CHAPTER

14

IGPs: RIP, OSPF, and IS–IS

What You Will Learn

In this chapter, you will learn about the role of IGPs and how these routing protocols are used in a routing domain or autonomous system (AS). We'll use OSPF and RIP, but mention IS–IS as well.

You will learn how a routing policy can distribute the information gathered from one routing protocol into another, where it can be used to build routing and forwarding tables, or *announced* (sent) to other routers. We'll create a routing policy to announce our IPv6 routes to the other routers.

As is true of many chapters in this book, this chapter's content is more than enough for a whole book by itself. Only the basics of IGPs are covered here, but they are enough to illustrate the function of an internal routing protocol on our network.

In this chapter, we'll configure an IGP to run on the Juniper Networks routers that make up the Illustrated Network. In Chapter 9 we saw output that showed OSPF running on router CE6 as part of Best ISP's AS. So first we'll show how OSPF was configured on the routers in AS 65127 and AS 65459. We could configure IS–IS on the other AS, but that would make an already long chapter even longer. Because we closed the last chapter with IPv6 ping messages not working, let's configure RIPng, the version of RIP that is for IPv6. This is not an endorsement of RIPng, especially given other available choices. It's just an example.

Why not add OSPFv3 (the version of OSPF used with IPv6) for IPv6 support? We certainly could, but suppose the smaller site routers only supported RIP or RIPng? (RIP is usually bundled with basic software, but other IGPs often have to be purchased.) Then we would have no choice but to run RIPng to distribute the IPv6 addresses. If we configure RIPng to run on the ASs between on-site routers CE0 and CE6, we can always extend RIPng support right to the Unix hosts (the IPv6 hosts just need to point to CE0 or CE6 as their default routers).

In this chapter, we'll use the routers heavily, as shown in Figure 14.1.

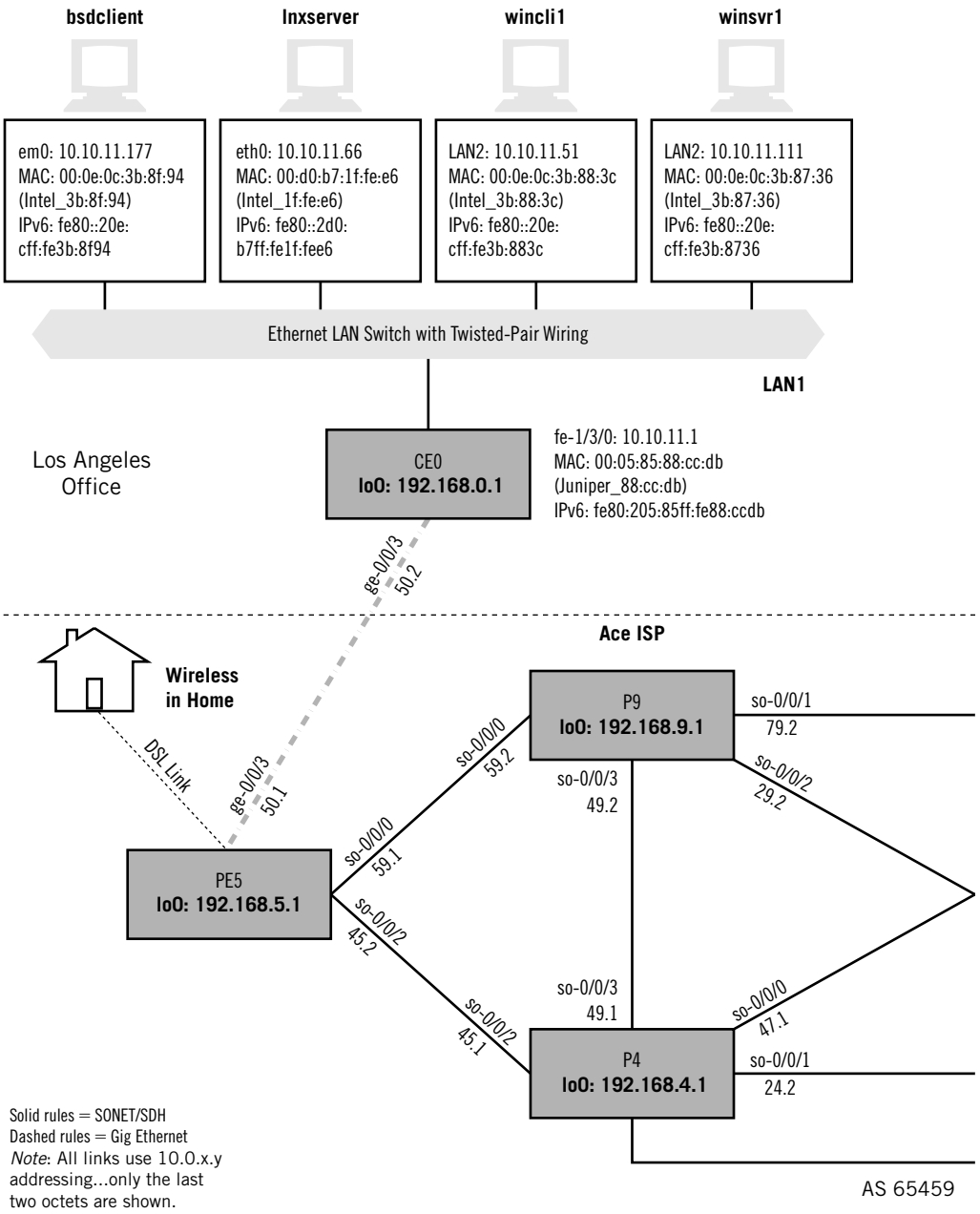
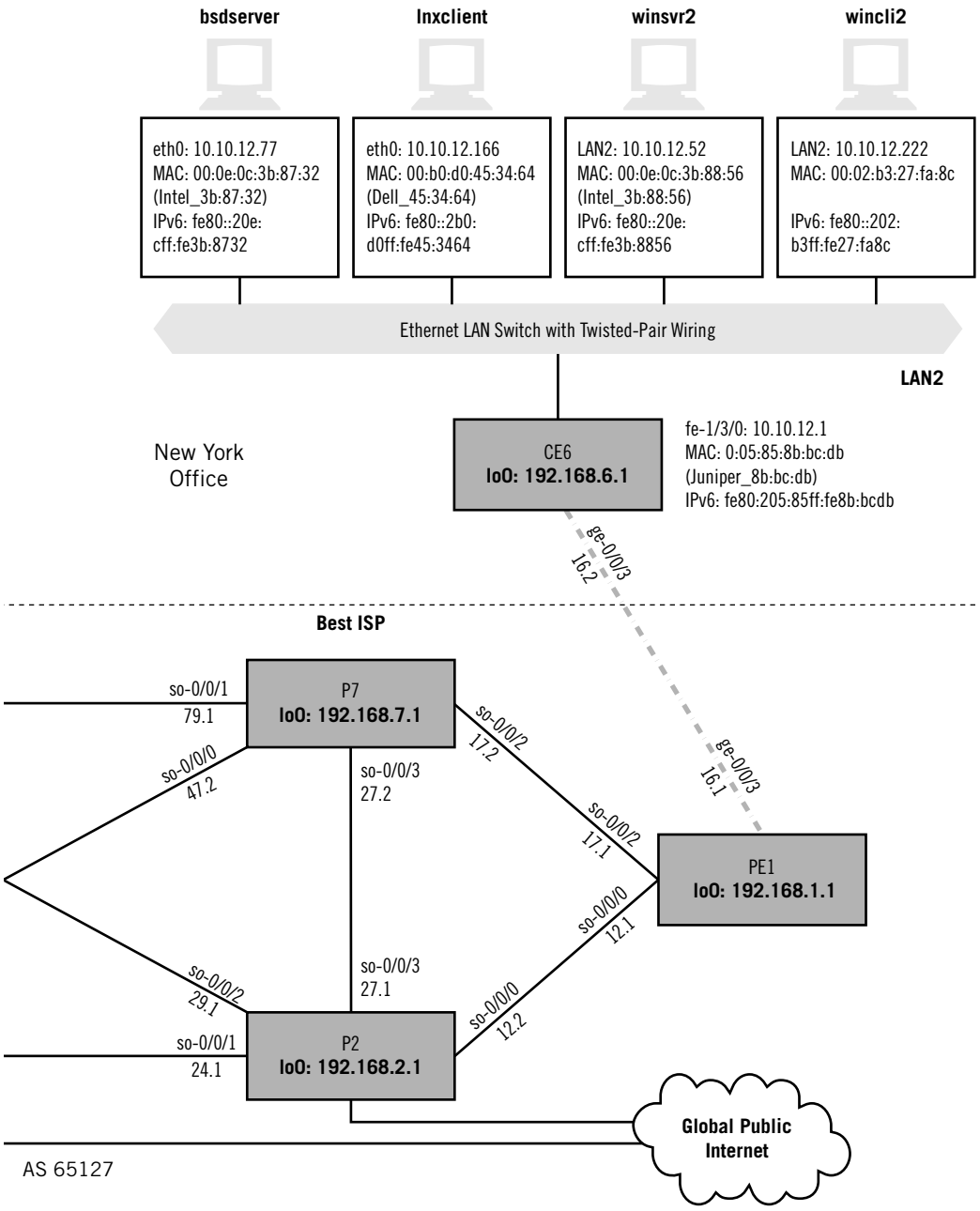


FIGURE 14.1

The routers on the Illustrated Network, showing routers on which OSPF and RIPng will be running. The IGPs will not be running between the two AS routing domains; instead, an EGP will run.


bsdserver
lnxclient
winsvr2
wincli2
eth0: 10.10.12.77
MAC: 00:0e:0c:3b:87:32
(Intel_3b:87:32)
IPv6: fe80::20e:
cff:fe3b:8732
eth0: 10.10.12.166
MAC: 00:b0:d0:45:34:64
(Dell_45:34:64)
IPv6: fe80::2b0:
d0ff:fe45:3464
LAN2: 10.10.12.52
MAC: 00:0e:0c:3b:88:56
(Intel_3b:88:56)
IPv6: fe80::20e:
cff:fe3b:8856
LAN2: 10.10.12.222
MAC: 00:02:b3:27:fa:8c
IPv6: fe80::202:
b3ff:fe27:fa8c
Ethernet LAN Switch with Twisted-Pair Wiring
LAN2
New York
Office
CE6
lo0: 192.168.6.1
fe-1/3/0: 10.10.12.1
MAC: 0:05:85:8b:bc:db
(Juniper_8b:bc:db)
IPv6: fe80:205:85ff:fe8b:bcdb
ge-0/0/3
16.2
Best ISP
so-0/0/1
79.1
P7
lo0: 192.168.7.1
so-0/0/2
17.2
so-0/0/0
47.2
so-0/0/3
27.2
ge-0/0/3
16.1
so-0/0/2
17.1
PE1
lo0: 192.168.1.1
so-0/0/0
12.1
so-0/0/3
27.1
so-0/0/2
29.1
so-0/0/0
12.2
so-0/0/1
24.1
P2
lo0: 192.168.2.1
Global Public
Internet
AS 65127

Unfortunately, when it comes to networks, a lot of things are interrelated, although we'd like to learn them sequentially. For example, we've already shown in Chapter 9 that OSPF is configured on the routers, although we didn't configure it. Also, although both ASs will run the same IGP (RIPng) in this chapter, the ASs are *not* running RIPng or any other IGP in between (e.g., on the links between routers P9 and P7). That's the job of the EGP, which we'll explore in the next chapter. There is a lot going on in this chapter, so let's list the topics covered here (and in Chapter 15), so we don't get lost.

1. We'll talk about ASs and the role of IGP and EGPs on a network.
2. We'll configure RIPng as the IGP in both ASs, starting with the IPv6 address on the interfaces and show that the routing information about LAN1 and LAN2 ends up everywhere. We will not talk about the role of the EGP in all this until Chapter 15.
3. We'll compare three major IGPs: RIP, OSPF, and IS–IS. In the OSPF section, we'll show how OSPF was configured in the two ASs for Chapter 9.

Internal and External Links

In this chapter, we'll add RIPng as an IGP on all but the links between AS 65459 and AS 65127. This affects routers P9 and P4 in AS 65459 and routers P7 and P2 in AS 65127. IGPs run on internal (*intra*-AS) links, and EGPs run on external (*inter*-AS) links.

In Chapter 15, we'll configure BGP as the EGP on those links. This chapter assumes that BGP is up and running properly on the external links between P9 and P4 in AS 65459 and P7 and P2 in AS 65127.

We'll use our Windows XP clients for this exercise, just to show that the "home version" of XP is completely comfortable with IPv6.

Autonomous System Numbers

Ace and Best ISP on the Illustrated Network use AS numbers (ASNs) in the private range, just as our IP addresses. IANA parcels them out to the various registries that assign them as needed to those who apply. Before 2007, AS numbers were 2-byte (16-bit) values with the following ranges of relevance:

- **0:** Reserved (can be used to identify nonrouted networks)
- **1–43007:** Allocated by ARIN, APNIC, AfriNIC, and RIPE NCC
- **43008–48127:** Held by IANA
- **48128–64511:** Reserved by IANA
- **64512–65534:** Designated by IANA for private use
- **65535:** Reserved

Since 2007, ASNs are allocated as 4-byte values. Because each field can run from 0 to 65535, the current way of designating ASNs is as two numbers in the form *nnnnn.nnnnnn*. The full range of ASNs now is from 0.0 to 65535.65535 (0 to 4,294,967,295 in decimal).

For example, 0.65525 is how the former 2-byte ASN 65535 would be written today. In this book, we'll drop the leading "0," and just use the "legacy" 2-byte AS format for Ace and Best ISP: 65459 and 65127.

Now, let's see what it takes to get RIPng up and running on these routers. So far, the link-local fe80 addresses have been fine for running ping and for neighbor discovery from router to host, but these won't be useful for LAN1 to LAN2 communications with IPv6. For this, we'll use routable `fc00` private ULA IPv6 addresses. Once we get RIPng up and running with routable addresses on our hosts and routers, we should be able to successfully ping from LAN1 to LAN2 using only IPv6 addresses. While we'll be configuring IGPs on both Ace and Best ISP's AS routing domains, we *won't* be running IGPs between them. That's the job of the EGP (Border Gateway Protocol, or BGP), and we'll add that in Chapter 15.

We need to create four routable IPv6 addresses and prefixes—two for the hosts and two for the router's LAN interfaces (both are `fe-1/3/0`). We've already done this in Chapter 4. The site IPv6 addresses, and the IPv4 and MAC addresses used on the same interfaces, are shown in Table 14.1. We don't need to change the link-local addresses on the link between the routers because, well, they are link-local.

We know from Chapter 13 that we have these IPv6 addresses configured on `wincli1` and `wincli2`. We have to do three things to enable RIPng on the routers:

- Configure routable addresses on interface `fe-1/3/0`
- Configure the RIPng protocol to run on the site (customer-edge) routers (CE0 and CE6), the provider-edge routers (PE5 and PE1), and the internal links on the provider-backbone routers (P9, P7, P4, and P2).
- Create and apply a routing policy on CE0 and CE6 to advertise the `fe-1/3/0` IPv6 addresses with RIPng.

Table 14.1 Routable IPv6 Addresses Used on the Network

System	IPv4 Network Address	MAC Address	IPv6 Address
wincli1	10.10.11/24	02:0e:0c:3b:88:3c	fc00:ffb3:d5:b:20e:cff:fe3b:883c
CE0 (fe-1/3/0)	10.10.11/24	00:05:85:88:cc:db	fc00:ffb3:d5:b:205:85ff:fe88:ccdb
CE6 (fe-1/3/0)	10.10.12/24	00:05:85:8b:bc:db	fc00:fe67:d4:c:205:85ff:fe8b:bcdb
Wincli2	10.10.12/24	00:02:b3:27:fa:8c	fc00:fe67:d4:c:202:b3ff:fe27:fa8c

The configurations are completely symmetrical, so one of each type will do for illustration purposes. Let's use router CE0 as the customer-edge router. First, the addresses for IPv4 (`family inet`) and IPv6 (`family net6`) must be configured on LAN interface `fe-1/3/0`.

```
set interfaces fe-1/3/0 unit 0 family inet address 10.10.11.1/24
set interfaces fe-1/3/0 unit 0 family inet6 address fe80::205:85ff:fe88:ccdb/64
set interfaces fe-1/3/0 unit 0 family inet6 address fc00:fe67:d4:c:205:85ff:fe88:
    ccdb/64
```

Note that the link-local address is fine as is. We usually have many addresses on an interface in most IPv6 implementations, including multicast. We just added the second address to it. Now we can configure RIPng itself on the link between CE0 and PE5. We have to explicitly tell RIPng to announce (export) the routing information specified in the `send-ipv6` routing policy (which we'll write shortly) and tell it the RIPng "neighbor" (routing protocol partner) is found on interface `ge-0/0/3` logical unit 0.

```
set protocols ripng group ripv6group export send-ipv6
set protocols ripng group ripv6group neighbor ge-0/0/3.0
```

Because RIPv2 and RIPng use multicast addresses, we specify the router's neighbor location with the *physical* address information (`ge-0/0/3`) instead of unicast address. And because Juniper Network's implementation of RIP always listens for routing information but never advertises or announces routes unless told, we'll have to write a routing policy to "export" the IPv6 addresses we want into RIPng. There's only one interface needed in this case, `fe-1/3/0.0` to LAN1. It seems odd to say `from` when sending, but in a Juniper Networks routing policy, `from` really means "out of"—"Out of all the interfaces, this applies to interface `fe-1/3/0`."

```
set policy-options policy-statement send-ipv6 from interface fe-1/3/0.0
set policy-options policy-statement send-ipv6 from family inet6
set policy-options policy-statement send-ipv6 then accept
```

All this routing policy says is that "if the routing protocol (which is RIPng) running on the LAN1 interface (`fe-1/3/0`) wants to advertise an IPv6 route (from family inet6), let it (accept)."

We also have to configure RIPng on the other routers. We know that we can't run RIPng on the external links on the border routers (P7, P9, P2, and P4), but we can show the full configurations on PE5 and PE1. These routers have to run RIPng on *three* interfaces, not just one, so that RIPng routing information flows from site router to backbone (and from backbone to site router). Let's look at PE5 (PE1 is about the same).

```
set interfaces fe-1/3/0 unit 0 family inet address 10.10.50.1/24
set interfaces fe-1/3/0 unit 0 family inet6 address fe80::205:85ff:fe85:aafe/64
set interfaces fe-1/3/0 unit 0 family inet6 address fc00:fe67:d4:c:205:85ff:fe85:
    aafe/64
```

We have IPv6 addresses on the SONET links to P9 and P4, `so-0/0/0` and `so-0/0/2`, but the details are not important. What is important is that we run RIPng on all three interfaces.

```
set protocols ripng group ripv6group export send-ipv6
set protocols ripng group ripv6group neighbor ge-0/0/3.0
set protocols ripng group ripv6group neighbor so-0/0/0.0
set protocols ripng group ripv6group neighbor so-0/0/2.0
```

The routing policy now will export the interface IPv6 addresses we want into RIPng. This policy has one *term* for each interface and is more complex than the one for the site routers.

```
set policy-options policy-statement send-ipv6 term A from interface ge-0/0/3.0
set policy-options policy-statement send-ipv6 term A from family inet6
set policy-options policy-statement send-ipv6 term A then accept
set policy-options policy-statement send-ipv6 term B from interface so-0/0/0.0
set policy-options policy-statement send-ipv6 term B from family inet6
set policy-options policy-statement send-ipv6 term B then accept
set policy-options policy-statement send-ipv6 term C from interface so-0/0/2.0
set policy-options policy-statement send-ipv6 term C from family inet6
set policy-options policy-statement send-ipv6 term C then accept
```

The policy simply means this: "Out of all interfaces, look at `ge-0/0/3`, `so-0/0/0`, and `so-0/0/2`. If the routing protocol running on those links (which is RIPng) wants to advertise an IPv6 route (from family inet6), let it (accept)."

The backbone routers run RIPng on their internal interfaces, but the configurations and policies are very similar to those on the provider-edge routers. We don't need to list those.

When all the configurations are committed and made active on the routers, we form an *adjacency* and exchange IPv6 routing information with each neighbor according to the policy. The IPv6 routing table on CE0 now shows the prefix of LAN2 (`fc00:fe67:d4:c::/64`) learned from CE6 with RIPng.

```
admin@CE0# show route table inet6 fc00:fe67:d4:c::/64

inet6.0: 38 destinations, 38 routes (38 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

fc00:ffbe:d5:b::/64 *[RIPng/100] 01:15:19, metric 6, tag 0
                to fc00:ffbe:d5:b::a00:3b01 via so-0/0/0.0
                > to fc00:ffbe:d5:b::a00:2d01 via so-0/0/2.0
```

What does all this mean? We've learned this route with RIPng, and its *preference* is 100 (high compared to local interfaces, which are 0). When routes are learned in different ways from different protocols, the route with the *lowest* preference will be the active route. The metric of 6 (hops) essentially shows that LAN2 is 6 routers away from LAN1. If there are different paths with different metrics through a collection of routers, the hop to the path with the lowest metric becomes the active route. More advanced routing protocols can compute metrics on the basis of much more than simply number of routers (hops).

Note the right angle bracket (>) to the left of the `so-0/0/2.0` link to router P9. Remember, there are *two* ways for PE5 to forward packets to LAN2: through router P4 at the end of link `so-0/0/0.0` and through router P9 at the end of link `so-0/0/0.0`. The > indicates that packets are being forwarded to router P9. (Usually, all other things being equal, a router chooses the link with the lower IP address.) However, the other link is available if the active link or router fails. (If we want to forward packets out *both* links, we can turn on *load balancing* and the links will be used in a round-robin fashion.)

But even with RIPng up and running among the routers, we still have to give non–link-local addresses to the hosts. Right now, if we try to use `ping6` on LAN2 to ping a different IPv6 private address on LAN1, we'll still get an error condition. Let's try it from `wincli2` on LAN2 to `wincl1` on LAN1.

```
C:\Documents and Settings\Owner>ping6 fe80::20c:cff:fe3b:883c

Pinging fe80::20c:cff:fe3b:883c with 32 bytes of data:

No route to destination.
  Specify correct scope-id or use -s to specify source address.
No route to destination.
  Specify correct scope-id or use -s to specify source address.
No route to destination.
  Specify correct scope-id or use -s to specify source address.
No route to destination.
  Specify correct scope-id or use -s to specify source address.

Ping statistics for fe80::20c:cff:fe3b:883c:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss)
```

Like the routers, the Windows XP hosts need routable addresses. We assign an interface (by index shown by `ipconfig`) that is a routable IPv6 address with the `ipv6 adu` command. But the address is still shown with `ipconfig`.

```
C:\Documents and Settings\Owner>ipconfig

Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix . :
        IP Address . . . . . . . . . . . . : 10.10.12.222
        Subnet Mask  . . . . . . . . . . . : 255.255.255.0
```

```
        IP Address . . . . . . . . . . . . : fc00:fe67:d5:c:202:b3ff:fe27:fa8c
        IP Address . . . . . . . . . . . . : fe80::202:b3ff:fe27:fa8c%5
        Default Gateway  . . . . . . . . . : 10.10.12.1
                                             fe80::5:85ff:fe8b:bcdb%5
                                             fc00:fe67:d5:c:205:85ff:fe8b:bcdb
```

How did the host know the default gateway to use for IPv6? We probed for neighbors earlier, but even if we had not, IPv6 router advertisement (which was configured with RIPng on the routers, and the main reason we did it) takes care of that.

Now we should be able to ping end to end from `wincli2` to `wincli1` by IPv6 address.

```
C:\Documents and Settings\Owner>ping6 fc00:ffb3:d4:b:20e:cff:fe3b:883c

Pinging fc00:ffb3:d.4:b:20e:cff:fe3b:883c
from fc00:fe67:d5:c:202:b3ff:fe27:fa8c with 32 bytes of data:

Reply from fc00:ffb3:d4:b:20e:cff:fe3b:883c: bytes=32 time<1ms
Reply from fc00:ffb3:d4:b:20e:cff:fe3b:883c: bytes=32 time<1ms
Reply from fc00:ffb3:d4:b:20e:cff:fe3b:883c: bytes=32 time<1ms
Reply from fc00:ffb3:d4:b:20e:cff:fe3b:883c: bytes=32 time<1ms

Ping statistics for fc00:ffb3:d4:b:20e:cff:fe3b:883c:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 4ms, Maximum = 5ms, Average = 4ms
```

The reverse also works as well. In the rest of this chapter, let's take a closer look at how the IGPs perform their task of distributing routing information within an AS. Remember, how the IGP routing information gets from AS to AS with an EGP is the topic of Chapter 15.

INTERIOR ROUTING PROTOCOLS

Routers initially know only about their immediate environments. They know the IP addresses and prefixes configured on their local interfaces, and at most a little more statically defined information. Yet all routers must know all the details about everything in their routing domain to forward packets rationally, hop by hop, toward a given destination. So routers offer to and ask their *neighbor routers* (adjacent routers one hop away) about the routing information they know. Little by little, each router then builds up a detailed routing information database about the TCP/IP network.

How do routers exchange this routing information within a domain and between routing domains? With *routing protocols*. Within a routing domain, several different routing protocols can be used. Between routing domains on the Internet, another routing protocol is used. This chapter focuses on the routing protocols used *within* a routing domain and the next chapter covers the routing protocol used *between* routing domains.

Interior routing protocols, or IGPs, run between the routers inside a single routing domain, or autonomous system (AS). A large organization or ISP can have a single AS, but many global networks divide their networks into one or more ASs. IGPs run within these routing domains and do not share information learned across AS boundaries except physical interface addresses if necessary.

Modern routing protocols require minimal configuration of *static routes* (routes configured and maintained by hand). Today, *dynamic* routing protocols allow adjacent (directly connected) routers to exchange routing table information periodically to build up the topology of the router network as a whole by passing information received by adjacent neighbors on to other routers.

IGPs essentially "bootstrap" themselves into existence, and then send information about their IP addresses and interfaces to other routers directly attached to the source router. These neighbor, or adjacent, routers distribute this information to their neighbors until the network has *converged* and all routers have the identical information available.

When changes in the network as a result of failed links or routers cause the routing tables to become outdated, the routing tables differ from router to router and are inconsistent. This is when routing loops and black holes happen. The faster a routing protocol converges, the better the routing protocol is for large-scale deployment.

THE THREE MAJOR IGPs

There are three main IGPs for IPv4 routing: RIP, OSPF, and IS–IS. The Routing Information Protocol (RIP), often declared obsolete, is still used and remains a popular routing protocol for small networks. The newer version of RIP, known as RIPv2, should always be used for IPv4 routing today. Open Shortest Path First (OSPF) and Intermediate System–Intermediate System (IS–IS) are similar and much more robust than RIP. There are versions of all three for IPv6: OSPFv3, RIPng (sometimes seen as RIPv6), and IS–IS works with either IPv4 or IPv6 today.

RIP is a *distance-vector* routing protocol, and OSPF and IS–IS are *link-state* routing protocols. Distance-vector routing protocols are simple and make routing decisions based on one thing: How many routers (hops) are there between here and the destination? To RIP, link speeds do not matter, nor does congestion near another router. To RIP, the "best" route always has the fewest number of hops (routers).

Link-state protocols care more about the network than simply the number of routers along the path to the destination. They are much more complex than distance-vector routing protocols, and link-state protocols are much more suited for networks with many different link speeds, which is almost always the case today. However, link-state protocols require an elaborate database of information about the network on each router. This database includes not only the local router addressing and interfaces, but each and every router in the immediate area and often the entire AS.

ROUTING INFORMATION PROTOCOL

The RIP is still used on all types of TCP/IP networks. The basics of RIP were spelled out in RFC 1058 from 1988, but this is misleading. RIP was in use long before 1988, but no one bothered to document RIP in detail. RIP is bundled with almost all implementations of TCP/IP, so networks often run only RIP. Why pay for something when RIP was available for free?

RIP version 1 (RIPv1) in RFC 1058 has a number of annoying limitations, but RIP is so popular that doing away with RIP is not a realistic consideration. RFC 1388 introduced RIP version 2 (RIPv2 or sometimes RIP-2) in 1993. RIPv2 addressed RIPv1 limitations, but could not turn a distance-vector protocol into a link-state routing protocol such as OSPF and IS–IS.

RIPv2 is backward compatible with RIPv1, and most RIP implementations run RIPv2 by default and allow RIPv1 to be configured. In this chapter, the term "RIP" by itself means "a version of RIP runs RIPv2 by default but can also be configured as RIPv1 as required."

Router vendor Cisco was deeply dissatisfied with RIPv1 limitations and so created its own vendor-specific (proprietary) version of an IGP routing protocol, which Cisco called the Interior Gateway Routing Protocol (IGRP). IGRP improved upon RIPv1 in several ways, but "pure" IGRP could only run between Cisco routers. As good as IGRP was, IGRP was still basically implemented as a distance-vector protocol. As networks grew more and more complex in terms of link speeds and router capacities, it was possible to switch to a link-state protocol such as OSPF or IS–IS, but many network administrators at the time felt these new protocols were not stable or mature enough for production networks. Cisco then invented Enhanced IGRP (EIGRP) as a sort of "hybrid" routing protocol that combined features of both distance-vector and link-state routing protocols all in one (proprietary) package.

Due to the proprietary nature of IGRP and EIGRP, only the basics of these routing protocols are covered in this chapter.

Distance-Vector Routing

RIP and related distance-vector routing protocols are classified as "Bellman–Ford" routing protocols because they all choose the "best" path to a destination based on the *shortest path* computation algorithm. It was first described by R. E. Bellman in 1957 and applied to a distributed network of independent routers by L. R. Ford, Jr. and D. R. Fulkerson in 1962. Every version of Unix today bundles RIP with TCP/IP, usually as the *routed* ("route management daemon") process, but sometimes as the *gated* process.

All routing protocols use a *metric* (measure) representing the relative "cost" of sending a packet from the current router to the destination. The lowest relative cost is the "best" way to send a packet. Distance-vector routing protocols have only one metric: distance. The distance is usually expressed in terms of the number of routers between the router with the packet and the router attached to the destination network. The

Table 14.2 Example RIP Routing Table

Network	Next Hop Interface	Cost
10.0.14.0	Ethernet 1 (E1)	2
172.16.15.0	Serial 1 (S1)	1
192.168.44.0	Ethernet 2 (E2)	3
192.168.66.0	Serial 2 (S2)	INF (15)
192.168.78.0	Locally attached	0

distance metric is carried between routers running the same distance-vector routing protocol as a *vector*, a field in a routing protocol update packet.

A simple example of how distance-vector, or hop-count, routing works will illustrate many of the principles that all routing protocols simple and complex must deal with. All routing protocols must pass along network information received from adjacent routers to all other routers in a routing domain, a concept known as *flooding*. Flooding is the easiest way to ensure consistency of routing tables, but convergence time might be high as routers at one end of a chain of routers wait for information from routers at the far end of the chain to make its way through the routers in between. Flooding also tends to maximize the bandwidth consumed by the routing protocol itself, but there are ways to reduce this.

RIP floods updates every 30 seconds. Note that routing information takes at least 30 seconds to reach the *closest* neighbor if that is the routing update interval used. Long chains of routers can take quite a long time to converge (several minutes) when a network address is added or when a link fails.

When this network converges, each routing table will be consistent and each router will be reachable from every other router over one of the interfaces. The network topology has been "discovered" by the routing protocol. An example of the information in one of these tables is shown in Table 14.2.

Routers can have alternatives other than those shown in the table. For example, the cost to reach network `192.168.44.0` from this router could be the same (3) over E1 as it is over E2. The E1 interface is most likely in the table because the update from the neighbor router saying "send `192.168.44.0` packets here" arrived before the update from another router saying the same thing, or the entry was already in the table. When costs are equal, routing tables tend to keep what they know.

Broken Links

The distance-vector information has now been exchanged and the routers all have a way to reach each other. Usually, the routing protocol will update an internal database in the router just for that routing protocol and one or more entries based on the database are made in the routing table, which might contain information from other routing protocols as well. The routing table information is then used to compute the "best" routes to be used in the forwarding table (sometimes called the switching table) of the

router. This chapter blurs the distinctions between routing protocol database, routing table, and forwarding table for the sake of simplicity and clarity.

What will happen to the network if a link "breaks" and can no longer be used to forward traffic? In a static routing world, this would be disastrous. But when using a dynamic routing protocol, even one as simple as a distance-vector routing protocol, the network should be able to converge around the new topology.

The routers at each end of the link, since they are locally connected to the interface (direct), will notice the outage first because routers constantly monitor the state of their interfaces at the physical level. Distance-vector protocols note this absent link by noting that the link now has an "infinite" cost. All routers formerly reachable through the link are now an infinite distance away.

Distance-Vector Consequences

In some cases, distance-vector updates are generated so closely in time by different routers that a link failure can cause a routing loop to occur, and packets can easily "bounce" back and forth between two adjacent routers until the packet TTL expires, even though the destination is reachable over another link. The "bouncing effect" will last until the network converges on the new topology.

However, this convergence can take some time, since routers not located at the end of a failed link have to gradually increase their costs to infinity one "hop" at a time. This is called "counting to infinity," and can drag out convergence time considerably if the value of "infinity" is set high enough. On the other hand, a low value of "infinity" will limit the maximum number of routers that can form the longest path through the network from source to destination.

In order to minimize the effects of bouncing and counting to infinity, most implementations of distance-vector routing protocols such as RIP also implement *split horizon* and *triggered updates*.

Split Horizon

If Router A is sending packets to Router B to reach Router E, then it makes no sense at all for Router B to try to reach Router E through Router A. All Router A will do is turn around and send the packet right back to Router B. So Router A should never advertise a way to reach Router E to Router B.

A more sophisticated form of split horizon is known as *split horizon with poison reverse*. Split horizon with poison reverse eliminates a lot of counting to infinity problems due to single link failures. However, many multiple link failures will still cause routing loops and counting to infinity problems even when split horizon with poison reverse is in use.

Triggered Updates

With triggered updates, a router running a distance-vector protocol such as RIP can remain silent if there are no changes to the information in the routing table. If a link failure is detected, triggered updates will send the new information. Triggered updates,

like split horizon, will not eliminate all cases of routing loops and counting to infinity. However, triggered updates always help the counting process to reach infinity much faster.

RIPv1

A RIP packet must be 512 bytes or smaller, including the header. RIP packets have no implied sequence, and each update packet is processed independently by the router receiving the update. A router is only required to keep *one* entry associated with each route. But in practice, routers might keep up to four or more routes (next hops) to the same destination so that convergence time is lowered.

RIPv1 required routers running RIP to *broadcast* the entire contents of their routing tables at fixed intervals. On LANs, this meant that the RIPv1 packets were sent inside broadcast MAC frames. But broadcast MAC frames tell not only every router on the LAN, but every *host* on the LAN, "pay attention to this frame." Inside the frame, the host would find a RIPv1 update packet, and probably ignore the contents. But every 30 seconds, every host on the LAN had to interrupt its own application processing and start throwing away RIPv1 packets.

Each host *could* keep the information inside the RIPv1 update packet. Some hosts on LANs with RIPv1 routers have as elaborate a routing table as the routers themselves. Hackers loved RIPv1: With a few simple coding changes, any host could impersonate a RIPv1 router and start pumping out fake routing information, as many college and university network administrators discovered in the late 1980s. (This is one reason you don't run RIP on host interfaces.)

Many people see RIP updates vary from 30 seconds and assume that timers are off. In fact, table updates in RIP are initiated on each router at *approximate* 30-second intervals. Strict synchronization is avoided because RIP traffic spikes can easily lead to discarded RIP packets. The *update timer* usually adds or subtracts a small amount of time to the 30-second interval to avoid RIP router synchronization.

Network devices running RIP can be either *active* or *passive (silent)* mode. Active RIP devices will listen for RIP update packets and also generate their own RIP update packets. Passive RIP devices will only listen for RIP updates and never generate their own update packets. Many hosts, for example, which must process the broadcast RIP updates sent on a LAN, are purely passive RIP devices.

RIPv1 Limitations

RIPv1 had a number of limitations that made RIPv1 difficult to use in large networks. The larger the routing domain, the more severe and annoying the limitations of RIPv1 become.

Wasted Space—All of the RIPv1 packet fields are larger than they need to be, sometimes many times larger. There are almost three times as many 0 bits as information bits in a RIP packet.

Limited Metrics—As a network grows, the distance-vector might require a metric greater than 15, which is unreachable (infinite).

No Link Speed Allowances—The simple hop count metric will always result in packets being sent (as an example) over two hops using low-speed, 64-kbps links rather than three hops using SONET/SDH links.

No Authentication—RIPv1 devices will accept RIPv1 updates from any other device. Hackers love RIPv1 for this very reason, but even an innocently misconfigured router can disrupt an entire network using RIPv1.

Subnet Masks—RIPv1 requires the use of the same subnet mask because RIPv1 updates do not carry any subnet mask information.

Slow Convergence—Convergence can be very slow with RIPv1, often 5 minutes or more when links result in long chains of routers instead of neat meshes. And "circles" of RIPv1 routers maximize the risk of counting to infinity.

RIPv2

RIPv2 first emerged as an update to RIPv1 in RFC 1388 issued in January 1993. This initial RFC was superseded by RFC 1723 in November 1994. The only real difference between RFC 1388 and RFC 1723 is that RFC 1723 deleted a 2-byte Domain field from the RIPv2 packet format, designating this space as unused. No one was really sure how to use the Domain field anyway. The current RIPv2 RFC is RFC 2453 from November 1998.

RIPv2 was not intended as a replacement for RIPv1, but to extend the functions of RIPv1 and make RIP more suitable for VLSM. The RIP message format was changed as well to allow for authentication and multicasting.

In spite of the changes, RIPv2 is still RIP and suffers from many of the same limitations as RIPv1. Most router vendors support RIPv2 by default, but allow interfaces or whole routers to be configured for backward compatibility with RIPv1. RIPv2 made major improvements to RIPv1:

- Authentication between RIP routers
- Subnet masks to be sent along with routes
- Next hop IP addresses to be sent along with routes
- Multicasting of RIPv2 messages

The RIPv2 packet format is shown in Figure 14.2.

Command Field (1 byte)—This is the same as in RIPv1: A value of 1 is for a Request and a value of 2 is for a Response.

Version Number (1 byte)—RIPv1 uses a value of 1 in this field, and RIPv2 uses a value of 2.

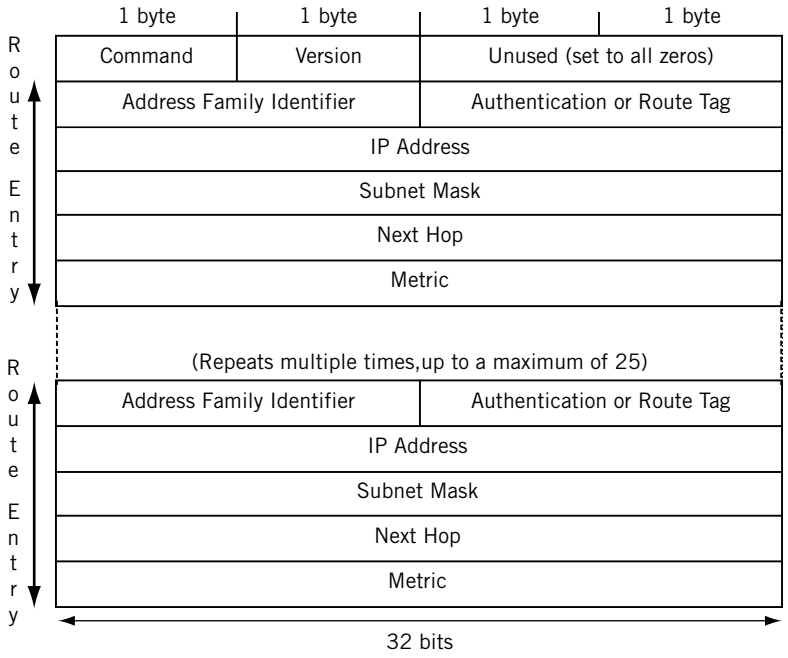


FIGURE 14.2

RIPv2 packet format, showing how the subnet mask is included with the routing information advertised.

Unused (2 bytes)—Set to all zero bits. This was the Domain field in RFC 1388. Now officially unused in RFC 1723, this field is ignored by routers running RIPv2 (but this field must be set to all 0 bits for RIPv1 routers).

Address Family Identifier (AFI) (2 bytes)—This field is set to a value of 2 when IP packet and routing information is exchanged. RIPv2 also defined a value of 1 to ask the receiver to send a copy of its entire routing table. When set to all 1s (0xFFFF), the AFI field is used to indicate that the 16 bits following the AFI field, ordinarily set to 0 bits, now carry information about the type of authentication being used by RIPv2 routers.

Authentication or Route Tag (2 bytes)—When the AFI field is not 0xFFFF, this is the Route Tag field. The Route Tag field identifies *internal* and *external* routes in RIPv2. Internal routes are those learned by RIP itself, either locally or through other RIP routers. External routes are routes learned from another routing protocol such as OSPF or BGP.

IPv4 Address (4 bytes)—This field and the three that follow can be repeated up to 25 times in the RIPv2 Response packet. This field is almost the same as in

RIPv1. This address can be a host route, a network address, or a default route. A RIPv2 Request packet has the IP address of the originator in this field.

Subnet Mask (4 bytes)—This field, the biggest change in RIPv2, contains the subnet mask that goes with the IP address in the previous field. If the network address does not use a subnet mask different from the natural classful major network mask, then this field can be set to all zeroes, just as in RIPv1.

Next Hop (4 bytes)—This field contains the next hop IP address that traffic to this IP address space should use. This was a vast improvement over the "implied" next hop used in RIPv1.

Metric (4 bytes)—Unfortunately, the metric field is unchanged. The range is still 1 to 15, and a metric value of 16 is considered unreachable.

RIPv2 is still RIP. But RIPv2's additions for authentication, subnet masks, next hops, and the ability to multicast routing information increase the sophistication of RIP and have extended RIP's usefulness.

Authentication

Authentication was added in RIPv2. The Response messages contain the routing update information, and authenticating the responder to a Request message is a good way to minimize the risk of a routing table becoming corrupted either by accident or through hacker activities. However, there were really only 16 bits available for authentication, hardly adequate for modern authentication techniques. So the authentication actually takes the place of one routing table entry and authenticates the entire update message. This gives 16 bytes (128 bits) for authentication, which is not state of the art, but is better than nothing.

The really nice feature of RIPv2 authentication is that router vendors can add their own Authentication Type values and schemes to the basics of RIPv2, and many do. For example, Cisco and Juniper Networks routers can be configured to use MD5 (Message Digest 5) authentication encryption to RIPv2 messages. Thus, most routers can have three forms of authentication on RIP interfaces: none, simple password, or MD5. Naturally, the MD5 authentication keys used must match up on the routers.

Subnet Masks

The biggest improvement from RIPv1 to RIPv2 was the ability to carry the subnet mask along with the route itself. This allowed RIP to be used in classless IP environments with VLSM.

Next Hop Identification

Consider a network where there are several site routers with only one or a few small LANs. The small routers run RIPv2 between themselves and their ISP's router, but might run a higher speed link to one router and a lower speed link to another. The higher speed link might be more hops away than the lower speed link.

The next hop field in RIPv2 is used to "override" the ordinary metric method of deciding active routes in RIP. RIPv2 routers check the next hop field in the routing update message. If the next hop field is set for a particular route, the RIP router will use this as the next hop for the route, regardless of distance-vector considerations.

This RIPv2 next hop mechanism is sometimes called *source routing* in some documents. But true source routing information is always set by a host, not a router. This is just RIPv2 *next hop identification*.

Multicasting

Multicasting is a kind of "halfway" distribution method between unicast (one source to one destination) and broadcast (one source to all possible destinations). Unlike broadcasts that are received by all nodes on the subnet, only devices that *join* the RIPv2 multicast group will receive packets for RIPv2. (We'll talk more about multicast in Chapter 16.) RIPv2 multicasting also offers a way to filter out RIPv2 messages from a RIPv1 only router. This can be important, since RIPv2 messages look very much like RIPv1 messages. But RIPv2 messages are all *invalid* by RIPv1 standards. RIPv1 devices would either discard RIPv2 messages because the mandatory all-zero fields are not all zeroes, or accept the routes and ignore the additional RIPv2 information such as the subnet mask. RIPv2 multicasting makes sure that only RIPv2 devices see the RIPv2 information. So RIPv1 and RIPv2 routers can easily coexist on the same LAN, for instance. The multicast group used for RIPv2 routers is `224.0.0.9`.

RIPv2 is still limited in several ways. The 15 maximum-hop count is still there, as well as counting to infinity to resolve routing loops. And RIPv2 does nothing to improve on the fixed distance-vector values that are a feature of all versions of RIP.

RIPng for IPv6

The version of RIP used with IPv6 is called *RIPng*, where "ng" stands for "next generation." (IPv6 itself was often called IPng in the mid-1990s.) RIPng uses exactly the same hop count metric as RIP as well as the same logic and timers. So RIPng is still a distance-vector RIP, with two important differences.

1. The packet formats have been extended to carry the longer IPv6 addresses.
2. IPv6 security mechanisms are used instead of RIPv2 authentication.

The overall format of the RIP packet is the same as the format of the RIPv2 packet (but RIPng cannot be used by IPv4). There is a 32-bit header followed by a set of 20-byte route entries. The header fields must be the same as those used in RIPv2: There is a 1-byte Command code field, followed by a 1-byte Version field (now 6), and then 2 unused bytes of bits that must still be set to all 0 bits. However, the 20-byte router entry fields in RIPng are totally different that those in RIPv2.

IPv6 addresses are 16 bytes long, leaving only 4 bytes for any other information that must be associated with the IPv6 route. First, there is a 2-byte Route Tag field with the same use as in RIPv2: The Route Tag field identifies *internal* and *external* routes. Internal routes are those learned by RIP itself, either locally or through other RIP routers.

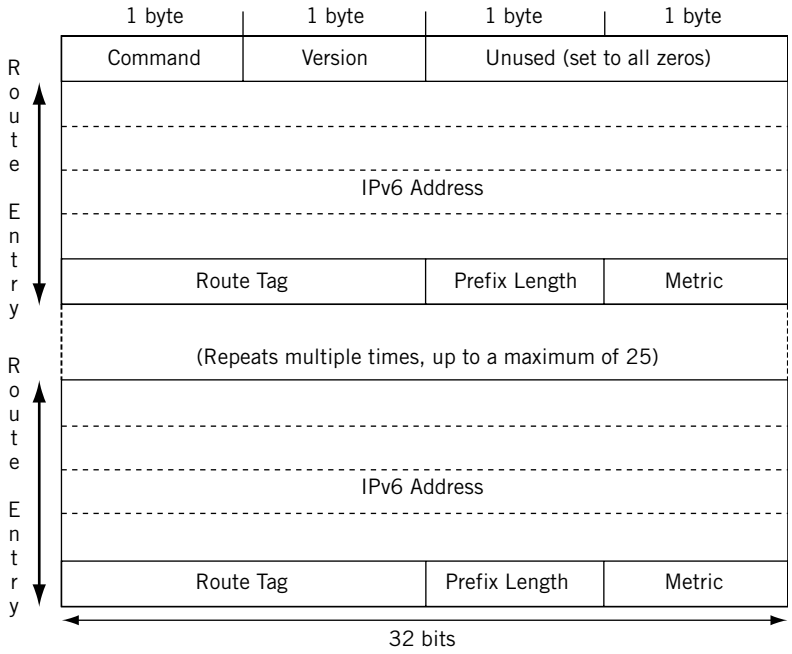


FIGURE 14.3

RIPng for IPv6 packet fields. Note the large address fields and different format than RIPv2 fields.

External routes are routes learned from another routing protocol such as OSPF or BGP. Then there is a 1-byte Prefix Length field that tells the receiver where the boundary between network and host is in the IPv6 address. Finally, there is a 1-byte Metric field (this field was a full 32 bits in RIPv1 and RIPv2). Since infinity is still 16 in RIPng, this is not a problem.

The fields of the RIPng packet are shown in Figure 14.3. The combination of IPv6 address and Prefix Length do away with the need for the Subnet Mask field in RIPv2 packets. The Address Format Identifier (AFI) field from RIPv2 is not needed in RIPng, since only IPv6 routing information can be carried in RIPng.

But IPv6 still needs a Next Hop field. This RIPv2 field contained the next-hop IP address that traffic to this IP address space should use, and was a vast improvement over the "implied" next hop used in RIPv1. Now, IPv6 does not always need this Next Hop information, but in many cases the next hop should be included in an IPv6 routing information update. An IPv6 Next Hop needs another 128 bits (16 bytes). The creators of RIPng decided to essentially reproduce the same route entry structure for the IPv6 Next Hop, but use a special value of the last field (the Metric) to indicate that the first 16 bytes in the route entry was an IPv6 Next Hop, not the route itself. The value chosen for the metric was 256 (0xFF) because this was far beyond the legal hop count limit (15) for RIP.

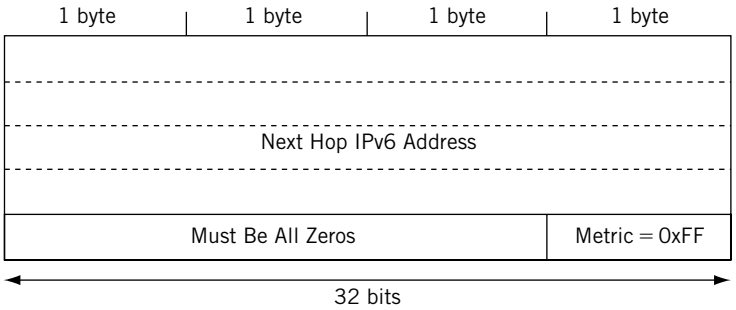


FIGURE 14.4

The Next Hop in IPv6 with RIPng. Note the use of the special metric value.

When the route entry used is an IPv6 Next Hop, the 3 bytes preceding the `0xFF` Metric must be set to all 0 bits. This is shown in Figure 14.4.

At first it might seem that the amount of the IPv6 routing information sent with RIPng must instantly double in size, since now each 20-byte IPv6 route requires a 20-byte IPv6 Next Hop field. This certainly would make IPv6 very unattractive to current RIP users. But it was not necessary to include a Next Hop entry for each and every IPv6 route because the creators of RIPng used a clever mechanism to optimize the use of the Next Hop entry.

A Next Hop always qualifies any IPv6 routes that follow it in the string of route entries until another Next Hop entry is reached or the packet stream ends. This keeps the number of "extra" Next Hop entries needed in RIPng to an absolute minimum. And due to the fact that the Next Hop field in RIPv2 has only specialized use, a lot of IPv6 routes need no Next Hop entry at all.

The decision to replace RIPv2 authentication with IPv6 security mechanisms was based on the superior security used in IPv6. When used with RIPng updates, the IPv6 Authentication Header protects both the data inside the packet and the IP addresses of the packet, but this is not the case with RIPv2 authentication no matter which method is used. And IPv6 encryption can be used to add further protection.

A NOTE ON IGRP AND EIGRP

Cisco routers often use a proprietary IGP known as the Interior Gateway Routing Protocol (IGRP) instead of RIP. Later, features were added to IGRP in the form of Enhanced IGRP (EIGRP). In spite of the name, EIGRP was a complete redesign of IGRP. This section will only give a brief outline of IGRP and EIGRP, since IGRP/EIGRP interoperability with Juniper Networks routers is currently impossible.

IGRP and EIGRP might appear to be open standards, but this is only due to the wide-ranging deployment of Cisco routers. Cisco has never published the details of IGRP internals (EIGRP is based on these), and is not likely to.

IGRP improves on RIP in several areas, but IGRP is still essentially a distance-vector routing protocol. EIGRP, on the other hand, is advertised by Cisco as a "hybrid" routing protocol that includes aspects of link-state routing protocols such as OSPF and IS–IS among the features of EIGRP. Today not many, even those with all-Cisco networks, would consider running EIGRP over OSPF or IS–IS.

Open Shortest Path First

OSPF is not a distance-vector protocol like RIP, but a *link-state* protocol with a set of metrics that can be used to reflect much more about a network than just the number of routers encountered between source and destination. In OSPF, a router attempts to route based on the "state of the links."

OSPF can be equipped with metrics that can be used to compute the "shortest" path through a group of routers based on link and router characteristics such as highest throughput, lowest delay, lowest cost (money), link reliability, or even more. OSPF is still used very cautiously, with default metrics based entirely on link bandwidth. Even with this conservative use, OSPF link states are an improvement over simple hop counts.

Distance-vector routing protocols like RIP were fine for networks comprised of equal speed links, but struggled when networks started to be built out of WAN links with a wide variety of available speeds. When RIP first appeared, almost all WANs were composed of low-speed analog links running at 9600 bps. Even digital links running at 56 or 64 kbps were mainly valued for their ability to carry five 9600-bps channels on the same physical link. Commercial T1s at 1.544 Mbps were not widely available until 1984, and then only in major metropolitan areas. Today, the quickest way to send packets from one router to another is not always through the fewest number of routers.

The "open" in OSPF is based on the fact that the Shortest Path First (SPF) algorithm was not owned by anyone and could be used by all. The SPF algorithm is often called the *Dijkstra algorithm* after the computer and network pioneer that first worked it out from graph theory. Dijkstra himself called the new method SPF, first described in 1959, because compared to a distance-vector protocol's counting to infinity to produce convergence, his algorithm always found the "shortest path first."

OSPF version 1 (OSPFv1), described in RFC 1131, never matured beyond the experimental stage. The current version of OSPF, OSPFv2, which first appeared as RFC 1247 in 1991, and is now defined by RFC 2328 issued in 1998, became the recommended replacement for RIP (although a strong argument could be made in favor of IS–IS, discussed later in this chapter).

Link States and Shortest Paths

Link-state protocols are all based on the idea of a *distributed map* of the network. All of the routers that run a link-state protocol have the same copy of this network map, which is built up by the routing protocol itself and not imposed on the network from an outside source. The network map and all of the information about the routers and links (and the routes) are kept in a *link-state database* on each router. The database

is not a "map" in the usual sense of the word: Records represent the topology of the network as a series of links from one router to another. The database must be identical on all of the routers in an *area* for OSPF to work.

Initially, each router only knows about a piece of the entire network. The local router knows only about itself and the local interfaces. So *link-state advertisements* (LSAs), the OSPF information sent to all other routers from the local router, always identify the local router as the source of the information.

The OSPF routing protocol "floods" this information to all of the other routers so that a complete picture of the network is generated and stored in the link-state database. OSPF uses *reliable flooding* so that OSPF routers have ways to find out if the information passed to another router was received or not.

The more routers and links that OSPF has to deal with, the larger the link-state database that has to be maintained. In large router networks, the routing information could slow traffic. OSPFv2 introduced the idea of *stub areas* into an OSPF routing domain. A stub area could function with a greatly reduced link-state database, and relied on a special *backbone area* to reach the entire network.

What OSPF Can Do

By 1992, OSPF had matured enough to be the recommended IGP for the Internet and had delivered on its major design goals.

Better Routing Metrics for Links

OSPF employs a configurable link metric with a range of valid values between 1 and 65,535. There is no limit on the total cost of a path between routers from source to destination, as long as all the routers are in the same AS. Network administrators, for example, could assign a metric of 10,000 to a low-bandwidth link and 10 to a very high-bandwidth Metro Ethernet or SONET/SDH link. In theory, these values could be manually assigned through a central authority. In practice, most implementations of OSPF divide a *reference bandwidth* by the actual bandwidth on the link, which is known through the router's interface configuration. The default reference bandwidth is usually 100 Mbps (Fast Ethernet). Since the metric cannot be less than 0, all links at 100 Mbps or faster use a 1 as a link metric and thus revert to a simple hop count when computing longest cost paths. The reference bandwidth is routinely raised to accommodate higher and higher bandwidths, but this requires a central authority to carry out consistently.

Equal-Cost Multipaths

There are usually multiple ways to reach the same destination network that the routing protocol will compute as having the same cost. When equal-cost paths exist, OSPF routers can find and use equal-cost paths. This means that there can be multiple next hops installed in a forwarding table with OSPF. OSPF does not specify how to use these multipaths: Routers can use simple round-robin per packet, round-robin per flow, hashing, or other mechanisms.

Router Hierarchies

OSPF made very large routing domains possible by introducing a two-level hierarchy of areas. With OSPF, the concepts of an "edge" and "backbone" router became common and well understood.

Internal and External Routes

It is necessary to distinguish between routing information that originated within the AS (internal routing information) and routing information that came from another AS (external routing information). Internal routing information is generally more trusted than external routing information that might have passed from ISP to ISP across the Internet.

Classless Addressing

OSPF was first designed in a classful Internet environment with Class A, B, and C addresses. However, OSPF is comfortable with the arbitrary network/host boundaries used by CIDR and VLSM.

Security

RIPv1 routers accepted updates from anyone, and even RIPv2 routers only officially used simple plain-text passwords that could be discovered by anyone with access to the link. OSPF allows not only for simple password authentication, but strong MD5 key mechanisms on routing updates.

ToS Routing

The original OSPF was intended to support the bit patterns established for the Type of Service (ToS) field in the IP packet header. Routers at the time had no way to enforce ToS routing, but OSPF anticipated the use of the Internet for all types of traffic such as voice and video and went ahead and built into OSPF ways to distribute multiple metrics for links. So OSPF routing updates can include ToS routing information for five IP ToS service classes, defined in RFC 1349. The service categories and OSPF ToS values are normal service (ToS = 0), minimize monetary cost (2), maximize reliability (4), maximize throughput (8), and minimize delay (16). Since all current implementations of OSPF support only a ToS value of 0, no more need be said about the other ToS metrics.

By the way, here's all we did on the customer- and provider-edge routers in each AS to configure OSPF to run on every router interface. Now, in a real network, we wouldn't necessarily configure OSPF to run on all of the router's internal or management interfaces, but it does no harm here.

```
set protocols ospf area 0.0.0.0 interface all
```

All OSPF routers do not have to be in the same area, and in most real router networks, they aren't. But this is a simple network and only configures an OSPF *backbone area*, 0.0.0.0. The provider routers in our ISP cores (P9, P7, P4 and P2), which are called

AS border routers, or ASBRs, run OSPF on the *internal* links within the AS, but not on the *external* links to the other AS (this is where we'll run the EGP).

The relationship between the OSPF use of a reference bandwidth and ToS routing should be clarified. Use of the OSPF link reference bandwidth is different from and independent of ToS support, which relies on the specific settings in the packet headers. OSPF routers were supposed to keep separate link-state databases for each type of service, since the least-cost path in terms of bandwidth could be totally different from the least-cost path computed based on delay or reliability. This was not feasible in early OSPF implementations, which struggled to maintain the single, normal ToS = 0 database. And it turned out that the Internet users did not want lots of bandwidth or low delay or high reliability when they sent packets. Internet users wanted lots of bandwidth *and* low delay *and* high reliability when they sent packets. So the reference bandwidth method is about all the link-state that OSPF can handle, but that is still better than nothing.

OSPF Router Types and Areas

OSPFv2 introduced areas as a way to cut down on the size of the link-state database, the amount of information flooded, and the time it takes to run the SPF algorithm, at least on areas other than the special backbone area.

An OSPF area is a logical grouping of routers sharing the same 32-bit Area ID. The Area ID can be expressed in dotted decimal notation similar to an IP address, such as 192.168.17.33. The Area ID can also be expressed as a decimal equivalent, so Area 261 is the same as Area 0.0.1.5. When the Area ID is less than 256, usually only a single number is used, but Area 249 is still really Area 0.0.0.249.

There are five OSPF area types. The position of a router with respect to OSPF areas is important as well. The area types are shown in Figure 14.5.

The OSPF Area `0` (`0.0.0.0`) is very special. This is the backbone area of an OSPF routing domain. An OSPF routing domain (AS) can consist of a single area, but in that case the single area must be Area `0`. Only the backbone area can generate the summary routing topology information that is used by the other areas. This is why all interarea traffic must pass through the backbone area. (There are *backdoor links* that can be configured on some routers to bypass the backbone area, but these violate the OSPF specification.) In a sense, the backbone area knows everything. Not so long ago, only powerful high-end routers could be used on an OSPF backbone. On the Illustrated Network, each AS consists of only an Area 0.

If an area is not the backbone area, it can be one of four other types of areas. All of these areas connect to the backbone area through an Area Border Router (ABR). An ABR by definition has links in two or more areas. In OSPF, routers always form the boundaries between areas. A router with links outside the OSPF routing domain is called an autonomous system boundary router (ASBR). Routing information about destination IP addresses not learned from OSPF are always advertised by an ASBR. Even when static routes, or RIP routes, are redistributed by OSPF, that router technically becomes an ASBR. ASBRs are the source of *external routes* that are outside of the

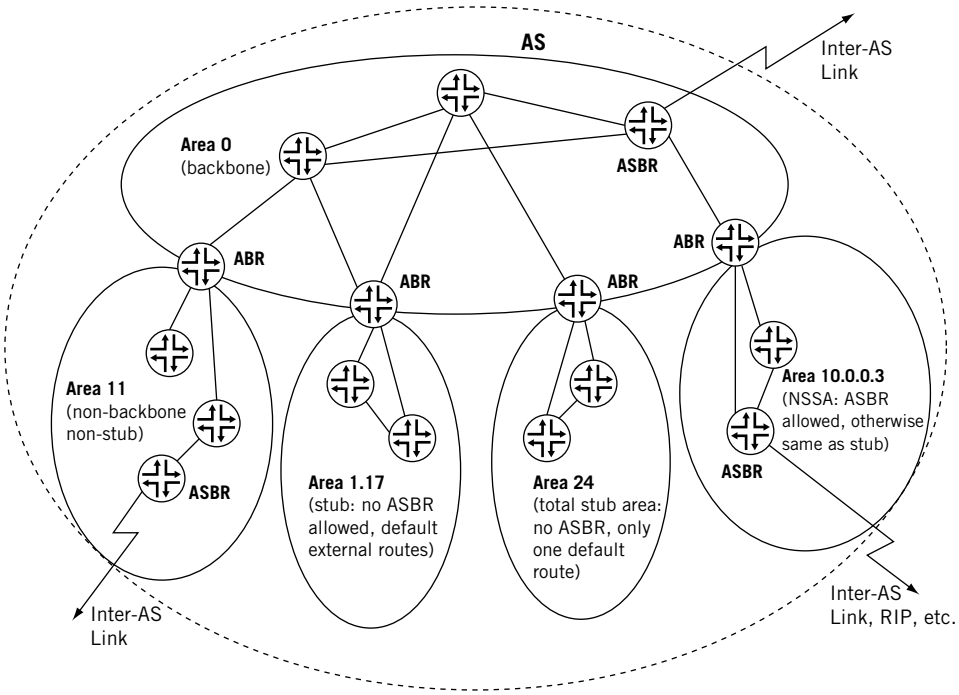


FIGURE 14.5

OSPF area types, showing the various ways that areas can be given numbers (decimal, IP address, or other). Note that ABRs connect areas and ASBRs have links outside the AS or to other routing protocols.

OSPF routing domain, and external routes are often very numerous in an OSPF routing domain attached to the global Internet. If a router is not an ABR or ASBR, it is either an *internal router* and has all of its interfaces within the same area, or a *backbone router* with at least one link to the backbone. However, these terms are not as critical to OSPF configurations as to ABRs or ASBRs. That is, not all backbone routers are ABRs or ASBRs; backbone routers can also be internal routers, and so on.

Non-backbone, Non-stub Areas

These areas are really smaller versions of the backbone area. There can be links to other routing domains (ASBRs) and the only real restriction on a non-backbone, non-stub area is that it cannot be Area 0. Area 11 in Figure 14.5 is a non-backbone, non-stub area.

Stub Area

Stub areas cannot have links outside the AS. So there can be no ASBRs in a stub area. This minimizes the amount of external routing information that needs to be distributed into the link-state databases of the stub area routers. Because an AS might be an ISP on the

Internet, the number of external routes required in an OSPF routing domain is usually many times larger than the internal routes of the AS itself. Stub area routers only obtain information on routes external to the AS from the ABR. Area 1.17 in Figure 14.5 is a stub area.

Total Stub Area

This is also called a "totally stubby area." Recall that stub areas cannot have ASBRs within them, by definition. But stub areas can only reach other ASBRs, which have the links leading to and from other ASs, through an ABR. So why include detailed external route information in the stub area router's link-state database? All that is really needed is the proper default route as advertised by the ABR. Total stub areas only know how to reach their ABR for a route that is not within their area. Area 24 in Figure 14.5 is a total stub area.

Not-So-Stubby Area

Banning ASBRs from stub areas was very restrictive. Even the advertisement of static routes into OSPF made a router an ASBR, as did the presence of a single LAN running RIP, if the routes were advertised by OSPF. And as ISPs merged and grew by acquiring smaller ISPs, it became difficult to "paste" the new OSPF area with its own ASBRs onto the backbone area of the other ISP. The easiest thing to do was to make the new former AS a stub area, but the presence of an ASBR prevented that solution. The answer was to introduce the concept of a not-so-stubby area (NSSA) in RFC 1587. An NSSA can have ASBRs, but the external routing information introduced by this ASBR into the NSSA is either kept within the NSSA or translated by the ABR into a form useful on the backbone Area 0 and to other areas. Area 10.0.0.3 in Figure 14.5 is an NSSA.

OSPF Designated Router and Backup Designated Router

An OSPF router can also be a Designated Router (DR) and Backup Designated Router (BDR). These have nothing to do with ABRs and ASBRs, and concern only the relationship between OSPF routers on links that deliver packets to more than one destination at the same time (mainly LANs).

There are two major problems with LANs and public data networks like ATM and frame relay (called non-broadcast multiple-access, or NBMA, networks). First is the fact that the link-state database represents links and routers as a directed graph. A simple LAN with five OSPF routers would need $N(N-1)/2$, or $5(4)/2 = 20$ link-state advertisements just to represent the links between the routers, even though all five routers are mutually adjacent on the LAN and any frame sent by one is received by the other four. Second, and just as bad, is the need for flooding. Flooding over a LAN with many OSPF routers is chaotic, as link-state advertisements are flooded and "reflooded" on the LAN.

To address these issues, multiaccess networks such as LANs always elect a designated router for OSPF. The DR solves the two problems by representing the multiaccess network as a single "virtual router" or "pseudo-node" to the rest of the network and managing the process of flooding link-state advertisements on the multiaccess

network. So each router on a LAN forms an OSPF adjacency only with the DR (and also the Backup DR [BDR] as mentioned later). All link-state advertisements go only to the DR (and BDR), and the DR forwards them on to the rest of the network and internetwork routers.

Each network that elects a DR also elects a BDR that will take over the functions of the DR if and when the DR fails. The DR and BDR form OSPF adjacencies with all of the other routers on the multiaccess network and the DR and BDR also form an adjacency with each other.

OSPF Packets

OSPF routers communicate using IP packets. OSPF messages ride directly inside of IP packets as IP protocol number 89. Because OSPF does not use UDP or TCP, the OSPF protocol is fairly elaborate and must reproduce many of the features of a transport protocol to move OSPF messages between routers.

There can be one of five OSPF packet types inside the IP packet, all of which share a common OSPF header. The structure of the common OSPF header is shown in Figure 14.6.

The version field is 2, for OSPFv2, and the type has one of the five values. The packet length is the length of the OSPF packet in bytes. The Router ID is the IP address selected as OSPF Router ID (usually the loopback interface address), and the Area ID is the OSPF area of the router that originates the message. The checksum is the same as the one used on IP packets and is computed on the whole OSPF packet.

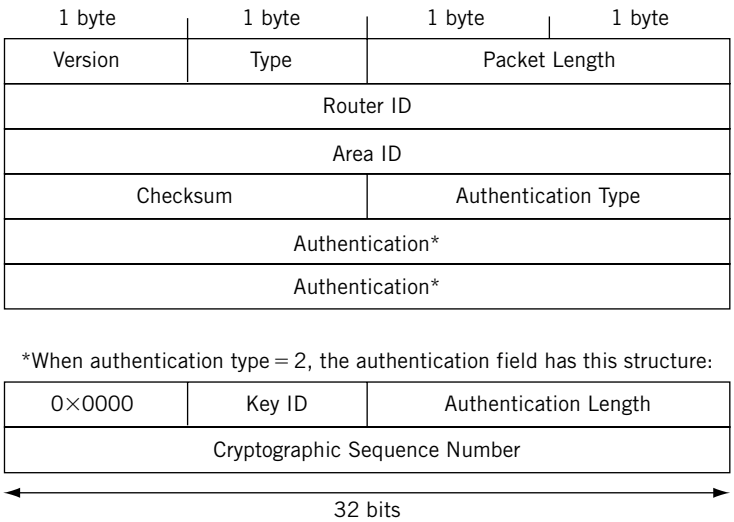


FIGURE 14.6

OSPF packet header fields, showing how the structure can vary with type.

The Authentication Type (or AuType) is either none (0), simple password authentication (1), or cryptographic authentication (2). The simple password is an eight-character plain-text password, but the use of AuType = 2 authentication gives the authentication field the structure shown in the figure. In this case, the Key ID identifies the secret key and authentication algorithm (MD5) used to create the message digest, the Authentication Data Length specifies the length of the message digest appended to the packet (which does not count as part of the packet length), and the Cryptographic Sequence Number always increases and prevents hacker "replay" attacks.

OSPFv3 for IPv6

The changes made to OSPF for IPv6 are minimal. It is easy to transition from OSPF for IPv4 to OSPF for IPv6. There is new version number, OSPF version 3 (OSPFv3), and some necessary format changes, but less than might be expected. The basics are described in RFC 2740.

OSPF for IPv6 (often called OSPFv6) will use link local IPv6 addresses and IPv6 multicast addresses. The IPv6 link-state database will be totally independent of the IPv4 link-state database, and both can operate on the same router.

Naturally, OSPFv6 must make some concessions to the larger IPv6 addresses and next hops. But the common LSA header has few changes as well. The Link State Identifier field is still there, but is now a pure identifier and not an IPv4 address. There is no longer an Options field, since this field also appears in the packets that need it, and the LSA Header Type field is enlarged to 16 bits. Naturally, when LSAs carry the details of IPv6 addresses, those fields are now large enough to handle the 128 bit IPv6 addresses.

INTERMEDIATE SYSTEM–INTERMEDIATE SYSTEM

OSPF is not the only link-state routing protocol that ISPs use within an AS. The other common link-state routing protocol is IS–IS (Intermediate System–Intermediate System). When IS–IS is used with IP, the term to use is *Integrated IS*. IS–IS is not really an IP routing protocol. IS–IS is an ISO protocol that has been adapted ("integrated") for IP in order to carry IP routing information inside non-IP packets.

IS–IS packets are not IP packets, but rather ConnectionLess Network Protocol (CLNP) packets. CLNP packets have ISO addresses, not IP source and destination addresses. CLNP packets are not normally used for the transfer of user traffic from client to server, but for the transfer of link-state routing information between routers. IS–IS does not have "routers" at all: Routers are called *intermediate systems* to distinguish them from the *end systems* (ES) that send and receive traffic.

The independence of IS–IS from IP has advantages and disadvantages. One advantage is that network problems can often be isolated to IP itself if IS–IS is up and running between two routers. One disadvantage is that there are now sources and destinations on the network (the ISO addresses) that are not even "ping-able." So if a link between

two routers is configured with incorrect IP addresses (such as `10.0.37.1/24` on one router and `10.0.38.2/24` on the other), IS–IS will still come up and exchange routing information over the link, but IP will not work correctly, leaving the network administrators wondering why the routing protocol is working but the routes are broken.

Our network does not use IS–IS, so much of this section will be devoted to introducing IS–IS terminology, such as link-state protocol (LSP) data unit instead of OSPF's link-state advertisement (LSA), and contrasting IS–IS behavior with OSPF.

The IS–IS Attraction

If IS–IS is used instead of OSPF as an IGP within an AS, there must be strong reasons for doing so. Why introduce a new type of packet and addressing to the network? And even the simple task of assigning ISO addresses to routers can be a complex task. Yet many ISPs see IS–IS as being much more flexible than OSPF when it comes to the structure of the AS.

IS–IS routers can form both Level 1 (L1) and Level 2 (L2) adjacencies. L1 links connect routers in the same IS–IS area, and L2 links connect routers in different areas. In contrast to OSPF, IS–IS does not demand that traffic sent between areas use a special backbone area (Area 0.0.0.0). IS–IS does not care if interarea traffic uses a special area or not, as long as it gets there. The same is true when a larger ISP acquires a smaller one and it is necessary to "paste" new areas onto existing areas. With IS–IS, an ISP can just paste the new area wherever it makes sense and configure IS–IS L1/L2 routers in the right places. IS–IS takes care of everything.

A backbone area in IS–IS is simply a contiguous collection of routers in different areas capable of running L2 IS–IS. The fact that the routers must be directly connected (contiguous) to form the backbone is not too much as a limitation (most core routers on the backbone usually have multiple connections). Each and every IS–IS backbone router can be in a different area. If an AS structure similar to centralized OSPF is desired, this is accomplished in IS–IS by running certain (properly connected) routers as L2-only routers in one selected area (the backbone), connecting areas adjacent to the central area with L1/L2 routers, and making the other the routers in the other areas L1-only routers. The IS–IS attraction is in this type of flexibility compared to OSPF.

IS–IS and OSPF

ISO's idea of a network layer protocol was CLNP. To distribute the routing information, ISO invented ES–IS to get routing information from routers to and from clients and servers, and IS–IS to move this information between routers.

IS–IS came from DEC as part of the company's effort to complete DECnet Phase V. Standardized as ISO 10589 in 1992, it was once thought that IS–IS would be the natural progression from RIP and OSPF to a better routing protocol. (OSPF was struggling at the time.) To ease the transition from IP to OSI-RM protocols, Integrated IS–IS (or Dual IS–IS) was developed to carry routing information for both IP and ISO-RM protocols.

OSPF rebounded, ironically by often borrowing what had been shown to work in IS–IS. Today OSPF is the recommended IGP to run on the Internet, but IS–IS still has adherents for reasons of flexibility. Of course, OSPF has much to recommend it as well.

Similarities of OSPF and IS–IS

- Both IS–IS and OSPF are link-state protocols that maintain a link-state database and run an SPF algorithm based on Dijkstra to compute a shortest path tree of routes.
- Both use Hello packets to create and maintain adjacencies between neighboring routers.
- Both use areas that can be arranged into a two-level hierarchy or into interarea and intraarea routes.
- Both can summarize addresses advertised between their areas.
- Both are classless protocols and handle VLSM.
- Both will elect a designated router on broadcast networks, although IS–IS calls it a designated intermediate system (DIS).
- Both can be configured with authentication mechanisms.

Differences between OSPF and IS–IS

Many of the differences between IS–IS and OSPF are terminology. The use of the terms IS and ES have been mentioned. IS–IS has a subnetwork point of attachment (SNPA) instead of an interface, protocol data units (PDUs) instead of packets, and other minor differences. OSPF LSAs are IS–IS link-state PDUs (LSPs), and LSPs are packets all on their own and do not use OSPF's LSA-OSPF header-IP packet encapsulation.

But all IS–IS and OSPF differences are not trivial. Here are the major ones.

Areas—In OSPF, ABRs sit on the borders of areas, with one or more interfaces in one area and other interfaces in other areas. In IS–IS, a router (IS) is either totally in one area or another, and it is the links between the routers that connect the areas.

Route Leaking—When L2 information is redistributed into L1 areas, it is called *route leaking*. Route leaking is defined in RFC 2966. A bit called the Up/Down bit is used to distinguish routes that are local to the L1 area (Up/Down = 0) from those that have been leaked in the area from an L1/L2 router (Up/Down = 1). This is necessary to prevent potential routing loops. Route leaking is a way to make IS–IS areas with LI only routers as "smart" as OSPF routers in not-so-stubby-areas (NSSAs).

Network Addresses—CLNP does not use IP addresses in its packets. IS–IS packets use a single *ISO area address* (Area ID) for the entire router because the router must be within one area or another. Every IS–IS router can have up to three different area ISO addresses, but this chapter uses one ISO address per router. The ISO Area ID is combined with an *ISO system address* (System ID) to give the ISO *Network Entity Title*, or NET. Every router must be given an ISO NET as described in ISO 8348.

Network Types—OSPF has five different link or network types that OSPF can be configured to run on: point-to-point, broadcast, non-broadcast multi-access (NBMA), point-to-multipoint, and virtual links. In contrast, IS–IS defines only two types of links or *subnetworks*: broadcast (LANs) and point-to-point (called "general topology"). This only distinguishes links that can support multicasting (broadcast) and use a designating router (DIS) and links that do not support multicasting.

Designated Intermediate System (DIS)—Although IS–IS technically uses a DIS, many still refer to these devices as a designated router (DR). The DIS or DR represents the entire multiaccess network link (such as a LAN) as a single *pseudo-node*. The pseudo-node (a "virtual node" in some documentation) does not really exist, but there are LSPs that are issued for the entire multiaccess network as if the pseudo-node were a real device. Unlike OSPF, all IS–IS routers on a pseudo-node (such as a LAN) are always fully adjacent to the pseudo-node. This is due to the lack of a backup DIS, and new DIS elections must take place quickly.

LSP Handling—IS–IS routers handle LSPs differently than OSPF routers handle LSAs. While OSPF LSAs age from zero to a maximum (MaxAge) value of 3600 seconds (1 hour), IS–IS LSPs age downward from a MaxAge of 1200 seconds (20 minutes) to 0. The normal refresh interval is 15 minutes. Since IS–IS does not use IP addresses, multicast addresses cannot be used in IS–IS for LSP distribution. Instead, a MAC destination address of `0180.c200.0014` (AllL1ISs) is used to carry L1 LSPs to L1 ISs (routers), and a MAC destination address of `0180.c200.0015` (AllL2ISs) is used to carry L2 LSPs to L2 ISs (routers).

Metrics—Like OSPF, IS–IS can use one of four different metrics to calculate least-cost paths (routes) from the link-state database. For IS–IS, these are default (all routers must understand the default metric system), delay, expense, and error (reliability in OSPF). Only the default metric system is discussed here, as with OSPF, and that is the only system that most router vendors support. The original IS–IS specification used a system of metric values that could only range from 0 to 63 on a link, and paths (the sum of all link costs along the route) could have a maximum cost of 1023. Today, IS–IS implementations allow for "wide metrics" to be used with IS–IS. This makes the IS–IS metrics 32 bits wide.

IS–IS for IPv6

One advantage that IS–IS has over OSPF is that IS–IS is not an IP protocol and is not as intimately tied up with IPv4 as OSPF. So IS–IS has fewer changes for IPv6: IPv4 is already strange enough.

With IPv6, the basic mechanisms of RFC 1195 are still used, but two new Type-Length-Vector (TLVs, which define representation) types are defined for IPv6.

IPv6 Interface Address (type 232)—This TLV just modifies the interface address field for the 16-byte IPv6 address space.

IPv6 Reachability (type 236)—This TLV starts with a 32-bit wide metric. Then there is an Up/Down bit for route leaking, an I/E bit for external (other routing protocol or AS) information, and a "sub-TLVs present?" bit. The last 5 bits of this byte are reserved and must be set to 0. There is then 1 byte of Prefix Length (VLSM) and from 0 to 16 bytes of the prefix itself, depending on the value of the Prefix Length field. Zero to 248 bytes of sub-TLVs end the TLV.

Both types have defined sub-TLVs fields, but none of these has yet been standardized.

QUESTIONS FOR READERS

Figure 14.7 shows some of the concepts discussed in this chapter and can be used to help you answer the following questions.

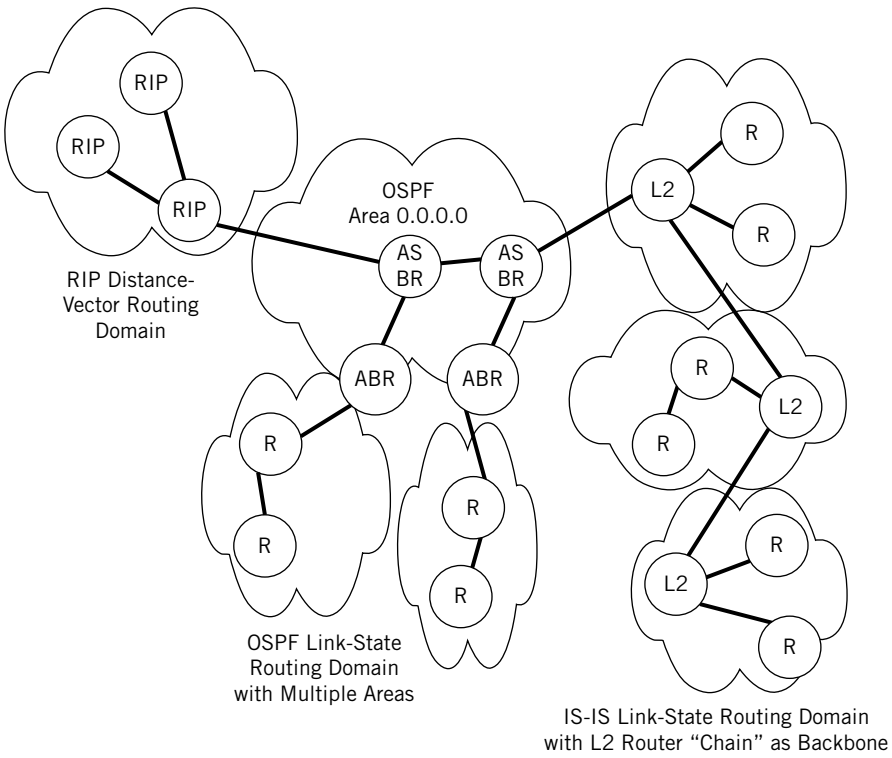


FIGURE 14.7

Three IGPs and some of their major characteristics.

1. Why does RIP continue to be used in spite of its limitations?
2. What is the difference between distance-vector and link-state routing protocols?
3. It is often said that it is easier to configure a backbone area in IS–IS than in OSPF. What is the basis for this statement?
4. What are the similarities between OSPF and IS–IS?
5. What are the major differences between OSPF and IS–IS?