This chapter covers the following subjects:

- Link-State Routing Protocol and OSPF Concepts
- Balanced Hybrid Routing Protocol and EIGRP Concepts
- OSPF Configuration
- EIGRP Configuration

CHAPTER 6

OSPF and EIGRP Concepts and Configuration

Routing protocols learn routes—the current best routes—and put those subnets in the IP routing table. In the last chapter, you saw how distance vector protocols accomplish that goal. In this chapter, you will read about how two different types of routing protocols, link-state and balanced hybrid, accomplish that same goal.

When people first created distance vector protocols, routers had slow processors connected to slow links (relative to today's technology). For perspective, RFC 1058, published as an Internet standard RFC in June 1988, defined the first version of RIP for IP. The underlying distance vector logic was defined far in advance of the Internet RFC for RIP, mainly in the early 1980s. Therefore, distance vector protocols were designed to advertise just the basic routing information across the network to save bandwidth. These protocols were also designed to use little processing and memory, because the routing devices of the day had, relative to today, only small amounts of memory and processing power.

Link-state and balanced hybrid protocols were developed mainly in the early to mid-1990s, and they were designed under the assumptions of faster links and more processing power in the routers. By sending more information, and requiring the routers to perform more processing, these newer types of routing protocols can gain some important advantages over distance vector protocols—mainly, faster convergence. The goal remains the same—to add the currently-best routes to the routing table—but these protocols use different methods to find and add those routes. This chapter outlines how link-state and balanced hybrid protocols do their work, as well as how to configure the most popular routing protocol of each type—Open Shortest Path First (OSPF) and Enhanced IGRP (EIGRP). Please refer to Appendix F for some further details about a few of the topics in this chapter.

"Do I Know This Already?" Quiz

The purpose of the "Do I Know This Already?" quiz is to help you decide if you need to read the entire chapter. If you intend to read the entire chapter, you do not necessarily need to answer these questions now.

The ten-question quiz, derived from the major sections in the "Foundation Topics" section, helps you determine how to spend your limited study time.

Table 6-1 outlines the major topics discussed in this chapter and the "Do I Know This Already?" quiz questions that correspond to those topics.

Table 6-1 *"Do I Know This Already?" Foundation Topics Section-to-Question Mapping*

Foundations Topics Section	Questions Covered in This Section
Link-State Routing Protocol and OSPF Concepts	3, 5
Balanced Hybrid Routing Protocol and EIGRP Concepts	2, 4, 5
OSPF Configuration	1, 6, 7, 8
EIGRP Configuration	9, 10

CAUTION The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you don't know the answer to a question or you're only partially sure of the answer, you should mark this question as wrong for purposes of the self-assessment. Giving yourself credit for an answer you guess correctly skews your self-assessment results and might give you a false sense of security.

1. Which of the following affects the calculation of OSPF routes when all possible default values are used?

   a. Bandwidth
   b. Delay
   c. Load
   d. Reliability
   e. MTU
   f. Hop count

2. Which of the following affects the calculation of EIGRP metrics when all possible default values are used?

   a. Bandwidth
   b. Delay
   c. Load
   d. Reliability
   e. MTU
   f. Hop count

3. OSPF runs an algorithm to calculate the currently-best route. Which of the following terms refer to that algorithm?

   a. SPF
   b. DUAL
   c. Feasible successor
   d. Dijkstra
   e. Good old common sense

4. EIGRP uses an algorithm to find routes when no backup route exists. Which of the following terms refers to that algorithm?

   a. SPF
   b. DUAL
   c. Feasible successor
   d. Dijkstra
   e. Good old common sense

5. How do OSPF and EIGRP notice when a neighboring router fails?

   a. The failing router sends a message before failing
   b. The failing router sends a "dying gasp" message
   c. The router notices a lack of routing updates for a period of time
   d. The router notices a lack of Hello messages for a period of time

6. Which of the following network commands, following the command **router ospf 1**, tell this router to start using OSPF on interfaces whose IP addresses are 10.1.1.1, 10.1.100.1, and 10.1.120.1?

   a. **network 10.0.0.0 255.0.0.0 area 0**
   b. **network 10.0.0.0 0.255.255.255 area 0**
   c. **network 10.0.0.1 255.0.0.255 area 0**
   d. **network 10.0.0.1 0.255.255.0 area 0**
   e. **network 10.0.0.0 255.0.0.0**
   f. **network 10.0.0.0 0.255.255.255**
   g. **network 10.0.0.1 255.0.0.255**
   h. **network 10.0.0.1 0.255.255.0**

7. Which of the following network commands, following the command **router ospf 1**, tells this router to start using OSPF on interfaces whose IP addresses are 10.1.1.1, 10.1.100.1, and 10.1.120.1?

   a. **network 0.0.0.0 255.255.255.255 area 0**
   b. **network 10.0.0.0 0.255.255.0 area 0**
   c. **network 10.1.1.0 0.x.1x.0 area 0**
   d. **network 10.1.1.0 255.0.0.0 area 0**
   e. **network 10.0.0.0 255.0.0.0 area 0**

8. Which of the following commands list the OSPF neighbors off interface serial 0/0?

   a. **show ip ospf neighbor**
   b. **show ip ospf interface**
   c. **show ip neighbor**
   d. **show ip interface**
   e. **show ip ospf neighbor interface serial 0/0**

9. In the **show ip route** command, what code designation implies that a route was learned with EIGRP?

   a. E
   b. I
   c. G
   d. R
   e. P
   f. A
   g. B
   h. C
   i. D

10. Which of the following network commands, following the command **router eigrp 1**, tells this router to start using EIGRP on interfaces whose IP addresses are 10.1.1.1, 10.1.100.1, and 10.1.120.1?

    a. **network 10.0.0.0**

    b. **network 10.1.1x.0**

    c. **network 10.0.0.0 0.255.255.255**

    d. **network 10.0.0.0 255.255.255.0**

The answers to the "Do I Know This Already?" quiz appear in Appendix A. The suggested choices for your next step are as follows:

- **8 or less overall score**—Read the entire chapter. This includes the "Foundation Topics," "Foundation Summary," and "Q&A" sections.
- **9 or 10 overall score**—If you want more review on these topics, skip to the "Foundation Summary" section and then go to the "Q&A" section. Otherwise, move to the next chapter.

Foundation Topics

Link-State Routing Protocol and OSPF Concepts

The ICND exam covers link-state protocol concepts and a single link-state routing protocol—OSPF. In this chapter, you will read about the basics. If you find yourself thinking that there has to be more to OSPF than what is covered here, you're right! If you move on to the CCNP certification, you will need to learn many more details about OSPF and link-state protocols. For CCNA, you just need to know the basics.

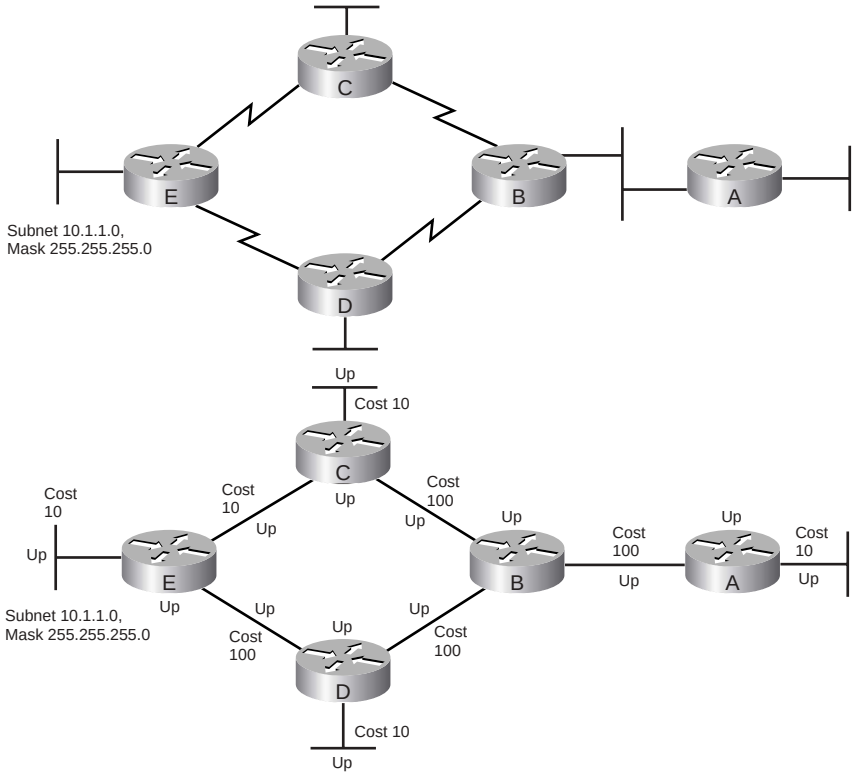
Link-state and distance vectors share a common goal—filling the routing tables with the currently-best routes. They differ significantly in how they accomplish this task. The largest difference between the two is that distance vector protocols advertise sparse information. In fact, distance vector protocols know that other routers exist only if the other router broadcasts a routing update to them. When a distance vector protocol in a router receives a routing update, the update says nothing about the routers beyond the neighboring router that sent the update. Conversely, link-state protocols advertise a large amount of topological information about the network, and the routers perform some CPU-intensive computation on the topological data. They even discover their neighbors before exchanging routing information.

Figure 6-1 illustrates what a router might advertise with a link-state protocol. The actual contents of the routing updates are not shown. This is a graphical representation.

The network topology is shown in the upper part of the figure. With a link-state protocol, the whole network topology is described in the routing update, as shown in the lower part of the figure. Rather than Router B's telling Router A what the metric (or cost) for the route should be, Router B tells Router A the metric associated with every link in the network. Router B also tells Router A about all the routers in the network, including which subnets they are attached to and their status. In effect, it is like Router A has been given a map of the network, along with the cost associated with each link. Of course, the map is not literally a drawn map—it is a mathematical model of the network based on the topology information.

The link-state protocol on Router A calculates the lowest-cost route to all subnets based on the topology information, including the route to subnet 10.1.1.0, mask 255.255.255.0. When more than one route to a subnet exists, the link-state routing protocol chooses the path with the lowest metric. Packets traveling to 10.1.1.0 from Router A go through Router C because this route has the lower cost.

Figure 6-1 *Content Advertised to a Neighboring Router: Link State*



Unlike distance vector protocols, link-state protocols must calculate the metric instead of simply being told the metric in the received routing update. For instance, with distance vector protocols, Router B tells Router A something like "subnet 10.1.1.0, metric 3." With link-state protocols, the topology information learned by a router includes a cost associated with each link in the network. A router totals the cost associated with each link in each route to find the metric associated with the route. For instance, Router A discovers two routes to subnet 10.1.1.10, with a metric of 220 for the route to 10.1.1.0 through Router C and a metric of 310 for the route to 10.1.1.0 through Router D. In both cases, Router A uses Router B as the next hop. Therefore, Router A puts a route to 10.1.1.0 in its routing table, using Router B's interface IP address as the next hop. Similarly, Router B calculates routes to 10.1.1.0 through Router C and Router D and places the better route (through Router C) in Router B's routing table.

The algorithm used to calculate routes with link-state protocols is called the *Shortest Path First (SPF) algorithm*. It is sometimes called the *Dijkstra SPF algorithm*, or simply *Dijkstra* after its inventor. You can look at Figure 6-1 and figure out the two routes and total the metrics to find the lowest-cost route. Routers, however, cannot just look at the figure. In fact,

routers really just know the list of routers and subnets and which routers are connected to which subnets. The SPF algorithm processes all that topology information to come up with the best route to each subnet.

Link-state protocols do not just start broadcasting topology information out every interface when the router first boots. Instead, link-state protocols first use a process by which they discover neighbors. (Neighbors can also be statically defined instead of being discovered.) *Neighbors* are other routers, also running the same link-state protocol, that share a common subnet. As soon as routers know that they are neighbors, they can exchange their respective copies of the topology information—called the *topology database*—and then run SPF to calculate new routes.

After a router identifies a neighbor, the routers exchange the information in their topology databases. OSPF sends several types of packets—*link-state updates* (LSUs) and *Database Description* (DD) packets—that contain topology information as well as individual link-state advertisements (LSAs). For instance, a link LSA describes a subnet number and mask, the cost (metric), and other information about the subnet. Also, OSPF uses a reliable protocol to exchange routing information, ensuring that lost LSU packets are retransmitted. So OSPF routers can know with certainty whether a neighbor has yet received all the LSAs when exchanging routing information.

The next several pages cover some of the details of how OSPF and link-state protocols work. The basic process of learning routes for the first time with OSPF goes something like this:

1. Each router discovers its neighbors on each interface. The list of neighbors is kept in a neighbor table.
2. Each router uses a reliable protocol to exchange topology information (LSAs) with its neighbors.
3. Each router places the learned topology information in its topology database.
4. Each router runs the SPF algorithm against its own topology database to calculate the best routes to each subnet in the database.
5. Each router places the best route to each subnet in the IP routing table.

Link-state protocols require more work from the routers, but the work is typically worth the effort. A router running a link-state protocol uses more memory and more processing cycles than do distance vector protocols. The topology updates take many more bytes compared to distance vector routing updates, although because OSPF does not advertise all routes every update interval like distance vector protocols do, the overall number of bytes sent can be smaller with OSPF. A link-state protocol uses a neighbor table and a topology database in addition to adding routes to the routing table. Also, the SPF algorithm must be used to recalculate routes when links go up or down, and the algorithm itself requires memory and

processing on each router. However, you can reduce the amount of memory and processing required by following some good design practices, some of which are covered in this section. Also, OSPF converges much more quickly than do distance-vector protocols—and fast convergence is the most important feature of a routing protocol.

OSPF Protocols and Operation

So far, the text has described a wide overview of how link-state protocols work, with a few specific details about OSPF. The next several sections of this chapter take a more detailed look at OSPF operation and protocols. The topics are listed in order with regards to what a router does with OSPF when that router first loads IOS.

Identifying OSPF Routers with a Router ID

The OSPF topology database consists of lists of subnet numbers (called *links*, hence the name *link-state database*). It also contains lists of routers, along with the links (subnets) to which each router is connected. Armed with the knowledge of links and routers, a router can run the SPF algorithm to compute the best routes to all the subnets.

The database entries for subnets can be easily identified with the subnet number and the associated prefix. (Remember, a subnet mask can be represented by a prefix value as well, for instance, /24, to mean the same thing as 255.255.255.0.) To uniquely identify each router in the database, OSPF uses a concept called the *OSPF router ID* (RID). The end goal is to have a way to uniquely identify each router in the database, and to make sure that no two routers have the same RID to avoid confusion. So, OSPF has each router use one of the routers' IP addresses, because the routers should not use duplicate IP addresses.

Of course, routers typically have several interfaces and several IP addresses. A Cisco router uses the following criteria to select its RID:

- The router first looks for the existence of any loopback interfaces that are up. If so, the router picks the highest numeric IP address among the loopback interfaces
- If no loopback is found, the router picks the highest numeric IP address from all its working (up and up) interfaces

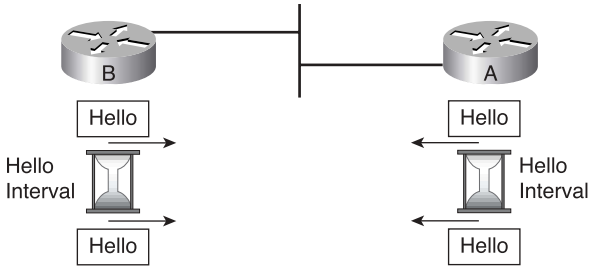
NOTE A loopback interface is a virtual interface that can be configured with the **interface loopback** *interface-number* command, where *interface-number* is an integer. Loopback interfaces are always in an "up and up" state unless administratively placed into a shutdown state. For instance, a simple configuration of the command **interface loopback 0**, followed by **ip address 192.168.200.1 255.255.255.0** would create a loopback interface, and assign it an IP address. Assuming the subnet on the loopback interface is advertised into the internetwork, an engineer can **ping**, **trace**, and **telnet** to the loopback IP address.

Each router chooses its OSPF RID when OSPF is initialized. Initialization happens during the initial load of IOS. So, if OSPF comes up, and later other interfaces come up that happen to have higher IP addresses, then the OSPF RID does not change until the OSPF process is restarted. (OSPF can be restarted with the **clear ip ospf process** command as well.)

Meeting Neighbors by Saying Hello

Once a router has picked its OSPF RID, and some interfaces come up, the router is ready to meet its OSPF neighbors. OSPF routers can become neighbors if they are connected to the same subnet. To discover other OSPF-speaking routers, a router multicasts OSPF Hello packets out to each interface, and hopes to receive OSPF Hello packets from other routers connected to those interfaces. Figure 6-2 outlines the basic concept.

Figure 6-2 *Link-State Hello Packets*



Router A and B both send Hello messages onto the LAN. Soon afterwards, the two routers can begin to exchange topology information with each other, and then run the Dijkstra algorithm in order to fill the routing table with the best routes. The Hello messages themselves have the following features:

- The Hello message follows the IP packet header, with the IP packet protocol type 89.
- Hello packets are sent to multicast IP address 224.0.0.5, which is intended for all OSPF-speaking routers.
- OSPF routers listen for packets sent to IP multicast address 224.0.0.5, in part hoping to receive Hello packets.

Routers learn several important pieces of information from looking at the received Hello packets. The Hello message includes the sending router's RID, Area ID, Hello interval, dead interval, router priority, designated router, backup designated router, and a list of neighbors that the sending router already knows about on the subnet. (More to come on most of these items.)

The list of neighbors is particularly important to the Hello process. For example, when Router A receives a Hello from Router B, Router A needs to somehow tell Router B that Router A got the Hello. To do so, Router A adds Router B's RID to the list of OSPF neighbors inside the next Hello that Router A multicasts onto the network. Likewise, when Router B receives Router A's Hello, Router B's next (and ongoing) Hellos include Router A's RID in the list of neighbors.

Once a router sees its own RID in a received Hello, the router believes that *two-way* communication has been established to that neighbor. The two-way state for a neighbor is important because at that point, more detailed information, such as LSAs, can be exchanged. Also, in some cases on LANs, neighbors might reach the two-way state and stop there—more on that in the section titled "Database Exchange and Becoming Fully Adjacent" coming up in a few pages.

Potential Problems in Becoming a Neighbor

Interestingly, receiving a Hello from a router on the same subnet does not always result in two routers becoming neighbors. It's like meeting a new neighbor in real life—if you happen to disagree about a lot of things, and not get along, you might literally live on the same street, but not really talk all that much. Similarly, with OSPF, routers on the same subnet must agree about several of the parameters exchanged in the Hello; otherwise, the routers simply do not become neighbors. Specifically, the following must match before a pair of routers will become neighbors:

- Subnet mask used on the subnet
- Subnet number (as derived using the subnet mask and each routers' interface IP address)
- Hello Interval
- Dead Interval
- OSPF Area ID

If any one of these parameters differs, the routers do not become neighbors. In short, if troubleshooting OSPF when routers should be neighbors, and they are not, check this list!

Now a quick review of the detailed steps so far—knowing that as of yet, two neighbors have not yet exchanged any routing information:

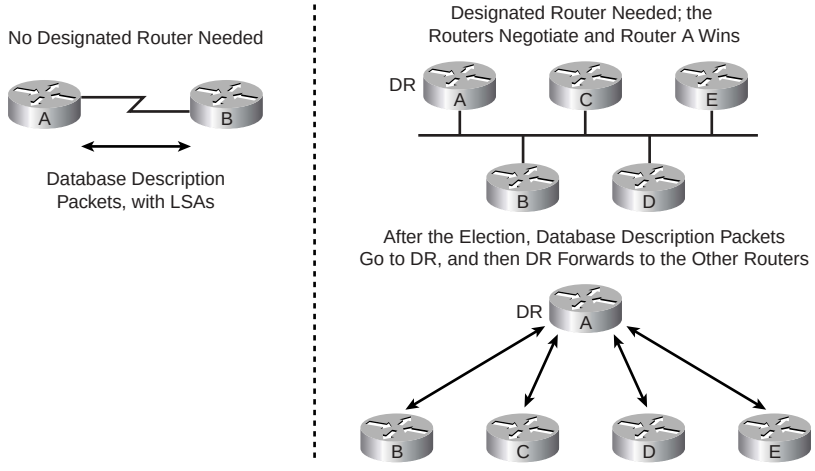
1. Each router initializes OSPF and picks its RID.
2. Routers discover each other as neighbors using Hello packets.
3. Routers reach a two-way communications state with a neighbor once they see their own RID in the Hello from that neighbor.

The neighbors might finally be ready to exchange topology information—but maybe not. The next step is to elect a *Designated Router* (DR) for LANs, and in some cases, for Frame Relay and ATM WANs.

Reducing OSPF Overhead Using Designated Routers

In some cases, a *Designated Router* (DR) must be elected for the subnet before *Database Description (DD)* packets, containing LSAs, can be exchanged between routers. DRs are always required on a LAN, and sometimes (depending on topology and configuration) required with Frame Relay and ATM. Figure 6-3 shows the classic example, with a DR being required on a LAN, and a DR not being required on a point-to-point serial link.

Figure 6-3 *No DR on a Point-to-Point Link, with a DR on the LAN*



When a DR is not required, neighboring routers can go ahead and start sending routing updates to each other, as shown on the left side of the figure. On the right side, the top figure shows a LAN topology, where a DR election has been held, with Router A winning the election. As a result, all routing updates flow to and from Router A, with Router A essentially distributing the routing updates (topology information in DD and LSU packets) to the other routers.

The DR concept prevents overloading a subnet with too much OSPF traffic when many routers are on a subnet. Of course, lots of routers could be attached to one LAN, which is why a DR is required for routers attached to a LAN. For instance, if 10 routers were attached to the same LAN subnet, and they were allowed to forward OSPF updates to each of the other nine routers, topology updates would flow between 45 different pairs of neighbors—with almost all the information being redundant. With the DR concept, as seen in Figure 6-3 on the right, that same LAN would require routing updates only between the DR and the nine other routers, significantly reducing flooding of OSPF information across the LAN.

A router decides whether it needs to elect a DR, plus some other details of operation, based on an interface's OSPF *network type*. For instance, a point-to-point link has a default OSPF network type of *point-to-point*, which does not require a DR. Similarly, LAN interfaces default to an OSPF network type of *broadcast*, which always requires a DR. For Nonbroadcast Multiaccess (NBMA) networks such as Frame Relay, OSPF allows for the configuration of five different variations of OSPF network types, with some options that require a DR, and with others that do not. (These types can be configured with the **ip ospf network** *type* command.)

Because the DR is so important to the exchange of routing information, the loss of the elected DR could cause delays in convergence. OSPF includes the concept of a *Backup DR* (BDR) on each subnet, so when the DR fails, or loses connectivity to the subnet, the BDR can take over as the DR. (All routers except for the DR and BDR are typically called "DROther" in IOS **show** command output.)

Electing the Designated Router

When a DR is required, the neighboring routers hold an election. To elect a DR, the neighboring routers look at two fields inside the Hello packets they receive, and choose the DR based on the following criteria:

- The router sending the Hello with the *highest OSPF Priority* setting becomes the DR.
- If two or more routers tie with the highest priority setting, the router sending the Hello with the *highest RID* wins.
- While not always the case, typically the router with the second highest priority becomes the BDR.
- A priority setting of 0 means that the router never can become DR.
- The range of priority values that allows a router to be a candidate are 1 through 255.
- If a DR has been elected, and a new router starts sending Hellos onto the same subnet, with a higher priority than the current DR, the new higher-priority router does *not* immediately take over as DR, but rather must wait until the DR and BDR fail.

Database Exchange and Becoming Fully Adjacent

Finally, neighboring routers can begin to exchange routing information with each other using Database Description packets, as well as LSA and LSU packets. First, a quick recap of the steps taken so far:

1. Each router initializes OSPF and picks its RID.
2. Routers discover each other as neighbors using Hello packets.

3. Routers reach a two-way communications state with a neighbor once they see their own RID in the Hello from that neighbor.
4. If the OSPF network type for the interface requires a DR, the DR is elected (as well as the BDR).

At this point, on an interface that does not use a DR, OSPF updates can be sent to all neighbors on that interface. These packets typically use unicast destination IP addresses of each neighbor to which the update is being sent.

On interfaces with an elected DR, the non-DR routers send updates to the DR and BDR, using the 224.0.0.6 multicast address as the destination. This special multicast address means "all OSPF DRs", which means that the DR, and the BDR, will be listening for the packets. Then, the DR relays the updates to all OSPF routers on the subnet, using a destination IP address of 224.0.0.5. Note that the BDR listens for and receives the updates, so it is ready to take over for the DR, but the BDR does not forward the updates to the non-DR routers.

The neighboring routers now exchange their entire topology database with their neighbor. Remember, it is a large amount of information in comparison to the sparse information sent in distance vector updates. Once a router has exchanged its entire link-state database with a neighbor, it transitions into a state called the *Full* state. In normal working operation, the **show ip ospf neighbor** command should list one or more neighbors in a Full state, indicating the expected final resting state of an OSPF neighbor.

NOTE OSPF considers a neighbor in the Full state to be *fully adjacent*. So, while a router might have several neighbors on an interface, only some of them might become *fully adjacent*.

Keep in mind that the Full state is for neighbors with which Link State Updates have been exchanged. On LANs, for instance, non-DR routers never exchange updates with each other, but they do exchange updates with the DR and BDR. As a result, a non-DR router will end up in a Full state with the DR and BDR, and a two-way state with other non-DRs. This concept is shown Example 6-4 later in the chapter.

Once a router has received a full database exchange from a neighbor, it can run the SPF algorithm and update the routing table.

Steady-State Operation

Hello packets serve the same purpose as timed, regular full routing updates serve for distance vector protocols. With distance vector protocols, when a router fails to hear routing updates from a neighbor for some multiple of the update interval, the router believes the silent router has failed. The router then marks all routes it learned from the now-silent router as having an infinite metric.

Similarly, with OSPF, when a router fails to hear Hellos from a neighbor for an interval called the *dead interval*, the router believes the silent router has failed. The dead interval defaults to four times the Hello interval. For instance, on Ethernet interfaces, Cisco routers default to a Hello interval of 10 seconds and a dead interval of 40 seconds. OSPF keeps working until the dead interval expires; after that, the router marks the now-silent router as "down" in its neighbor table. Then the router that stopped receiving the Hellos runs Dijkstra to calculate new routes, based on the fact that one of the network's routers is now out of service. Also, the router floods topology updates to its neighbors to let them know about the failure, with the other routers also running the Dijkstra algorithm again to compute new routes.

Loop Avoidance

The SPF algorithm prevents loops as a natural part of the processing of the topology database with the SPF algorithm. Unlike distance vector protocols, link-state protocols do not need loop-avoidance features such as split horizon, poison reverse, and hold-down timers.

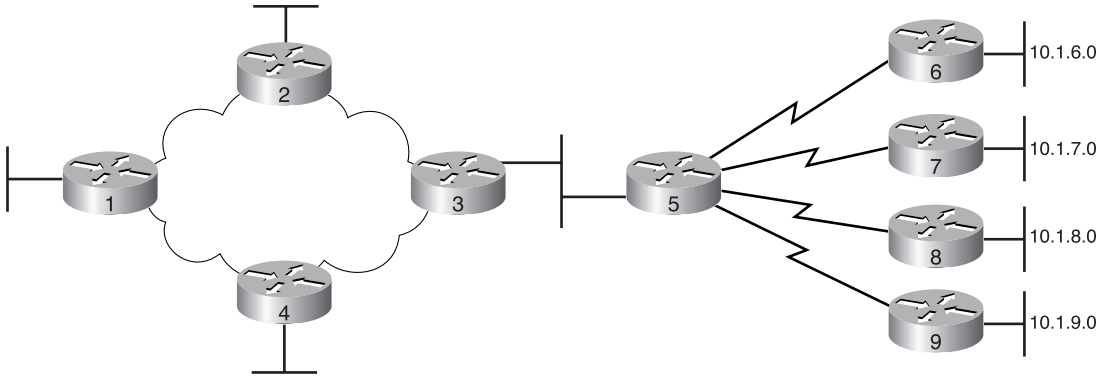
Link-state protocols rely on the rapid dissemination of information about failed routers and subnets to prevent loops. Therefore, when a link or router fails, a router noticing the failure immediately floods the new router or link status to its neighbors, with those routers forwarding the updated status to their neighbors, eventually flooding the new status information to all the routers in the network. (In a way, this feature works like "triggered updates" for distance vector protocols, but this behavior is just a feature of link-state protocols and does not have a specific name.)

Interestingly, the convergence time of most distance vector protocols consists of the time taken by the loop-avoidance features. For instance, the hold-down timer alone accounts for several minutes of convergence time. With link-state protocols, none of the time-consuming loop-avoidance features are needed, which means that link-state protocols can converge very quickly. With proper design, OSPF can converge as quickly as 5 seconds after a router notices a failure in most cases.

Scaling OSPF Through Hierarchical Design

OSPF can be used in some networks with very little thought as to design issues. You just turn on OSPF in all the routers, and it works! However, in large networks, engineers need to think about and plan how to use several OSPF features that allow it to scale well in larger networks. To appreciate the issues behind OSPF scalability, and the need for good design to allow scalability, examine Figure 6-4.

Figure 6-4 *Single-Area OSPF*



In the network shown in Figure 6-4, the topology database on all nine routers is the same full topology that matches the figure. With a network that size, you can just enable OSPF, and it works fine. But imagine a network with 900 routers instead of only nine, and several thousand subnets. In that size of network, OSPF convergence time might be slow, and the routers might experience memory shortages and processor overloads. The problems can be summarized as follows:

- A larger topology database requires more memory on each router.
- Processing the larger-topology database with the SPF algorithm requires processing power that grows exponentially with the size of the topology database.
- A single interface status change (up to down or down to up) forces every router to run SPF again!

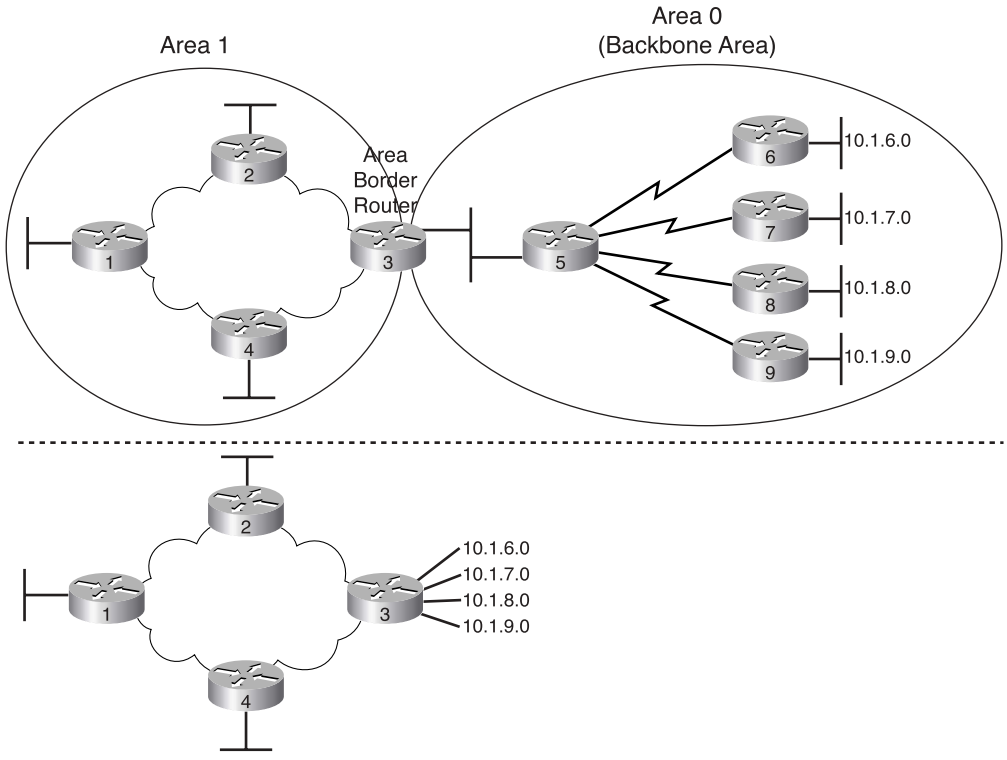
Although there is no exact definition of "large" in this context, in networks with at least 50 routers and at least a few hundred subnets, engineers should use OSPF scalability features to reduce the problems just described. These numbers are gross generalizations. They depend largely on the network design, models or routers, and so on.

OSPF Areas

Using OSPF areas solves many, but not all, of the most common problems with running OSPF in larger networks. OSPF areas break up the network so that routers in one area know less topology information about the subnets in the other area and they do not know about the routers in the other area at all. With smaller-topology databases, routers consume less memory and take less processing time to run SPF.

Figure 6-5 shows the same network as Figure 6-4, but with two OSPF areas, labeled Area 1 and Area 0.

Figure 6-5 *Two-Area OSPF*



The same topology is shown in the upper part of the figure, but the lower part of the figure shows the topology database on Routers 1, 2, and 4. By placing part of the network in another area, the routers inside Area 1 are shielded from some of the details. Router 3 is known as an OSPF Area Border Router (ABR), because it is on the border between two different areas. Router 3 does not advertise full topology information about the part of the network in Area 0 to Routers 1, 2, and 4. Instead, Router 3 advertises summary information about the subnets in Area 0, effectively making Routers 1, 2, and 4 think the topology looks like the lower part of Figure 6-5. Therefore, Routers 1, 2, and 4 view the world as if it has fewer routers. As a result, the SPF algorithm takes less time, and the topology database takes less memory.

It is very important to note that the summarized information shown in Figure 6-5 does not change the number of subnets known inside Area 1 or Area 0. The summarized information just means that a router inside one area receives routing updates that use fewer bytes, thereby decreasing the amount of memory needed to store the information. Later you will learn about a feature called *route summarization*, in which the number of subnets advertised into another area is reduced as well. The terms are similar, and both happen in Area Border Routers, but the concepts are different.

NOTE Although the perspectives of the routers in Area 1 are shown in Figure 6-5, the same thing happens in reverse—routers in Area 0 do not know the details of Area 1's topology.

Using areas improves all three of the scaling problems that were stated earlier. By making the topology databases smaller, the databases take less memory. With smaller databases, the SPF algorithm takes less time and converges more quickly. Also, although it is not obvious, when links in Area 0 change state, the routers that are totally in Area 1 do not need to run SPF again. So, with areas, many fewer SPF calculations are required in the network's routers.

Notice that the dividing line between areas is not a link, but a router. In Figure 6-5, Router 3 is in both area 1 and Area 0. OSPF uses the term Area Border Router (ABR) to describe a router that sits in both areas. An ABR has the topology database for both areas and runs SPF when links change status in either area. Using areas does not actually reduce memory requirements or the number of SPF calculations for ABRs like Router 3.

Stub Areas

OSPF includes other features to improve how it works in larger networks. The CCNP routing test expects you to know the topics covered in Cisco's Building Scalable Cisco Internetworks (BSCI) course, and you might guess from the course's name that OSPF scalability features are covered in depth. You can also refer to Tom Thomas's *OSPF Network Design Solutions* from Cisco Press, or search on "OSPF Design Guide" at Cisco.com for a great reference document.

When you move on to the CCNP certification, you should pay particular attention to the topic of stub areas. OSPF allows you to define an area as a stub area; as a result, the size of the topology database for routers in that area can be reduced even further. OSPF allows for other variants of areas—called Totally Stubby and Not-So-Stubby areas—that affect the size of the topology database, which in turn affects how fast the SPF algorithm runs. A new OSPF area type, Totally Not-So-Stubby Area (TNSSA), provides yet another type of area. You should be aware of the various types when working with OSPF in a real network.

Summary: Comparing Link-State and OSPF to Distance Vector Protocols

Link-state protocols have a major advantage over distance vector protocols in how fast they converge and in how they prevent loops. With today's networks, a 3-minute wait for a distance vector routing protocol to converge typically is perceived as a network outage. A 10-second convergence time for OSPF might simply be perceived as an irritation. Also, link-state protocols easily prevent loops. In addition, OSPF is publicly defined in RFC 2328, so you can use routers from multiple vendors with some confidence that they will work reasonably well together.

Link-state protocols do have some drawbacks. The biggest negative relates to the planning and design effort that is required for larger networks. Depending on the network's physical topology, OSPF might or might not be a natural fit. For instance, OSPF defines area 0 as the "backbone" area. All nonbackbone areas must connect to each other through the backbone area only, making OSPF designs hierarchical. Many networks work well with a hierarchical OSPF design, but others do not. The other drawbacks are more obvious. Link-state protocols can consume memory and CPU to the point of impacting overall router performance, depending on the network and the OSPF design.

Table 6-2 summarizes some of the key points of comparison between the two types of routing protocols.

Table 6-2 *Comparing Link-State and Distance Vector Protocols*

Feature	Link-State	Distance Vector
Convergence Time	Fast	Slow, mainly because of loop-avoidance features
Loop Avoidance	Built into the protocol	Requires extra features such as split horizon
Memory and CPU Requirements	Can be large; good design can minimize	Low
Requires Design Effort for Larger Networks	Yes	No
Public Standard or Proprietary	OSPF is public	RIP is publicly defined; IGRP is not

Balanced Hybrid Routing Protocol and EIGRP Concepts

Cisco uses the term *balanced hybrid* to describe the category of routing protocols in which EIGRP resides. Cisco supports two distance vector IP routing protocols—RIP and IGRP. It also supports two link-state IP routing protocols—OSPF and Intermediate System-to-Intermediate System (IS-IS). Furthermore, Cisco supports a single balanced hybrid IP routing protocol—EIGRP.

Cisco uses the term balanced hybrid because EIGRP has some features that act like distance vector protocols and some that act like link-state protocols. Cisco also sometimes refers to EIGRP as an advanced distance vector protocol.

EIGRP Features and Comparison with IGRP

EIGRP is an enhanced version of IGRP, so some level of comparison with IGRP is useful. As it turns out, there are more differences than similarities between the two protocols. Table 6-3 summarizes these similarities and differences.

Table 6-3 *EIGRP and IGRP Similarities and Differences*

Similarities	Differences
Both are Cisco proprietary protocols.	EIGRP converges significantly faster than IGRP
They use the same logic for multiple equal-cost paths	EIGRP sends routing information once to a neighbor, and then only sends new or updated information; IGRP repeats the entire routing table every 90 seconds. Therefore, EIGRP has much less overhead than IGRP.
The metrics are practically identical, with EIGRP's formula for calculating the metric simply including a multiplier of 256.	EIGRP can be used to exchange routing information for Novell IPX and Apple Computer AppleTalk Layer 3 protocols, in addition to IP.

Internal processing details differ significantly as well, and will be described in the next few sections of this chapter.

EIGRP Processes and Tables

EIGRP follows three general steps to be able to add routes to the IP routing table. In its basic form, these three steps are similar to OSPF, but with large differences in the underlying detail. The steps are as follows:

1. EIGRP routers discover other EIGRP routers that are attached to the same subnet, and then the routers form a *neighbor relationship* with each other. Each router keeps a list of the neighbors in its *EIGRP neighbor table*.
2. EIGRP then exchanges network topology information with known neighbors, placing the information in the *EIGRP topology table*. (There is no requirement for a DR or BDR concept like OSPF.)
3. EIGRP analyzes the topology information, and puts the lowest-metric routes into the IP routing table.

As a result of these three steps, EIGRP actually works with three tables:

- **The EIGRP neighbor table**—Viewed with the **show ip eigrp neighbor** command
- **The EIGRP topology table**—Viewed with the **show ip eigrp topology** command
- **The IP routing table**—Viewed with the **show ip route** or **show ip route eigrp** commands

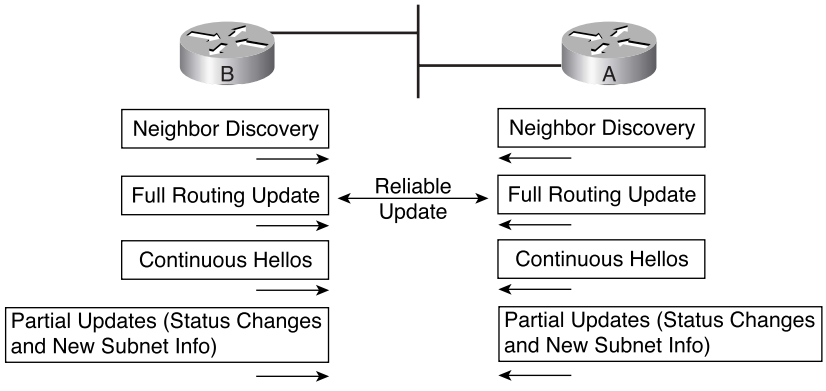
EIGRP on a single router could end up creating and updating nine tables due to its support of IP, IPX, and AppleTalk. If configured for all three Layer 3 protocols, EIGRP would have a neighbor table, topology table, and routing table for each of the three Layer 3 protocols. For instance, the **show ipx eigrp topology** command would display EIGRP's topology table used to store information about IPX network numbers, with different information than is shown with the **show ip eigrp topology** command.

The next few sections describe some details about how EIGRP forms neighbor relationships, exchanges routes, and adds entries to the IP routing table.

Neighbors and Sending Topology Information

Figure 6-6 shows the typical sequence used by two EIGRP routers that connect to the same subnet. They discover each other as neighbors, and they reliably exchange full routing information. The process is different from OSPF, but the same goal of reliably ensuring that all neighbors receive all routing information is achieved. EIGRP sends and receives EIGRP hello packets to ensure that the neighbor is still up and working—like OSPF, but with a different Hello packet than OSPF. When link status changes, or new subnets are discovered, reliable routing updates are sent, but only with the new information—again, like OSPF.

Figure 6-6 *Sequence of Events for EIGRP Exchange of Routing Information*



Hello messages are used to perform neighbor discovery, and are continually sent between neighbors to allow neighbors to notice when connectivity has failed. The *Hello interval* defines how often a router should send Hellos, and how often to expect to receive them. The Hello interval can be changed, but it is 5 seconds on LANs and point-to-point WAN links, and 60 seconds on multipoint WANs like Frame Relay, by default.

EIGRP uses EIGRP *update messages* to actually convey topology information to neighbors. These update messages can be sent to multicast IP address 224.0.0.10 if the sending router needs to update multiple routers on the same subnet; otherwise, the updates are sent to the unicast IP address of the particular neighbor. (Hello messages are always sent to the 224.0.0.10 multicast address.)

The update messages are sent using the *Reliable Transport Protocol (RTP).* The significance of RTP is that, like OSPF, EIGRP will resend routing updates that are lost in transit. By using RTP, EIGRP can better avoid loops.

Updating the Routing Table While Avoiding Loops

Loop avoidance poses one of the most difficult problems with any dynamic routing protocol. Distance vector protocols overcome this problem with a variety of tools, some of which create a large portion of the minutes-long convergence time after a link failure. Link-state protocols overcome this problem by having each router keep a full topology of the network, so that by running a rather involved mathematical model, a router can avoid any loops.

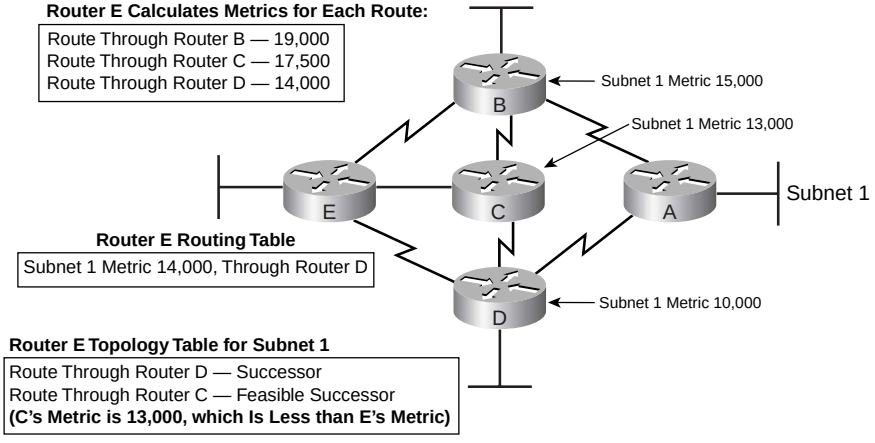
EIGRP avoids loops by keeping some basic topological information but not full information. When a router learns multiple routes to the same subnet, it puts the best route in the routing table. EIGRP keeps some topological information for the same reason as OSPF—so that it can very quickly converge and use a new route without causing a loop. EIGRP keeps its internal algorithms simple by using sparser topology information than OSPF; however, as a result, only some alternate routes can be easily and quickly used without causing loops, and some require more work. Failed routes that have an EIGRP *feasible successor* can be used immediately after the route fails. Failed routes without a feasible successor require EIGRP to use a *Query and Response* process to confirm that no loop exists before an alternate route can be used. Both processes result in fast convergence, typically quicker than 10 seconds, but the query and response process does take slightly longer.

EIGRP Successors and Feasible Successors

Of the other suboptimal routes, some may be used immediately if the currently-best route fails, without fear of having a loop occur. EIGRP runs a simple algorithm to identify which routes could be used immediately after a route failure, without causing a loop. EIGRP then keeps these loop-free backup routes in its topology table and uses them if the currently-best route fails.

Figure 6-7 illustrates how EIGRP figures out which routes can be used after a route fails without causing loops.

Figure 6-7 *Successors and Feasible Successors with EIGRP*



In the figure, Router E learns three routes to Subnet 1, from Routers B, C, and D. After calculating each route's metric based on bandwidth and delay information received in the routing update, Router E finds that the route through Router D has the lowest metric, so Router E adds that route to its routing table, as shown.

EIGRP builds a topology table that includes the currently-best route plus the alternative routes that would not cause loops if they were used when the currently-best route through Router D failed. EIGRP calls the best route (the route with the lowest metric) the *successor.* Any backup routes that could be used without causing a loop are called *feasible successors*. In Figure 6-7, the route through Router C would not cause a loop, so Router E lists the route through Router C as a feasible successor. Router E thinks that using the route through Router B could cause a loop, so that route is not listed as a feasible successor.

EIGRP decides if a route can be a feasible successor if the computed metric for that route on the neighbor is less than its own computed metric. When that neighbor has a lower metric for its route to the subnet in question, that route is said to have met the *feasibility condition*. For example, Router E computes a metric of 14,000 on its best route (through Router D). Router C s computed metric is lower than 14,000 (it s 13,000), so Router E believes that if the existing route failed, it could use the route through Router C and not cause a loop. As a result, Router E adds a route through Router C to the topology table as a feasible successor route. Conversely, Router B's computed metric is 15,000, which is larger than Router E's computed metric of 14,000, so Router E does not consider the route through Router B a feasible successor.

If the route to Subnet 1 through Router D fails, Router E can immediately put the route through Router C into the routing table, without fear of creating a loop. Convergence occurs almost instantly in this case.

The Query and Reply Process

When a route fails and the route has no feasible successor, EIGRP uses a distributed algorithm called Diffusing Update Algorithm (DUAL). DUAL sends queries looking for a loop-free route to the subnet in question. When the new route is found, DUAL adds it to the routing table.

The EIGRP DUAL process simply uses messages to confirm that a route exists, and would not create a loop, before deciding to replace a failed route with an alternate route. For instance, in Figure 6-7, imagine that both routers C and D fail. Router E does not have a feasible successor route to subnet 1, but there is an obvious physically-available path through Router B. In order to use the route, Router E sends EIGRP *query* messages to his working neighbors (in this case, Router B). Router B's route to subnet 1 is still working fine, so Router B replies to Router E with an EIGRP *reply* message, simply stating the details of the working route to subnet 1, and confirming that it is still viable. Router E can then add a new route to subnet 1 to its routing table, without fear of a loop.

Replacing a failed route with a feasible successor takes a very short amount of time, typically less than a second or two. When queries and replies are required, convergence can take slightly longer, but in most networks, convergence can still occur in less than 10 seconds.

EIGRP Summary

EIGRP converges quickly while avoiding loops. EIGRP does not have the same scaling issues as link-state protocols, so no extra design effort is required. EIGRP takes less memory and processing than link-state protocols.

EIGRP converges much more quickly than do distance vector protocols, mainly because EIGRP does not need the loop-avoidance features that slow down distance vector convergence. By sending only partial routing updates, after full routing information has been exchanged, EIGRP reduces overhead on the network.

The only significant disadvantage of EIGRP is that it is a Cisco-proprietary protocol. So, if you want to be prepared to use multiple vendors' routers in a network, you should probably choose an alternative routing protocol. Alternatively, you could use EIGRP in the Cisco

routers and OSPF in the others and perform a function called *route redistribution,* in which a router exchanges routes between the two routing protocols inside the router.

Table 6-4 summarizes some of the key comparison points between EIGRP, IGRP, and OSPF.

Table 6-4 *EIGRP Features Compared to OSPF and IGRP*

Feature	EIGRP	IGRP	OSPF
Discovers neighbors before exchanging routing information	Yes	No	Yes
Builds some form of topology table in addition to adding routes to the routing table	Yes	No	Yes
Converges quickly	Yes	No	Yes
Uses metrics based on bandwidth and delay by default	Yes*	Yes	No
Sends full routing information on every routing update cycle	No	Yes	No
Requires distance vector loop-avoidance features	No	Yes	No
Public standard	No	No	Yes
Uses DUAL Algorithm	Yes	No	No

*EIGRP uses the same metric as IGRP, except that EIGRP scales the metric by multiplying by 256.

OSPF Configuration

OSPF includes many configuration options as a result of its complexity. Tables 6-5 and 6-6 summarize the OSPF configuration and troubleshooting commands, respectively.

Table 6-5 *IP OSPF Configuration Commands*

Command	Configuration Mode
router ospf *process-id*	Global
network *ip-address wildcard-mask* **area** *area-id*	Router subcommand
ip ospf cost *interface-cost*	Sets the OSPF cost associated with the interface
bandwidth *bandwidth*	Sets the interface bandwidth, from which OSPF derives the cost based on the formula 10^8 / bandwidth
auto-cost reference bandwidth *number*	Router subcommand that tells OSPF the numerator in the formula used to calculate the OSPF cost based on the interface bandwidth

continues

Table 6-5 *IP OSPF Configuration Commands (Continued)*

Command	Configuration Mode
ip ospf hello *number*	Interface subcommand that sets the OSPF Hello interval, and also resets the Dead interval to 4 times this number
ip ospf network *type*	Interface subcommand that defines the OSPF network type

Table 6-6 *IP OSPF EXEC Commands*

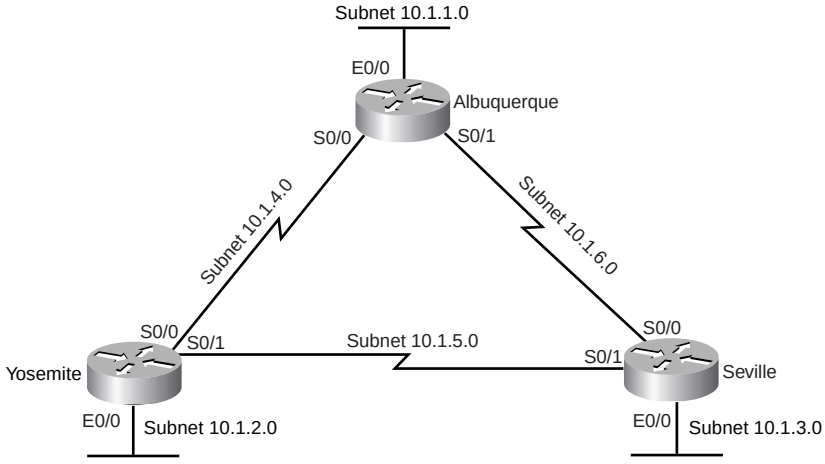
Command	Description
show ip route [*ip-address* [*mask*] [**longer-prefixes**]] | [*protocol* [*process-id*]]	Shows the entire routing table, or a subset if parameters are entered.
show ip protocols	Shows routing protocol parameters and current timer values.
show ip ospf interface	Lists the area in which the interface resides, and neighbors adjacent on this interface.
show ip ospf neighbor	Lists neighbors and current status with neighbors, per interface.
show ip route ospf	Lists routes in the routing table learned by OSPF.
debug ip ospf events	Issues log messages for each OSPF packet.
debug ip ospf packet	Issues log messages describing the contents of all OSPF packets.
debug ip ospf hello	Issues log messages describing Hellos and Hello failures.

This section includes two sample configurations using the same network diagram. The first example shows a configuration with a single OSPF area, and the second example shows multiple areas, along with some **show** commands.

OSPF Single-Area Configuration

When only a single area is used, OSPF configuration differs only slightly from RIP and IGRP configuration. The best way to describe the configuration, and the differences with the configuration of the other routing protocols, is to use an example. Figure 6-8 shows a sample network, and Example 6-1 shows the configuration on Albuquerque.

Figure 6-8 *Sample Network for OSPF Single-Area Configuration*



Example 6-1 *OSPF Single-Area Configuration on Albuquerque*

```
interface ethernet 0/0
ip address 10.1.1.1 255.255.255.0
interface serial 0/0
ip address 10.1.4.1 255.255.255.0
interface serial 0/1
ip address 10.1.6.1 255.255.255.0
!
router ospf 1
network 10.0.0.0 0.255.255.255 area 0
```

The configuration correctly enables OSPF on all three interfaces on Albuquerque. First, the **router ospf 1** global command puts the user in OSPF configuration mode. The **router ospf** command has a parameter called the OSPF *process-id*. In some instances, you might want to run multiple OSPF processes in a single router, so the **router** command uses the *process-id* to distinguish between the processes. Although the *process-id* used on the three routers is the same, the actual value is unimportant, and the numbers do not have to match on each router.

The **network** command tells Albuquerque to enable OSPF on all interfaces that match the **network** command and, on those interfaces, to place the interfaces into Area 0. The OSPF **network** command matches interfaces differently than does the **network** command for RIP and IGRP. The OSPF **network** command includes a parameter called the *wildcard mask*. The wildcard mask works just like the wildcard mask used with Cisco access control lists (ACLs), which are covered in more depth in Chapter 12, "IP Access Control List Security."

The wildcard mask represents a 32-bit number. When the mask has a binary 1 in one of the bit positions, that bit is considered a wildcard bit, meaning that the router should not care what binary value is in the corresponding numbers. For that reason, binary 1s in the wildcard mask are called *don't care bits*. If the wildcard mask bit is 0 in a bit position, the corresponding bits in the numbers being compared must match. You can think of these bits as the *do care bits*. So, the router must examine the two numbers and make sure that the values match for the bits that matter—in other words, the do care bits.

For instance, the wildcard mask in Example 6-1 is 0.255.255.255. When converted to binary, this number is 0000 0000 1111 1111 1111 1111 1111 1111—in other words, eight 0s and 24 1s. The **network** command tells Cisco IOS software to compare 10.0.0.0, which is the number in the **network** command, to the IP addresses of each interface on the router. The wildcard mask tells Cisco IOS software to compare only the first octet; the last three octets are wildcards, and anything matches. So, all three interface IP addresses are matched.

Example 6-2 shows an alternative configuration for Albuquerque that also enables OSPF on every interface. In this case, the IP address for each interface is matched with a different **network** command. The wildcard mask of 0.0.0.0 means that all 32 bits must be compared, and they must match—so the **network** commands include the specific IP address of each interface, respectively. Many people prefer this style of configuration in production networks, because it removes any ambiguity as to the interfaces on which OSPF is running.

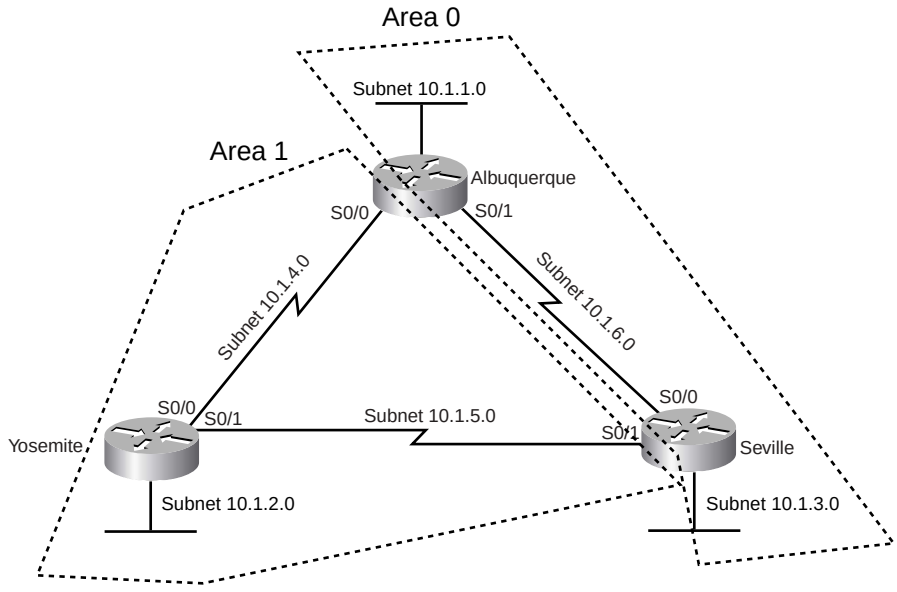
Example 6-2 *OSPF Single-Area Configuration on Albuquerque Using Three* **network** *Commands*

```
interface ethernet 0/0
ip address 10.1.1.1 255.255.255.0
interface serial 0/0
ip address 10.1.4.1 255.255.255.0
interface serial 0/1
ip address 10.1.6.1 255.255.255.0
!
router ospf 1
network 10.1.1.1 0.0.0.0 area 0
network 10.1.4.1 0.0.0.0 area 0
network 10.1.6.1 0.0.0.0 area 0
```

OSPF Configuration with Multiple Areas

Configuring OSPF with multiple areas is simple once you understand OSPF configuration in a single area. Designing the OSPF network by making good choices as to which subnets should be placed in which areas is the hard part! After the area design is complete, the configuration is easy. For instance, consider Figure 6-9, which shows some subnets in Area 0 and some in Area 1.

Figure 6-9 *Multiarea OSPF Network*



Multiple areas are not needed in such a small network, but two areas are used in this example to show the configuration. Note that Albuquerque and Seville are both ABRs, but Yosemite is totally inside area 1, so it is not an ABR.

Examples 6-3 and 6-4 show the configuration on Albuquerque and Yosemite, along with several **show** commands.

Example 6-3 *OSPF Multiarea Configuration and* **show** *Commands on Albuquerque*

```
!
! Only the OSPF configuration is shown to conserve space
!
router ospf 1
network 10.1.1.1 0.0.0.0 area 0
network 10.1.4.1 0.0.0.0 area 1
network 10.1.6.1 0.0.0.0 area 0
Albuquerque#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route
```

continues

Example 6-3 *OSPF Multiarea Configuration and* **show** *Commands on Albuquerque (Continued)*

```
Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 6 subnets
O       10.1.3.0 [110/65] via 10.1.6.3, 00:01:04, Serial0/1
O       10.1.2.0 [110/65] via 10.1.4.2, 00:00:39, Serial0/0
C       10.1.1.0 is directly connected, Ethernet0/0
C       10.1.6.0 is directly connected, Serial0/1
O       10.1.5.0 [110/128] via 10.1.4.2, 00:00:39, Serial0/0
C       10.1.4.0 is directly connected, Serial0/0

Albuquerque#show ip route ospf
     10.0.0.0/24 is subnetted, 6 subnets
O       10.1.3.0 [110/65] via 10.1.6.3, 00:01:08, Serial0/1
O       10.1.2.0 [110/65] via 10.1.4.2, 00:00:43, Serial0/0
O       10.1.5.0 [110/128] via 10.1.4.2, 00:00:43, Serial0/0
Albuquerque#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
10.1.6.3          1   FULL/  -        00:00:35    10.1.6.3        Serial0/1
10.1.5.2          1   FULL/  -        00:00:37    10.1.4.2        Serial0/0
Albuquerque#show ip ospf interface
Serial0/1 is up, line protocol is up
  Internet Address 10.1.6.1/24, Area 0
  Process ID 1, Router ID 10.1.6.1, 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
  Index 2/3, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 2, maximum is 2
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 10.1.6.3
  Suppress hello for 0 neighbor(s)
Ethernet0/0 is up, line protocol is up
  Internet Address 10.1.1.1/24, Area 0
  Process ID 1, Router ID 10.1.6.1, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 10.1.6.1, Interface address 10.1.1.1
  No backup designated router on this network
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:08
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 0, maximum is 0
  Last flood scan time is 0 msec, maximum is 0 msec
```

Example 6-3 *OSPF Multiarea Configuration and* **show** *Commands on Albuquerque (Continued)*

```
  Neighbor Count is 0, Adjacent neighbor count is 0
  Suppress hello for 0 neighbor(s)
Serial0/0 is up, line protocol is up
  Internet Address 10.1.4.1/24, Area 1
  Process ID 1, Router ID 10.1.6.1, 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 00:00:01
  Index 1/2, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 10.1.5.2
  Suppress hello for 0 neighbor(s)
```

Example 6-4 *OSPF Multiarea Configuration and* **show** *Commands on Yosemite*

```
!
! Only the OSPF configuration is shown to conserve space
!
router ospf 1
network 10.0.0.0 255.255.255.255 area 1
Yosemite#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 6 subnets
IA      10.1.3.0 [110/65] via 10.1.5.1, 00:00:54, Serial0/1
IA      10.1.1.0 [110/65] via 10.1.4.1, 00:00:49, Serial0/0
C       10.1.2.0 is directly connected, Ethernet0/0
C       10.1.5.0 is directly connected, Serial0/1
IA      10.1.6.0 [110/128] via 10.1.4.1, 00:00:38, Serial0/0
C       10.1.4.0 is directly connected, Serial0/0
```

The configuration only needs to show the correct area number on the **network** command matching the appropriate interfaces. For instance, the **network 10.1.4.1 0.0.0.0 area 1** command matches Albuquerque's Serial 0/0 interface IP address, placing that interface in Area 1. The **network 10.1.6.1 0.0.0.0 area 0** and **network 10.1.1.1 0.0.0.0 area 0** commands

place Serial 0/1 and Ethernet 0/0, respectively, in Area 0. Unlike Example 6-1, Albuquerque cannot be configured to match all three interfaces with a single **network** command, because one interface (Serial 0/0) is in a different area than the other two interfaces.

The **show ip route ospf** command just lists OSPF-learned routes, as opposed to the entire IP routing table. The **show ip route** command lists all three connected routes, as well as the three OSPF learned routes. Note that Albuquerque's route to 10.1.2.0 has the **O** designation beside it, meaning *intra-area*, because that subnet resides in Area 1, and Albuquerque is part of Area 1 and Area 0.

In Example 6-4, notice that the OSPF configuration in Yosemite requires only a single **network** command. Because all interfaces in Yosemite are in Area 1, and all three interfaces are in network 10.0.0.0, the command can just match all IP addresses in network 10.0.0.0 and put them in Area 1. Also note that the routes learned by Yosemite from the other two routers show up as *interarea (IA) routes*, because those subnets are in Area 0, and Yosemite is in Area 1.

The OSPF topology database includes information about routers and the subnets, or links, to which they are attached. To identify the routers in the neighbor table's topology database, OSPF uses a router ID (RID) for each router. A router's OSPF RID is that router's highest IP address on a physical interface when OSPF starts running. Alternatively, if a loopback interface has been configured, OSPF uses the highest IP address on a loopback interface for the RID, even if that IP address is lower than some physical interface's IP address. Also, you can set the OSPF RID using the **router-id** command in router configuration mode.

NOTE If you're not familiar with it, a loopback interface is a special virtual interface in a Cisco router. If you create a loopback interface using the **interface loopback** *x* command, where *x* is a number, that loopback interface is up and operational as long as the router IOS is up and working. You can assign an IP address to a loopback interface, you can ping the address, and you can use it for several purposes—including having a loopback interface IP address as the OSPF router ID.

Many commands refer to the OSPF RID, including the **show ip ospf neighbor** command. This command lists all the neighbors, using the neighbors' RIDs to identify them. For instance, in Example 6-3, the first neighbor in the output of the **show ip ospf neighbor** command lists **Router ID 10.1.5.2**, which is Yosemite's RID.

Finally, the **show ip ospf interface** command lists more-detailed information about OSPF operation on each interface. For instance, this command lists the area number, OSPF cost, and any neighbors known on each interface. The timers used on the interface, including the Hello and dead timer, are also listed.

OSPF uses the cost to determine the metric for each route. You can set the cost value on an interface using the **ip ospf cost** *x* interface subcommand. You can also set the OSPF cost of an interface using the **bandwidth** interface subcommand. If you do not set an interface's cost, IOS defaults to use the formula 10^8 / *bandwidth*, where *bandwidth* is the interface's bandwidth. For instance, Cisco IOS software defaults to a bandwidth of 10,000 (the unit in the **bandwidth** command is kbps, so 10,000 means 10 Mbps) on Ethernet interfaces, so the default cost is 10^8 / 10^7, or 10. Higher-speed serial interfaces default to a bandwidth of 1544, giving a default cost of 10^8 / 1,544,000, which is rounded to 64, as shown in the example. If you change the interface's bandwidth, you change the OSPF cost as well.

You might have noticed that the cost for a Fast Ethernet interface (100 Mbps) is calculated as a cost of 1. For Gigabit interfaces (1000 Mbps), the calculation yields .1, but only integer values can be used, so OSPF uses a cost of 1 for Gigabit interfaces as well. So Cisco lets you change the *reference bandwidth*, which is the value in the numerator of the calculation in the preceding paragraph. For instance, using the router OSPF subcommand **auto-cost reference-bandwidth 1000**, you change the numerator to 1000 Mbps, or 10^9. The calculated cost on a Gigabit interface is then 1, and the cost on a Fast Ethernet is 10.

OSPF Troubleshooting

This section contains two examples, one showing a problem with forming neighbor relationships due to a misconfigured Hello interval, and another showing some information about the DR election process on a LAN. Figure 6-10 depicts the network used for both examples.

Figure 6-10 *LAN with Four Routers, Used in OSPF Troubleshooting Examples*

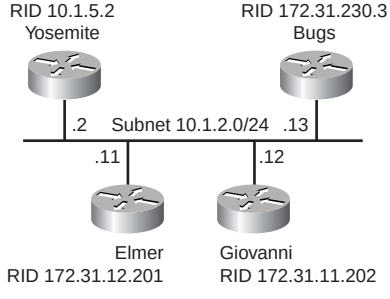


Figure 6-10 shows four routers that will become neighbors in the next two examples. The examples show the network from the perspective of the Yosemite router. Both examples list several commands, with the comments about the commands interspersed in comment lines inside the example.

Example 6-5 lists information in the following sequence:

1. The **show ip ospf neighbor** command confirms that Yosemite has three neighbors on its fa0/0 interface.
2. The **show ip ospf interface fa 0/0** command confirms that the current Hello interval is 10, and the current dead interval is 40.
3. The Hello interval on interface fa0/0 is changed to 4.
4. Yosemite's fa0/0 interface is brought down and back up in order to force Yosemite to attempt to form new neighbor relationships.
5. The **debug ip ospf hello** output shows why Yosemite cannot find any neighbors on that interface anymore.

Example 6-5 *Hello Problems as a Result of Mis-Matched Hello Intervals*

```
Yosemite#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
172.31.11.202     1   FULL/DROTHER    00:00:33    10.1.2.12       FastEthernet0/0
172.31.12.201     1   FULL/BDR        00:00:31    10.1.2.11       FastEthernet0/0
172.31.230.3      1   FULL/DROTHER    00:00:36    10.1.2.13       FastEthernet0/0
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Above, Yosemite has three neighbors over interface Fa0/0.
! Below, Yosemite's Fa0/0 interface has default Hello and Dead
! Intervals of 10 and 40 seconds, respectively.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Yosemite#show ip ospf interface fa 0/0
FastEthernet0/0 is up, line protocol is up
  Internet Address 10.1.2.2/24, Area 0
  Process ID 2, Router ID 10.1.5.2, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State BDR, Priority 255
  Designated Router (ID) 172.31.230.3, Interface address 10.1.2.13
  Backup Designated router (ID) 10.1.5.2, Interface address 10.1.2.2
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:00
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 0, maximum is 1
  Last flood scan time is 0 msec, maximum is 4 msec
  Neighbor Count is 3, Adjacent neighbor count is 3
    Adjacent with neighbor 172.31.11.202
    Adjacent with neighbor 172.31.12.201
    Adjacent with neighbor 172.31.230.3  (Designated Router)
  Suppress hello for 0 neighbor(s)
```

Example 6-5 *Hello Problems as a Result of Mis-Matched Hello Intervals (Continued)*

```
Yosemite#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Yosemite(config)#interface fastethernet 0/0
Yosemite(config-if)#ip ospf hello 4
Yosemite(config-if)#^Z

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Above, Yosemite's fa0/0 is changed to have a Hello interval of 4.
! Below, the Hello interval has changed, and the Dead interval has
! been reset to 4 times the new Hello interval.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Yosemite#show ip ospf interface fast 0/0
FastEthernet0/0 is up, line protocol is up
  Internet Address 10.1.2.2/24, Area 0
  Process ID 2, Router ID 10.1.5.2, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State BDR, Priority 255
  Designated Router (ID) 172.31.230.3, Interface address 10.1.2.13
  Backup Designated router (ID) 10.1.5.2, Interface address 10.1.2.2
  Timer intervals configured, Hello 4, Dead 16, Wait 16, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:01
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 0, maximum is 1
  Last flood scan time is 0 msec, maximum is 4 msec
  Neighbor Count is 3, Adjacent neighbor count is 3
    Adjacent with neighbor 172.31.11.202
    Adjacent with neighbor 172.31.12.201
    Adjacent with neighbor 172.31.230.3  (Designated Router)
  Suppress hello for 0 neighbor(s)
Yosemite#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Yosemite(config)#interface fast 0/0
Yosemite(config-if)#shutdown
Yosemite(config-if)#no shutdown
Yosemite(config-if)#^Z
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Above, Yosemite's fa0/0 interface is brought down and back up.
! Below, Yosemite does not form any neighbor relationships; there
! were three neighbors listed before this change. Debug ip ospf hello
! at the end of the example shows a message stating mismatched Hello
! parameters.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Yosemite#show ip ospf neighbor

Yosemite#debug ip ospf hello
OSPF hello events debugging is on
```

continues

Example 6-5 *Hello Problems as a Result of Mis-Matched Hello Intervals (Continued)*

```
Yosemite#
Mar  1 09:48:42.000: OSPF: Rcv hello from 172.31.230.3 area 0 from FastEthernet0/0 10.1.2.13
Mar  1 09:48:42.000: OSPF: Mismatched hello parameters from 10.1.2.13
Mar  1 09:48:42.000: OSPF: Dead R 40 C 16, Hello R 10 C 4  Mask R 255.255.255.0 C 255.255.255.0
Mar  1 09:48:43.034: OSPF: Send hello to 224.0.0.5 area 0 on FastEthernet0/0 from 10.1.2.2
```

Example 6-6 shows an example proving that a when a new router is added to a LAN, even with a higher OSPF priority, it does not take over for the existing DR or BDR. Example 6-6's sequence is as follows:

1. The **show ip ospf neighbor** and **show ip ospf interface** command confirm that Yosemite is currently not the DR or BDR.
2. Yosemite's Fa0/0 OSPF priority is changed to 255.
3. The next couple **show** commands reconfirm that Yosemite is not the current DR or BDR.
4. Yosemite's fa0/0 is brought down, and back up, forcing Yosemite to re-discover neighbors and re-attempt to become DR.
5. **show** commands confirm that Yosemite still did not become the DR or BDR.

Example 6-6 *Yosemite Router with Priority 255 Does Not Pre-empt Existing DR*

```
Yosemite#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
172.31.11.202     1   2WAY/DROTHER    00:00:32    10.1.2.12       FastEthernet0/0
172.31.12.201     1   FULL/DR         00:00:30    10.1.2.11       FastEthernet0/0
172.31.230.3      1   FULL/BDR        00:00:35    10.1.2.13       FastEthernet0/0
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Above, Yosemite has three neighbors on interface fa0/0. Each line
! shows Yosemite's status with each neighbor (2way or full), and what
! kind of router the neighbor is. For instance, the second neighbor
! is the DR, and the third neighbor listed in the BDR.
! Below, Yosemite is confirmed as a non-DR (shown as "DROTHER"), with
! the actual DR and BDR listed. Note that Yomseite's RID is shown as
! well. Also note that the network type is "BROADCAST", which implies
! that a DR and BDR must be elected.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Yosemite#show ip ospf interface fast 0/0
FastEthernet0/0 is up, line protocol is up
  Internet Address 10.1.2.2/24, Area 0
  Process ID 2, Router ID 10.1.5.2, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State DROTHER, Priority 255
  Designated Router (ID) 172.31.12.201, Interface address 10.1.2.11
```

Example 6-6 *Yosemite Router with Priority 255 Does Not Pre-empt Existing DR (Continued)*

```
  Backup Designated router (ID) 172.31.230.3, Interface address 10.1.2.13
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:00
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
Last flood scan time is 0 msec, maximum is 4 msec
  Neighbor Count is 3, Adjacent neighbor count is 2
    Adjacent with neighbor 172.31.12.201  (Designated Router)
    Adjacent with neighbor 172.31.230.3   (Backup Designated Router)
  Suppress hello for 0 neighbor(s)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Above, note that the last two highlighted lines show Yosemite
! adjacent with the DR and BDR, but not the "DROTHER" neighbor.
! Yosemite is not a DR or BDR (it is a "DROTHER"), and non-DR
! routers on a LAN do not become fully adjacent with other non-DR
! routers.
! Below, Yosemite is brought down and back up, but the DR and BDR
! do not change, according to the show ip ospf interface command.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Yosemite#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Yosemite(config)#interface fast 0/0
Yosemite(config-if)#shutdown
Yosemite(config-if)#no shutdown
Yosemite(config-if)#^Z
Yosemite#show ip ospf interface fast 0/0
FastEthernet0/0 is up, line protocol is up
  Internet Address 10.1.2.2/24, Area 0
  Process ID 2, Router ID 10.1.5.2, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State DROTHER, Priority 255
  Designated Router (ID) 172.31.12.201, Interface address 10.1.2.11
  Backup Designated router (ID) 172.31.230.3, Interface address 10.1.2.13
! The rest of the lines are omitted for brevity
```

EIGRP Configuration

If you remember how to configure IGRP, EIGRP configuration is painless. You configure EIGRP exactly like IGRP, except that you use the **eigrp** keyword instead of the **igrp** keyword in the **router** command.

Tables 6-7 and 6-8 summarize the EIGRP configuration and troubleshooting commands, respectively. After these tables, you will see a short demonstration of EIGRP configuration, along with **show** command output.

Table 6-7 *IP EIGRP Configuration Commands*

Command	Configuration Mode
router eigrp *autonomous-system*	Global
network *network-number* [*network-mask*]	Router subcommand
network *network-number*	Router subcommand
maximum-paths *number-paths*	Router subcommand
variance *multiplier*	Router subcommand
traffic-share {*balanced* | *min*}	Router subcommand

Table 6-8 *IP EIGRP EXEC Commands*

Command	Description
show ip route [*ip-address* [*mask*] [**longer-prefixes**]] | [*protocol* [*process-id*]]	Shows the entire routing table, or a subset if parameters are entered.
show ip protocols	Shows routing protocol parameters and current timer values.
show ip eigrp neighbors	Lists EIGRP neighbors and status.
show ip eigrp topology	Lists the contents of the EIGRP topology table, including successors and feasible successors.
show ip route eigrp	Lists only EIGRP-learned routes from the routing table.
show ip eigrp traffic	Lists traffic statistics about EIGRP.

Example 6-5 shows a sample EIGRP configuration, along with **show** commands, on Albuquerque, in the same network used in the OSPF examples (Figure 6-7). The EIGRP configuration required on Yosemite and Seville matches the two lines of EIGRP configuration on Albuquerque.

Example 6-7 *Sample Router Configuration with EIGRP Partially Enabled*

```
router eigrp 1
network 10.0.0.0
Albuquerque#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
```

Example 6-7 *Sample Router Configuration with EIGRP Partially Enabled (Continued)*

```
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 6 subnets
D       10.1.3.0 [90/2172416] via 10.1.6.3, 00:00:43, Serial0/1
D       10.1.2.0 [90/2172416] via 10.1.4.2, 00:00:43, Serial0/0
C       10.1.1.0 is directly connected, Ethernet0/0
C       10.1.6.0 is directly connected, Serial0/1
D       10.1.5.0 [90/2681856] via 10.1.6.3, 00:00:45, Serial0/1
                 [90/2681856] via 10.1.4.2, 00:00:45, Serial0/0
C       10.1.4.0 is directly connected, Serial0/0

Albuquerque#show ip route eigrp
     10.0.0.0/24 is subnetted, 6 subnets
D       10.1.3.0 [90/2172416] via 10.1.6.3, 00:00:47, Serial0/1
D       10.1.2.0 [90/2172416] via 10.1.4.2, 00:00:47, Serial0/0
D       10.1.5.0 [90/2681856] via 10.1.6.3, 00:00:49, Serial0/1
                 [90/2681856] via 10.1.4.2, 00:00:49, Serial0/0

Albuquerque#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address                 Interface   Hold Uptime   SRTT   RTO  Q  Seq Type
                                        (sec)         (ms)       Cnt Num
0   10.1.4.2                Se0/0         11 00:00:54   32   200  0  4
1   10.1.6.3                Se0/1         12 00:10:36   20   200  0  24

Albuquerque#show ip eigrp interfaces
IP-EIGRP interfaces for process 1

                    Xmit Queue   Mean   Pacing Time   Multicast    Pending
Interface    Peers  Un/Reliable  SRTT   Un/Reliable   Flow Timer   Routes
Et0/0            0        0/0       0       0/10           0           0
Se0/0            1        0/0      32       0/15          50           0
Se0/1            1        0/0      20       0/15          95           0
```

For EIGRP configuration, all three routers must use the same AS number on the **router eigrp** command. For instance, they all use **router eigrp 1** in this example. The actual number used doesn't really matter, as long as it is the same number on all 3 routers. (The range of valid AS numbers is 1 through 65,535, as is the range of valid Process IDs with the **router ospf** command). The **network 10.0.0.0** command enables EIGRP on all interfaces whose IP addresses are in network 10.0.0.0, which includes all three interfaces on Albuquerque. With the identical two EIGRP configuration statements on the other two routers, EIGRP is enabled on all three interfaces on those routers as well, because those interfaces are also in network 10.0.0.0.

The **show ip route** and **show ip route eigrp** commands both list the EIGRP-learned routes with a **D** beside them. **D** signifies EIGRP. The letter **E** was already being used for Exterior Gateway Protocol (EGP) when Cisco created EIGRP, so it chose the next-closest letter to denote EIGRP-learned routes.

You can see information about EIGRP neighbors with the **show ip eigrp neighbors** command, and the number of active neighbors (called peers in the command output) with the **show ip eigrp interfaces** command, as seen in the last part of the example. These commands also provide some insight into EIGRP's underlying processes, like the use of RTP for reliable transmission. For instance, the **show ip eigrp neighbors** command lists a "Q Cnt" (short for Queue Count) column, listing the number of packets either waiting to be sent to a neighbor, or packets that have been sent but for which no acknowledgement has been received. The **show ip eigrp interfaces** command lists similar information in the "Xmit Queue Un/Reliable" column, which separates statistics for EIGRP messages that are sent with RTP (reliable) or without (unreliable).

Example 6-8 shows an alternative style of **network** command for EIGRP configuration, along with a detailed look at **show** and **debug** commands related to the feasible successor concept. The beginning of the example shows how to configure EIGRP **network** commands using a wildcard mask, with the same meaning as the wildcard mask used with the OSPF style of **network** command. In the upcoming example, three **network** commands are used on Albuquerque, one matching each of the three interfaces.

After that, Example 6-8 goes on to make the following points (in sequence) about Albuquerque's EIGRP status:

1. Albuquerque is shown to have a single IP route to 10.1.3.0/24, and two equal-cost routes to 10.1.5.0/24.
2. Next, the **show ip eigrp topology** shows information correlating to the ip routing table, with 1 successor route to 10.1.3.0/24, and two successor routes to 10.1.5.0/24.
3. After changing Yosemite's configuration to a higher bandwidth, the results from Albuquerque's **show ip eigrp topology** command reveal a feasible successor route to subnet 10.1.3.0.
4. Finally, Albuquerque's current route to 10.1.3.0 will fail, with **debug** messages showing Albuquerque's fail over to use the feasible successor route.

Example 6-8, with annotations, follows. The details can be a little tricky, but it does shed some light on the internal workings of EIGRP.

Example 6-8 *Using Wildcard Masks with EIGRP Configuration, and Feasible Successor Examination*

```
Albuquerque#router eigrp 1
Albuquerque(config-router)#network 10.1.1.0 0.0.0.255
Albuquerque(config-router)#network 10.1.4.0 0.0.0.255
Albuquerque(config-router)#network 10.1.6.0 0.0.0.255
Albuquerque(config-router)#^z
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Above, EIGRP has been changed to use the wildcard mask option. Note
! that a separate network command was used to match each of the three
! interfaces.
! Below, note the single route to subnet 10.1.3.0, and the two
! equal-metric routes to 10.1.5.0.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Albuquerque#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 6 subnets
D       10.1.3.0 [90/2172416] via 10.1.6.3, 00:00:57, Serial0/1
D       10.1.2.0 [90/2172416] via 10.1.4.2, 00:00:57, Serial0/0
C       10.1.1.0 is directly connected, Ethernet0/0
C       10.1.6.0 is directly connected, Serial0/1
D       10.1.5.0 [90/2681856] via 10.1.4.2, 00:00:57, Serial0/0
                 [90/2681856] via 10.1.6.3, 00:00:57, Serial0/1
C       10.1.4.0 is directly connected, Serial0/0
Albuquerque#show ip eigrp topology
IP-EIGRP Topology Table for AS(1)/ID(10.1.6.1)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.1.3.0/24, 1 successors, FD is 2172416
        via 10.1.6.3 (2172416/28160), Serial0/1
P 10.1.2.0/24, 1 successors, FD is 2172416
        via 10.1.4.2 (2172416/28160), Serial0/0
P 10.1.1.0/24, 1 successors, FD is 281600
        via Connected, Ethernet0/0
P 10.1.6.0/24, 1 successors, FD is 2169856
        via Connected, Serial0/1
P 10.1.5.0/24, 2 successors, FD is 2681856
        via 10.1.4.2 (2681856/2169856), Serial0/0
```

continues

Example 6-8 *Using Wildcard Masks with EIGRP Configuration, and Feasible Successor Examination (Continued)*

```
         via 10.1.6.3 (2681856/2169856), Serial0/1
P 10.1.4.0/24, 1 successors, FD is 2169856
         via Connected, Serial0/0
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Above, the EIGRP topology table shows one successor for the route
! to 10.1.3.0, and two successors for 10.1.5.0, reconfirming that
! EIGRP installs successor routes (not feasible successor routes)
! into the IP routing table.
! Below, the bandwidth of Yosemite's link to Seville (Yosemite's
! S0/1 interface) is changed from 1544 to 2000, which lowers
! Yosemite's metric to 10.1.3.0.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Yosemite(config-if)#bandwidth 2000
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Below, back in Albuquerque, the EIGRP topology table shows a single
! successor route for 10.1.3.0, but two entries listed - the new
! entry is a feasible successor route. The new entry shows a route
! to 10.1.3.0 through 10.1.4.2 (which is Yosemite). See the text
! following this example for more detail.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Albuquerque#show ip eigrp topology
IP-EIGRP Topology Table for AS(1)/ID(10.1.6.1)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.1.3.0/24, 1 successors, FD is 2172416
         via 10.1.6.3 (2172416/28160), Serial0/1
         via 10.1.4.2 (2684416/1794560), Serial0/0
! the rest of the lines omitted for brevity
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! The next command is on Yosemite!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Yosemite#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, 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, 5 subnets
D       10.1.3.0 [90/1794560] via 10.1.5.3, 00:40:14, Serial0/1
C       10.1.2.0 is directly connected, FastEthernet0/0
D       10.1.1.0 [90/2195456] via 10.1.4.1, 00:42:19, Serial0/0
```

Example 6-8 *Using Wildcard Masks with EIGRP Configuration, and Feasible Successor Examination (Continued)*

```
C       10.1.5.0 is directly connected, Serial0/1
C       10.1.4.0 is directly connected, Serial0/0
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Above, Yosemite's route to 10.1.3.0 lists a metric of 1794560.
! Refer to the notes at the end of the example for more information.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! The next command is on Albuquerque!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Below, debug eigrp fsm is enabled, and then Seville's link to
! Albuquerque (Seville's S0/0 interface) will be disabled, but not
! shown in the example text. SOME DEBUG MESSAGES are omitted to
! improve readability.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Albuquerque#debug eigrp fsm
EIGRP FSM Events/Actions debugging is on
Albuquerque#
*Mar  1 02:35:31.836: %LINK-3-UPDOWN: Interface Serial0/1, changed state to down
*Mar  1 02:35:31.848: DUAL: rcvupdate: 10.1.6.0/24 via Connected metric 42949672
95/4294967295
*Mar  1 02:35:31.848: DUAL: Find FS for dest 10.1.6.0/24. FD is 2169856, RD is 2
169856
*Mar  1 02:35:31.848: DUAL: 0.0.0.0 metric 4294967295/4294967295 not found D
min is 4294967295
*Mar  1 02:35:31.848: DUAL: Peer total/stub 2/0 template/full-stub 2/0
*Mar  1 02:35:31.848: DUAL: Dest 10.1.6.0/24 entering active state.
*Mar  1 02:35:31.852: DUAL: Set reply-status table. Count is 2.
*Mar  1 02:35:31.852: DUAL: Not doing split horizon
*Mar  1 02:35:31.852: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 10.1.6.3 (Serial0/1) is
down: interface down
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! The first message following these comments means that EIGRP will
! now start acting on the fact that neighbor 10.1.6.3 is down.
! The second message states that EIGRP is considering route 10.1.3.0.
! The third message means that it is looking for an "FS" (Feasible
! Successor) for this route.
! The next two highlighted messages imply that the old route to
! 10.1.3.0 is removed, and the new successor route (previously the
! feasible successor route) is added to the "RT" (routing table).
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
*Mar  1 02:35:31.852: DUAL: linkdown: start - 10.1.6.3 via Serial0/1
*Mar  1 02:35:31.852: DUAL: Destination 10.1.3.0/24
*Mar  1 02:35:31.852: DUAL: Find FS for dest 10.1.3.0/24. FD is 2172416, RD is 2172416
*Mar  1 02:35:31.856: DUAL: 10.1.6.3 metric 4294967295/4294967295
*Mar  1 02:35:31.856: DUAL: 10.1.4.2 metric 2684416/1794560 found Dmin is 2684416
*Mar  1 02:35:31.856: DUAL: Removing dest 10.1.3.0/24, nexthop 10.1.6.3
*Mar  1 02:35:31.856: DUAL: RT installed 10.1.3.0/24 via 10.1.4.2
*Mar  1 02:35:31.856: DUAL: Send update about 10.1.3.0/24.  Reason: metric chg
*Mar  1 02:35:31.860: DUAL: Send update about 10.1.3.0/24.  Reason: new if
! Rest of beug messages omitted for brevity
```

Most of the explanations for Example 6-7 are shown inside the example as comments. The output of the **show ip eigrp topology** command, when the feasible successor route to 10.1.3.0 exists, bears a closer look. That's because this command output identifies successor and feasible successor routes, but it is not obvious.

NOTE Example 6-8 has two **show ip eigrp topology** commands; the one described next is the second occurrence, a little past half-way through the example.

Small portions of three lines of output from the **show ip eigrp topology** command are highlighted. The first of these lines lists "FD is 2172416", which means that the Feasible Distance (FD) is 2,172,416. The *Feasible Distance* is Albuquerque's calculated metric for the route.

Next, look at the two numbers in parentheses in the second highlighted line from the **show ip eigrp topology** command. The first of these is this router's calculated metric for the route. Notice that the first number is 2,172,416, the same as the FD on the previous line. The fact that the first number in parenthesis matches the FD means that the route on this line is the *successor route*—the one that should be currently installed in the routing table. Note that the successor route's next hop address (10.1.6.3) matches next hop address shown for the route to 10.1.3.0/24 in the IP routing table, which confirms that the successor route is the route installed into the IP routing table.

The third line highlighted for this command references the second number in parentheses. (For reference, the highlighted line is "via 10.1.4.2 (2684416/1794560), Serial0/0"). This number (1,794,560 in this case) is the metric for this route *as calculated by the neighbor*. The neighbor on that line of output is 10.1.4.2, or Yosemite. As seen in the **show ip route** command output from Yosemite, which follows the output of the **show ip eigrp topology** command, Yosemite's route to 10.1.3.0 indeed has a metric of 1,794,560.

So, what does all this mean? Well, the requirement for a feasible successor route (in other words, the feasibility condition) is that the next hop neighbor's metric must be smaller than that router's calculated metric for the current best (successor) route. The analysis of the **show ip eigrp topology** command output can be summarized as follows:

- The route to 10.1.3.0 through 10.1.6.3 (Seville) is the successor route, because the calculated metric (2,172,416), shown as the first of the two numbers in parenthesis, has the same value as the "FD" (feasible distance). FD is by definition the metric of the best route (successor route).

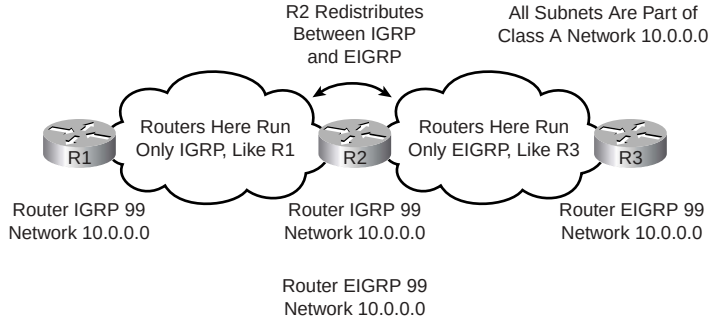
- The route to 10.1.3.0 through 10.1.4.2 (Yosemite) is a feasible successor route, because the neighbor's calculated metric (1,794,560, shown as the second number in parenthesis) is lower than the successor route's calculated metric.

IGRP to EIGRP Migration

In the early days of routing, the main options for routing protocol were RIP (Version 1) and IGRP. IGRP is appreciably better than RIP Version 1, particularly with regards to the metric and convergence time. As a nice side effect for Cisco stockholders, a choice for IGRP was a choice to use only Cisco routers.

When Cisco later announced EIGRP, they wanted to make migration simple. If a network is small enough to allow migration of all routers at one time, the routers could simply be configured with the new EIGRP configuration, the old IGRP configuration deleted, and the process was complete. However, most medium to large networks might want to migrate a subset of the routers to EIGRP, and migrate other routers later, just due to the size of the project. To make migration simple, Cisco created a feature of EIGRP called *automatic redistribution*. Figure 6-11 depicts the idea, and shows the configurations required on all three routers in order to enable automatic redistribution.

Figure 6-11 *Example Automatic Redistribution Between IGRP and EIGRP*



When using any other pair of IP routing protocols in different parts of an internetwork, a more detailed manual configuration of a feature called *Route Redistribution* is required. However, with IGRP and EIGRP, R2's routes learned with IGRP are automatically advertised with EIGRP, and visa versa. The only requirement is that both IGRP and EIGRP be configured on the same router at the border between the two routing protocols, and that they both use the same AS number (AS numbers range between 1 and 65,535).

Foundation Summary

The "Foundation Summary" section lists the most important facts from the chapter. Although this section does not list everything that will be on the exam, a well-prepared CCNA candidate should at a minimum know all the details in each Foundation Summary before taking the exam.

The basic process of learning routes for the first time with OSPF goes something like this:

1. Each router discovers its neighbors on each interface. The list of neighbors is kept in a neighbor table.
2. Each router uses a reliable protocol to exchange topology information (LSAs) with its neighbors.
3. Each router places the learned topology information in its topology database.
4. Each router runs the SPF algorithm against its own topology database to calculate the best routes to each subnet in the database.
5. Each router places the best route to each subnet in the IP routing table.

Figure 6-12 shows a network with multiple areas, along with the perspective of the topology for routers inside Area 1.

Figure 6-12 *Two-Area OSPF*

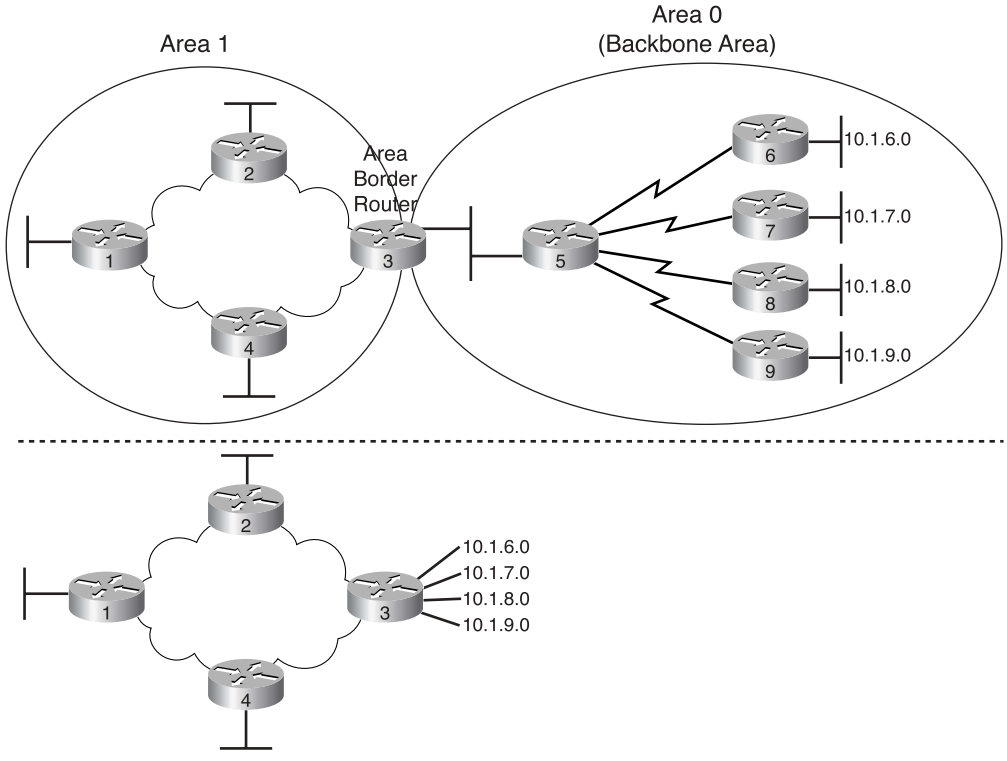


Table 6-9 summarizes some of the key points of comparison between the two types of routing protocols.

Table 6-9 *Comparing Link-State and Distance Vector Protocols*

Feature	Link State	Distance Vector
Convergence Time	Fast	Slow, mainly because of loop-avoidance features
Loop Avoidance	Built into the protocol	Requires extra features such as split horizon
Memory and CPU Requirements	Can be large; good design can minimize	Low
Requires Design Effort for Larger Networks	Yes	No
Public Standard or Proprietary	OSPF is public	RIP is publicly defined; IGRP is not

Table 6-10 summarizes some of the key comparison points between EIGRP, IGRP, and OSPF.

Table 6-10 *EIGRP Features Compared to OSPF and IGRP*

Feature	EIGRP	IGRP	OSPF
Discovers neighbors before exchanging routing information	Yes	No	Yes
Builds some form of topology table in addition to adding routes to the routing table	Yes	No	Yes
Converges quickly	Yes	No	Yes
Uses metrics based on bandwidth and delay by default	Yes*	Yes	No
Sends full routing information on every routing update cycle	No	Yes	No
Requires distance vector loop-avoidance features	No	Yes	No
Public standard	No	No	Yes

*EIGRP uses the same metric as IGRP, except that EIGRP scales the metric by multiplying by 256.

Example 6-9 shows two alternative OSPF configurations for a router. All interfaces are in Area 0.

Example 6-9 *OSPF Single-Area Configuration Alternatives*

```
interface ethernet 0/0
ip address 10.1.1.1 255.255.255.0
interface serial 0/0
ip address 10.1.4.1 255.255.255.0
interface serial 0/1
ip address 10.1.6.1 255.255.255.0
!
router ospf 1
network 10.0.0.0 0.255.255.255 area 0
!
! Alternately:
!
router ospf 1
network 10.1.1.1 0.0.0.0 area 0
network 10.1.4.1 0.0.0.0 area 0
network 10.1.6.1 0.0.0.0 area 0
```

The wildcard mask represents a 32-bit number. When the mask has a binary 1 in one of the bit positions, that bit is considered a wildcard bit, meaning that the router should not care what binary value is in the corresponding numbers. For that reason, binary 1s in the wildcard

mask are called *don't care bits*. If the wildcard mask bit is 0 in a bit position, the corresponding bits in the numbers being compared must match. You can think of these bits as the *do care bits*. The router must examine the two numbers and make sure that the values match for the bits that matter—in other words, the do care bits.

For instance, the first wildcard mask in Example 6-8 is 0.255.255.255. When converted to binary, this number is 0000 0000 1111 1111 1111 1111 1111 1111—in other words, eight 0s and 24 1s. The **network** command tells the IOS to compare 10.0.0.0, which is the number in the **network** command, to the IP addresses of each interface on the router. The wildcard mask tells IOS to compare only the first octet; the last three octets are wildcards, and anything matches. So, all three interface IP addresses are matched.

With the wildcard mask of 0.0.0.0, entire numbers must be compared—hence the specific **network** commands referring to the specific IP addresses of the interfaces in the alternative configuration.

Q & A

As mentioned in the Introduction, you have two choices for review questions. The following questions give you a bigger challenge than the exam because they are open-ended. By reviewing with this more-difficult question format, you can exercise your memory better and prove your conceptual and factual knowledge of the topics covered in this chapter. The answers to these questions are found in Appendix A.

For more practice with exam-like question formats, including multiple-choice questions and those using a router simulator, use the exam engine on the CD.

1. Create a minimal configuration enabling IP on each interface on a 2600 series router (two serial, one Ethernet). The Network Information Center (NIC) assigns you network 192.168.1.0. Your boss says that you need, at most, 60 hosts per LAN subnet. You also have point-to-point links attached to the serial interfaces. When choosing the IP address values and subnet numbers, you decide to start with the lowest numerical values. Assume that point-to-point serial links will be attached to this router and that EIGRP is the routing protocol.
2. Write down the steps you would take to migrate from RIP to OSPF in a router whose current RIP configuration includes only **router rip** followed by a **network 10.0.0.0** command. Assume a single OSPF area, and use as few **network** commands as possible.
3. Create a configuration for EIGRP on a router with these interfaces and addresses: e0 using 10.1.1.1, e1 using 224.1.2.3, s0 using 10.1.2.1, and s1 using 199.1.1.1. Use AS number 5.
4. Create a configuration for EIGRP on a router with these interfaces and addresses: e0 using 200.1.1.1, e1 using 128.1.3.2, s0 using 192.0.1.1, and s1 using 223.254.254.1.
5. From a router's user mode, without using debugs or privileged mode, how can you determine what routers are sending you EIGRP routing updates?
6. If the command **router eigrp 1,** followed by **network 10.0.0.0**, with no other network commands, is configured in a router that has an Ethernet0 interface with IP address 168.10.1.1, does EIGRP send updates out Ethernet0?
7. If the command **router ospf 1,** followed by **network 10.0.0.0 0.255.255.255 area 0**, with no other network commands, is configured in a router that has an Ethernet0 interface with IP address 10.10.1.1, does OSPF send updates out Ethernet0?

8. If the commands **router eigrp 1** and **network 10.0.0.0** are configured in a router that has an Ethernet0 interface with IP address 168.10.1.1, mask 255.255.255.0, does this router have a route to 168.10.1.0?
9. Which command lists all IP routes learned via OSPF?
10. Compare and contrast the type of information exchanged in routing updates sent by distance vector routing protocols versus link-state protocols.
11. Define balanced hybrid and give an example of a balanced hybrid protocol.
12. Describe how balanced hybrid protocols differ from distance vector protocols in terms of how a router notices that a neighboring router has failed.
13. List the distance vector loop-avoidance features used by OSPF, such as split horizon.
14. List two OSPF features that help decrease the size of the OSPF topology database.
15. Assume that you must choose between OSPF and EIGRP for a routing protocol in a new network you are building. List and explain the most compelling reason to choose OSPF and the most compelling reason to choose EIGRP.