,ch01.21583 Page 1 Wednesday, January 9, 2002 12:23 PM

In this chapter:

- What Is IP Routing?
- Directly Connected Networks
- Static Routing
- Dynamic Routing
- The Routing Table
- Underlying Processes
- Summing Up

CHAPTER 1
Starting Simple

What Is IP Routing?

A young woman boards a commuter train in a small town in Quebec, changes trains a couple of times, and, a day later, arrives in New York City. She walks up the stairs from the platform into Grand Central Terminal, looks up above her head, and, for the first time, sees the constellations, hundreds of feet above on the ceiling.

A high school student in New Zealand downloads maps of Sri Lanka from a local (Sri Lankan) web site. The maps show the natural features, the political boundaries, the flora and fauna, rainfall, ancient kingdoms, languages, and religions. The download takes thousands of IP packets that find their way from Sri Lanka to the student's PC in New Zealand.

Just as our Canadian friend changed trains at several stations along the way, the IP packets from the Sri Lankan web site may have bounced through dozens of routers before arriving at the student's machine.

The routing of IP packets in an IP network is the set of tasks required to move an IP packet from router to router to its destination, as specified in the IP header field. This book is about the set of tasks that accomplish IP routing.

There are similarities in routing concepts between IP networks, transportation systems, and mail delivery operations. Throughout this text, we will often illustrate IP routing concepts by comparison with these other systems.

Directly Connected Networks

When our Canadian visitor finally picks up her bags and is ready to head out of Grand Central Terminal, she looks around for the exit signs. On one end, below a row of immense windows, is a sign saying "Vanderbilt Avenue." Below the opposite row of tall windows is a sign saying "Lexington Avenue." Under the large stone arches is a sign reading "42nd Street" (Figure 1-1).

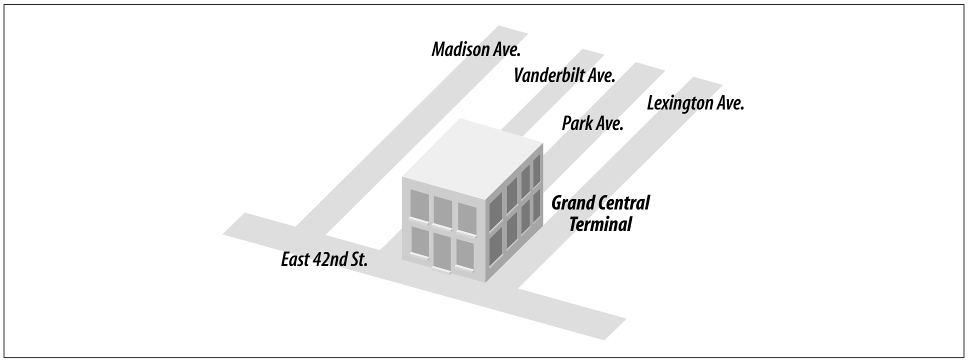

Figure 1-1. Grand Central Terminal and the adjoining streets

Just as the streets around Grand Central Terminal are immediately accessible to any traveler, a router has directly attached networks that are immediately accessible (in other words, that do not require any specific routing mechanism to discover). Consider router *R*, in the following example. Networks `1.0.0.0`, `10.1.1.0`, and `10.1.2.0` are directly connected to the router:

```
hostname R
!
interface Ethernet0
ip address 1.1.1.1 255.0.0.0
!
interface Ethernet1
ip address 10.1.1.4 255.255.255.0
!
interface Ethernet2
ip address 10.1.2.4 255.255.255.0
...
```

In fact, the moment these networks are connected to the router they are visible in *R*'s routing table. Note in the following output that the command to display the routing table is *show ip route* (in EXEC mode). Also note the "C" that is prepended to the entries in the routing table, indicating that the routes were discovered as directly connected to the router:

```
R#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
```

```
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

C    1.0.0.0/8 is directly connected, Ethernet0
     10.0.0.0/8 is subnetted, 2 subnets
C       10.1.1.0/24 is directly connected, Ethernet1
C       10.1.2.0/24 is directly connected, Ethernet2
```

Directly connected networks are automatically installed in the routing table if the interface to the network is up. Figure 1-2 shows router *R* with its directly connected networks. (The EXEC command *show interface* will show the state of the interfaces). In the previous example, it is assumed that all three interfaces to the directly connected networks are up. If an interface to a directly connected network goes down, the corresponding route is removed from the routing table.

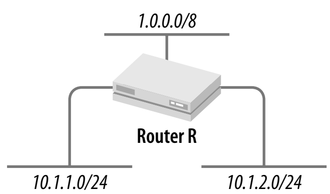

Figure 1-2. Router R with its directly connected networks

If multiple IP addresses are attached to an interface (using secondary addresses), all the associated networks are installed in the routing table.

Static Routing

Our Canadian friend has always wanted to see the New York Public Library. She gets directions at the information booth: "Make a right on 42nd Street; walk three blocks; look for the lions in front of the library." The information-booth attendant may have no idea that the library is closed that day, or that the sidewalk on 42nd Street is blocked just then because of fire trucks and 41st Street may be the preferable route. The information booth has given the same directions to the library for the last hundred years and hopefully will for hundreds more—the route from Grand Central Station to the library, in other words, is static.

In a similar vein, a network administrator can create a static route. So, to reach network `146.1.0.0`, we may add the command:

```
ip route 146.1.0.0 255.255.0.0 1.1.1.2
```

which says to get to network `146.1.0.0/16`, go to the next hop of `1.1.1.2`. This specifies a fixed path to `146.1.0.0/16`, as shown here, where the contents of the routing table are displayed using the EXEC command *show ip route*:

```
  R#sh ip route
  ...
1 S    146.1.0.0/16 [1/0] via 1.1.1.2
```

Even if `1.1.1.2` goes down, an alternate path—shown via *R2* in Figure 1-3—cannot be used until a second static route is specified:

```
ip route 146.1.0.0 255.255.0.0 1.1.1.3
```

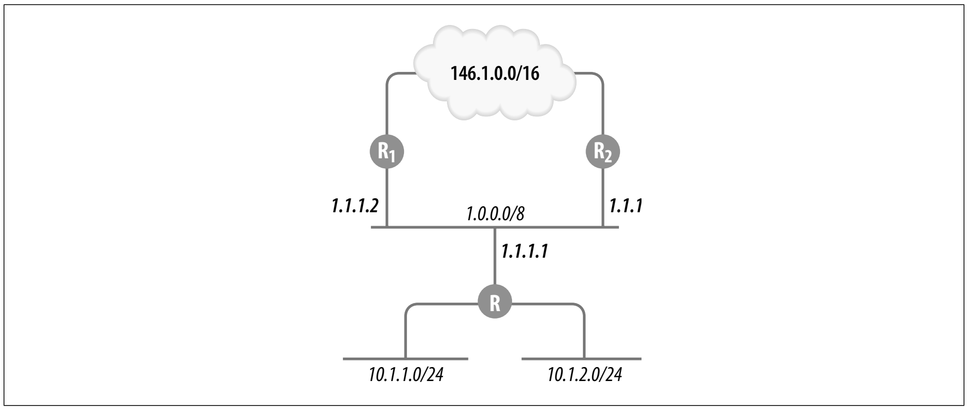

Figure 1-3. Router R's connectivity to 146.1.0.0

The syntax of the static route command is:

```
ip route network [mask] {address | interface} [distance]
```

where *network* and *mask* specify the IP address and mask of the destination. The next hop may be specified by its IP address or by the interface on which to send the packet. To point a static route to an interface (*Ethernet0* in this case), use:

```
ip route 146.1.0.0 255.255.0.0 interface Ethernet0
```

Static routes are smart to the extent that if the next hop (interface or IP address) specified goes down, the router will remove the static route entry from the routing table.

In line 1, the static route in the routing table is accompanied by "[1/0]". This specifies the administrative distance and the metric associated with the route. We'll discuss distance and metrics in the next section.

As should be obvious, static routing does not scale well. As the network grows, the task of maintaining static routes becomes more and more horrendous.

Dynamic Routing

After the public library, our Canadian visitor jumps into a taxi to go crash at a friend's place in Brooklyn. "Go over the Brooklyn Bridge," she tells the driver. They head downtown. Suddenly, the driver slams on his brakes and makes an abrupt turn. Cars all around jam on their brakes, and pedestrians run hither and thither. "The radio said it is an hour to go over the bridge! We will take the tunnel!" the driver shouts to the back seat. This is an example of dynamic routing in a transportation system. What is dynamic routing in IP networks? Dynamic routing protocols allow each router to automatically discover one or more paths to each destination in the network. When the network topology changes, such as when new paths are added or when paths go out of service, dynamic routing protocols automatically adjust the contents of the routing table to reflect the new network topology.

Dynamic routing relies on (frequent!) updates to discover changes in network topology. In the example in Figure 1-3, when the path *R3* → *R4* is added to the network it can be automatically discovered by a routing protocol, such as RIP, EIGRP, or OSPF.

The routing protocols in use today are based on one of two algorithms: *Distance Vector* or *Link State*. Distance Vector (DV) algorithms broadcast routing information to all neighboring routers. In other words, each router tells all of its neighbors the routes it knows. When a router receives a route (from a neighbor) that is not in its routing table, it adds the route to its table; if the router receives a route that is already in its routing table, it keeps the *shorter* route in its table. DV algorithms are sometimes also described as routing by rumor: bad routing information propagates just as quickly as good information. Link State algorithms operate on a different paradigm. First, each router constructs its own topological map of the entire network, based on updates from neighbors. Next, each router uses Dijkstra's algorithm to compute the *shortest* path to each destination in this graph. Both DV and Link State algorithms are described in further detail in the chapters that follow.

In the previous paragraph, we spoke of the "shorter" or "shortest" path in the context of both DV and Link State algorithms. Since a router may know of multiple paths to a destination, each routing protocol must provide a mechanism to discover the "shorter" or "shortest" path based on one or more of the following criteria: number of hops, delay, throughput, traffic, reliability, etc. A *metric* is usually attached to this combination; lower metric values indicate "shorter" paths. For each routing protocol discussed in the chapters that follow, we will describe how the route metric is computed.

A network under a single administrative authority is described as an *autonomous system* (AS) in routing parlance. *Interior gateway protocols* (IGPs) are designed to support the task of routing internal to an AS. IGPs have no concept of political boundaries between ASs or the metrics that may be used to select paths between ASs. RIP, IGRP, EIGRP, and OSPF are IGPs. *Exterior gateway protocols* (EGPs) are designed to support routing between ASs. EGPs deploy metrics to select one inter-AS path over another. BGP is the most commonly used EGP.

Routing architectures may be broadly classified as *flat* or *hierarchical*. Flat routing implies that all routes are known to all peers—all routers in the network are equal, possessing the same routing information. Hierarchical routing implies that some routers possess only local routes, whereas others possess a little bit more information, and still others possess even more.

Let's draw an analogy to the postal system. When I write a letter to a friend in India, the postman in the U.S. may have no idea where India is. He forwards all foreign mail to a designated post office in his state. That designated post office must know every postal system in the world. Such a system, in which some post offices are regional and some handle foreign mail, could be described as hierarchical.

In large IP networks, only a few routers need to know every route in the network. These routers are sometimes described as *core* routers. Around the core routers is a layer of *distribution* routers that need not possess the complete routing table. When a distribution router receives a packet whose destination IP address does not appear in its local routing table, the distribution router simply forwards the packet to a core router.

In the earlier example of the high school student in New Zealand accessing a web site in Sri Lanka, the small router in the high school in New Zealand probably has only a tiny routing table, with no routing entries for Sri Lanka. The high school router will forward all traffic for unknown destinations to another router, which in turn may forward the traffic to another one. Large IP networks exhibit several layers of hierarchy.

As we will see in the chapters that follow, some routing protocols have features that make it easier to build hierarchies. These features include route aggregation, classlessness, the use of default routes, and the flexibility with which routes can be exchanged with other routing protocols.

RIP is an example of an almost completely flat routing protocol. OSPF exhibits several features that permit the design of hierarchical networks.

As with any other algorithm, routing algorithms may also be categorized based on their complexity, flexibility, overhead, memory and CPU utilization, robustness, and stability. These properties of routing algorithms are of interest to the routing engineer, since he provides the (router) infrastructure to execute these algorithms.

The Routing Table

At Grand Central Terminal, a big wall lists all the destinations and their corresponding track numbers (see Figure 1-4). Passengers find their destination on this wall and then proceed to the indicated platforms. Similarly, a routing table must contain at least two pieces of information: the destination network and the next hop toward that destination. This reflects a fundamental paradigm of IP routing: hop-by-hop routing. In other words, a router does not know the full path to a destination, but only the next hop to reach the destination.

Departures

Time	*Destination*	*Track number*
9:18	New Haven	17
9:21	Cos Cob	22
9:24	Valhalla	11
9:31	Dover Plains	19
9:42	Bronxville	12

Figure 1-4. Destinations and track numbers at Grand Central Terminal

Routes are installed in the routing table as they are learned through the mechanisms we have been discussing: directly connected networks, static routes, and dynamic routing protocols. A typical routing table in a Cisco router looks like this:

```
   Router>show ip route
   Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
          D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
          N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
          E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
          i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default

   Gateway of last resort is 0.0.0.0 to network 0.0.0.0


2       177.130.0.0/30 is subnetted, 2 subnets
   C       177.130.17.152 is directly connected, Serial1
   C       177.130.17.148 is directly connected, Serial0
3       10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
4  S       10.0.0.0/8 [1/0] via 160.4.115.74
5  S       10.254.101.0/24 [1/0] via 160.4.101.4
6       162.162.0.0/24 is subnetted, 2 subnets
   O IA    162.162.101.0 [110/3137] via 11.175.238.4, 02:16:02, Ethernet0
                         [110/3137] via 11.175.238.3, 02:16:02, Ethernet0
   O IA    162.162.253.0 [110/3127] via 11.175.238.4, 02:25:43, Ethernet0
                         [110/3127] via 11.175.238.3, 02:25:43, Ethernet0
7  O E2 192.188.106.0/24 [110/20] via 11.175.238.33, 20:49:59, Ethernet0
   ...
```

Note that the first few lines of the output attach a code to the source of the routing information: “C” and “S” denote “connected” and “static”, respectively, as we saw earlier, “I” denotes IGRP, etc. This code is prepended to each routing entry in the routing table, signifying the source of that route.

The body of the routing table essentially contains two pieces of information: the destination and the next hop. So, 177.130.0.0 (line 2) has two subnets, each with a 30-bit mask. The two subnets are listed in the following two lines.

Line 3 shows an interesting case. 10.0.0.0 has two subnets: 10.0.0.0/8 and 10.254.101.0/24. Not only are the subnet masks different, but the subnets are overlapping. A destination address of 10.254.101.1 would match both route entries! So, should a packet for 10.254.101.1 be routed to 160.4.115.74 or 160.4.101.4? Routing table lookups follow the rule of *longest prefix match*. 10.254.101.1 matches 8 bits on line 4 and 24 bits on line 5—the longer prefix wins, and the packet is forwarded to 160.4.101.4. 162.162.0.0 (line 6) has two subnets, each of which is known via two paths. 192.188.106.0 (line 7) is not subnetted.

What if a route is learnt via multiple sources—say, via OSPF and as a static entry? Each source of routing information has an attached measure of its trustworthiness, called *administrative distance* in Cisco parlance. The lower the administrative distance, the more trustworthy the source.

Table 1-1 shows the default administrative distances.

Table 1-1. Default administrative distances

Route source	Default distance
Connected interface	0
Static route	1
External BGP	20
IGRP	100
OSPF	110
IS-IS	115
RIP	120
EGP	140
Internal BGP	200
Unknown	255

Thus, if a route is known both via OSPF and as a static entry, the static entry, not the entry known via OSPF, will be installed in the routing table.

Note that distance information and the route metric appear in the output of *show ip route* inside square brackets with the distance information first, followed by a "/" and the route metric: [distance/metric].

Administrative distance is only considered internally within a router; distance information is not exchanged in routing updates.

Underlying Processes

Behind the scenes, there are three key sets of processes running on each router that make up IP routing. I have already discussed examples from each of these three sets in the preceding sections. These processes may be organized into three categories:

1. Processes associated with the discovery of paths to various destinations in the network. These processes include dynamic routing protocols, such as RIP and IGRP, as well as static route entries. This text describes these processes in detail.
2. Processes that maintain the IP routing table. These processes receive updates from all dynamic routing protocols running on the router as well as from static route entries. By attaching administrative distance values to each routing information source, these processes break ties when multiple sources (e.g., OSPF and static route entries) report paths to the same destination. I discussed the use of administrative distance values in the previous section. Other examples from this group of processes will be discussed in Chapter 8.
3. Processes involved with the forwarding of IP packets. These processes are invoked when a router receives a packet to forward. The result of the match between the destination IP address in the packet and the contents of the IP routing table may be a match with one entry in the routing table, a match with more than one entry in the routing table, a match with a default route, etc. One general rule here is the rule of longest prefix match—if there is more than one match, the match with the longest subnet mask (or prefix) wins. Further, the outcome of these processes depends on whether the router is configured for classful or classless route lookups.

Several concepts that have not yet been discussed were thrown into the preceding discussion. For instance, we have not yet talked about classful versus classless route lookups or about default routes. These concepts will be addressed in later chapters. However, this early lesson in the division of processes should help you to understand and classify concepts more quickly.

Summing Up

Dynamic routing protocols are the mainstay of IP routing. Thus, without ado, I will begin with RIP and then, moving on in order of complexity, will discuss IGRP, EIGRP, OSPF, and BGP-4.