CHAPTER 9
Routing and Routers

Routing is a term with multiple meanings in different disciplines. In general, it refers to determining a path for something. In telecom, a call may be routed based on the number being dialed or some other identifier. In either case, a path is determined for the call.

Mail is also routed—I'm not talking about email here (though email is routed, too)—but rather, snail mail. When you write an address and a zip code on a letter, you are providing the means for the post office to route your letter. You provide a destination and, usually, a source address, and the post office determines the best path for the letter. If there is a problem with the delivery of your letter, the return address is used to route it back to you. The exact path the letter takes to get from its source to its destination doesn't really matter; all you care about is that it (hopefully) makes it in a timely fashion, and in one piece.

In the IP world, packets or frames are forwarded on a local network by switches, hubs, or bridges. If the address of the destination is not on the local network, the packet must be forwarded to a *gateway*. The gateway is responsible for determining how to get the packet to where it needs to be. RFC 791, titled INTERNET PROTOCOL, defines a gateway thusly:

> 2.4. Gateways
>
> Gateways implement Internet protocol to forward datagrams between networks. Gateways also implement the Gateway to Gateway Protocol (GGP) [7] to coordinate routing and other Internet control information.
>
> In a gateway the higher level protocols need not be implemented and the GGP functions are added to the IP module.

When a station on a network sends a packet to a gateway, the station doesn't care *how* the packet gets to its destination—just that it does (at least, in the case of TCP). Much like a letter in the postal system, each packet contains its source and destination addresses so routing can be accomplished with ease.

In the realm of semantics and IP, a gateway is a device that forwards packets to a destination other than the local network. For all practical purposes, a gateway is a

router. Router is the term I will generally use in this book, although you will also see the phrase *default gateway*.

In the olden days of data communication, a "gateway" was a device that translated one protocol to another. For example, if you had a device that converted a serial link into a parallel link, that device would be called a gateway. Similarly, a device that converted Ethernet to Token Ring might be called a gateway. Nowadays, such devices are called *media converters*. (This wouldn't be the first time I've been accused of harkening back to the good old days—pull up a rocker here on the porch, and have a mint julep while I spin you a yarn.)

Routers usually communicate with each other by means of one or more *routing protocols*. These protocols let the routers learn information about networks other than the ones directly connected to them.

Network devices used to be limited to bridges and routers. Bridges, hubs, and switches operated only on Layer 2 of the OSI stack, and routers only on Layer 3. Now these devices are often merged into single devices, and routers and switches often operate on all seven layers of the OSI stack.

In today's world, where every device seems to be capable of anything, when should you pick a router rather than a switch? Routers tend to be WAN-centric, while switches tend to be LAN-centric. If you're connecting T1s, you probably want a router. If you're connecting Ethernet, you probably want a switch.

Routing Tables

Routing is a fundamental process common to almost every network in use today. Still, many engineers don't understand how routing works. While the Cisco certification process should help you understand how to configure routing, in this section, I'll show you what you need to know about routing in the real world. I'll focus on the foundations, because that's what most engineers seem to be lacking—we spend a lot of time studying the latest technologies and sometimes forget the core principles on which everything else is based.

In a Cisco router, the routing table is called the *route information base* (RIB). When you execute the command `show ip route`, the output you receive is a formatted view of the information in the RIB.

Each routing protocol has its own table of information. For example, EIGRP has the topology table, and Open Shortest Path First (OSPF) has the OSPF database. Each protocol makes decisions about which routes will be held in its database. Routing protocols use their own metrics to determine which route is best, and the metrics vary widely. The metric value is determined by the routing protocol from which the route

was learned. Thus, the same link may have very different metrics depending on the protocol used. For example, the same path may be described with a metric of 2 in the Routing Information Protocol (RIP), 200 in OSPF, and 156160 in EIGRP.

Routing protocols and metrics are covered in more detail in Chapter 10.

If the same route is learned from two sources within a single routing protocol, the one with the best metric will win. Should the same route be learned from two routing protocols within a single router, the protocol with the lowest administrative distance will win. The *administrative distance* is the value assigned to each routing protocol to allow the router to prioritize routes learned from multiple sources. The administrative distances for the various routing protocols are shown in Table 9-1.

Table 9-1. Routing protocols and their administrative distances

Route type	Administrative distance
Connected interface	0
Static route	1
Enhanced Interior Gateway Routing Protocol (EIGRP) summary route	5
External Border Gateway Protocol (BGP)	20
Internal EIGRP	90
Interior Gateway Routing Protocol (IGRP)	100
Open Shortest Path First (OSPF)	110
Intermediate System-Intermediate System (IS-IS)	115
Routing Information Protocol (RIP)	120
Exterior Gateway Protocol (EGP)	140
On Demand Routing (ODR)	160
External EIGRP	170
Internal BGP	200
Unknown	255

Spanning tree, discussed in Chapter 8, isn't really a routing protocol, because the protocol doesn't care about the data being passed; spanning tree is only concerned with loops and with preventing them from a physical and Layer-2 perspective. In other words, spanning tree is concerned more with determining that all possible paths within its domain are loop-free than with determining the paths along which data should be sent.

When a packet arrives at a router, the router determines whether the packet needs to be forwarded to another network. If it does, the router checks the RIB to see whether it contains a route to the destination network. If there is a match, the packet is adjusted and forwarded out the proper interface to where it belongs (see Chapter 15 for more information on this process). If no match is found in the RIB, the packet is forwarded to the default gateway, if one exists. If no default gateway exists, the packet is dropped.

Originally, the destination network was described by a network address and a subnet mask. Today, destination networks are often described by a network address and a prefix length. The network address is an IP address that references a network. The prefix length is the number of bits set to 1 in the subnet mask. Networks are described in the format *network-address/prefix-length*. For example, the network 10.0.0.0 with a subnet mask of 255.0.0.0 would be described as 10.0.0.0/8. When shown in this format, the route is called simply a *prefix*. The network 10.0.0.0/24 is said to be a longer prefix than the network 10.0.0.0/8. The more bits that are used to identify the network portion of the address, the longer the prefix is said to be.

The RIB may include multiple routes to the same network. For example, in Figure 9-1, R2 learns the network 10.0.0.0 from two sources: R1 advertises the route 10.0.0.0/8, and R3 advertises the route 10.0.0.0/24. Because the prefix lengths are different, these are considered to be different routes. As a result, they will both end up in the routing table.

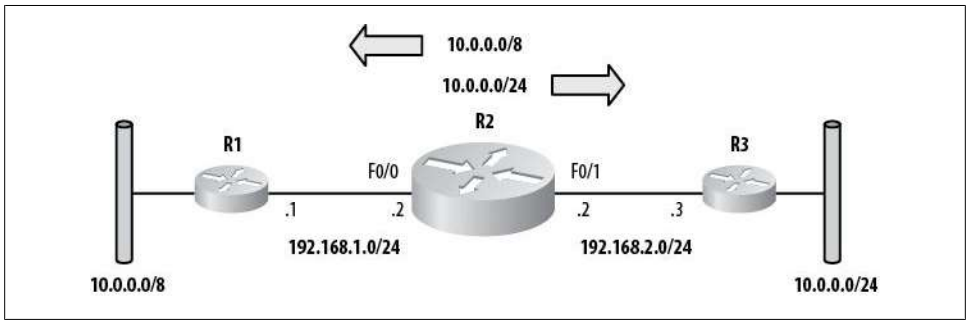

Figure 9-1. Same network with different prefix lengths

Here are the routes as seen in R2:

```
     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
D       10.0.0.0/8 [90/30720] via 192.168.1.1, 00:12:01, FastEthernet0/0
D       10.0.0.0/24 [90/30720] via 192.168.2.3, 00:12:01, FastEthernet0/1
```

When a packet is received in R2, the destination IP address is matched against the routing table. If R2 receives a packet destined for 10.0.0.1, which route will it choose? There are two routes in the table that seem to match: 10.0.0.0/8 and 10.0.0.0/24. The route with the longest prefix length (also called the *most specific* route) is the more desirable route. Thus, when a packet destined for 10.0.0.1 arrives on R2, it will be forwarded to R3. The important thing to realize about this example is that there may

be legitimate addresses within the 10.0.0.0/24 range behind R1 that R2 will never be able to access.

Technically, 10.0.0.0/8 is a network and 10.0.0.0/24 is a subnet. Read on for further clarification.

Route Types

The routing table can contain six types of routes:

Host route
: A host route is a route to a host. In other words, the route is not to a network. Host routes have a subnet mask of 255.255.255.255 and a prefix length of /32.

Subnet
: A subnet is a portion of a major network. The subnet mask is used to determine the size of the subnet. 10.10.10.0/24 (255.255.255.0) is a subnet.

Summary (group of subnets)
: A summary route is a single route that references a group of subnets. 10.10.0.0/16 (255.255.0.0) would be a summary, provided that subnets with longer masks (such as 10.10.10.0/24) existed.

Major network
: A major network is any classful network, along with its native mask. 10.0.0.0/8 (255.0.0.0) is a major network.

Supernet (group of major networks)
: A supernet is a single route that references a group of major networks. For example, 10.0.0.0/7 is a supernet that references 10.0.0.0/8 and 11.0.0.0/8.

Default route
: A default route is shown as 0.0.0.0/0 (0.0.0.0). This route is also called the *route of last resort*. This is the route that is used when no other route matches the destination IP address in a packet.

The IP Routing Table

To show the IP routing table, use the `show ip route` command:

```
R2#sho 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, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route
```

```
Gateway of last resort is 11.0.0.1 to network 0.0.0.0

     172.16.0.0/16 is variably subnetted, 6 subnets, 2 masks
D       172.16.200.0/23 is a summary, 00:56:18, Null0
C       172.16.200.0/24 is directly connected, Loopback2
C       172.16.201.0/24 is directly connected, Serial0/0
C       172.16.202.0/24 is directly connected, Loopback3
C       172.16.100.0/23 is directly connected, Loopback4
D       172.16.101.0/24 [90/2172416] via 11.0.0.1, 00:53:07, FastEthernet0/1
C    10.0.0.0/8 is directly connected, FastEthernet0/0
C    11.0.0.0/8 is directly connected, FastEthernet0/1
     192.168.1.0/32 is subnetted, 1 subnets
D       192.168.1.11 [90/156160] via 11.0.0.1, 00:00:03, FastEthernet0/1
S*   0.0.0.0/0 [1/0] via 11.0.0.1
D    10.0.0.0/7 is a summary, 00:54:40, Null0
```

The first block of information is shown every time the command is executed. In the interest of brevity, I will remove it from most of the examples in this book. This block is a key that explains the codes listed down the left side of the routing table.

The next line lists the default gateway, if one is present:

```
Gateway of last resort is 11.0.0.1 to network 0.0.0.0
```

If there are two or more default gateways, they will all be listed. This is common when the default gateway is learned from a routing protocol that allows equal-cost load sharing. If two links provide access to the advertised default and they both have the same metric, they will both be listed as default routes. In this case, packets will be equally balanced between the two links using per-packet load balancing.

If no default gateway has been configured or learned, you'll instead see this message:

```
Gateway of last resort is not set
```

The next block of text contains the rest of the routing table:

```
     172.16.0.0/16 is variably subnetted, 6 subnets, 2 masks
D       172.16.200.0/23 is a summary, 00:56:18, Null0
C       172.16.200.0/24 is directly connected, Loopback2
C       172.16.201.0/24 is directly connected, Serial0/0
C       172.16.202.0/24 is directly connected, Loopback3
C       172.16.100.0/23 is directly connected, Loopback4
D       172.16.101.0/24 [90/2172416] via 11.0.0.1, 00:53:07, FastEthernet0/1
C    10.0.0.0/8 is directly connected, FastEthernet0/0
C    11.0.0.0/8 is directly connected, FastEthernet0/1
     192.168.1.0/32 is subnetted, 1 subnets
D       192.168.1.11 [90/156160] via 11.0.0.1, 00:00:03, FastEthernet0/1
S*   0.0.0.0/0 [1/0] via 11.0.0.1
D    10.0.0.0/7 is a summary, 00:54:40, Null0
```

Let's examine a single entry from the routing table, so you can see what's important:

```
D       172.16.101.0/24 [90/2172416] via 11.0.0.1, 00:53:07, FastEthernet0/1
```

First is the route code. In this case it's `D`, which indicates the route was learned via EIGRP (you can look this up in the block of codes at the top of the `show ip route` output).

Next is the route itself. In this example, the route is to the subnet 172.16.101.0/24. After that are two numbers in brackets: the first number is the administrative distance (see Table 9-1) and the second number is the metric for the route. The metric is determined by the routing protocol from which the route was learned (in this case, EIGRP).

The next piece of information is the next hop the router needs to send packets to in order to reach this subnet. In this case, `via 11.0.0.1` indicates that packets destined for the subnet 172.16.101.0/24 should be forwarded to the IP address 11.0.0.1. Finally, you have the age of the route (`00:53:07`), followed by the interface out which the router will forward the packet (`FastEthernet0/1`).

I've built the sample router so that the routing table will have one of each type of route. Again, those route types are *host*, *subnet*, *summary*, *major network*, *supernet*, and *default*. The following sections explain the types in more detail. I'll show the routing table entries for each type in bold.

Host Route

A host route is simply a route with a subnet mask of all ones (255.255.255.255), or a prefix length of /32. In the sample routing table, the route to 192.168.1.11 is a host route:

```
      172.16.0.0/16 is variably subnetted, 6 subnets, 2 masks
D        172.16.200.0/23 is a summary, 00:56:18, Null0
C        172.16.200.0/24 is directly connected, Loopback2
C        172.16.201.0/24 is directly connected, Serial0/0
C        172.16.202.0/24 is directly connected, Loopback3
C        172.16.100.0/23 is directly connected, Loopback4
D        172.16.101.0/24 [90/2172416] via 11.0.0.1, 00:53:07, FastEthernet0/1
C     10.0.0.0/8 is directly connected, FastEthernet0/0
C     11.0.0.0/8 is directly connected, FastEthernet0/1
         192.168.1.0/32 is subnetted, 1 subnets
D        192.168.1.11 [90/156160] via 11.0.0.1, 00:00:03, FastEthernet0/1
S*    0.0.0.0/0 [1/0] via 11.0.0.1
D     10.0.0.0/7 is a summary, 00:54:40, Null0
```

Notice that the route is shown to be a part of a larger network (in this case, 192.168.1.0). We know this because the host route is indented under the major network. The router will attempt to show you which classful (major) network contains the route. If the router knows about only a single subnet mask, it will assume the network has been divided equally with that mask. In this case, the router has assumed that the major network 192.168.1.0/24 has been equally subnetted, with each subnet having a /32 mask. Hence, the natively /24 network 192.168.1.0 is shown as 192.168.1.0/32.

Subnet

Subnets are indented under their source major networks. In our example, the major network 172.16.0.0/16 has been subnetted; in fact, it has been subnetted under the rules of Variable Length Subnet Masks (VLSM), which allow each subnet to have a different subnet mask (within certain limits—see Chapter 34 for more detail). The route in the middle that is not in bold is a summary route, which I'll cover next:

```
172.16.0.0/16 is variably subnetted, 6 subnets, 2 masks
D       172.16.200.0/23 is a summary, 00:56:18, Null0
C       172.16.200.0/24 is directly connected, Loopback2
C       172.16.201.0/24 is directly connected, Serial0/0
C       172.16.202.0/24 is directly connected, Loopback3
C       172.16.100.0/23 is directly connected, Loopback4
D       172.16.101.0/24 [90/2172416] via 11.0.0.1, 00:53:07, FastEthernet0/1
C    10.0.0.0/8 is directly connected, FastEthernet0/0
C    11.0.0.0/8 is directly connected, FastEthernet0/1
     192.168.1.0/32 is subnetted, 1 subnets
D       192.168.1.11 [90/156160] via 11.0.0.1, 00:00:03, FastEthernet0/1
S*   0.0.0.0/0 [1/0] via 11.0.0.1
D    10.0.0.0/7 is a summary, 00:54:40, Null0
```

Summary (Group of Subnets)

The term *summary* is used in the routing table to represent any group of routes. Technically, according to the Cisco documentation, a summary is a group of subnets, while a supernet is a group of major networks. Both are called summaries in the routing table. Thus, while the example routing table shows two summary entries, only the first is technically a summary route:

```
     172.16.0.0/16 is variably subnetted, 6 subnets, 2 masks
D       172.16.200.0/23 is a summary, 00:56:18, Null0
C       172.16.200.0/24 is directly connected, Loopback2
C       172.16.201.0/24 is directly connected, Serial0/0
C       172.16.202.0/24 is directly connected, Loopback3
C       172.16.100.0/23 is directly connected, Loopback4
D       172.16.101.0/24 [90/2172416] via 11.0.0.1, 00:53:07, FastEthernet0/1
C    10.0.0.0/8 is directly connected, FastEthernet0/0
C    11.0.0.0/8 is directly connected, FastEthernet0/1
     192.168.1.0/32 is subnetted, 1 subnets
D       192.168.1.11 [90/156160] via 11.0.0.1, 00:00:03, FastEthernet0/1
S*   0.0.0.0/0 [1/0] via 11.0.0.1
D    10.0.0.0/7 is a summary, 00:54:40, Null0
```

The last entry in the routing table, which is also reported as a summary, is a group of major networks and is technically a supernet.

The differentiation between supernets and summary routes is primarily an academic one. In the real world, both are routinely called summary routes or aggregate routes. Different routing protocols use different terms for groups of routes, be they subnets or major networks—BGP uses the term "aggregate," while OSPF uses the term "summary."

The destination for both summary routes is `Null0`. `Null0` as a destination indicates that packets sent to this network will be dropped. The summary routes point to `Null0` because they were created within EIGRP on this router.

The `Null0` route is there for the routing protocol's use. The more specific routes must also be included in the routing table because the local router must use them when forwarding packets. The specific routes will not be advertised in the routing protocol; only the summary will be advertised. We can see this if we look at an attached router:

```
     172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
D       172.16.200.0/23 [90/156160] via 11.0.0.2, 04:30:21, FastEthernet0/1
D       172.16.202.0/24 [90/156160] via 11.0.0.2, 04:30:21, FastEthernet0/1
D       172.16.100.0/23 [90/156160] via 11.0.0.2, 04:30:21, FastEthernet0/1
C       172.16.101.0/24 is directly connected, Serial0/0
```

On the connected router, the summary route for 172.16.200.0/23 is present, but the more specific routes 172.16.200.0/24 and 172.16.201.0/24 are not.

Major Network

A major network is a network in its native form. For example, the 10.0.0.0/8 network has a native subnet mask of 255.0.0.0. The network 10.0.0.0/8 is therefore a major network. Referencing 10.0.0.0 with a prefix mask longer than /8 changes the route to a subnet, while referencing it with a mask shorter than /8 changes the route to a supernet.

Two major networks are shown in the routing table:

```
     172.16.0.0/16 is variably subnetted, 6 subnets, 2 masks
D       172.16.200.0/23 is a summary, 00:56:18, Null0
C       172.16.200.0/24 is directly connected, Loopback2
C       172.16.201.0/24 is directly connected, Serial0/0
C       172.16.202.0/24 is directly connected, Loopback3
C       172.16.100.0/23 is directly connected, Loopback4
D       172.16.101.0/24 [90/2172416] via 11.0.0.1, 00:53:07, FastEthernet0/1
C    10.0.0.0/8 is directly connected, FastEthernet0/0
C    11.0.0.0/8 is directly connected, FastEthernet0/1
     192.168.1.0/32 is subnetted, 1 subnets
D       192.168.1.11 [90/156160] via 11.0.0.1, 00:00:03, FastEthernet0/1
S*   0.0.0.0/0 [1/0] via 11.0.0.1
D    10.0.0.0/7 is a summary, 00:54:40, Null0
```

172.16.0.0/16 is also shown, but only as a reference to group all of the subnets underneath it. The entry for 172.16.0.0/16 is not a route.

Supernet (Group of Major Networks)

A supernet is a group of major networks. In this example, there is a route to 10.0.0.0/7, which is a group of the major networks 10.0.0.0/8 and 11.0.0.0/8:

```
     172.16.0.0/16 is variably subnetted, 6 subnets, 2 masks
D       172.16.200.0/23 is a summary, 00:56:18, Null0
C       172.16.200.0/24 is directly connected, Loopback2
C       172.16.201.0/24 is directly connected, Serial0/0
C       172.16.202.0/24 is directly connected, Loopback3
C       172.16.100.0/23 is directly connected, Loopback4
D       172.16.101.0/24 [90/2172416] via 11.0.0.1, 00:53:07, FastEthernet0/1
C    10.0.0.0/8 is directly connected, FastEthernet0/0
C    11.0.0.0/8 is directly connected, FastEthernet0/1
     192.168.1.0/32 is subnetted, 1 subnets
D       192.168.1.11 [90/156160] via 11.0.0.1, 00:00:03, FastEthernet0/1
S*   0.0.0.0/0 [1/0] via 11.0.0.1
D    10.0.0.0/7 is a summary, 00:54:40, Null0
```

Notice the route is again destined to `Null0`. Sure enough, on a connected router we will only see the summary and not the more specific routes:

```
D    10.0.0.0/7 [90/30720] via 11.0.0.2, 04:30:22, FastEthernet0/1
```

Default Route

The default route, or "route of last resort," is displayed in a special place above the routing table, so it can easily be seen:

```
Gateway of last resort is 11.0.0.1 to network 0.0.0.0

     172.16.0.0/16 is variably subnetted, 6 subnets, 2 masks
D       172.16.200.0/23 is a summary, 00:56:18, Null0
C       172.16.200.0/24 is directly connected, Loopback2
C       172.16.201.0/24 is directly connected, Serial0/0
C       172.16.202.0/24 is directly connected, Loopback3
C       172.16.100.0/23 is directly connected, Loopback4
D       172.16.101.0/24 [90/2172416] via 11.0.0.1, 00:53:07, FastEthernet0/1
C    10.0.0.0/8 is directly connected, FastEthernet0/0
C    11.0.0.0/8 is directly connected, FastEthernet0/1
     192.168.1.0/32 is subnetted, 1 subnets
D       192.168.1.11 [90/156160] via 11.0.0.1, 00:00:03, FastEthernet0/1
S*   0.0.0.0/0 [1/0] via 11.0.0.1
D    10.0.0.0/7 is a summary, 00:54:40, Null0
```

In this case, the default route is a static route, as indicated by the `S` in the first column, but it could be learned from a routing protocol as well. The asterisk next to the `S` indicates that this route is a candidate for the default route. There can be more than one candidate, in which case there will be multiple entries with asterisks. There can even be multiple default routes, but only one will be listed in the first line.

This output shows a router with two active default gateways, though only one is listed in the first line:

```
Gateway of last resort is 10.0.0.1 to network 0.0.0.0

     20.0.0.0/24 is subnetted, 1 subnets
S       20.0.0.0 [1/0] via 10.0.0.1
     10.0.0.0/24 is subnetted, 3 subnets
C       10.0.0.0 is directly connected, FastEthernet0/0
C    192.168.1.0/24 is directly connected, FastEthernet0/1
S*   0.0.0.0/0 [1/0] via 10.0.0.1
                [1/0] via 10.0.0.2
```

When in doubt, look at the 0.0.0.0/0 entry in the routing table, as it will always have the most accurate information.

Virtual Routing and Forwarding

Nexus switches and newer versions of IOS support something called Virtual Routing and Forwarding (VRF) instances. Each VRF is a self-contained routing table within the same router. Within a Nexus 7000, you can have multiple VRFs within a single Virtual Device Context (VDC). On the Nexus 5000, which as of this writing does not support VDCs or routing, you can still have multiple VRFs. In fact, by default, the management network resides in a management VRF, while other traffic is in the default VRF. In IOS 15.x, you may also configure VRFs.

On the Nexus platform, two VRFs exist by default—*management* and *default*:

```
NX-7K-1-Daisy# sho vrf
VRF-Name                  VRF-ID    State   Reason
default                        1    Up      --
management                     2    Up      --
```

Creating a new VRF is as simple as specifying one with the `vrf context` *`vrf-name`* command. Here, I'll create two new VRFs, named Earth and Mars:

```
NX-7K-1-Daisy(config)# vrf context Earth
NX-7K-1-Daisy(config)# vrf context Mars
```

Now we can see the additional VRFs with the `show vrf` command:

```
NX-7K-1-Daisy(config)# sho vrf
VRF-Name                  VRF-ID    State   Reason
Earth                          3    Up      --
Mars                           5    Up      --
default                        1    Up      --
management                     2    Up      --
```

Interfaces can be assigned to VRFs. Remember that in Nexus 7000 switches, interfaces default to routed mode, so we don't need to configure them as such. You apply interfaces to VRFs with the `vrf member` *`vrf-name`* interface command:

```
NX-7K-1-Daisy(config)# int e3/25
NX-7K-1-Daisy(config-if)# vrf member Earth
NX-7K-1-Daisy(config-if)# ip address 10.0.0.1/24
```

Now I'll assign a different interface to another VRF:

```
NX-7K-1-Daisy(config-if)# int e3/26
NX-7K-1-Daisy(config-if)# vrf member Mars
NX-7K-1-Daisy(config-if)# ip address 10.0.0.1/24
```

Notice that I have two interfaces in the same router configured with the same IP address. This is possible because they each belong to a different VRF. The routing tables in VRFs Earth and Mars are completely separate from each other. This is a pretty powerful feature that adds another layer of virtualization to the Nexus platform.

VRFs are a distinctly Layer-3 idea, so you cannot assign a VLAN to a VRF, but you can assign a VLAN interface to one.

VRFs can be frustrating if you're not used to them. In our example, the command `sho ip route` won't show any of our configured interfaces. This is because the default VRF is named *default*. Since there are multiple routing tables now, if you want to see the routing table within a specific VRF, you must specify the VRF within your show command:

```
NX-7K-1-Daisy# sho ip route vrf Earth
IP Route Table for VRF "Earth"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]

10.0.0.0/24, ubest/mbest: 1/0, attached
    *via 10.0.0.1, Eth3/25, [0/0], 02:49:03, direct
10.0.0.1/32, ubest/mbest: 1/0, attached
    *via 10.0.0.1, Eth3/25, [0/0], 02:49:03, local
```

Most Layer-3 relayed commands support the VRF keyword in the Nexus, as shown here:

```
NX-7K-1-Daisy# sho ip eigrp neighbors vrf Earth
IP-EIGRP neighbors for process 0 VRF Earth

IP-EIGRP neighbors for process 100 VRF Earth
H   Address                 Interface       Hold  Uptime  SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
0   10.0.0.2                Eth3/25         11   00:00:50  3    200   0   2
```

Even the `ping` command supports the VRF keyword:

```
NX-7K-1-Daisy# ping 10.0.0.2 vrf Earth
PING 10.0.0.2 (10.0.0.2): 56 data bytes
Request 0 timed out
64 bytes from 10.0.0.2: icmp_seq=1 ttl=254 time=0.884 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=254 time=0.538 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=254 time=0.597 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=254 time=0.6 ms
```

To make things easier, you can change the current VRF so that all subsequent commands apply to that VRF. You do this with the `routing-context` command, which is not a configuration command:

```
NX-7K-1-Daisy# routing-context vrf Earth
NX-7K-1-Daisy%Earth#
```

This changes our prompt by appending the name of the VRF after a percent sign to remind us where we are. Now when we run the `show ip route` (or any other) command, it is applied to the current routing context, which is Earth:

```
NX-7K-1-Daisy%Earth# sho ip route
IP Route Table for VRF "Earth"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]

10.0.0.0/24, ubest/mbest: 1/0, attached
    *via 10.0.0.1, Eth3/25, [0/0], 05:01:14, direct
10.0.0.1/32, ubest/mbest: 1/0, attached
    *via 10.0.0.1, Eth3/25, [0/0], 05:01:14, local
200.200.200.0/24, ubest/mbest: 1/0
    *via 10.0.0.2, Eth3/25, [90/130816], 00:21:50, eigrp-100, internal
```

To change back to the default VRF, use the `routing-context vrf default` command, which results in our prompt returning to normal:

```
NX-7K-1-Daisy%Earth# routing-context vrf default
NX-7K-1-Daisy#
```

CHAPTER 10
Routing Protocols

A *routing protocol* is a means whereby devices exchange information about the state of the network. The information collected from other devices is used to make decisions about the best path for packets to flow to each destination network.

Routing protocols are both protocols and applications. The protocols themselves sit at Layer 3 in the OSI model, while the applications that make the routing decision run at Layer 7. Many routing protocols exist, though only a few are in common use today. Older protocols are rarely used, though some networks may contain legacy devices that support only those protocols. Some firewalls and servers may support a limited scope of routing protocols—most commonly Routing Information Protocol (RIP) and Open Shortest Path First (OSPF)—but for the sake of simplicity, I will refer to all devices that participate in a routing protocol as *routers*.

Routing protocols allow networks to be dynamic and resistant to failure. If all routes in a network were static, the only form of dynamic routing we would be able to employ would be the *floating static route*. A floating static route becomes active only if another static route is removed from the routing table. Here's an example:

```
ip route 0.0.0.0 0.0.0.0 192.168.1.1 1
ip route 0.0.0.0 0.0.0.0 10.0.0.1 2
```

The primary default route points to 192.168.1.1 and has a metric of 1. The second default route points to 10.0.0.1 and has a metric of 2.

Routes with the best metrics are inserted into the routing table, so in this case, the first route will win. Should the network 192.168.1.0 become unavailable, all routes pointing to it will be removed from the routing table. At this time, the default route to 10.0.0.1 will be inserted into the routing table, since it now has the best metric for the 0.0.0.0/0 network.

The floating static route allows routes to change if a directly connected interface goes down, but it cannot protect routes from failing if a remote device or link fails. *Dynamic* routing protocols usually allow all routers participating in the protocol to learn about any failures on the network through regular communication between routers.

Communication Between Routers

Routers need to communicate with one another to learn the state of the network. One of the original routing protocols, the RIP, sent out updates about the network using broadcasts. This was fine for smaller networks, but as networks grew, these broadcasts became troublesome. Every host on a network listens to broadcasts, and with RIP, the broadcasts can be quite large.

Most modern routing protocols communicate on broadcast networks using *multicast packets*. Multicast packets have specific IP and corresponding MAC addresses that reference predetermined groups of devices.

Because routing is usually a dynamic process, existing routers must be able to discover new routers to add their information into the tables that describe the network. For example, all EIGRP (Enhanced Internal Gateway Routing Protocol) routers within the same domain must be able to communicate with each other. Defining specific neighbors is not necessary with this protocol, because they are discovered dynamically.

Most interior gateway protocols discover neighbors dynamically. BGP (Border Gateway Protocol) does not discover neighbors. Instead, BGP must be configured to communicate with each neighbor manually.

The Internet Assigned Numbers Authority (IANA) shows all multicast addresses in use at *http://www.iana.org/assignments/multicast-addresses*. Some of the more common multicast addresses include:

```
224.0.0.0  Base Address (Reserved)                  [RFC1112,JBP]
224.0.0.1  All Systems on this Subnet               [RFC1112,JBP]
224.0.0.2  All Routers on this Subnet                      [JBP]
224.0.0.4  DVMRP    Routers                         [RFC1075,JBP]
224.0.0.5  OSPFIGP  OSPFIGP All Routers            [RFC2328,JXM1]
224.0.0.6  OSPFIGP  OSPFIGP Designated Routers     [RFC2328,JXM1]
224.0.0.9  RIP2 Routers                           [RFC1723,GSM11]
224.0.0.10 IGRP Routers                               [Farinacci]
224.0.0.12 DHCP Server / Relay Agent                   [RFC1884]
224.0.0.18 VRRP                                       [RFC3768]
224.0.0.102 HSRP                                       [Wilson]
```

The list shows that all IGRP (Internal Gateway Routing Protocol) routers, including Enhanced IGRP routers, will listen to packets sent to the address 224.0.0.10.

Not all routing protocols use multicasts to communicate. Because BGP does not discover neighbors, it has no need for multicasts, and instead uses unicast packets. Many other routing protocols can also be configured to assign neighbors statically. This usually results in unicast messages, instead of multicasts, being sent to specific routers.

There may be more than one type of routing protocol on a single network. In the Ethernet network shown in Figure 10-1, for example, there are five routers, three of which are running OSPF, and two of which are running EIGRP. There is no reason for the EIGRP routers to receive OSPF updates, or vice versa. Using multicasts ensures that only the routers running the same routing protocols communicate with and discover each other.

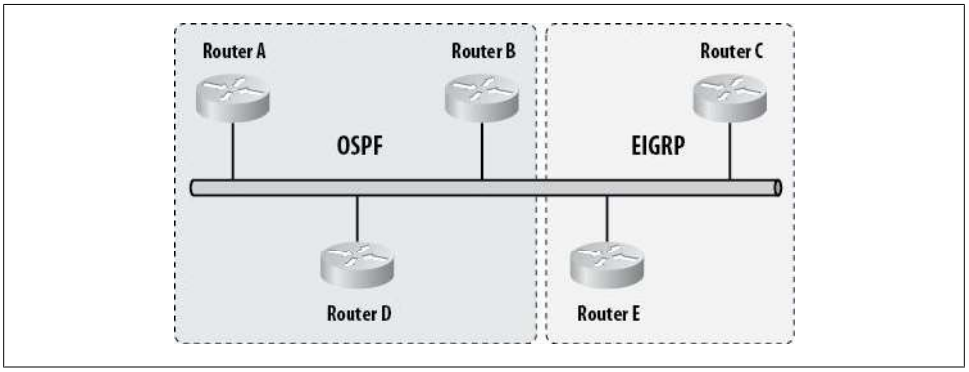

Figure 10-1. Multiple routing protocols on a single Ethernet network

A network may also contain multiple instances of the same routing protocol. These separate areas of control are called *autonomous systems* in EIGRP, and *domains* in OSPF (although the term autonomous system is often used incorrectly). In EIGRP, each instance is referenced with an autonomous system number (ASN). OSPF is more complicated to configure in this way, and is outside the realm of this book. OSPF on Cisco routers is configured with a *process-ID*, but this ID is locally significant to the router. Within a single router, routes between multiple OSPF processes are not automatically redistributed, but two OSPF routers, each with a different process-ID, will form a neighbor adjacency (assuming everything else required is in place). The process-ID in OSPF does not reference the domain.

Figure 10-2 shows a network with two EIGRP processes active. Because the multicast packets sent by an EIGRP router will be destined for all EIGRP routers, all EIGRP routers will listen to the updates. The updates contain the autonomous-system-ID, so the individual routers can determine whether to retain or discard them. RIP does not support the idea of separate processes, so any router running RIP will receive and process updates from all other RIP routers on the network.

When there are two autonomous systems on the same network, the routes learned in each may or may not be shared between the processes by default. One of the features of EIGRP is automatic redistribution between processes. For routes to be shared with other protocols, one of the routers must participate in both processes and be configured to share the routes between them.

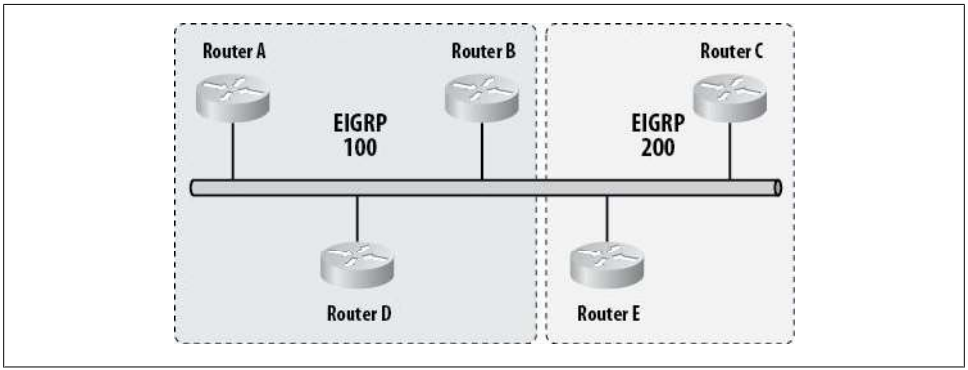

Figure 10-2. Two EIGRP processes on a single network

The act of passing routes from one process or routing protocol to another process or routing protocol is called *redistribution*. An example of multiple EIGRP routing processes being redistributed is shown in Figure 10-3.

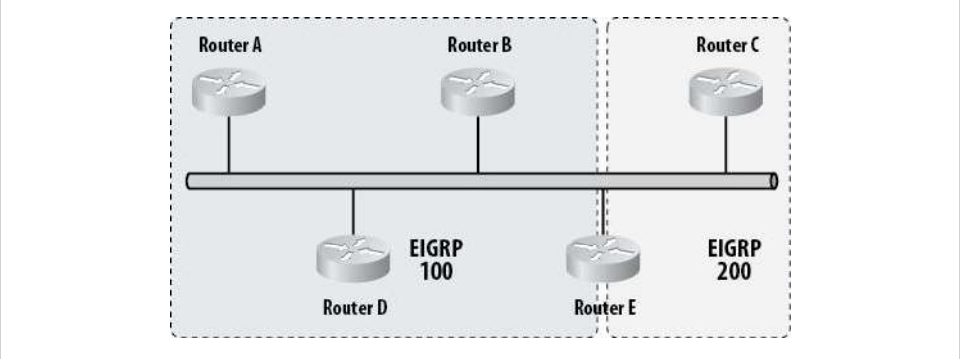

Figure 10-3. Routing protocol redistribution

In Figure 10-2, we have two EIGRP autonomous systems, but there is no way for the processes to learn each other's routes. In Figure 10-3, Router E is configured to be a member of EIGRP 100 and EIGRP 200. Router E thus redistributes routes learned on each AS into the other process.

When a route is learned within a routing process, the route is said to be *internal*. When a route is learned outside the routing process and redistributed into the process, the route is said to be *external*. Internal routes are usually considered to be more reliable than external routes, based on a metric called *administrative distance* (described later in the chapter). Exceptions include BGP, which prefers external routes over internal ones, and OSPF, which does not assign different administrative distances to internal versus external routes.

Metrics and Protocol Types

The job of a routing protocol is to determine the best path to a destination network. The best route is chosen based on a protocol-specific set of rules. RIP uses the number of hops (routers) between networks, whereas OSPF calculates the cost of a route based on the bandwidth of all the links in the network. EIGRP uses links' reported bandwidths and delays to determine the best path, by default, and you can configure it to use a few more factors as well. Each of these protocols determines a value for each route. This value is usually called a *metric*. Routes with lower metrics are more desirable.

Perhaps the simplest form of metric to understand is the one used by RIP: hop count. In RIP, the hop count is simply the number of routers between the router determining the path and the network to be reached.

Let's consider an example. In Figure 10-4, there are two networks, labeled 10.0.0.0 and 20.0.0.0. Router A considers 20.0.0.0 to be available via two paths: one through Router B and one through Router E. The path from Router A through Router B traverses Routers B, C, and D, resulting in a hop count of three for this path. The path from Router A through Router E traverses routers E, F, G, and D, resulting in a hop count of four for this path.

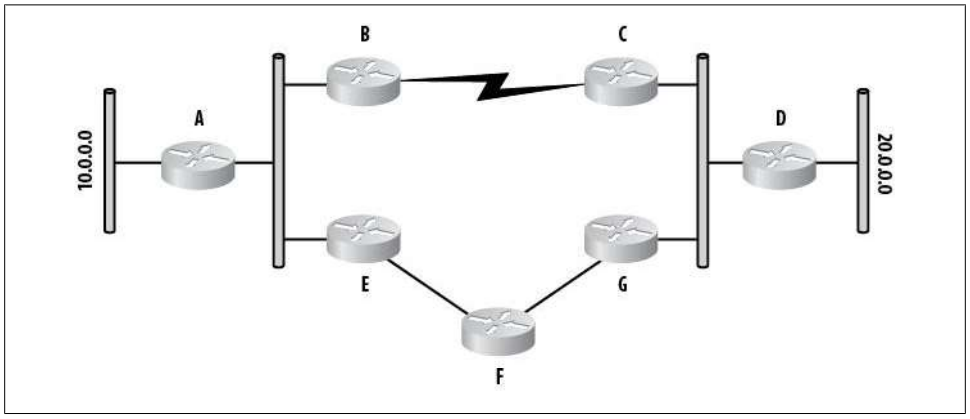

Figure 10-4. Example of metrics in routing protocols

Lower metrics always win, so Router A will consider the path through Router B to be the better path. This router will be added to the routing table with a metric of 3.

Using hop count as a metric has a limitation that can cause suboptimal paths to be chosen. Looking at Figure 10-5, you can see that the link between Routers B and C is a T1 running at 1.54 Mbps, while the links between Routers E, F, and G are all direct fiber links running at 1 Gbps. That means the path through Routers E, F, and G will be substantially faster than the link between Routers B and C, even though that link has fewer hops. However, RIP doesn't know about the bandwidth of the links in use, and takes into account only the number of hops.

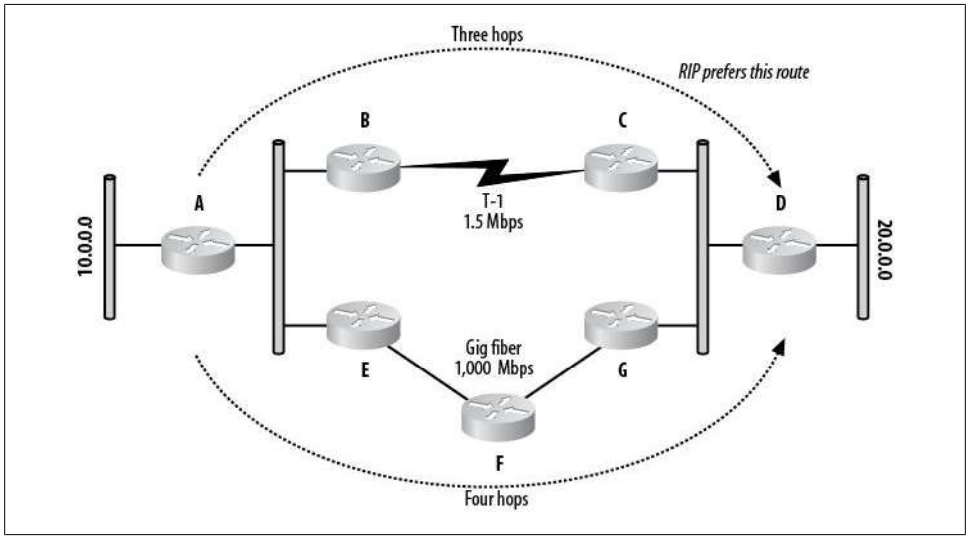

Figure 10-5. RIP uses hops to determine the best routes

A protocol such as RIP is called a *distance-vector* routing protocol, as it relies on the distance to the destination network to determine the best path. Distance-vector protocols suffer from another problem, called *counting to infinity*. Protocols such as RIP place an upper limit on the number of hops allowed to reach a destination. Hop counts that exceed this number are considered to be unreachable. In RIP, the maximum hop count is 15, with a hop count of 16 being unreachable. As you might imagine, this does not scale well in modern environments, where there may easily be more than 16 routers in a given path. A more modern version of RIP, called RIP version 2 (RIPv2 or RIP2), raises the limit to 255 hops, with 256 being unreachable. However, since RIPv2 still doesn't understand the states and capabilities of the links that join the hops together, most networks employ newer, more robust routing protocols instead.

Routing protocols such as OSPF are called *link-state* routing protocols. These protocols include information about the links between the source router and destination network, as opposed to simply counting the number of routers between them.

OSPF adds up the *cost* of each link. You determine the cost of a link by dividing 100,000,000 by the bandwidth of the link in bits per second (bps). The costs of some common links are therefore:

100 Mbps (100,000,000 / 100,000,000 bps) = 1
10 Mbps (100,000,000 / 10,000,000 bps) = 10
1.5 Mbps (100,000,000 / 1,540,000 bps) = 64 (results are rounded)

Figure 10-6 shows the same network used in the RIP example. This time, OSPF is determining the best path to the destination network using bandwidth-based metrics. The metric for the T1 link is 64, and the metric for the gigabit link path is 4. Because

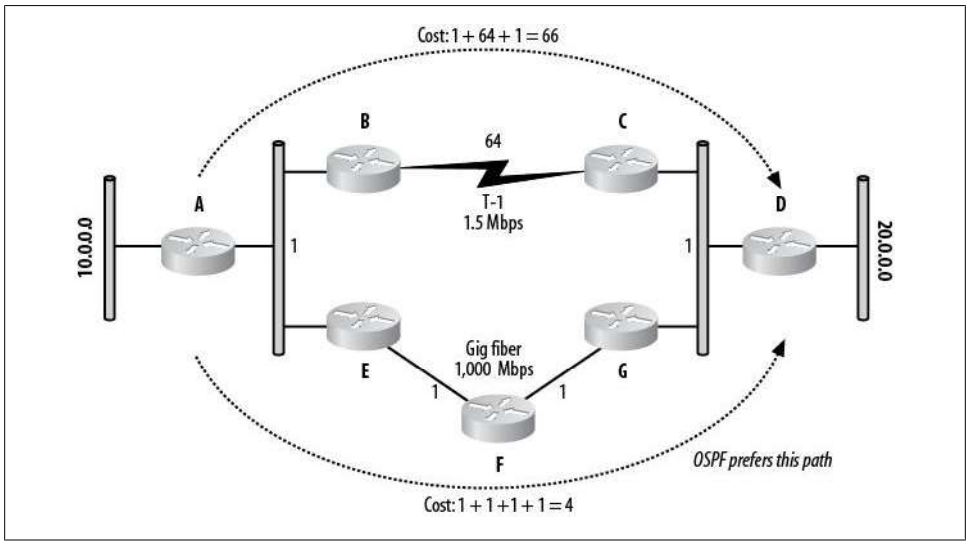

Figure 10-6. OSPF uses bandwidth to determine the best routes

the metric for the link through Routers E, F, and G is lower than that for the link through Routers B and C, this path is inserted into the routing table.

EIGRP uses a more complicated formula for determining costs. It can include bandwidth, delay, reliability, effective bandwidth, and maximum transmission unit (MTU) in its calculation of a metric. EIGRP is considered to be a *hybrid* protocol.

Administrative Distance

Networks often have more than one routing protocol active. In such situations, there is a high probability that the same networks will be advertised by multiple routing protocols. Figure 10-7 shows a network in which two routing protocols are running: the top half of the network is running RIP, and the bottom half is running OSPF. Router A will receive routes for the network 20.0.0.0 from RIP and OSPF. RIP's route has a better metric, but, as we've seen, OSPF has a better means of determining the proper path. So, how is the best route determined?

Routers choose routes based on a predetermined set of rules. One of the factors in deciding which route to place in the routing table is *administrative distance*. Administrative distance is a value assigned to every routing protocol. In the event of two protocols reporting the same route, the routing protocol with the lowest administrative distance will win, and its version of the route will be inserted into the RIB.

The administrative distances of the various routing protocols are shown in Table 10-1.

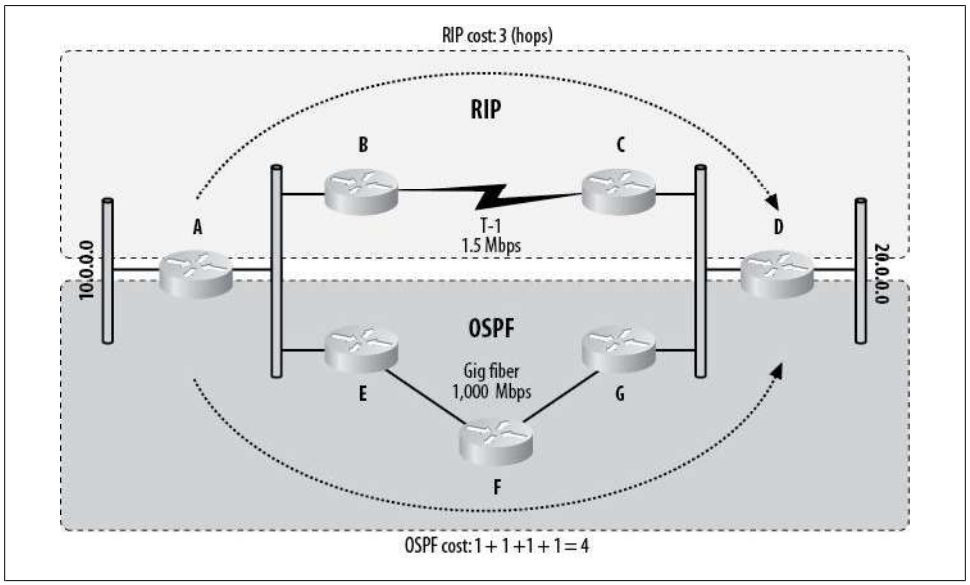

Figure 10-7. Competing routing protocols

Table 10-1. Administrative distances of routing protocols

Route type	Administrative distance
Connected interface	0
Static route	1
EIGRP summary route	5
External BGP	20
Internal EIGRP	90
IGRP	100
OSPF	110
IS-IS	115
RIP	120
EGP	140
ODR	160
External EIGRP	170
Internal BGP	200
Unknown	255

A static route to a connected interface has an administrative distance of 0, and is the only route that will override a normal static route. A route sourced with an administrative distance of 255 is not trusted, and will not be inserted into the routing table.

Looking at Table 10-1, you can see that RIP has an administrative distance of 120, while OSPF has an administrative distance of 110. This means that even though the RIP route has a better metric in Figure 10-7, the route inserted into the routing table will be the one provided by OSPF.

Specific Routing Protocols

Entire books have been written on each of the routing protocols discussed in this chapter. My goal is not to teach you everything you need to know about the protocols, but rather to introduce them and show you what you need to know to get them operational. I'll also include some of the commands commonly used to troubleshoot these protocols.

Routing protocols are divided into types based on their purpose and how they operate. The major division between routing protocols is that of internal gateway protocols versus external gateway protocols.

An *internal gateway protocol*, or IGP, is designed to maintain routes within an *autonomous system*. An autonomous system is any group of devices controlled by a single entity. An example might be a company or a school, but the organization does not need to be that broad—an autonomous system could be a floor in a building or a department in a company. Examples of IGPs include RIP, EIGRP, and OSPF.

An *external gateway protocol*, or EGP, is designed to link autonomous systems together. The Internet is the prime example of a large-scale EGP implementation. The autonomous systems—groups of devices controlled by individual service providers, schools, companies, etc.—are each self-contained. They are controlled internally by IGPs and are interconnected using an EGP (in the case of the Internet, BGP).

Figure 10-8 shows how different autonomous systems might be connected. Within each circle is an autonomous system. The IGP running in each autonomous system is unrelated to the external gateway protocol. The EGP knows only that a certain network is owned by a certain autonomous system. Let's say that 1.0.0.0/8 is within ASN 1, 2.0.0.0/8 is within ASN 2, 3.0.0.0/8 is within ASN 3, and so on. For a device in ASN 1 to get to the network 10.0.0.0/8, the path might be through autonomous systems 1, 2, 3, 9, and 10. It might also be through autonomous systems 1, 2, 7, 8, 9, and 10, or even 1, 2, 7, 8, 3, 9, and 10. As with a distance-vector IGP counting hops, the fewer the number of autonomous systems traversed, the more appealing the route.

The important thing to remember with EGPs is that they really don't care how many routers there are or what the speeds of the links are. The only thing an EGP cares about is traversing the least possible number of autonomous systems to arrive at a destination.

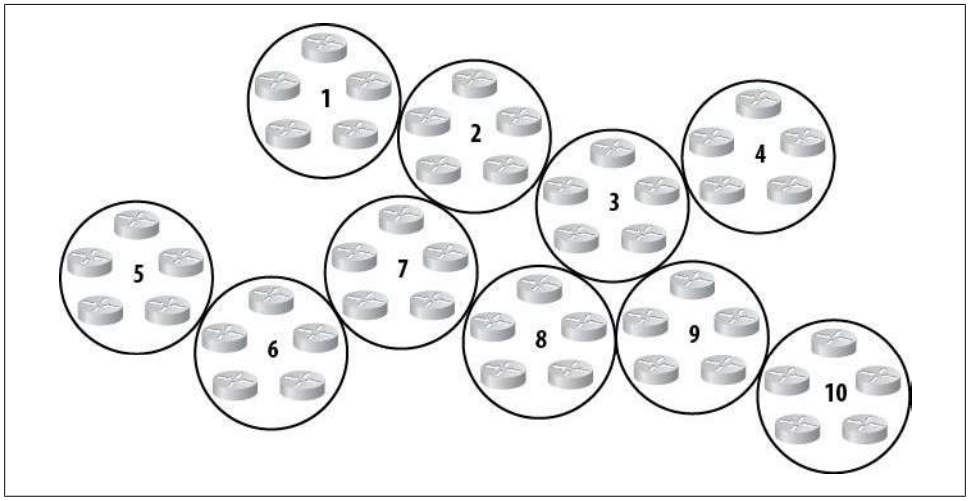

Figure 10-8. Interconnected autonomous systems

Before we go any further, let's define some key routing terms:

Classful routing protocol
: A classful routing protocol is one that has no provision to support subnets. The natural state of the network is always advertised. For example, the network 10.0.0.0 will always be advertised with a subnet mask of 255.0.0.0 (/8), regardless of what subnet mask is actually in use. RIPv1 and IGRP are classful routing protocols.

Classless routing protocol
: A classless routing protocol is one that includes subnet masks in its advertisements. All modern protocols are classless. EIGRP and OSPF are classless routing protocols.

Poison reverse
: If a router needs to tell another router that a network is no longer viable, one of the methods it might employ is *route poisoning*. Consider RIPv1 as an example. Recall that a metric of 16 is considered unreachable. A router can send an update regarding a network with a metric of 16, thereby *poisoning* the entry in the routing table of the receiving router. When a router receives a poison update, it returns the same update to the sending router. This reflected route poisoning is called *poison reverse*. Distance-vector routing protocols (including the hybrid protocol EIGRP) use route poisoning, while link-state protocols such as OSPF do not.

Split horizon
: Split horizon is a technique used by many routing protocols to prevent routing loops. When split horizon is enabled, routes that the routing protocol learns are not advertised out the same interfaces from which they were learned. This rule can be problematic in virtual circuit topologies, such as Frame Relay or ATM. If a route

is learned on one permanent virtual circuit (PVC) in a Frame Relay interface, chances are the other PVC needs the update but will never receive it, because both PVCs exist on the same physical interface. Frame Relay subinterfaces are often the preferred method of dealing with split horizon issues.

Convergence
: A network is said to be *converged* when all of the routers in the network have received and processed all updates. Essentially, this condition exists when a network is stable. Anytime a link's status changes, the routing protocols must propagate that change, whether through timed updates or triggered updates. With timed updates, if updates are sent but no changes need to be made, the network has converged.

As mentioned earlier, many routing protocols exist, but luckily, only a few are in widespread use. Each has its own idiosyncrasies. In the following sections, I'll cover the basic ideas behind the more common protocols and show how to configure them for the most typical scenarios. There is no right or wrong way to configure routing protocols, though some ways are certainly better than others. When designing any network, remember that simplicity is a worthy goal that will save you countless hours of troubleshooting misery.

RIP

RIP is the simplest of the routing protocols in use today. While I like to tell my clients that *simple is good*, I don't consider RIP to be simple goodness.

RIP broadcasts all the routes it knows about every 30 seconds, regardless of the statuses of any other routers in the network. Because RIP uses broadcasts, every host on the network listens to the updates, even though few can process them. On larger networks, the updates can be quite large and consume a lot of bandwidth on expensive WAN links.

Another issue with RIP is the fact that it does not use triggered updates. A *triggered update* is one that is sent when the network changes; *nontriggered* (*timed*) *updates* are sent on a regular schedule. Coupled with the fact that updates are sent only every 30 seconds, this behavior causes RIP networks to converge very slowly. Slow convergence is not acceptable in most modern networks, which require failover and convergence in seconds. A RIP network with only five routers may take two minutes or more to converge.

RIP is a classful protocol, which means subnet masks are not advertised. This is also an unacceptable limitation in most networks. Figure 10-9 illustrates one of the most common pitfalls of using classful protocols such as RIP. Router A is advertising its directly connected network 10.10.10.0/24, but because RIP is classful, it advertises the network without the subnet mask. Without a subnet mask, the receiving router must assume the entire network is included in the advertisement. Consequently, upon

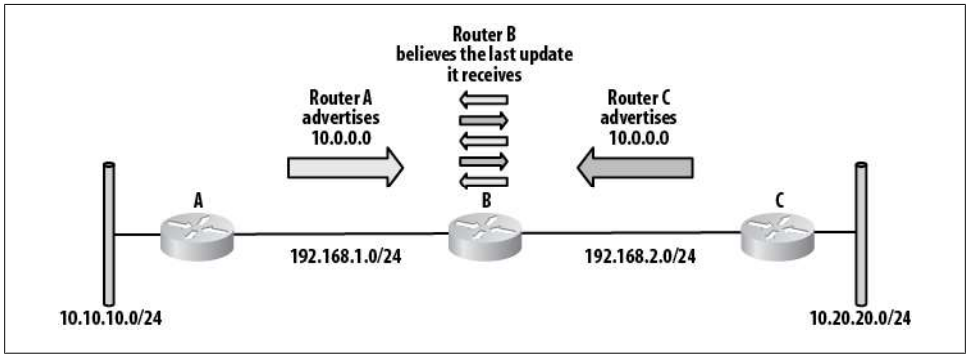

Figure 10-9. RIP classful design problem

receiving the advertisement for 10.0.0.0 from Router A, Router B inserts the entire 10.0.0.0/8 network into its routing table. Router C has a different 10 network attached: 10.20.20.0/24. Again, RIP advertises the 10.0.0.0 network from Router C without a subnet mask. Router B has now received another advertisement for 10.0.0.0/8. The network is the same, the protocol is the same, and the hop count is the same. When a newer update is received for a route that has already been inserted into the routing table, the newer update is considered to be more reliable and is inserted into the routing table, overwriting the previous entry. This means each time Router B receives an update from Router A or Router C, it will change its routing table to show that network 10.0.0.0 is behind the router from which it received the update.

You might be tempted to say that the networks behind Routers A and C in Figure 10-9 are different, but from RIP's point of view, you would be wrong. Technically, the *networks* behind Routers A and C are the same. They are both part of the 10.0.0.0/8 network. The routers are connected to different *subnets* within the 10.0.0.0/8 network, which is why RIP has a problem with the design.

The only other type of network that RIP understands is a host network. RIP can advertise a route for a /32 or 255.255.255.255 network. Because RIP does not include subnet masks in its updates, a route is determined to be a host route when the address of the network is anything other than a normal network address.

Configure routing protocols in IOS using the `router` command. The protocol name is included in the command, which puts the router into router configuration mode:

```
Router-A (config)#router rip
Router-A (config-router)#
```

On modern routers that support RIPv2, if you wish to use RIPv1, you must specify it explicitly in the router configuration, because RIPv2 is used by default:

```
router rip
 version 1
```

By default, no interfaces are included in the routing protocol. This means no interfaces will have routing updates sent on them, and any routing updates received on the interfaces will be ignored.

To enable interfaces in a routing protocol, specify the networks that are configured on the interfaces you wish to include by using the `network` command in router configuration mode:

```
Router-A (config)#router rip
Router-A (config-router)# network 10.10.10.0
```

With a classful protocol like RIP, you must be careful, because, as an example, including the network 10.0.0.0 will include every interface configured with a 10.x.x.x IP address, regardless of subnet mask. RIP does not allow the inclusion of a subnet or inverse mask in the `network` statements. You can enter a network other than 10.0.0.0, but IOS will convert the entry into the full classful network.

The preceding entry results in the following being displayed in the configuration:

```
router rip
 network 10.0.0.0
```

The configuration that would include all interfaces for Router A as shown in Figure 10-10 is as follows:

```
router rip
 version 1network 10.0.0.0
 network 192.168.1.0
 network 192.168.2.0
```

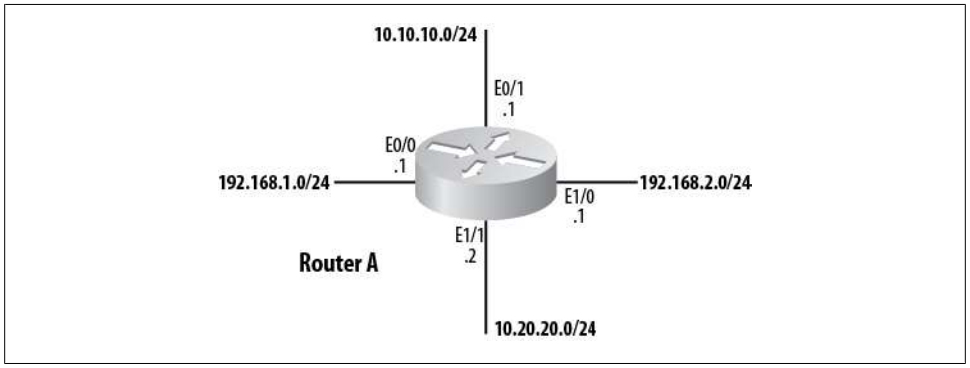

Figure 10-10. Routing protocol network interfaces

One entry covers both the 10.10.10.0 and 10.20.20.0 networks, but the 192.168 networks each require their own `network` statements. This is because 192.x.x.x networks are class C networks, while 10.x.x.x networks are class A networks.

You won't always want to include every interface that the `network` statement encompasses. In the preceding example, we might want to allow RIP on E0/0, but not on

E1/1. This can be accomplished with the use of the `passive-interface` command, which removes an interface from the broader range specified by the `network` command:

```
router rip
 version 1
 passive-interface Ethernet1/1
 network 10.0.0.0
```

The `passive-interface` command causes RIP to stop sending updates on the specified interface. The router will continue to receive and process RIP updates received on the interface, though.

Routes learned via RIP are identified in the routing table with an `R` in the first column. This example shows the network 172.16.0.0/16 learned via RIP. The actual network in use is 172.16.100.0/24, but because RIP is classful, the router assumes the entire 172.16.0.0/16 network is there as well:

```
R3#sho ip route
[text removed]

Gateway of last resort is 192.168.1.2 to network 0.0.0.0
R    172.16.0.0/16 [120/1] via 10.10.10.4, 00:00:21, Ethernet0/0
     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.10.10.0/24 is directly connected, Ethernet0/0
C       10.100.100.100/32 is directly connected, Loopback0
C    192.168.1.0/24 is directly connected, Ethernet1/0
S*   0.0.0.0/0 [254/0] via 192.168.1.2
```

Here we see an example of a host route being received by RIP:

```
R4#sho ip route
[text removed]

Gateway of last resort is not set
     172.16.0.0/32 is subnetted, 1 subnets
C       172.16.1.1 is directly connected, Loopback0
     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.10.10.0/24 is directly connected, Ethernet0/0
R       10.100.100.100/32 [120/1] via 10.10.10.3, 00:00:21, Ethernet0/0
C    192.168.1.0/24 is directly connected, Ethernet0/1
```

RIPv2

RIP was updated in the mid-1990s to reflect the widespread use of Classless Internet Domain Routing (CIDR) and Variable Length Subnet Masks (VLSM). The new protocol, RIP version 2, operates similarly to RIP version 1 in that it still uses hops as its only metric. However, it does have some significant advantages over RIPv1, including:

- RIPv2 supports classless routing by including subnet masks in network advertisements.
- RIPv2 supports triggered updates.

- Updates in RIPv2 are sent using the multicast address 224.0.0.9 instead of as broadcasts.
- Neighbors can be configured with RIPv2. When a neighbor is configured, updates are sent to that neighbor using unicasts, which can further reduce network traffic.
- RIPv2 supports authentication between routers.

Even though RIPv2 supports subnets, it still only accepts classful addresses in the `network` command, so be careful when determining which networks and interfaces you've included. Use the `passive-interface` command to limit the scope of the `network` command, if necessary.

RIPv2 is classless and advertises routes including subnet masks, but it summarizes routes by default. This means that if you have a 10.10.10.0/24 network connected to your router, it will still advertise 10.0.0.0/8, just like RIPv1. The first thing you should do when configuring RIPv2 is turn off autosummarization with the router command `no auto-summary`:

```
R3(config)#router rip
R3(config-router)# no auto-summary
```

The routing table in a Cisco router makes no distinction between RIPv1 and RIPv2. Both protocols are represented by a single `R` in the routing table. Cisco routers default to RIPv1, so you should enable version 2 if you want it, and if you're using RIP, you probably do.

EIGRP

EIGRP is a classless enhancement to IGRP, which supports only classful networks. EIGRP, like IGRP, is a Cisco-proprietary routing protocol, which means that only Cisco routers can use this protocol. If you throw a Juniper or Nortel router into your network, it will not be able to communicate with your Cisco routers using EIGRP.

EIGRP is a very popular routing protocol because it's easy to configure and manage. With minimal configuration and design, you can get an EIGRP network up and running that will serve your company for years to come.

The ease of configuring EIGRP is also the main reason I see so many misbehaving EIGRP networks in the field. A network engineer builds a small network for his company. As time goes on, the network gets larger and larger, and the routing environment gets more and more complicated. EIGRP manages the routing on the network quite nicely, until one day things start to go wrong. The engineer who built the network can't figure out what's wrong, and consultants are called in who completely redesign the network.

This is not to say that EIGRP is not a good routing protocol; I believe it is a very strong protocol. My point is that it's almost too easy to configure. You can throw two EIGRP routers on an Ethernet LAN with minimal configuration, and they will communicate

and share routes. You can do the same with 10 or 20 or 100 routers, and they will all communicate and share routes. You can add 100 serial links with remote sites using EIGRP, and they will all communicate and share routes. The routing table will be a mess and the routers may be converging constantly, but the packets will flow. Eventually, however, the default settings may fail to work properly because the default configurations are not designed to scale in massive networks.

When EIGRP is configured properly on a network with a well-designed IP address scheme, it can be an excellent protocol for even a large network. When configured with multiple processes, it can scale very well.

EIGRP is a hybrid routing protocol that combines features from distance-vector protocols with features usually seen in link-state protocols. EIGRP uses triggered updates, so updates are sent only when changes occur. Bandwidth and delay are used as the default metrics, and although you can add other attributes to the equation, it is rarely a good idea to do so. EIGRP converges very quickly, even in large networks. A network that might take minutes to converge with RIP will converge in seconds with EIGRP.

To configure EIGRP, enter into router configuration mode with the `router eigrp` *`autonomous-system-number`* command. The autonomous system number identifies the instance of EIGRP. A router can have multiple instances of EIGRP running on it, each with its own database containing routes. The router will choose the best route based on criteria such as metrics, administrative distance, and so on. This behavior is different from that of RIP in that RIP runs globally on the router.

Figure 10-11 shows a router with two instances of EIGRP active. Each instance is referenced by an ASN. Routes learned in one process are not shared with the other process, by default. Each process is essentially its own routing protocol. For a route learned in one process to be known to the other, the router must be configured for redistribution. EIGRP will redistribute IGRP routes automatically within the same ASN (redistribution is covered in detail in Chapter 11).

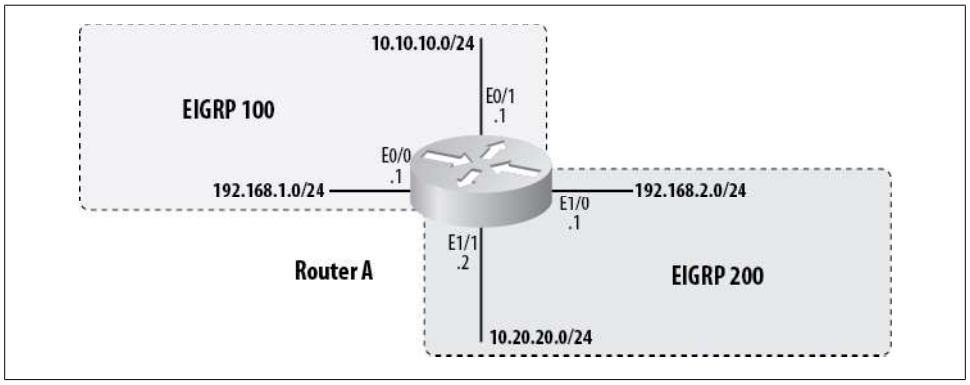

Figure 10-11. Multiple EIGRP instances

As with all IGPs, you list the interfaces you wish to include using the `network` command. EIGRP, like RIP, will automatically convert a classless network into the classful equivalent. The difference is that with EIGRP, you can add an inverse subnet mask to make the entry more specific. The following commands add all interfaces with addresses in the 10.0.0.0 network to the EIGRP 100 process:

```
Router-A(config)#router eigrp 100
Router-A(config-router)# network 10.0.0.0
```

But in the example in Figure 10-11, we'd like to add only the interface with the network 10.10.10.0/24. The subnet mask for a /24 network is 255.255.255.0, and the inverse subnet mask is 0.0.0.255 (inverse subnet masks are also called wildcard masks, and are discussed in Chapter 23). So, to add only this interface, we'd use the following `network` command:

```
Router-A(config-router)#network 10.10.10.0 0.0.0.255
```

After executing this command, the running configuration will still contain the less specific 10.0.0.0 network statement:

```
router eigrp 100
 network 10.10.10.0 0.0.0.255
 network 10.0.0.0
```

Both commands will take effect. Be careful of this, as it can cause no end of frustration. In this example, it will cause the interface E1/1 to be included in EIGRP 100, which is not what we want. We need to remove the less specific `network` command by negating it:

```
router eigrp 100
 no network 10.0.0.0
```

It's a very good practice to enable only the specific interface you wish to add in any routing process that supports it. You can do this by specifying the IP address on the interface with an all-zeros mask. In our example, the command is `network 10.10.10.1 0.0.0.0`. This prevents surprises should network masks change or interfaces be renumbered. Thus, my preferred configuration for EIGRP on the router shown in Figure 10-11 is:

```
router eigrp 100
 network 10.10.10.1 0.0.0.0
 network 192.168.1.1 0.0.0.0
!
router eigrp 200
 network 10.20.20.1 0.0.0.0
 network 192.168.2.1 0.0.0.0
```

EIGRP summarizes routes the same way RIP does, but because EIGRP is a classless protocol, we can disable this behavior with the `no auto-summary` command:

```
Router-A(config-router)#no auto-summary
```

There are very few instances where you'd want to leave autosummary on, so you should get into the habit of disabling it.

EIGRP operates by sending out hello packets using the multicast IP address 224.0.0.10 on configured interfaces. When a router running EIGRP receives these hello packets, it checks to see if the hello contains a process number matching an EIGRP process running locally. If it does, a handshake is performed. If the handshake is successful, the routers become *neighbors*.

Unlike RIP, which broadcasts routes to anyone who'll listen, EIGRP routers exchange routes only with neighbors. Once a *neighbor adjacency* has been formed, update packets are sent to the neighbor directly using unicast packets.

A useful command for EIGRP installations is `eigrp log-neighbor-changes`. This command displays a message to the console/monitor/log (depending on your logging configuration) every time an EIGRP neighbor adjacency changes state:

```
1d11h: %DUAL-5-NBRCHANGE: IP-EIGRP 100: Neighbor 10.10.10.4 (Ethernet0/0) is up:
new adjacency
```

On large networks, this can be annoying during a problem, but it can easily be disabled if needed.

To see the status of EIGRP neighbors on a router, use the `show ip eigrp neighbors` command:

```
R3#sho ip eigrp neighbors
IP-EIGRP neighbors for process 100
H   Address                 Interface   Hold Uptime   SRTT   RTO  Q  Seq Type
                                        (sec)         (ms)       Cnt Num
1   10.10.10.5              Et0/0         14 00:00:19    4   200  0  1
0   10.10.10.4              Et0/0         13 00:02:35    8   200  0  3
```

This command's output should be one of the first things you look at if you're having problems, because without a neighbor adjacency, EIGRP routers will not exchange routes.

Routes learned via internal EIGRP have an administrative distance of 90 and are marked with a single `D` in the first column of the routing table. Routes learned via external EIGRP have an administrative distance of 170 and are marked with the letters `D EX` at the beginning of the route:

```
R3#sho ip route
[text removed]

Gateway of last resort is 192.168.1.2 to network 0.0.0.0

     5.0.0.0/32 is subnetted, 1 subnets
D EX    5.5.5.5 [170/409600] via 10.10.10.5, 00:00:03, Ethernet0/0
     10.0.0.0/8 is variably subnetted, 2 masks
C       10.10.10.0/24 is directly connected, Ethernet0/0
C       10.100.100.100/32 is directly connected, Loopback0
C    192.168.1.0/24 is directly connected, Ethernet1/0
```

```
D    192.168.3.0/24 [90/2195456] via 10.10.10.5, 00:08:42, Ethernet0/0
S*   0.0.0.0/0 [254/0] via 192.168.1.2
```

EIGRP stores its information in three databases: the route database, the topology database, and the neighbor database. Viewing the topology database can be a tremendous help when troubleshooting routing problems. Not only can you see what EIGRP has put into the routing table, but you can also see what EIGRP considers to be alternate possibilities for routes:

```
R3#sho ip eigrp topology
IP-EIGRP Topology Table for AS(100)/ID(10.100.100.100)

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

P 5.5.5.5/32, 1 successors, FD is 409600
         via 10.10.10.5 (409600/128256), Ethernet0/0
P 10.10.10.0/24, 1 successors, FD is 281600
         via Connected, Ethernet0/0
P 192.168.3.0/24, 1 successors, FD is 2195456
         via 10.10.10.5 (2195456/2169856), Ethernet0/0
```

OSPF

In a nutshell, the premise of the OSPF routing protocol is that the shortest or fastest path that is available is the one that will be used.

OSPF is the routing protocol of choice when:

- There are routers from vendors other than Cisco in the network.
- The network requires segmentation into areas or zones.
- You want to avoid proprietary protocols.

OSPF is a link-state routing protocol. The metric it uses is bandwidth. The bandwidth of each link is calculated by dividing 100,000,000 by the bandwidth of the link in bits per second. Thus, a 100 Mbps link has a metric or "cost" of 1, a 10 Mbps link has a cost of 10, and a 1.5 Mbps link has a cost of 64. A 1 Gbps (or faster) link also has a cost of 1 because the cost cannot be lower than 1. The costs for each link in the path are added together to form a metric for the route.

In networks that include links faster than 100 Mbps, the formula for link cost can be changed using the `auto-cost reference-bandwidth` command. The default reference bandwidth is 100. In other words, by default, a 100 Mbps link has a cost of 1. To make a 1,000 Mbps link have a cost of 1, change the reference bandwidth to 1,000:

```
R3(config)#router ospf 100
R3(config-router)# auto-cost reference-bandwidth 1000
```

If you change the reference bandwidth, you must change it on every router communicating in the OSPF process. Failure to do so will cause unstable networks and unpredictable routing behavior.

OSPF classifies routers according to their function in the network. These are the types of OSPF routers:

Internal router
: An internal router resides completely within a single area within a single OSPF autonomous system.

Area border router (ABR)
: An ABR resides in more than one area within a single OSPF autonomous system.

Autonomous system border router (ASBR)
: An ASBR connects to multiple OSPF autonomous systems, or to an OSPF autonomous system and another routing protocol's autonomous system.

Backbone routers
: Backbone routers in area zero. Area zero is considered the backbone in an OSPF network.

Designated router (DR)
: The DR is the router on a broadcast network that is elected to do the brunt of the OSPF processing. The DR will update all the other routers in the area with routes.

Backup designated router (BDR)
: The BDR is the router most eligible to become the DR should the DR fail.

Unlike other routing protocols, OSPF does not send routes, but rather *link state advertisements* (LSAs). Each OSPF router determines which routes to use based on an internal database compiled from these LSAs. There are six LSA types:

Router LSAs (type 1)
: Router LSAs are sent by every OSPF router into each connected area. These advertisements describe the router's links within the area.

Network LSAs (type 2)
: Network LSAs are sent by DRs and describe the routers connected to the network from which the LSA was received.

Summary LSAs for ABRs (type 3)
: Summary LSAs for ABRs are sent by ABRs. These advertisements describe interarea routes for networks. They are also used to advertise summary routes.

Summary LSAs for ASBRs (type 4)
: Summary LSAs for ASBRs are sent by ASBRs and ABRs. These advertisements describe links to ASBRs.

Autonomous System External (ASE) LSAs (type 5)
: ASE LSAs are sent by ASBRs and ABRs. These advertisements describe networks external to the autonomous system. They are sent everywhere except to stub areas.

Not So Stubby Area (NSSA) LSAs (type 7)
: NSSA LSAs are sent by NSSA ASBRs. Since type-5 LSAs are not allowed within the NSSA, type-7 LSAs are sent instead and stay within the NSSA. Type-7 LSAs get converted to type-5 LSAs by NSSA ASRs for advertisement into the backbone.

OSPF separates networks into areas. The core area, which all other areas must connect with, is area zero. One of OSPF's perceived benefits is that it forces you to design your network in such a way that there is a core with satellite areas. You can certainly build an OSPF network with only an area zero, but such a design usually doesn't scale well.

There are two main types of areas: *backbone* and *nonbackbone*. Area zero is the backbone area; all other areas are nonbackbone areas. Nonbackbone areas are further divided into the following types:

Normal area
: An OSPF area that is not area zero and is not configured as one of the other types. No special configuration is required.

Stub area
: An OSPF area that does not allow ASE LSAs. When an area is configured as a stub, no O E1 or O E2 routes will be seen in the area.

Totally stubby area (TSA)
: An OSPF area that does not allow type-3, -4, or -5 LSAs, except for the default summary route. TSAs see only a default route and routes local to the areas themselves.

Not so stubby area (NSSA)
: No type-5 LSAs are allowed in an NSSA. Type-7 LSAs that convert to type 5 at the ABR are allowed.

NSSA totally stub area
: NSSA totally stub areas are a combination of totally stubby and not so stubby areas. This area type does not allow type-3, -4, or -5 LSAs, except for the default summary route; it does allow type-7 LSAs that convert to type 5 at the ABR.

On Ethernet and other broadcast networks, OSPF elects a router to become the designated router and another to be the backup designated router. Calculating OSPF routes can be CPU-intensive, especially in a dynamic network. Having one router that does the brunt of the work makes the network more stable and allows it to converge faster. The DR calculates the best paths, then propagates that information to its neighbors within the network that are in the same area and OSPF process.

OSPF dynamically elects the DR through a relatively complicated process. The first step involves the router interface's OSPF *priority*. The default priority is 1, which is the lowest value an interface can have and still be elected the DR. A value of 0 indicates

that the router is ineligible to become the DR on the network. Setting the priority higher increases the chances that the router will be elected the DR. The OSPF interface priority is configured using the interface command `ip ospf priority`. The valid range is 0–255.

Ideally, you should plan which router is to become the DR and set its priority accordingly. Usually, there is an obvious choice, such as a hub or core router, or perhaps just the most powerful router on the network. The designated router will be doing more work than the other routers, so it should be the one with the most horsepower. If your design includes a hub router, that router will need to be the DR, because it will be the center of the topology.

If the OSPF interface priority is not set, resulting in a tie, the router will use the OSPF router ID to break the tie. Every router has an OSPF router ID. This ID can be configured manually with the `router-id` command. If the router ID is not configured manually, the router will assign it based on the IP address of the highest-numbered loopback address, if one is configured. If a loopback address is not configured, the router ID will be the highest IP address configured on the router. The only ways to change the router ID are to remove and reinstall the OSPF configuration or to reboot the router. Be careful, and think ahead when planning your network IP scheme.

When first deploying OSPF, engineers commonly make the mistake of neglecting the priority and router ID when configuring the routers. Left to its own devices, OSPF will usually pick routers that you would not choose as the DR and BDR.

A common network design using OSPF is to have a WAN in the core as area zero. Figure 10-12 shows such a network. Notice that all of the areas are designated with the same OSPF process number. Each area borders on area zero, and there are no paths between areas other than via area zero. This is a proper OSPF network design.

Area zero does not have to be fully meshed when you're using technologies such as Frame Relay. This is in part because OSPF recognizes the fact that there are different types of networks. OSPF knows that networks supporting broadcasts act differently from networks that are point-to-point and thus have only two active IP addresses. OSPF supports the following network types:

Point-to-point
: A point-to-point network is one with only two nodes on it. A common example is a serial link between routers, such as a point-to-point T1. No DR is chosen in a point-to-point network, because there are only two routers on the network. This is the default OSPF network type on serial interfaces with PPP or HDLC encapsulation.

Point-to-multipoint
: In a point-to-multipoint network, one hub router connects to all the other routers, but the other routers only connect to the hub router. Specifically, the remote

routers are assumed to be connected with virtual circuits, though only one IP network is used. No neighbors are configured and no DR is chosen. Area 0 in Figure 10-12 could be configured as a point-to-multipoint OSPF network.

Broadcast
: A broadcast network is an Ethernet, Token Ring, or FDDI network. Any number of hosts may reside on a broadcast network, and any host may communicate directly with any other host. A DR must be chosen and neighbors must be discovered or configured on a broadcast network. A broadcast network uses multicasts to send hello packets to discover OSPF routers. This is the default OSPF network type for Ethernet and Token Ring networks.

Nonbroadcast multiaccess (NBMA)
: In a nonbroadcast multiaccess network, all nodes may be able to communicate with one another, but they do not share a single medium. Examples include Frame Relay, X.25, and Switched Multimegabit Data Service (SMDS) networks. Because NBMA networks do not use multicasts to discover neighbors, you must manually configure them. Area 0 in Figure 10-12 could be configured as an NBMA network. This is the default OSPF network type on serial interfaces with Frame Relay encapsulation.

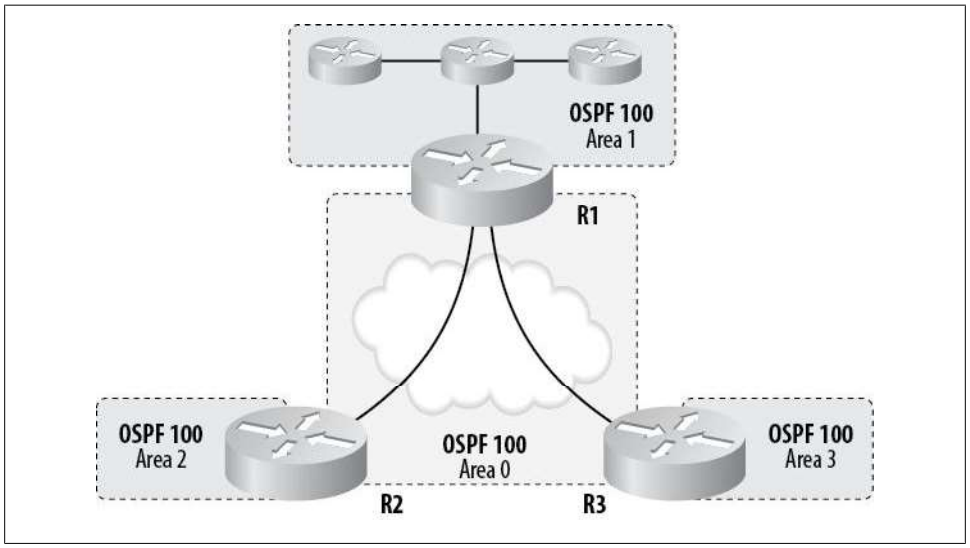

Figure 10-12. Simple OSPF network

OSPF enables interfaces using the `network` router command. It is a classless protocol, so you must use inverse subnet masks to limit the interfaces included. Unlike with EIGRP, you must include the inverse mask. If you do not, OSPF will not assume a classful network, but will instead report an error:

```
R3(config-router)#network 10.10.10.0
% Incomplete command.
```

In addition to the inverse mask, you must also specify the area in which the network resides:

```
R3(config-router)#network 10.10.10.0 0.0.0.255 area 0
```

My preference is to specifically configure interfaces so there are no surprises. You do this with an inverse mask of 0.0.0.0:

```
R3(config-router)#network 10.10.10.1 0.0.0.0 area 0
```

OSPF routes are marked by the letter `O` in the first column of the routing table:

```
R3#sho ip route
[Text Removed]

Gateway of last resort is 192.168.1.2 to network 0.0.0.0

     192.192.192.0/30 is subnetted, 1 subnets
C       192.192.192.4 is directly connected, Serial0/0
     172.16.0.0/32 is subnetted, 1 subnets
O IA    172.16.1.1 [110/11] via 10.10.10.4, 00:00:09, Ethernet0/0
     10.0.0.0/24 is subnetted, 1 subnets
C       10.10.10.0 is directly connected, Ethernet0/0
C    192.168.1.0/24 is directly connected, Ethernet1/0
S*   0.0.0.0/0 [254/0] via 192.168.1.2
```

Various OSPF route types are described in the routing table. They are: `O` (OSPF), `O IA` (OSPF interarea), `O N1` (OSPF NSSA external type 1), and `O N2` (OSPF NSSA external type 2).

OSPF stores its routes in a database, much like EIGRP. The command to show the database is `show ip ospf database`:

```
R3#sho ip ospf database

            OSPF Router with ID (192.192.192.5) (Process ID 100)

                Router Link States (Area 0)
Link ID         ADV Router      Age         Seq#       Checksum Link count
192.192.192.5   192.192.192.5   1769        0x8000002A 0x00C190 1

                Summary Net Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum
192.192.192.4   192.192.192.5   1769        0x8000002A 0x003415

                Router Link States (Area 1)

Link ID         ADV Router      Age         Seq#       Checksum Link count
192.192.192.5   192.192.192.5   1769        0x8000002A 0x00B046 1

                Summary Net Link States (Area 1)

Link ID         ADV Router      Age         Seq#       Checksum
10.10.10.0      192.192.192.5   1769        0x8000002A 0x0002A2
```

```
OSPF Router with ID (192.168.1.116) (Process ID 1)
```

If all of this seems needlessly complicated to you, you're not alone. The complexity of OSPF is one of the reasons many people choose EIGRP instead. If you're working in a multivendor environment, however, EIGRP is not an option.

BGP

BGP is a very different protocol from the others described here. The most obvious difference is that BGP is an external gateway protocol, while all the previously discussed protocols were internal gateway protocols. BGP can be hard to understand for those who have only ever dealt with internal protocols like EIGRP and OSPF, because the very nature of the protocol is different. As BGP is not often seen in the corporate environment, I'll cover it only briefly here.

BGP does not deal with hops or links, but rather with autonomous systems. A network in BGP is referred to as a *prefix*. A prefix is advertised from an autonomous system. BGP then propagates that information through the connected autonomous systems until all the autonomous systems know about the prefix.

Routes in BGP are considered most desirable when they traverse the least possible number of autonomous systems. When a prefix is advertised, the autonomous system number is prefixed onto the autonomous system *path*. This path is the equivalent of a route in an internal gateway protocol. When an autonomous system learns of a prefix, it learns of the path associated with it. When the autonomous system advertises that prefix to another autonomous system, it prepends its own ASN to the path. As the prefix is advertised to more and more autonomous systems, the path gets longer and longer. The shorter the path, the more desirable it is.

Figure 10-13 shows a simple example of BGP routing in action. The network 10.0.0.0/8 resides in AS 105, which advertises this prefix to AS 3 and AS 2. The path for 10.0.0.0/8 within AS 3 and AS 2 is now 10.0.0.0/8 AS105. AS 2 in turn advertises the prefix to AS 1, prepending its own ASN to the path. AS 1 now knows the path to 10.0.0.0/8 as 10.0.0.0/8 AS2, AS105. Meanwhile, AS 3 advertises the prefix to AS 100, which then knows the path to 10.0.0.0/8 as 10.0.0.0/8 AS3, AS105.

On the other side of the world, AS 102 receives two paths:

```
> 10.0.0.0/8 AS1, AS2, AS105
  10.0.0.0/8 AS101, AS100, AS3, AS105
```

The > on the first line indicates that BGP considers this the preferred path. The path is preferred because it is the shortest path among the known choices.

What makes BGP so confusing to newcomers is the many attributes that can be configured. A variety of weights can be attributed to paths, with names like local preference, weight, communities, and multiexit discriminator. To make matters worse, many of these attributes are very similar in function.

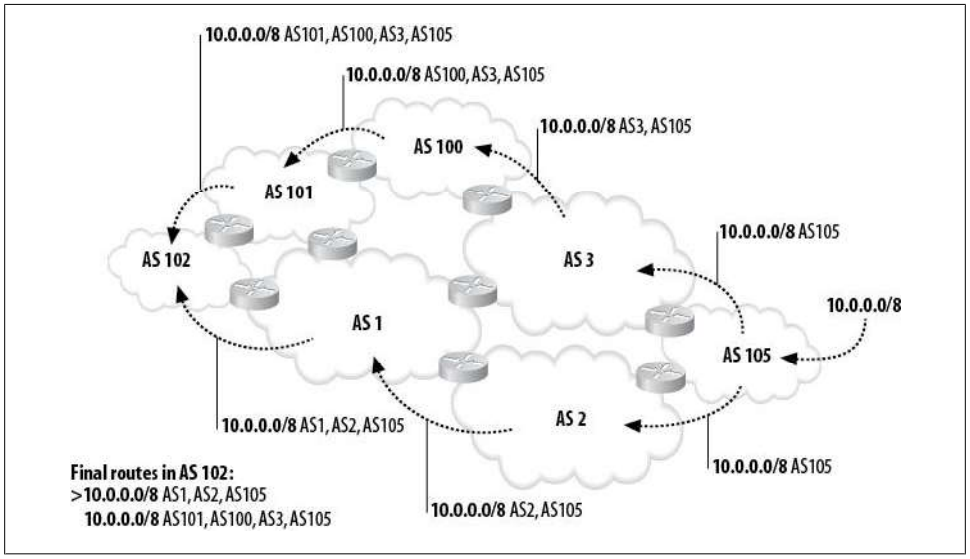

Figure 10-13. Routing in BGP

The protocol also functions differently from other protocols. For example, the `network` statement, which is used to enable interfaces in other protocols, is used in BGP to list the specific networks that can be advertised.

BGP does not discover neighbors; they must be configured manually. There can only be one autonomous system on any given router, though it may communicate with neighbors in other autonomous systems.

BGP is the routing protocol of the Internet. Many of the major service providers allow anonymous Telnet into *route servers* that act just like Cisco routers. Do an Internet search for the term "looking-glass routers," and you should find plenty of links. These route servers are an excellent way to learn more about BGP, as they are a part of the largest network in the world and have active routes to just about every public network on Earth. Unless you're working at a tier-1 service provider, where else could you get to poke around with a BGP router that has 20 neighbors, 191,898 prefixes, and 3,666,117 paths? I have a pretty cool lab, but I can't compete with that! Here is the output from an actual route server:

```
route-server>sho ip bgp summary
BGP router identifier 10.1.2.5, local AS number 65000
BGP table version is 208750, main routing table version 208750
191680 network entries using 19359680 bytes of memory
3641563 path entries using 174795024 bytes of memory
46514 BGP path attribute entries using 2605064 bytes of memory
42009 BGP AS-PATH entries using 1095100 bytes of memory
4 BGP community entries using 96 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 197854964 total bytes of memory
```

```
Dampening enabled. 2687 history paths, 420 dampened paths
191529 received paths for inbound soft reconfiguration
BGP activity 191898/218 prefixes, 3666117/24554 paths, scan interval 60 secs

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.0.0.2        4  7018       0       0        0    0    0 never    Idle (Admin)
12.0.1.63       4  7018   45038     188   208637    0    0 03:04:16        0
12.123.1.236    4  7018   39405     189   208637    0    0 03:05:02   191504
12.123.5.240    4  7018   39735     189   208637    0    0 03:05:04   191504
12.123.9.241    4  7018   39343     189   208637    0    0 03:05:03   191528
12.123.13.241   4  7018   39617     188   208637    0    0 03:04:20   191529
12.123.17.244   4  7018   39747     188   208637    0    0 03:04:58   191505
12.123.21.243   4  7018   39441     188   208637    0    0 03:04:28   191528
12.123.25.245   4  7018   39789     189   208637    0    0 03:05:07   191504
12.123.29.249   4  7018   39602     188   208637    0    0 03:04:16   191505
12.123.33.249   4  7018   39541     188   208637    0    0 03:04:16   191528
12.123.37.250   4  7018   39699     188   208637    0    0 03:04:26   191529
12.123.41.250   4  7018   39463     188   208637    0    0 03:04:19   191529
12.123.45.252   4  7018   39386     188   208637    0    0 03:04:20   191505
12.123.133.124  4  7018   39720     188   208637    0    0 03:04:20   191528
12.123.134.124  4  7018   39729     188   208637    0    0 03:04:22   191529
12.123.137.124  4  7018   39480     188   208637    0    0 03:04:15   191528
12.123.139.124  4  7018   39807     188   208637    0    0 03:04:24   191528
12.123.142.124  4  7018   39748     188   208637    0    0 03:04:22   191505
12.123.145.124  4  7018   39655     188   208637    0    0 03:04:23   191529
```

These route servers can get pretty busy and very slow. If you find yourself waiting too long for a response to a query, either wait a bit and try again, or try another route server.

Choose your favorite public IP network (doesn't everyone have one?) and see how the paths look from the looking-glass router. If you don't have a favorite, choose one that you can easily figure out, like one in use by www.cisco.com or www.oreilly.com:

```
[bossman@myserver bossman]$nslookup www.oreilly.com
Server:  localhost
Address:  127.0.0.1

Name:    www.oreilly.com
Addresses:  208.201.239.36, 208.201.239.37
```

Once you have the address, you can do a lookup for the network:

```
route-server>sho ip bgp 208.201.239.0
BGP routing table entry for 208.201.224.0/19, version 157337
Paths: (19 available, best #15, table Default-IP-Routing-Table)
  Not advertised to any peer
  7018 701 7065, (received & used)
    12.123.137.124 from 12.123.137.124 (12.123.137.124)
      Origin IGP, localpref 100, valid, external, atomic-aggregate
      Community: 7018:5000
  7018 701 7065, (received & used)
    12.123.33.249 from 12.123.33.249 (12.123.33.249)
```

```
    Origin IGP, localpref 100, valid, external, atomic-aggregate
    Community: 7018:5000
7018 701 7065, (received & used)
  12.123.29.249 from 12.123.29.249 (12.123.29.249)
    Origin IGP, localpref 100, valid, external, atomic-aggregate
    Community: 7018:5000
7018 701 7065, (received & used)
  12.123.41.250 from 12.123.41.250 (12.123.41.250)
    Origin IGP, localpref 100, valid, external, atomic-aggregate
    Community: 7018:5000
7018 701 7065, (received & used)
  12.123.1.236 from 12.123.1.236 (12.123.1.236)
    Origin IGP, localpref 100, valid, external, atomic-aggregate, best
    Community: 7018:5000
```