C     172.16.2.0 is directly connected, Ethernet2
C     172.16.3.0 is directly connected, Ethernet3
   10.0.0.0/30 is subnetted, 1 subnet
C     10.1.1.0 is directly connected, Serial1
```

The preceding output confirms the expected result: Router B knows only its directly connected subnets (the routes printed in boldface).

Configuring RIP

Configuring RIP is easy and requires a knowledge of just two IOS commands: **router rip** and **network**. Starting with Router A, here's how you configure the example network from the previous section (Figure 2-2) with RIP:

```
RTA#conf t
Enter configuration commands, one per line. End with CNTL/Z.
RTA(config)#router rip
RTA(config-router)#network 192.168.2.0
RTA(config-router)#network 192.168.3.0
RTA(config-router)#network 10.0.0.0
```

The command **router rip** activates the RIP routing service on the router and changes the prompt to router configuration mode as indicated by the prompt **config-router**.

The command **network 192.168.2.0** tells the router to enable RIP processing on major net 192.168.2.0. This means the router will send and receive RIP messages on all interfaces that are part of this major net—in this case, just one interface, Ethernet2. The command also tells the router to advertise this network, 192.168.2.0, to other routers.

NOTE With the **network** command you must always specify a major net number, not a subnet number.

The commands **network 192.168.3.0** and **network 10.0.0.0** enable RIP processing for Router A's other two directly connected major nets.

Similarly, configure Router B with RIP and configure the major nets connected to Router B that should run RIP:

```
RTB#conf t
Enter configuration commands, one per line. End with CNTL/Z.
RTB(config)#router rip
RTB(config-router)#network 172.16.0.0
RTB(config-router)#network 10.0.0.0
```

The commands are equal in meaning to the commands used to configure Router A, except the major net numbers are consistent with Router B's directly connected networks. You should note that only one command, **network 172.16.0.0**, is entered to enable RIP for both subnets 172.16.2.0 and 172.16.3.0. This is because the network command specifies the major net (not subnets) that should run RIP—with this command, all interfaces that are part of the major net are RIP-enabled.

As mentioned earlier in "A Brief Review of Internetworking," routing protocols are the languages of routers. Routers talk to each other over routing protocols and exchange information about the internetwork. In the case of RIP, a RIP router sends to its neighboring routers a list of all the networks it knows about. The list includes its directly attached networks and any networks it has learned about from other routers. In the current example (Figure 2-2), Router B must tell Router A about 172.16.0.0 and Router A must tell Router B about 192.168.2.0 and 192.168.3.0. This communication is accomplished with RIP messages called *updates* or *advertisements*.

Verifying RIP Configuration

With RIP enabled on both routers, you can issue **show ip route**, examine the routing tables again, and verify RIP is working. Here is the output for Router A in Figure 2-2:

```
RTA#sh ip ro
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
    D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
```

continues

```
      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, * - candidate default
      U - per-user static route, o - ODR
      T - traffic engineered route

Gateway of last resort is not set

R  172.16.0.0/16 [120/1] via 10.1.1.2, 00:00:13, Serial1
   10.0.0.0/30 is subnetted, 1 subnet
C     10.1.1.0 is directly connected, Serial1
C  192.168.2.0/24 is directly connected, Ethernet2
C  192.168.3.0/24 is directly connected, Ethernet3
```

The first line of the routing table (shown in boldface) is new and is a route to major net 172.16.0.0. This is the major net advertised by Router B to Router A with the RIP routing protocol (as indicated by the code **R** at the beginning of the line). The output shows that Router A now knows how to reach 172.16.0.0: It has a route to that destination. When Router A receives a packet destined for 172.16.0.0 (perhaps a packet originated by someone in network 192.168.2.0), it can now properly forward it.

The output **[120/1]** provides the *administrative distance* (120) and metric (1 hop) for the route. The administrative distance (covered in Chapter 3) is the priority level of a route and is used to prioritize routes when they are learned from multiple routing protocols. The metric for RIP is a simple hop count and in this example tells you that 172.16.0.0 is one hop (one router) away.

The output **via 10.1.1.2** specifies the address of the neighboring router that is in the direction of the destination (Router B). This is called the *next hop router* or *next hop address*. Packets destined for 172.16.0.0 are sent to this address.

The output **00:00:13** tells you the age of the route: 13 seconds ago, Router A received an advertisement for this route. Because RIP advertises approximately every 30 seconds, this number should stay between 00:00:00 and 00:00:30 under normal circumstances. When a destination becomes unreachable because of an outage or other reason, the age of the route increases until it reaches a maximum age (240 seconds for RIP), at which time it is removed from the routing table.

NOTE OSPF and EIGRP send routing updates only when changes occur in the network. For these routing protocols, it is perfectly valid to have routes that are many hours old.

The output **Serial1** tells you Router A's Serial1 interface points to the destination network. You can think of this as the exit interface: All packets to 172.16.0.0 are sent out this interface.

Similarly, issuing **show ip route** on Router B confirms that Router B is receiving RIP advertisements from Router A:

```
RTB#sh ip ro
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, * - candidate default
       U - per-user static route, o - ODR
       T - traffic engineered route

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 2 subnets
C       172.16.2.0 is directly connected, Ethernet2
C       172.16.3.0 is directly connected, Ethernet3
     10.0.0.0/30 is subnetted, 1 subnet
C       10.1.1.0 is directly connected, Serial1
R    192.168.2.0/24 [120/1] via 10.1.1.1, 00:00:11, Serial1
R    192.168.3.0/24 [120/1] via 10.1.1.1, 00:00:11, Serial1
```

The two RIP routes shown in boldface in the preceding output are advertised by Router A. Router B now has routes to 192.168.2.0 and 192.168.3.0 and can forward packets destined for them.

Deploying IGRP

IGRP is a routing protocol invented by Cisco that addresses some of the scaling problems with RIP. The following is a list of some key points on IGRP:

- Like RIP, IGRP is a distance vector routing protocol.
- Like RIP, IGRP is a classful routing protocol.
- Unlike RIP, IGRP can support large internetworks and is not limited to a 15-hop network diameter. An IGRP internetwork can have a maximum diameter of 255 hops.
- Instead of a hop count, IGRP uses a sophisticated metric (called a *composite* metric) to select optimal routes through an internetwork. The path characteristics included in the metric are bandwidth, delay, load, and reliability. Optionally, you may adjust the weighting of each characteristic for a user-defined formula. See http://www.cisco.com/warp/public/103/index.shtml for more information.
- IGRP can send traffic to a destination over multiple paths in a load-balancing fashion, even if the paths have different metrics. This is called *unequal-cost load balancing*. By default, RIP, IGRP, EIGRP, and OSPF all support equal-cost load balancing, although IGRP and EIGRP support both equal- and unequal-cost load balancing. This is configured with the **variance** router configuration mode command (see the IOS Configuration Guide for IP Routing Protocols or search the Cisco Web site for *variance*).

- IGRP supports autonomous systems (identified by an autonomous system number). An internetwork can support multiple IGRP autonomous systems, and a router can run multiple IGRP processes with one process for each autonomous system. See "A Brief Review of Internetworking" earlier in this chapter for more on autonomous systems.
- IGRP is a Cisco proprietary routing protocol, so you need to use Cisco routers or, for interoperability with RIP and other protocols, routing protocol redistribution (covered in Chapter 3).
- IGRP sends routing updates less frequently than RIP and thus creates less overhead traffic than RIP. But because IGRP has longer timers, it can sometimes converge more slowly than RIP.

The following sections describe IGRP configuration for the example topology introduced earlier in "Deploying RIP."

Configuring IGRP

Configuring IGRP is just as easy as configuring RIP. The two basic IOS commands are **router igrp** and **network**. Starting with Router A, here's how you use IGRP to configure the example network shown in Figure 2-2:

```
RTA#conf t
Enter configuration commands, one per line. End with CNTL/Z.
RTA(config)#router igrp 100
RTA(config-router)#network 192.168.2.0
RTA(config-router)#network 192.168.3.0
RTA(config-router)#network 10.0.0.0
```

The command **router igrp 100** activates the IGRP routing service on the router and changes the prompt to router configuration mode as indicated by the prompt **config-router**. The number **100** is the autonomous system number and is a required parameter of the **router igrp** command. Instead of 100, you may use any number between 1 and 65535 to identify the autonomous system. Routers within the same autonomous system exchange routing protocol information with each other.

The command **network 192.168.2.0** tells the router to enable RIP processing on major net 192.168.2.0. The meaning is the same as when using the **network** command with RIP: It means the router will send and receive IGRP messages on all interfaces that are part of 192.168.2.0. The command also tells the router to advertise this network, 192.168.2.0, to other routers. Like RIP, the **network** command specifies a major net number, not a subnet.

The commands **network 192.168.3.0** and **network 10.0.0.0** enable IGRP processing for Router A's other two connected networks.

Similarly, configure Router B with IGRP and configure the major nets connected to Router B that should run IGRP:

```
RTB(config)#router igrp 100
RTB(config-router)#network 172.16.0.0
RTB(config-router)#network 10.0.0.0
```

The preceding commands are equal in meaning to the commands used to configure Router A. The autonomous system number (100) in **router igrp 100** must match Router A; otherwise, Router A and Router B will not exchange routing updates—they will be in separate autonomous systems.

Verifying IGRP Configuration

With IGRP enabled on both routers, you can issue **show ip route** to verify its operation. Here is the output for Router A:

```
RTA#sh ip ro
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, * - candidate default
       U - per-user static route, o - ODR
       T - traffic engineered route

Gateway of last resort is not set

I  172.16.0.0/16 [100/7500] via 10.1.1.2, 00:00:08, Serial1
   10.0.0.0/30 is subnetted, 1 subnet
C     10.1.1.0 is directly connected, Serial1
C  192.168.2.0/24 is directly connected, Ethernet2
C  192.168.3.0/24 is directly connected, Ethernet3
```

The first line of the routing table (shown in boldface) is a route to major net 172.16.0.0. This route was advertised by Router B to Router A with IGRP (as indicated by the code **I** at the beginning of the line).

The output **[100/7500]** provides the administrative distance (100) and IGRP metric (7500) for the route. As mentioned earlier, the administrative distance is a priority level the router uses to rank routes it receives from multiple routing protocols (covered in Chapter 3). The metric is a composite of bandwidth and delay by default. By tweaking IGRP with the **metric** command, you can incorporate reliability and load into the metric. See the following resources for more information on IGRP metric details:

- IGRP information: http://www.cisco.com/warp/public/103/index.shtml
- IOS Documentation, Configuration Guides: http://www.cisco.com/univercd/cc/td/doc/product/software/index.htm

The output **via 10.1.1.2, 00:00:08, Serial1** is just like the output seen earlier with RIP. It specifies the next hop router (**10.1.1.2**), the age of the route (**00:00:08**), and the exit interface (**Serial1**).

Similarly, issuing **show ip route** on Router B confirms that Router B is receiving IGRP advertisements from Router A:

```
RTB#sh ip ro
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, * - candidate default
       U - per-user static route, o - ODR
       T - traffic engineered route

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 2 subnets
C       172.16.2.0 is directly connected, Ethernet2
C       172.16.3.0 is directly connected, Ethernet3
     10.0.0.0/30 is subnetted, 1 subnet
C       10.1.1.0 is directly connected, Serial1
I    192.168.2.0/24 [100/7500] via 10.1.1.1, 00:00:33, Serial1
I    192.168.3.0/24 [100/7500] via 10.1.1.1, 00:00:34, Serial1
```

In the preceding output, the IGRP routes (**192.168.2.0/24** and **192.168.3.0/24**) advertised by Router A are shown in boldface.

Deploying Enhanced IGRP

EIGRP is an advanced routing protocol invented by Cisco that provides scaling for large internetworks, fast convergence, classless routing features, and low overhead.

EIGRP is a hybrid routing protocol. Fundamentally, it's a distance vector protocol, but it has characteristics of a link-state protocol. EIGRP combines the advantages found in both distance vector protocols and link-state protocols, and does away with many of their respective disadvantages.

The following is a list of key points for EIGRP:

- Although its name implies a close relation to IGRP, EIGRP shares little in common with IGRP.
- Like distance vector protocols, EIGRP is generally easy to deploy.
- Unlike distance vector protocols, EIGRP converges quickly and has low overhead. These are advantages often found in link-state protocols such as OSPF (the most popular link-state protocol). EIGRP sends partial updates instead of its entire routing table—and only when changes occur. Also, EIGRP sends an update to only the neighbors that need it.
- Unlike RIP and IGRP, EIGRP is a classless routing protocol and supports VLSM, supernetting, and route summarization. See Chapter 1 for more information on VLSM, supernetting, and summarization.

- Related to VLSM, EIGRP supports discontiguous subnets. This means subnets of a major net can be separated from each other by different, intermediary major nets.
- Like most link-state protocols, EIGRP scales well in large networks and supports multiple autonomous systems.
- EIGRP internetworks are flexible for changes in topology and reengineering. On the contrary, OSPF internetworks are generally more difficult to reengineer because of the OSPF rules on backbones and areas (see "Deploying OSPF," later in this chapter).
- EIGRP uses the same composite metric as IGRP, except it differs by a factor of 256. Multiply an IGRP metric by 256 and you get the equivalent EIGRP metric.
- Like IGRP, EIGRP is a Cisco proprietary routing protocol, so you need to use Cisco routers or—for interoperability with other protocols—routing protocol redistribution (covered in Chapter 3).
- Like IGRP, EIGRP supports unequal-cost load balancing with the **variance** command.
- EIGRP supports route authentication for enhanced security. This means you can configure a router to accept routing updates only from trusted sources. See the IOS Configuration Guide for IP Routing Protocols for more information.
- In addition to IP, EIGRP supports Novell IPX and AppleTalk. You can use EIGRP instead of IPX-RIP or AppleTalk RTMP (Routing Table Maintenance Protocol) and leverage the advanced features of EIGRP.

The following sections describe EIGRP configuration for the example topology shown in Figure 2-2.

Configuring EIGRP

Configuring EIGRP is just as easy as configuring RIP or IGRP. You use the familiar **router** and **network** commands. Starting with Router A, here's how you configure the example network (Figure 2-2) with EIGRP. Notice that the configuration is almost identical to that of IGRP:

```
RTA(config)#router eigrp 100
RTA(config-router)#network 192.168.2.0
RTA(config-router)#network 192.168.3.0
RTA(config-router)#network 10.0.0.0
```

The command **router eigrp 100** activates the EIGRP routing service on the router and changes the prompt to router configuration mode as indicated by the prompt **config-router**. The number **100** is the autonomous system number and is a required parameter of the command.

As with the RIP and IGRP configurations, the commands **network 192.168.2.0**, **network 192.168.3.0**, and **network 10.0.0.0** enable EIGRP processing for Router A's three connected major nets.

Similarly, you can configure Router B with EIGRP and configure the major nets connected to Router B that should run EIGRP:

```
RTB(config)#router eigrp 100
RTB(config-router)#network 172.16.0.0
RTB(config-router)#network 10.0.0.0
```

The preceding commands are equal in meaning to the commands used to configure Router A. The autonomous system number (100) in **router eigrp 100** must match Router A; otherwise, Router A and Router B will not exchange routing updates.

Verifying EIGRP Configuration

Use the command **show ip route** to verify EIGRP operation. Here is the output for Router A:

```
RTA#sh ip ro
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, * - candidate default
       U - per-user static route, o - ODR
       T - traffic engineered route

Gateway of last resort is not set

D  172.16.0.0/16 [90/1920000] via 10.1.1.2, 00:09:27, Serial1
   10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
D     10.0.0.0/8 is a summary, 00:09:29, Null0
C     10.1.1.0/30 is directly connected, Serial1
C  192.168.2.0/24 is directly connected, Ethernet2
C  192.168.3.0/24 is directly connected, Ethernet3
```

The first line of the routing table (shown in boldface) is a route to major net 172.16.0.0. This route was advertised by Router B to Router A with EIGRP (as indicated by the code **D** at the beginning of the line).

Other information you can extract from the output are the route's administrative distance (**90**), EIGRP metric (**1920000**), next hop address (**10.1.1.2**), age of the route (**00:09:27**), and exit interface (**Serial1**).

NOTE The EIGRP metric differs from the IGRP metric by a factor of 256. The output of **show ip route** in the example proves this: The EIGRP metric (1920000) divided by 256 equals the IGRP metric (7500) from "Verifying IGRP Configuration" earlier in this chapter.

The other EIGRP route printed in boldface, **10.0.0.0/8**, is called a *summary route*. As the name implies, this route summarizes all of the routes in major net 10.0.0.0 with a single, general route—notice the broad **/8** mask, appropriate for generalizing an entire class A network. By default, EIGRP creates a summary route at every major net boundary (a place where two or more major nets meet). Router A is a major net boundary because three major nets meet there: 192.168.2.0, 192.168.3.0, and 10.0.0.0. EIGRP auto-summarization is covered in Chapter 3.

It might seem weird that the summary route points to the **Null0** interface (a logical interface). The Null0 interface is like a trash can or *bit bucket* that leads to nowhere: A packet sent to Null0 is simply discarded by the router. So why have it? This route to Null0 is used to originate a summary route and advertise it to other routers. Router A tells other routers (in the example, just Router B) that it knows how to reach major net 10.0.0.0—that is, it advertises 10.0.0.0/8. When Router A receives a packet destined for 10.0.0.0, it doesn't actually use the summary route to forward the packet. Instead, it uses a route within 10.0.0.0 that describes the destination more exactly. This is called *using the most specific route*. The router uses the most specific route when selecting a route to a destination. If the router doesn't have a specific route to a particular destination, the summary route is the only choice—and in that case, the packet does indeed go to Null0 (it's discarded). This could happen when a client erroneously sends a packet to an unknown destination or when there are routing problems on the network.

For completeness, the following is the output of **show ip route** for Router B:

```
RTB#sh ip ro
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, * - candidate default
       U - per-user static route, o - ODR
       T - traffic engineered route

Gateway of last resort is not set

     172.16.0.0/16 is variably subnetted, 3 subnets, 2 masks
D       172.16.0.0/16 is a summary, 01:30:35, Null0
C       172.16.2.0/24 is directly connected, Ethernet2
C       172.16.3.0/24 is directly connected, Ethernet3
     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
D       10.0.0.0/8 is a summary, 01:30:35, Null0
C       10.1.1.0/30 is directly connected, Serial1
D    192.168.2.0/24 [90/1920000] via 10.1.1.1, 01:30:08, Serial1
D    192.168.3.0/24 [90/1920000] via 10.1.1.1, 01:30:08, Serial1
```

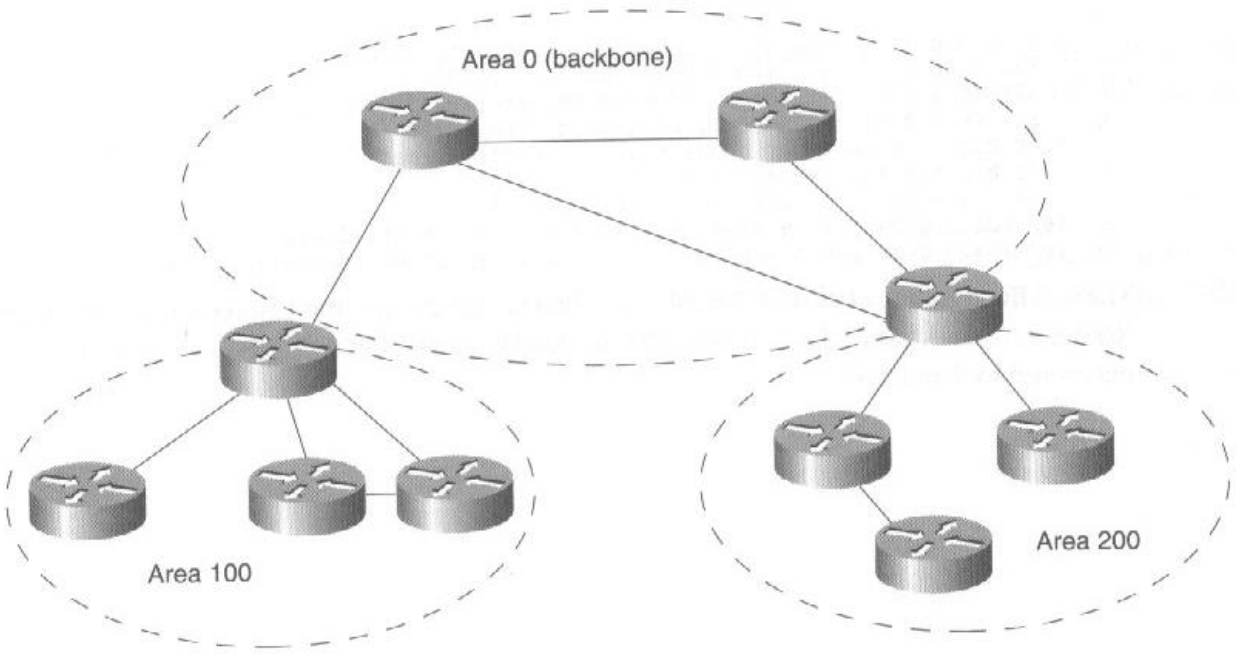
The boldface routes (**192.168.2.0/24** and **192.168.3.0/24**) are the EIGRP routes advertised by Router A. The other EIGRP routes (**172.16.0.0/16** and **10.0.0.0/8**) are summary routes generated by Router B.

Deploying OSPF

OSPF is a standards-based, link-state routing protocol defined by RFC 2328. The following is a list of some key points for OSPF:

- OSPF is a link-state routing protocol. An OSPF router sends small updates (called *link-state advertisements*, or LSAs) that include information for its attached links only, not for all known routes. These updates propagate a portion of the internetwork called an *area*. Each router in an area builds a database from all of the LSAs it receives. From the database, a router can calculate a shortest (least-cost) path to every known destination by using the *Dijkstra algorithm*.
- Like EIGRP, OSPF is a classless protocol that supports VLSM, supernetting, summarization, and discontiguous subnets.
- OSPF converges quickly and creates low overhead over network links (updates are sent only when they are necessary). However, the LSA database and Dijkstra algorithm require more memory and CPU resources than other routing protocols, resulting in more system resource overhead at the router level.
- OSPF's metric is *cost*. By default, the cost across a link is 10^8 divided by the link bandwidth (10^8/BW). You can set the cost on an interface with the **ip osfp cost** interface configuration command.
- An OSPF autonomous system is built of areas joined in a hierarchical fashion. One area, called *area 0* ("area zero") or the *backbone area*, is required. Inter-area traffic must traverse area 0. Figure 2-3 illustrates a simple arrangement of areas and routers with a couple of user-defined areas (areas 100 and 200) connected to area 0.

Figure 2-3 *A Basic OSPF Hierarchy with Area 0 and Two Non-Backbone Areas*



- All OSPF areas must connect to area 0. If, for some reason, an area cannot be directly connected to area 0, a *virtual link* may be configured. A virtual link joins the stray area to area 0 via a transit area (another non-backbone area).
- OSPF generally scales well because of its fast convergence, low overhead, and hierarchical design.
- OSPF generally requires more initial planning and design than other protocols. Very few real-world networks fit into nice hierarchical OSPF areas. A thoughtfully defined area plan that considers growth, router sizing, and area sizing will lead to a more stable and manageable internetwork. It is recommended that you seek the advice of an experienced consultant or Cisco engineer before swinging an OSPF design into action.
- Depending on the original design and the degree of changes on your network, OSPF might be less flexible to topological changes and network reengineering. Select and size your areas wisely. Use the Bibliography resources and recommendations from experienced OSPF engineers. Nothing beats real-world experience when it comes to OSPF network design.
- OSPF supports equal-cost load balancing but not unequal-cost load balancing.
- OSPF supports route authentication for enhanced security.
- More complete information on OSPF can be found in Cisco's *OSPF Design Guide* at http://www.cisco.com/warp/public/104/1.html and in the resources listed in the Bibliography.

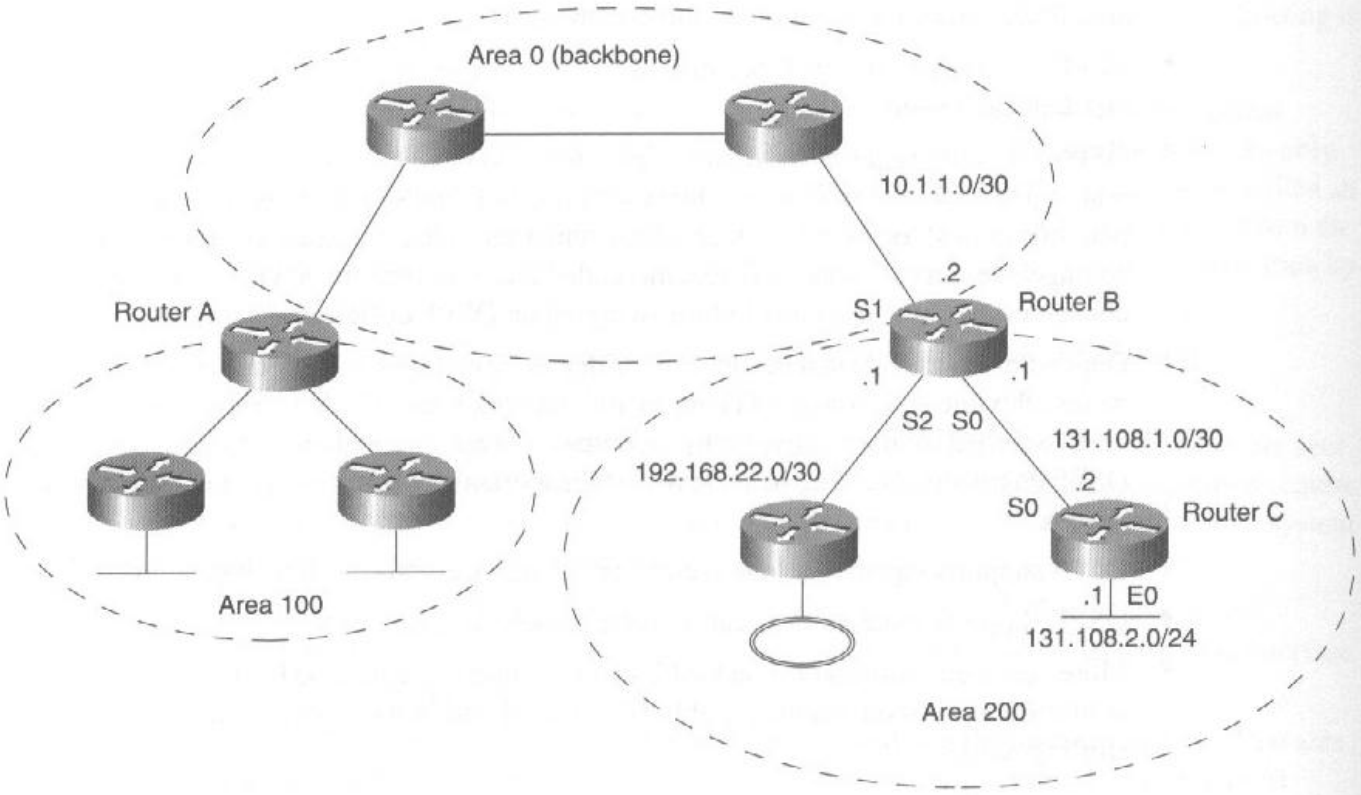
Configuring OSPF

For the purpose of illustrating basic OSPF configuration, consider the example topology depicted in Figure 2-4.

Figure 2-4 consists of an area 0 and two non-backbone areas, area 100 and area 200. Router A is called an *area border router (ABR)* because it attaches to both area 0 and area 100. Router B is also an ABR because it connects to area 0 and area 200. Router C is completely contained within area 200 and is called an *internal router.*

Figure 2-4 illustrates a key point for OSPF configuration: You assign *links*, not routers themselves, to areas. Router B, for example, has three links: one in area 0 (Serial1) and two in area 200 (Serial2 and Serial0). The importance of keeping this perspective will be apparent during configuration.

Figure 2-4 *An Example Internetwork for OSPF Configuration*



For the purpose of illustrating basic OSPF configuration, consider Router B and Router C. Router C is a good router to start with because it's an internal router in area 200. The following commands configure OSPF on Router C:

```
RTC(config)#router ospf 10
RTC(config-router)#network 131.108.2.0 0.0.0.255 area 200
RTC(config-router)#network 131.108.1.0 0.0.0.3 area 200
```

The command **router ospf 10** starts the OSPF software process on the router. The number **10** denotes a process identification number (*process ID*, for short). Unlike IGRP and EIGRP, this is not an autonomous system number. The process ID uniquely identifies an OSPF routing process when there are multiple OSPF processes running on the router (typically, you configure only one OSPF process per router). The OSPF process ID is locally significant and is not shared with other routers—Router C's process ID does not have to match process IDs on other routers—but if you use a consistent process ID across all routers, you'll probably make configuration maintenance a little easier. Therefore, pick any number that looks good to you.

The command **network 131.108.2.0 0.0.0.255 area 200** assigns Router C's Ethernet0 interface to area 200. This is not readily apparent until you break up the command.

The **network** command has two main pieces:

- The matching criteria: **131.108.2.0 0.0.0.255**
- The area assignment: **area 200**

The area assignment is straightforward: Whatever is matched by the matching criteria is placed into area 200.

The matching criteria requires a look at the IP addresses on the router. As depicted in Figure 2-4, the IP address of Ethernet0 is 131.108.2.1/24. The **network** command contains an address pattern to match 131.108.2.0 and a *wildcard mask* 0.0.0.255. The wildcard mask looks like the inverse of a subnet mask, and that is indeed the case for this example (but it does not have to be). The wildcard mask defines the bits in the address pattern that the router must match when comparing the pattern to its interface addresses. Any interfaces that match the criteria are put into area 200. A zero bit in the mask means to match the corresponding bit of the pattern. A one bit in the mask is a "don't care" bit and means the corresponding bit of the pattern is ignored.

NOTE Recall from Chapter 1 that there are 32 bits in an IP address but the address is written in dotted decimal notation.

The criteria **131.108.2.0 0.0.0.255** means *match the interfaces that start with 131.108.2 and the last octet (the last 8 bits) can be anything*. This criteria matches Router C's Ethernet0; therefore, Ethernet0 is a link in area 200 (the area assignment). The criteria does not match Serial0, whose address is 131.108.1.2/30, so another command is needed.

NOTE OSPF's **network** command is a logical assignment rather than a physical one. This means the router's hardware interfaces can change without affecting the assignment of areas as long as the addressing stays intact.

The second **network** command, **network 131.108.1.0 0.0.0.3 area 200**, means all interfaces that match the criteria **131.108.1.0 0.0.0.3** are assigned to area 200. The mask 0.0.0.3 means the first 30 bits of the pattern 131.108.1.0 are matched—convert 0.0.0.3 to binary and you will see 30 zeros followed by two ones. This criteria matches Router C's Serial0; therefore, the link between Router C and Router B is assigned to area 200.

Instead of entering two **network** commands, you could enter a single **network** command that matches all interfaces on Router C and puts them in area 200, like this:

```
RTC(config)#router ospf 10
RTC(config-router)#network 0.0.0.0 255.255.255.255 area 200
```

where **0.0.0.0 255.255.255.255** matches all addresses because the wildcard mask is all ones—instead of 0.0.0.0, you could type any address and get the same result. This works fine for the example because Router C has only two interfaces and both of them are in area 200; however, such a broad match provides the least amount of control over your area assignments. This could be a factor if you add new interfaces to a router or change specific area assignments.

Router B is an ABR (refer to Figure 2-4). The following commands configure OSPF on Router B:

```
RTB(config)#router ospf 10
RTB(config-router)#network 131.108.1.0 0.0.0.3 area 200
RTB(config-router)#network 10.1.1.0 0.0.0.3 area 0
RTB(config-router)#network 192.168.22.0 0.0.0.3 area 200
```

The command **router ospf 10** starts the OSPF process on Router B with a process ID of 10.

The command **network 131.108.1.0 0.0.0.3 area 200** assigns Router B's Serial0 interface to area 200. This agrees with Router C's configuration; that is, both routers agree that the link between them is in area 200. If the routers do not agree, they will not become OSPF neighbors and will not exchange LSAs.

The command **network 10.1.1.0 0.0.0.3 area 0** assigns Router B's Serial1 interface to area 0, the backbone area. This makes Router B an ABR: It's connected to area 0 and at least one other non-backbone area.

The command **network 192.168.22.0 0.0.0.3 area 200** assigns Router B's Serial2 interface to area 200. This agrees with the design described in Figure 2-4.

NOTE When configuring OSPF on so-called nonbroadcast multiaccess (NBMA) networks such as frame relay and ATM, you might need to configure the **ip ospf network** interface command or **neighbor** router configuration command. Consult the sources in the Bibliography for more information.

Verifying OSPF Configuration

Several good **show** commands exist for verifying OSPF operation. One of the first you should enter is **show ip ospf neighbors**:

```
RTC#sh ip ospf nei

Neighbor ID     Pri   State           Dead Time   Address         Interface
172.16.3.1      1     FULL/  -        0:00:33     131.108.1.1     Serial0
```

The preceding output is a list of OSPF routers neighboring Router C. The key field to examine is **State**. When OSPF is functioning properly, the neighbor state is **FULL**.

The **Neighbor ID** 172.16.3.1 is the OSPF router ID of Router B. 172.16.3.1 is an address assigned to a *loopback interface* on Router B. A loopback interface is a virtual, software interface on the router from which OSPF borrows an address to use for an ID. This is configured with the **interface loopback***N* command, where *N* represents a user-defined number:

```
RTB(config)#int loopback0
RTB(config-if)#ip address 172.16.3.1 255.255.255.255
```

Configuring a loopback interface is optional. The benefit of using a loopback interface is that the router's interface is always up—the network on a loopback interface never goes down unless you force it down with the **shutdown** interface command. Because the loopback is always up, the OSPF router ID never changes (unless you add a new loopback with a higher address). Having a constant router ID is important for some OSPF features, such as virtual links, that must be configured with explicit router IDs. Consistent router IDs also make OSPF troubleshooting easier.

Another good **show** command for validating OSPF configuration is **show ip ospf interface**:

```
RTB#sh ip ospf int
Serial0 is up, line protocol is up
 Internet Address 131.108.1.1/30, Area 200
 Process ID 10, Router ID 172.16.3.1, Network Type POINT_TO_POINT, Cost: 64
 Transmit Delay is 1 sec, State POINT_TO_POINT,
 Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
  Hello due in 00:00:07
 Neighbor Count is 1, Adjacent neighbor count is 1
  Adjacent with neighbor 131.108.2.1
 Suppress hello for 0 neighbor(s)
Serial1 is up, line protocol is up
 Internet Address 10.1.1.2/30, Area 0
 Process ID 10, Router ID 172.16.3.1, Network Type POINT_TO_POINT, Cost: 50
 Transmit Delay is 1 sec, State POINT_TO_POINT,
 Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
  Hello due in 00:00:08
 Neighbor Count is 1, Adjacent neighbor count is 1
  Adjacent with neighbor 171.70.240.253
 Suppress hello for 0 neighbor(s)
<lines deleted for brevity>
```

With the preceding output, you can check that your OSPF **network** commands with the wildcard masks are assigning interfaces to the expected areas (see the lines in boldface).

Finally, you want to ensure that routers are receiving and registering the OSPF routes. The familiar **show ip route** command does this. The following is a partial listing:

```
RTB#sh ip ro
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, * - candidate default
       U - per-user static route, o - ODR
       T - traffic engineered route

Gateway of last resort is not set

     10.0.0.0/30 is subnetted, 1 subnet
C       10.1.1.0 is directly connected, Serial1
     131.108.0.0/16 is variably subnetted, 2 subnets, 2 masks
O       131.108.2.0/24 [110/74] via 131.108.1.2, 05:06:36, Serial0
C       131.108.1.0/30 is directly connected, Serial0
     192.168.2.0/24 is subnetted, 1 subnet
O IA    192.168.2.0 [110/51] via 10.1.1.1, 05:06:26, Serial1
<lines deleted for brevity>
```

The preceding output for Router B contains two OSPF routes shown in boldface. Route 131.108.2.0/24 is coded with a lone letter **O**. This means the route is an intra-area route: The route and Router B are part of the same area. Route 192.168.2.0/24, on the other hand, is coded with **O IA**. This means the route is an inter-area route: The route is from a different area, not attached to this router. The inter-area route originated from another area and arrived at Router B through the backbone area, area 0.

NOTE You might notice from the key at the top of the **show ip route** output that there are other kinds of OSPF routes. External routes are routes redistributed into OSPF (see Chapter 3). For information on Not-So-Stubby-Area routes, see the IOS Configuration Guide for IP Routing Protocols.

Summary

This chapter covered basic theory and configuration for the most common interior routing protocols: RIP, IGRP, EIGRP, and OSPF. The next chapter, Chapter 3, builds on this baseline study and focuses on how to manage these routing protocols.

The following are the key concepts of this chapter:

- Internetworking is the practice of connecting multiple individual networks so they function as a single large internetwork or internet.
- Routing, which is the process of finding a path to a destination, makes internetworking possible. It is the crucial network service that governs the flow of traffic through your organization.
- A routing protocol is a language for routers. Routers use routing protocols to exchange information about the topology and health of the internetwork.
- RIP, one of the oldest and simplest of routing protocols, is a classful, distance vector protocol limited to a 15-hop metric. RIP generally does not scale well in large internets.
- IGRP, a routing protocol invented by Cisco, is a classful, distance vector protocol with a composite metric. IGRP supports autonomous systems and is designed to scale to larger internets than RIP.
- EIGRP, another routing protocol invented by Cisco, is a classless, hybrid protocol with a composite metric. EIGRP converges quickly, generates low traffic overhead, and scales well in large internets.
- Enabling RIP, IGRP, or EIGRP is fairly simple. To establish one of these services you need two fundamental commands: **router** and **network**. For example, **router eigrp 100** and **network 172.16.0.0**.
- OSPF is a classless, link-state protocol with a cost metric. OSPF converges quickly, generates low traffic overhead, and scales well in large internets. It requires more system resources and more up-front planning than the other protocols.