CHAPTER 6

Open Shortest Path First (OSPF)

In this chapter:

Last year I flew from New York to Osaka for a conference. My journey began when I hailed a cab on Broadway in downtown New York. "JFK," I told the cabbie, telling her my destination was John F. Kennedy Airport. I was still pushing my luggage down the seat so I could pull my door shut when the cab started to move. The cabbie changed lanes twice before I got it shut. I did make it to JFK in one piece, where I presented my ticket and boarded a flight to Osaka. At Osaka Airport, the taxi driver bowed to me as he took my luggage from my hand. Once the luggage was properly stowed, he asked for my destination. "New Otani Hotel," I told him, and he bowed again and closed my side door.

This everyday story of a passenger in transit illustrates how a traveler is able to complete a journey in spite of the fact that the whereabouts of his destination are not known to every element in the system. The cabbie in New York knows only local destinations and so knows how to get to JFK but not to the New Otani Hotel. The airline routes passengers between major airports. The taxi driver in Osaka also knows only local destinations, so, when returning to New York, I tell the driver that my destination is "Osaka Airport," not "New York." Any single element of the transportation system knows only the *local* geography. This leads to obvious efficiencies: the cabbie in New York needs to know only the New York metropolitan area, and the taxi driver in Osaka needs to know only the area in and around Osaka; the airline is the backbone linking JFK to Osaka.

Much like the transportation system just described, Open Shortest Path First (OSPF) is a *hierarchical* routing protocol, implying that the IP network has a geography with each *area* possessing only local routing information. In contrast, RIP and IGRP are *flat,* implying that there is no hierarchy in the network—every router possesses

routes to every destination in the network. Right away, you can see that a flat routing protocol has inherent inefficiencies—in our analogy, if the architecture of the transportation system was flat, the cabbie in New York would have to learn directions to the New Otani Hotel.

A hierarchical architecture, whether that of a transportation system or that of OSPF, allows the support of large systems because each area is responsible only for its local routes. RIP and IGRP cannot support very large networks because the routing overhead increases linearly with the size of the network.

Another radical difference from RIP and IGRP is that OSPF is not a DV protocol—OSPF is based on a Link State algorithm, Dijkstra. What is a Link State algorithm? *Link* refers to a router interface; in other words, the attached network. *State* refers to characteristics of the link such as its IP address, subnet mask, cost (or metric), and operational status (up or down). Routers executing OSPF describe the state of their directly connected links in *link state advertisement* (LSA) packets that are then flooded to all other routers. Using all the LSAs it receives, each router builds a topology of the network. The network topology is described mathematically in the form of a graph.

This topological database is the input to Dijkstra's Shortest Path First (SPF) algorithm. With itself as the root, each router runs the SPF algorithm to compute the shortest path to each network in the graph. Each router then uses its shortest-path tree to build its routing table. Compare this with DV protocols: DV protocols propagate routes from router to router (this is sometimes called routing by rumor) and each router chooses the best route (to each destination) from all the routes (to that destination) that it hears.

DV protocols have to set up special mechanisms to guard against bad routing information that could propagate from router to router. In contrast, routers running the SPF algorithm need to ensure the accuracy of their LS databases; as long as each router has the correct topology information, it can use the SPF algorithm to find the shortest path.

Dijkstra's algorithm is a wonderful tool but, as we shall see in more detail later, the SPF algorithm is expensive in terms of CPU utilization. The cost of running the algorithm increases quickly as the network topology grows. This would be a problem but, given OSPF's hierarchical structure, the network is divided into "small" areas, and the SPF algorithm is executed by each router only on its intra-area topology. So how do routers in two different areas communicate with each other? All areas summarize their routes to a special area called the *backbone area* or *area 0*. The backbone area in turn summarizes routes to all attached areas. Hence, traffic between any two areas must pass through the backbone area (see Figure 6-1).

OSPF derives its name from Dijkstra's SPF algorithm; the prefix "O" signifies that it's an "open" protocol and so is described in an "open" book that everyone can access. That open book is RFC 2328, thanks to John Moy. In contrast, IGRP and EIGRP are Cisco *proprietary* protocols. Multiple vendors support OSPF.

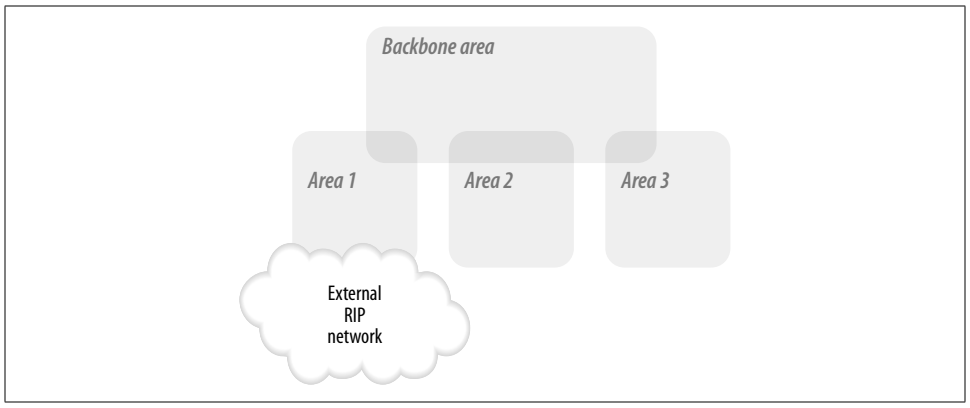

Figure 6-1. Overview of OSPF areas

Getting OSPF Running

Getting RIP, IGRP, and EIGRP running is easy, as we saw in earlier chapters. When TraderMary's network grew to London, Shannon, Ottawa, etc., the DV routing protocols adapted easily to the additions. Getting OSPF running on a small network is also easy, as we will see in this chapter. However, unlike RIP, IGRP, and EIGRP, OSPF is a hierarchical protocol. OSPF does not work well if the network topology grows as a haphazard mesh.

In this section, we will configure OSPF on a small network. In later sections, we will learn how to build hierarchical OSPF networks.

TraderMary's network, shown in Figure 6-2, can be configured to run OSPF as follows.

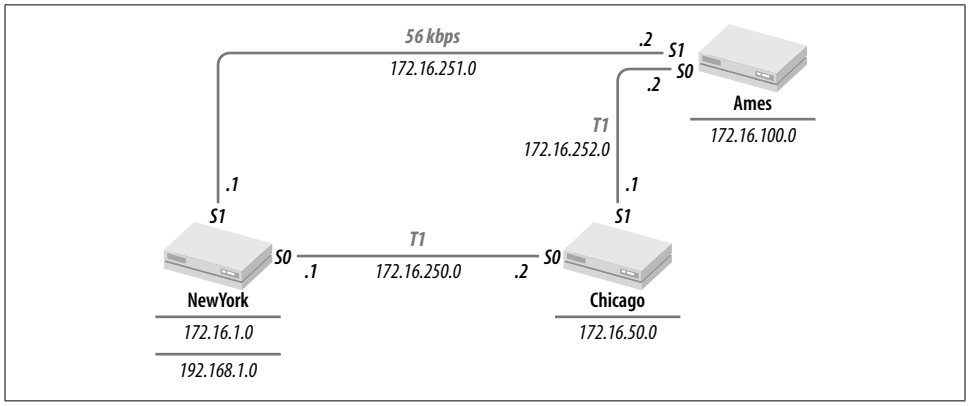

Figure 6-2. TraderMary's network

Like RIP and IGRP, OSPF is a distributed protocol that needs to be configured on every router in the network:

```
hostname NewYork
...
interface Ethernet0
ip address 172.16.1.1 255.255.255.0
!
interface Serial0
description New York to Chicago link
ip address 172.16.250.1 255.255.255.0
!
interface Serial1
description New York to Ames link
1 bandwidth 56
ip address 172.16.251.1 255.255.255.0
...
router ospf 10
network 172.16.0.0 0.0.255.255 area 0
```

The *router ospf* command starts the OSPF process on the router. The syntax of this command is:

```
router ospf process-id
```

The *process-id*, which should be between 1 and 65,535, is used to identify the instance of the OSPF process. The *process-id* configured in the previous example is 10. Router *Chicago* is similarly configured with the same *process-id*:

```
hostname Chicago
...
interface Ethernet0
ip address 172.16.50.1 255.255.255.0
!
interface Serial0
description Chicago to New York link
ip address 172.16.250.2 255.255.255.0
!
interface Serial1
description Chicago to Ames link
ip address 172.16.252.1 255.255.255.0
...

router ospf 10
network 172.16.0.0 0.0.255.255 area 0
```

Router *Ames* is also configured with OSPF:

```
hostname Ames
...
interface Ethernet0
ip address 172.16.100.1 255.255.255.0
!
interface Serial0
```

```
   description Ames to Chicago link
   ip address 172.16.252.2 255.255.255.0
   !
   interface Serial1
   description Ames to New York link
2  bandwidth 56
   ip address 172.16.251.2 255.255.255.0
   ...

   router ospf 10
   network 172.16.0.0 0.0.255.255 area 0
```

We next identify the networks that will be participating in the OSPF process and associate an area ID with each network. The syntax of this command is:

```
network address wildcard-mask area area-id
```

The *address* and *wildcard-mask* fields identify a network by its IP address. Networks that match the *address* and *wildcard-mask* fields are associated with the area *area-id*. How is a network's IP address matched against *address* and *wildcard-mask*?

wildcard-mask is a string of zeros and ones. An occurrence of a zero in *wildcard-mask* implies that the IP address being checked must exactly match the corresponding bit in *address*. An occurrence of a one in *wildcard-mask* implies that the corresponding bit in the IP address field is a "don't care bit"—the match is already successful.

Thus, the following clause can be read as stating that the first 16 bits of an IP address must be exactly "172.16" for the address to match the clause and be associated with area 0 and that the next 16 bits of the IP address are "don't care bits":

```
network 172.16.0.0 0.0.255.255 area 0
```

Any IP address, such as `172.16.x.y`, will match this *address/wildcard-mask* and be assigned the area ID of 0. Any other address, such as `10.9.x.y`, will not match this *address/wildcard-mask*.

If an interface IP address does not match the *address/wildcard-mask* on a network statement, OSPF will check for a match against the next network statement, if there is another statement. Hence, the order of network statements is important. If an interface IP address does not match the *address/wildcard-mask* on any network statement, that interface will not participate in OSPF.

There is more than one method of assigning area IDs to networks. The most rigorous method specifically lists every network when making a match. The wildcard mask contains only zeros:

```
hostname NewYork
...
router ospf 10
network 172.16.1.1 0.0.0.0 area 0
network 172.16.250.1 0.0.0.0 area 0
network 172.16.251.1 0.0.0.0 area 0
```

The most loose method is an all-ones wildcard mask:

```
hostname NewYork
...
router ospf 10
network 0.0.0.0 255.255.255.255 area 0
```

Note that in the second (loose) method, network `192.168.1.1` also belongs to area 0.

If an IP address does not match an area-ID specification, the match continues to the next statement. So, for example, a router may be configured as follows:

```
network 172.16.0.0 0.0.255.255 area 0
network 192.0.0.0 0.255.255.255 area 1
```

An IP address of `192.168.1.1` will not match the first statement. The match will then continue to the next statement. All IP addresses with "192" in the first 8 bits will match the second clause and hence will fall into area 1. A network with the address `10.9.1.1` will not match either statement and hence will not participate in OSPF.

The *area-id* field is 32 bits in length. You can specify the area ID in the decimal number system, as we did earlier, or in the dotted-decimal notation that we use for expressing IP addresses. Thus, the area ID `0.0.0.0` (in dotted decimal) is identical to the area ID 0 (in decimal); the area ID `0.0.0.100` (in dotted decimal) is identical to 100 (in decimal); and the area ID `0.0.1.0` (in dotted decimal) is identical to 256 (in decimal). The area ID of 0 is reserved for the backbone area. The area ID for nonbackbone areas can be in the range 1 to 4,294,967,295 (or, equivalently, `0.0.0.1` to `255.255.255.255`).

The *show ip ospf interface* command shows the assignment of area IDs to network interfaces:

```
NewYork#sh ip ospf interface
...
Ethernet0 is up, line protocol is up
3   Internet Address 172.16.1.1/24, Area 0
4   Process ID 10, Router ID 172.16.251.1, Network Type BROADCAST, Cost: 10
    ...
Serial0 is up, line protocol is up
  Internet Address 172.16.250.1/24, Area 0
  Process ID 10, Router ID 172.16.251.1, Network Type POINT_TO_POINT, Cost: 64
...
Serial1 is up, line protocol is up
  Internet Address 172.16.251.1/24, Area 0

  Process ID 10, Router ID 172.16.251.1, Network Type POINT_TO_POINT, Cost: 1785
...
```

The routing tables for *NewYork*, *Chicago*, and *Ames* will show all `172.16.0.0` subnets. Here is *NewYork*'s table:

```
NewYork#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
```

```
          E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
          i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default

   Gateway of last resort is not set

5        172.16.0.0/16 is variably subnetted, 6 subnets, 2 masks
6  O        172.16.252.0/24 [110/128] via 172.16.250.2, 01:50:18, Serial0
   C        172.16.250.0/24 is directly connected, Serial0
   C        172.16.251.0/24 is directly connected, Serial1
7  O        172.16.50.1/32 [110/74] via 172.16.250.2, 01:50:18, Serial0
   C        172.16.1.0/24 is directly connected, Ethernet0
8  O        172.16.100.1/32 [110/138] via 172.16.250.2, 01:50:18, Serial0
```

The OSPF-derived routes in this table are labeled with an "O" in the left margin. Note that the routing table provides summary information (as in line 5). This line contains subnet mask information (24 bits, or `255.255.255.0`) and the number of subnets in `172.16.0.0` (6).

OSPF Metric

Each OSPF router executes Dijkstra's SPF algorithm to compute the shortest-path tree from itself to every subnetwork in its area. However, RFC 2328 does not specify how a router should compute the cost of an attached network—this is left to the vendor. Cisco computes the cost of an attached network as follows:

$$Cost = 10^8/bandwidth\ of\ interface\ in\ bits\ per\ second$$

Using this definition, the OSPF cost for some common media types is shown in Table 6-1. Table 6-1 assumes default interface bandwidth. Note that the cost is rounded down to the nearest integer.

Table 6-1. Default OSPF costs

Media type	Default bandwidth	Default OSPF cost
Ethernet	10 Mbps	10
Fast Ethernet	100 Mbps	1
FDDI	100 Mbps	1
T-1 (serial interface)[a]	1,544 kbps	64
56 kbps (serial interface)	1,544 kbps	64
HSSI	45,045 kbps	2

[a] All serial interfaces on Cisco routers are configured with the same default bandwidth (1,544 kbits/s) and delay (20,000 ms) parameters.

The OSPF cost computed by a router can be checked with the command *show ip ospf interface*, as in line 4 in the code block in the previous section, where the cost of the Ethernet segment is 10. The composite cost of reaching a destination is the sum of the individual costs of all networks in the path to the destination and can be seen as output of the *show ip route* command in lines 6, 7, and 8.

The default value of the OSPF metric may not be adequate in some situations. For example, in TraderMary's configuration, the *NewYork → Ames* link runs at 56 kbps, but the default metric makes it appear to be a T-1. This was fixed by modifying the interface bandwidth, as in lines 3 and 4 in the previous section. The command to modify a bandwidth is:

```
bandwidth kilobits
```

Keep in mind that modifying the interface bandwidth impacts other protocols that utilize the bandwidth parameter, such as IGRP. Modifying bandwidth may not always be viable. In such situations, the OSPF cost of an interface may be directly specified:

```
ip ospf cost value
```

where *value* is an integer in the range 1 to 65,535 (OSPF sets aside two octets to represent interface cost, as we will see later in the section "How OSPF Works").

This approach to calculating OSPF costs does not work well for network speeds greater than 100 Mbps. The OSPF cost for all speeds greater than the reference bandwidth is rounded up to 1, and there is no way to distinguish between one network and another. The network engineer has two approaches to choose from here. First, manually configure the OSPF cost for all interfaces equal to or faster than 100 Mbps. For example, all FE interfaces may be configured with a cost of 8, OC-3 interfaces with a cost of 6, and GE interfaces with a cost of 4. Second, redefine the reference bandwidth with the following command:

```
ospf auto-cost reference-bandwidth reference-bandwidth
```

where *reference-bandwidth* is in Mbps. When this command is used, the cost of an interface is calculated as:

Cost = reference-bandwidth-in-bps/bandwidth of interface in bits per second

This command is available in Cisco IOS Releases 11.2 and later. If the reference bandwidth is modified, it must be modified on all routers in the OSPF domain. The default value of *reference-bandwidth* is 10^8.

The developers of OSPF envisaged (as an optional feature) multiple *types of service* (TOS) with differing metrics for each TOS. Using this concept, bulk data may be routed, say, over a satellite link, whereas interactive data may be routed under the sea. However, the TOS concept has not been carried into any major implementations—Cisco supports only one TOS.

Definitions and Concepts

Dijkstra's algorithm solves the problem of discovering the shortest path from a single source to all vertices in a graph where the edges are each represented with a cost. For example, a car driver could use Dijkstra's algorithm to find the shortest paths from New York to major cities in the northeastern U.S. and Canada. The input to Dijkstra would be a graph that could be represented by a matrix like that shown in Table 6-2.

Table 6-2. Driving distances

Town name	Town name	Driving distance (miles)
New York	Washington	236
New York	Boston	194
Boston	Chicago	996
Washington	Chicago	701
New York	Toronto	496
Detroit	Chicago	288
Washington	Detroit	527
Boston	Toronto	555
Toronto	Detroit	292

The output would be the shortest paths from New York to all other cities in the graph. A geographical view of Table 6-2 is contained in Figure 6-3.

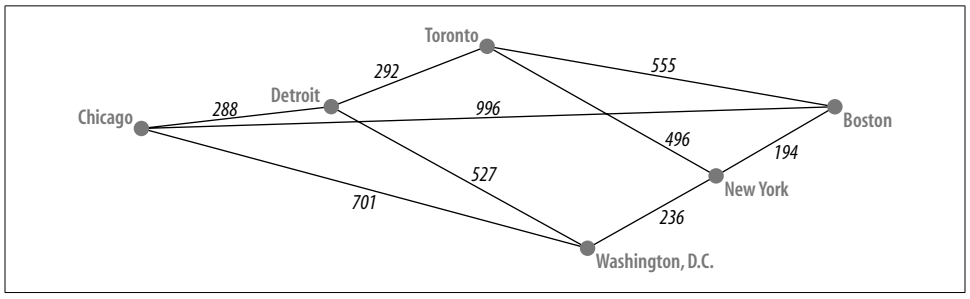

Figure 6-3. Geographical view of driving distances

There are six nodes in this graph: New York, Chicago, Boston, Toronto, Detroit, and Washington. There are nine edges in the graph, each represented by the distance between a pair of vertices. The SPF algorithm works as follows:

1. Starting at the source node—New York—build a list of one-segment paths originating at the source node. This list will be New York → Washington, New York → Boston, and New York → Toronto.
2. Sort this list in increasing order. The sorted list will be New York → Boston (194), New York → Washington (236), and New York → Toronto (496).
3. Pick the shortest path from this list—New York → Boston—and move Boston to the list of vertices for which the shortest path has been identified.
4. Next, append a new list of paths to the list that was defined in step 1. The list to be appended consists of one-segment paths starting from Boston. This list will be Boston → Chicago and Boston → Toronto. The composite list will be New York → Washington, New York → Toronto, Boston → Chicago, and Boston → Toronto.

The algorithm continues, as in step 2, and the composite list is sorted in increasing order with distances from the source node: New York → Washington (236), New York → Toronto (496), New York → Boston → Toronto (194 + 555 = 749), and New York → Boston → Chicago (194 + 996 = 1,190). In step 3, the shortest path is again picked from the top of the list and Washington is added to the list of vertices for which the shortest path has been identified. The algorithm continues until the shortest paths to all cities have been identified.

OSPF employs Dijkstra's SPF algorithm to compute the shortest path from a router to every network in the graph. In OSPF terminology, this graph of the network topology (similar to Table 6-2) is referred to as the topological database or the *link state database*. Each router executes the SPF algorithm with itself as the source node. The results of the SPF algorithm are the shortest paths to each IP network from the source node; hence, this constitutes the IP routing table for the router.

Although the database of Table 6-2 is relatively static—driving distances change only when new roads are built or old roads are closed—the LS database for a network is quite dynamic because of changes in the state of subnetworks. A link may go down or come up. A network administrator may make changes to the status of a link, such as shutting it down or changing its cost. Every time there is any change in a router's LS database, Dijkstra's SPF algorithm needs to be run again. It can be shown that the SPF algorithm takes ElogE time to run, where E is the number of edges in the graph.

As the size of a network grows, Dijkstra will consume more and more memory and CPU resources at each router. In other words, Dijkstra does not scale for large topologies. Fortunately, OSPF has a clever solution to this problem: break the network into *areas* and execute Dijkstra only on each *intra-area* topology.

An area is a collection of *contiguous* networks and routers that share a unique area ID. Each area maintains its own topological database: other areas do not see this topological information. The SPF algorithm is executed on each intra-area topology by the intra-area routers.

Containing the number of routers and networks in an area allows OSPF to scale to support large networks. The network can grow almost without bounds with the addition of new areas. If a single area becomes too large, it can be split into two or more areas.

Before a router can execute the SPF algorithm, it must have the most recent topological database for its area(s). Note the plural: a router may have interfaces in multiple areas. A topological change in an area will cause SPF to recompute on all routers with interfaces in that area. Routers in other areas will not be affected by the change. Breaking a network into areas is thus akin to breaking a network into smaller, independent networks.

Unlike flat networks such as RIP and IGRP in which each router has the same responsibilities and tasks, OSPF's hierarchy imposes a structure in which routers and even areas are differentiated with respect to their roles.

Backbone Area

The *backbone area* is of special significance in OSPF because all other areas must connect to it. The area ID of 0 (or `0.0.0.0`) is reserved for the backbone. Figure 6-4 shows an OSPF network comprised of a backbone area and three other areas—areas 1, 2, and 3. Note that all inter-area traffic must pass through the backbone area, which implies that backbone routers must possess the complete topological database for the network.

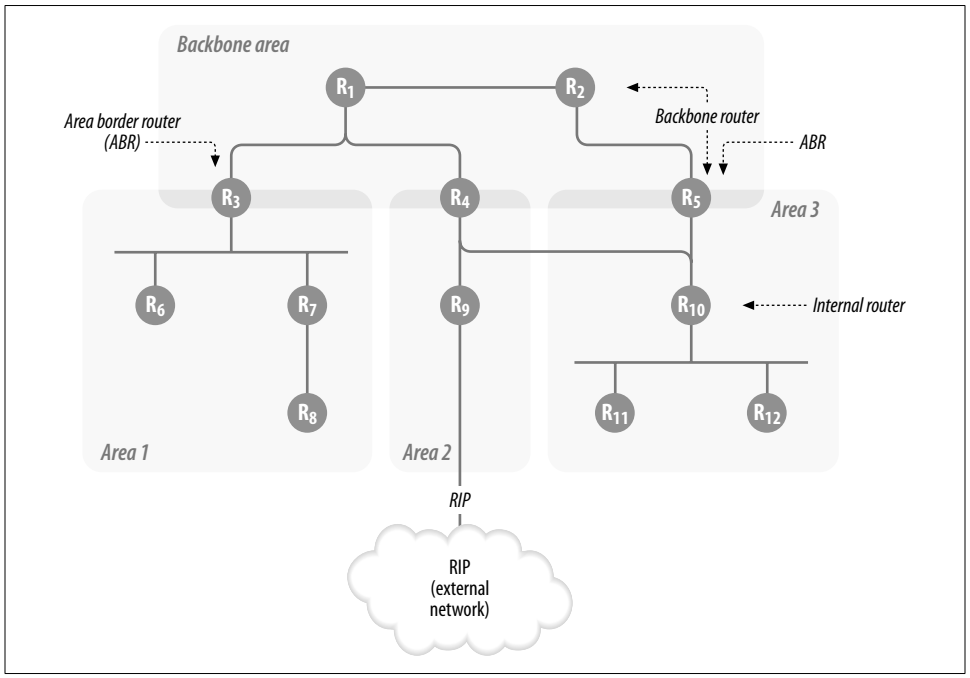

Figure 6-4. OSPF architecture: a high-level view

Backbone Router

A router with an interface in area 0 is referred to as a *backbone router*. A backbone router may also have interfaces in other areas. Routers *R1*, *R2*, *R3*, *R4*, and *R5* in Figure 6-4 are backbone routers.

The backbone routers hold a topological database that describes the state of all backbone links, summary links describing IP networks in areas 1, 2, and 3, and external links that describe the IP network in the RIP network.

Area or Regular Area

A *regular area* has a unique area ID in the range 1 (or `0.0.0.1`) to 4,294,967,295 (`255.255.255.255`).

A router in, say, area 1 will hold topological information for the state of all area 1 links, summary links that describe IP networks in areas 0, 2, and 3, and external links that describe IP networks in those networks.

Internal Router

An *internal router* has interfaces in one area only. Routers *R6*, *R7*, and *R8* in Figure 6-4 are internal routers in area 1.

Area Border Router (ABR)

An *area border router* has interfaces in more than one area. Routers *R3*, *R4*, and *R5* in Figure 6-4 are ABRs.

An ABR has topological information for multiple areas. Router *R3* is an ABR that holds topological databases for areas 0 and 1. Router *R4* holds topological databases for areas 0, 2, and 3. Router *R5* holds topological databases for areas 0 and 3.

An ABR can summarize the topological database for one of its areas. Router *R3* may summarize the topological database for area 1 into area 0. Summarization is key in reducing the computational complexity of the OSPF process.

Autonomous System Boundary Router (ASBR)

An *autonomous system boundary router* imports routing information from another AS into OSPF. The routes imported into OSPF from the other AS are referred to as *external routes*.

Router *R9* in Figure 6-4 is an ASBR. *R9* imports RIP routes from an external network into OSPF. An ASBR may be configured to summarize external routes into OSPF.

Stub Area

Consider an area with no direct connections to any external networks. Importing external records into this area may be unnecessary because all traffic to external networks must be routed to the ABRs. Such an area can use a default route (in place of external routes) to send all external IP traffic to its ABRs.

Configuring an area as a *stub area* blocks the advertisement of external IP records at the ABRs and instead causes the ABRs to generate default routes into the stub area.

Routers in a stub area hold a topological database that describes the state of all local links, summary links describing IP networks in other areas, but no external networks. This reduction in the size of the topological database saves on processor and memory resources. A stub area may use routers with less memory/CPU power or use the spare memory/CPU resources to build a *large* stub area.

There is a potential disadvantage to configuring an area as a stub area. For example, if area 3 in Figure 6-4 is configured as a stub area, *R4* and *R5* will each advertise a default route into the stub area. An external route may be closer to *R4*, but routers in the stub area will lose that information and route all external traffic to *R4* or *R5*, depending on which one is closer. Stub areas cannot support external connections since stub routers do not carry external LSAs. Stub areas cannot support virtual links, which I'll discuss later in this chapter, for similar reasons.

Totally Stubby Area

A *totally stubby area* carries the concept of a stub area further by blocking summary records for IP networks in other areas at the ABRs. All inter-area and external traffic is matched to the default route announced by the ABR(s).

In terms of LSA types, routers in totally stubby areas hold a topological database that describes the state of all local links only.

Just like a stub area, a totally stubby area cannot support connections to external networks.

Not So Stubby Area (NSSA)

Not so stubby areas are stub areas with one less restriction: NSSAs can support external connections. In all other respects, NSSAs are just like stub areas—routers in NSSAs do not carry external LSAs, nor do they support virtual links.

Any area that can be configured as a stub area but needs to support an external network can be changed into an NSSA.

OSPF Topological Database

The OSPF topological database is composed of link state advertisements (LSAs). OSPF routers originate LSAs describing a piece of the network topology; these LSAs are flooded to other routers that then compose a database of LSAs. There are several types of LSAs, each originating at a different router and describing a different component of the network topology. The various types of LSAs are:

Router LSA (type 1)
: A router LSA describes a router's links (or interfaces). All routers originate router LSAs. A router LSA is flooded to all intra-area routers.

Network LSA (type 2)
: A network LSA describes a broadcast network (such as an Ethernet segment) or a non-broadcast multi-access (NBMA) network (such as Frame Relay). All routers attached to the broadcast/NBMA network are described in the LSA. A network LSA is flooded to all intra-area routers.

Summary LSA (type 3)
: A summary LSA describes IP networks in another area. The summary LSA is originated by an ABR and flooded outside the area. Summary LSAs are flooded to routers in all OSPF areas except totally stubby areas.

ASBR summary LSA (type 4)
: ASBR summary LSAs describe the route to an ASBR. The mask associated with these LSAs is 32 bits long because the route they advertise is to a host—the IP address of the ASBR. ASBR summary LSAs originate at ASBRs. ASBR summary LSAs are flooded to routers in all OSPF areas except stub areas.

External LSA (type 5)
: External LSAs describe routes external to the OSPF process (in another autonomous system). An external route can be a default route. External LSAs originate at the ASBR. External LSAs are flooded throughout the OSPF network, except to stub areas.

NSSA external LSA (type 7)
: NSSA external LSAs describe routes to external networks (in another autonomous system) connected to the NSSA. Unlike type 5 external LSAs, NSSA external LSAs are flooded only within the NSSA. Optionally, type 7 LSAs may be translated to type 5 LSAs at the ABR and flooded as type 5 LSAs.

OSPF Route Types

Every router in OSPF uses its local topological database as input to the SPF algorithm. The SPF algorithm yields the shortest path to every known destination, which is then used to populate the IP routing table as one of four route types:

Intra-area route
: An intra-area route describes the route to a destination within the area.

Inter-area route
: An inter-area route describes the route to a destination in another area. The path to the destination comprises an intra-area path, a path through the backbone area and an intra-area path in the destination network's area. An inter-area route is sometimes referred to as a summary route.

External route (type 1)
: An external route describes the route to a destination outside the AS. The cost of a type 1 external route is the sum of the costs of reaching the destination in the external network and the cost of reaching the ASBR advertising the route.

External route (type 2)
: An external route describes the route to a destination outside the AS. The cost of a type 2 external route is the cost of reaching the destination in the external network only; it does not include the cost of reaching the ASBR advertising the route.

When routing a packet, the routing table is scanned for the most specific match. For example, say that the destination IP address in the packet is `10.1.1.254` and the routing table contains entries for `10.1.1.0/24` and `10.1.1.192/26`. The most specific match will be the route `10.1.1.192/26`. Now, what if `10.1.1.192/26` was known as an intra-area route and an inter-area route? OSPF prefers routes in the following order: intra-area routes (most preferred), inter-area routes, type 1 external routes, and type 2 external routes (least preferred).

Note the order in which the rules were applied: first the route with the most specific match was identified and then the OSPF preferences were applied. Thus, when routing the packet with the destination address `10.1.1.254`, if the routing table shows `10.1.1.0/24` as an intra-area route and `10.1.1.192/26` as a type 2 external route, the most specific match (`10.1.1.192/26`) will win. If OSPF has multiple equal-cost routes to a destination, it will load-balance traffic over those routes.

How OSPF Works

OSPF routers must first discover each other before they can exchange their topological databases. Once each router has the complete topological database, it can use the SPF algorithm to compute the shortest path to every network. This section focuses on neighbor discovery and the exchange of topological databases.

Let's begin at the beginning. OSPF packets are encapsulated directly in IP with the protocol field set to 89. The destination IP address in OSPF depends on the network type. OSPF uses two IP multicast addresses on broadcast and point-to-point networks: `225.0.0.5` for all OSPF routers and `224.0.0.6` for all DR/BDR (designated router/backup designated router) routers. Using IP multicast addresses is more efficient than using broadcast addresses. If broadcast addresses are used, all attached devices must receive the broadcast packet, unwrap it, and then discard the contents if they are not running OSPF. NBMA networks and virtual links use unicast addresses because they do not support multicast addresses.

Following the IP header is the OSPF header (see Figure 6-5). The OSPF header is common to all types of OSPF packets. The following list defines the format of the OSPF header and the five types of OSPF packets:

Version
: The OSPF version in use. The current version number is 2.

Type
: There are five types of OSPF packets:

  Type 1
  : Hello packets, described in the next section.

  Type 2
  : Database description packets, described later under "Database Exchange."

Type 3
: Link state requests, described in "Database Exchange."

Type 4
: Link state updates, described in "Database Exchange."

Type 5
: Link state acknowledgments, described in "Database Exchange."

Packet length
: The length of the OSPF packet, including the header.

Router ID
: The router ID of the router originating the OSPF packet.

Area ID
: The area ID of the network on which this packet is being sent.

Checksum
: The checksum for the entire packet, including the header.

Au type
: The type of authentication scheme in use. The possible values for this field are:

0 No authentication

1 Clear-text password authentication

2 MD5 checksum

Authentication data
: The authentication data.

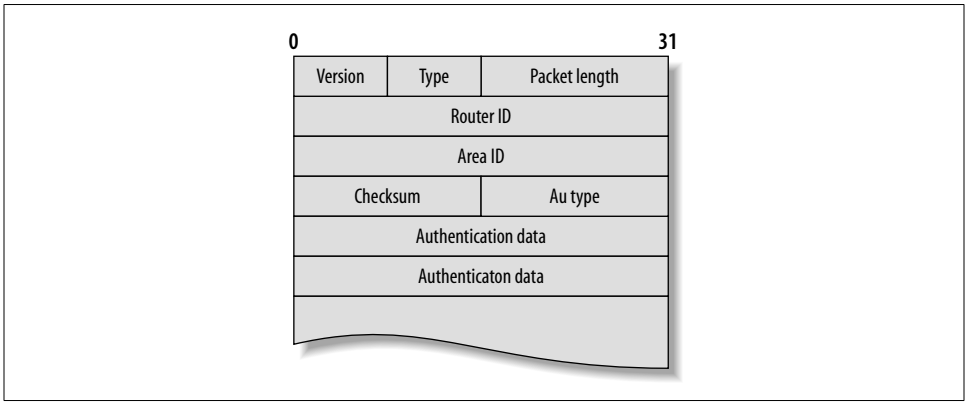

Figure 6-5. Format of an OSPF header

Neighbor Discovery: The Hello Protocol

Every router generates OSPF hello packets on every OSPF-enabled interface. Hello packets are sent every 10 seconds on broadcast media and every 30 seconds on non-

broadcast media. Routers discover their neighbors by listening to hellos. The output of the command *show ip ospf neighbor* lists the neighbors that have been discovered.

Each hello packet contains the fields described in the following sections. The format of a hello packet is shown in Figure 6-6.

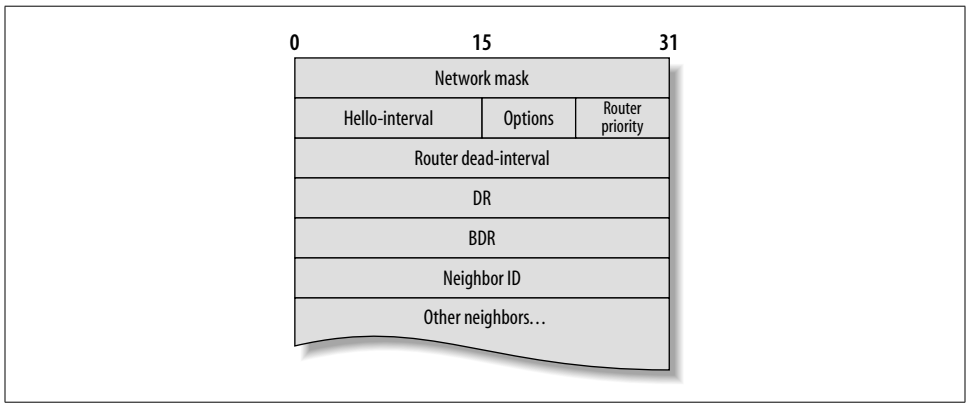

Figure 6-6. Format of hello packet

Router ID

When the OSPF process first starts on a router (e.g., when the router is powered up) it attempts to establish a *router ID*. The router ID is the name or label that will be attached to the node representing the router in the SPF topology graph. If OSPF cannot establish a router ID, the OSPF process aborts.

How does a router choose its router ID? There are two situations to consider here:

- If a router has one or more loopback interfaces, it chooses the highest IP address from the pool of loopback interfaces as its router ID. Loopback interfaces are always active.
- If a router has no loopback interfaces, it chooses the highest IP address from any of its active interfaces as its router ID. If a router has no active interface with an IP address, it will not start the OSPF process.

The router ID is chosen when the OSPF process first starts: the addition or deletion of interfaces or addresses on a router after the router ID has been selected does not change the router ID. A new router ID is picked only when the router is restarted (or when the OSPF process is restarted).

So, for example, the router ID of *NewYork* can be checked as follows:

```
NewYork#sh ip ospf
 Routing Process "ospf 10" with ID 172.16.251.1
 Supports only single TOS(TOS0) routes
 SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
......
```

In this example, the router ID was derived using the router's highest IP address. It is usually preferable to configure loopback interfaces to assign predictable router IDs to OSPF routers (since a loopback interface is a virtual interface and will not go down, as a physical interface would). The router ID must be unique within the topology database.

The configuration on *NewYork* may be modified as follows:

```
hostname NewYork
!
interface Loopback0
 ip address 192.168.1.1 255.255.255.255
...
```

After *NewYork* is rebooted, its router ID will change as follows:

```
NewYork#sh ip ospf
 Routing Process "ospf 10" with ID 192.168.1.1
...
```

Since the router ID is critical to the OSPF process, it is important for the network engineer to maintain a table of all router IDs.

Note the following points:

1. Since the router ID is needed only to represent the router in the SPF graph, it is not required that OSPF advertise the router ID. However, if the router ID is advertised, it will be represented as a stub link in a router LSA.
2. A mask of `255.255.255.255` may be chosen for the loopback interface to conserve on network addresses, as in the earlier example.
3. If the router ID is not advertised, any unique address can be used to represent the router ID—the use of nonreserved IP addresses will not cause any routing-table conflicts.

Area ID

The area ID of the interface on which the OSPF packet is being sent.

Checksum

The checksum pertaining to the hello packet.

Authentication

The authentication method and authentication data.

Network mask

The network mask of the interface on which the hello packet is being sent.

Hello-interval

The duration between hello packets. The default value of hello-interval is 10 seconds on most interfaces.

The hello-interval can be modified with the following command in interface configuration mode:

```
ip ospf hello-interval seconds
```

Options

OSPF defines several optional capabilities that a router may or may not support. The options field is one octet long, as shown in Figure 6-7.

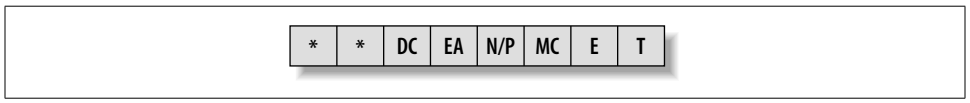

Figure 6-7. Format of the options field

Routers that support demand circuits set the DC bit; NSSA support is signified using the N bit. The E bit signifies that the router accepts external LSAs—stub routers turn off this bit. The T bit signifies the support of multiple types of service.

Router priority

A router with a higher priority takes precedence in the DR election algorithm. A value of 0 makes the router ineligible for DR/BDR election. The default value of this field is 1.

Router dead-interval

If no hello packets are received for the duration of the dead-interval, the neighbor is declared dead. This value can be altered with the following command in interface configuration mode:

```
ip ospf dead-interval value
```

Designated router (DR)

The designated router for multi-access networks. This field is set to `0.0.0.0` if no DR has been elected on the network.

Backup designated router

The IP address of the backup designated router's interface on this network. This field is set to `0.0.0.0` if no BDR has been elected on the network.

Neighbor router ID list

The neighbor router ID list is the list of neighboring routers from which this router has received hellos within the last dead-interval seconds. Before a router lists its neighbor in its hello packet, the two routers must agree on the following: area ID, authentication mechanism, network mask, hello-interval, router dead-interval, and options fields. If these values match, the routers become neighbors and start listing each other in their hello packets.

The following output shows *NewYork*'s neighbors:

```
NewYork#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.1.2       1   FULL/  -        00:00:31    172.16.250.2    Serial0
192.168.1.3       1   FULL/  -        00:00:32    172.16.251.2    Serial1
```

Note that the state of *NewYork*'s relationship with both neighbors is "Full," implying that the neighbors have exchanged LS databases to become adjacent. Under normal, stable conditions, the state of each neighbor relationship should be "2-way" or "Full." "2-way" implies that the neighbors have seen each other's hello packets but have not exchanged LSAs. In the process of maturing into a "Full" relationship, neighbors transition through the states "Exstart," "Exchange," and "Loading," indicating that neighbors have seen each other's hello packets and are attempting to exchange their LS databases. These are transitory states, all being well.

Then there are the problem states. "Down" indicates that a hello packet has not been received from the neighbor in the last router dead-interval. "Attempt" applies to NBMA networks and indicates that a hello has not yet been received from the neighbor. "Init" implies that a hello was received from the neighbor but its neighbor router ID list did not include the router ID of this router.

DR/BDR Election

Consider n routers on a broadcast network (such as Ethernet). If a router exchanged its topological database with every other router on the network, $(n \times (n - 1)) / 2$ adjacencies would be formed on the segment. This would create a lot of OSPF overhead traffic. OSPF solves this problem by electing a *designated router* (DR) and a *backup designated router* (BDR) on each broadcast network. Each router on a broadcast network establishes an adjacency with only the DR and the BDR. The DR and the BDR flood this topology information to all other routers on the segment.

DR/BDR election can be described in the following steps. Remember that the DR/BDR election process occurs on every multi-access network (not router). A router may be the DR on one interface but not another.

The following description assumes that a router *R* has just been turned up on a multi-access network:

1. On becoming active on a multi-access network, the OSPF process on router *R* begins receiving hellos from neighbors on its interface to the multi-access network. If the hellos indicate that there already are a DR and a BDR, the DR/BDR election process is terminated (even if *R*'s OSPF priority is higher than the current DR/BDR priority).
2. If hellos from neighbors indicate that there is no active BDR on the network, the router with the highest priority is elected the BDR. If the highest priority is shared by more than one router, the router with the highest router ID wins.
3. If there is no active DR on the network, the BDR is promoted to DR.

The following can be stated as corollaries of the above rules:

1. If a DR and BDR have already been elected, bringing up a new router (even with a higher priority) will not alter the identities of the DR/BDR.
2. If there is only one DR-eligible router on a multi-access network, that router will become the DR.
3. If there are only two DR-elegible routers on a multi-access network, one will be the DR and the other, the BDR.

A router with a higher priority takes precedence during DR election. A priority value of 0 indicates that the router is ineligible for DR election. The default priority value is 1. Routers with low memory and CPU resources should be made ineligible for DR election.

The router interface priority may be modified with the following command in interface configuration mode:

```
ip ospf priority number
```

where *number* is between 0 and 255.

The state of an OSPF interface (including the result of the DR/BDR election process) can be seen as output of the *show ip ospf interface* command:

```
NewYork#sh ip ospf interface
Ethernet0 is up, line protocol is up
  Internet Address 172.16.1.1/24, Area 0
  Process ID 10, Router ID 172.16.251.1, Network Type BROADCAST, Cost: 10
9   Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 172.16.251.1, Interface address 172.16.1.1
10  No backup designated router on this network
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:02
11  Neighbor Count is 0, Adjacent neighbor count is 0
  Suppress hello for 0 neighbor(s)
...
```

Note that *NewYork* is the DR on *Ethernet0*. Since there is no other router on this network, there is no BDR (line 10) and the router has not established any adjacencies (line 11).

Interface State

The state of an interface can have one of the following values:

Down
: The interface state is down as indicated by lower-level protocols, and no OSPF traffic has been sent or received yet.

Loopback
: The interface is looped and will be advertised in LSAs as a host route.

Point-to-point
: The interface is up and is recognized as a serial interface or a virtual link. After entering the point-to-point state, the neighbors will attempt to establish adjacency.

Waiting
: This state applies only to broadcast/NBMA networks on which the router is attempting to identify the DR/BDR.

DR
: This router is the DR on the attached network.

Backup
: This router is the BDR on the attached network.

DRother
: This router is neither the DR nor the BDR on the attached network. The router will form adjacencies with the DR and BDR (if they exist).

As an example, the state of *NewYork*'s interface to *Chicago* is point-to-point (line 12) and *NewYork* and *Chicago* have established adjacency (lines 13 and 14):

```
    NewYork#sh ip ospf interface
    ...
    Serial0 is up, line protocol is up
      Internet Address 172.16.250.1/24, Area 0
      Process ID 10, Router ID 172.16.251.1, Network Type POINT_TO_POINT, Cost: 64
12    Transmit Delay is 1 sec, State POINT_TO_POINT,
      Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
        Hello due in 00:00:01
13    Neighbor Count is 1, Adjacent neighbor count is 1
14      Adjacent with neighbor 69.1.1.1
      Suppress hello for 0 neighbor(s)
```

Neighbor Relationship

Not all neighbors establish adjacency. Neighbors may stay at "2-way" or enter into a "Full" relationship, depending on the type of network, as follows:

Point-to-point networks
: Routers on point-to-point networks always establish adjacency.

Broadcast networks
: Routers on broadcast networks establish adjacency only with the DR and the BDR, maintaining a 2-way relationship with the other routers on the network.

Non-broadcast multi-access (NBMA) networks
: Routers on NBMA networks establish adjacency only with the DR and the BDR.

Virtual links
: Routers on virtual links always establish adjacency.

Database Exchange

The *database description (DD) packet* is used to describe the contents of the LS database to a peer OSPF router. Only LSA headers are sent in DD packets; the peer router responds by sending its own LSA headers in DD packets.

The LSA header (Figure 6-8) uniquely identifies a piece of the OSPF network topology. The key fields in the LSA header are the *advertising router*, *LS type*, and *link state ID*. The advertising router is the router ID of the originator of the LSA. The LS type identifies the type of the LSA that follows. The link state ID depends on the LS type, as shown in Table 6-3.

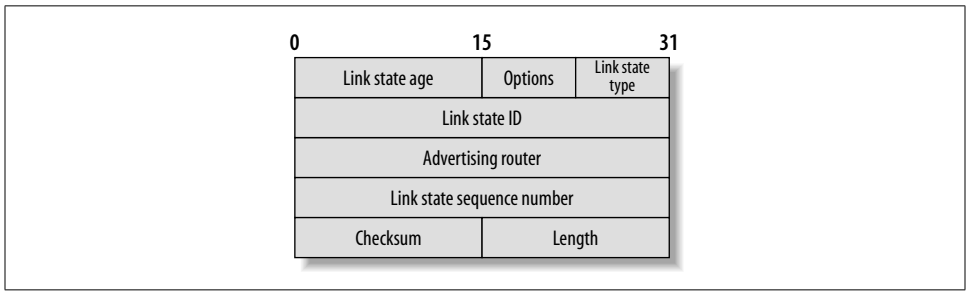

Figure 6-8. Format of an LSA header

Table 6-3. LS type and link state ID

LS type	Link state ID
1 (router LSA)	Router ID of the originator of the LSA
2 (network LSA)	IP address of the DR's interface to the multi-access network
3 (summary LSA)	IP address of the destination network
4 (ASBR summary LSA)	Router ID of the ASBR
5 (external LSA)	IP address of the destination network

Several copies of an LSA may be circulating in a network. The *LS sequence number*, a signed 32-bit integer, helps identify the most recent LSA. The first instance of an LSA record contains a sequence number field of 0x80000001. Each new instance of the LSA contains a sequence number that is one higher. The maximum sequence

number is 0x7fffffff, after which the sequence numbers are recycled. The sequence number helps identify the most recent instance of an LSA.

Upon receiving LSA headers in DD packets, both routers check to see if this piece of the OSPF topology is already contained in their LS databases. In this process, the advertising router, LS type, and link state ID fields (from the LSA header) are compared against the router's LS database. If no matching records are found or if a matching record is found with a lower sequence number, the complete LSA is requested using the *link state request packet*. The LS request packet contains the LSA header to help identify the record being sought.

In response to a link state request, a router issues a link state update containing the LSA. The LSA completely describes the piece of OSPF topology in question. LS updates are issued (a) in response to an LS request, as just described; (b) because of a change in the state of the link; and (c) every 30 minutes, with a new sequence number and the age field set to 0.

All LS updates are acknowledged in *link state acknowledgment packets* (see Figure 6-9).

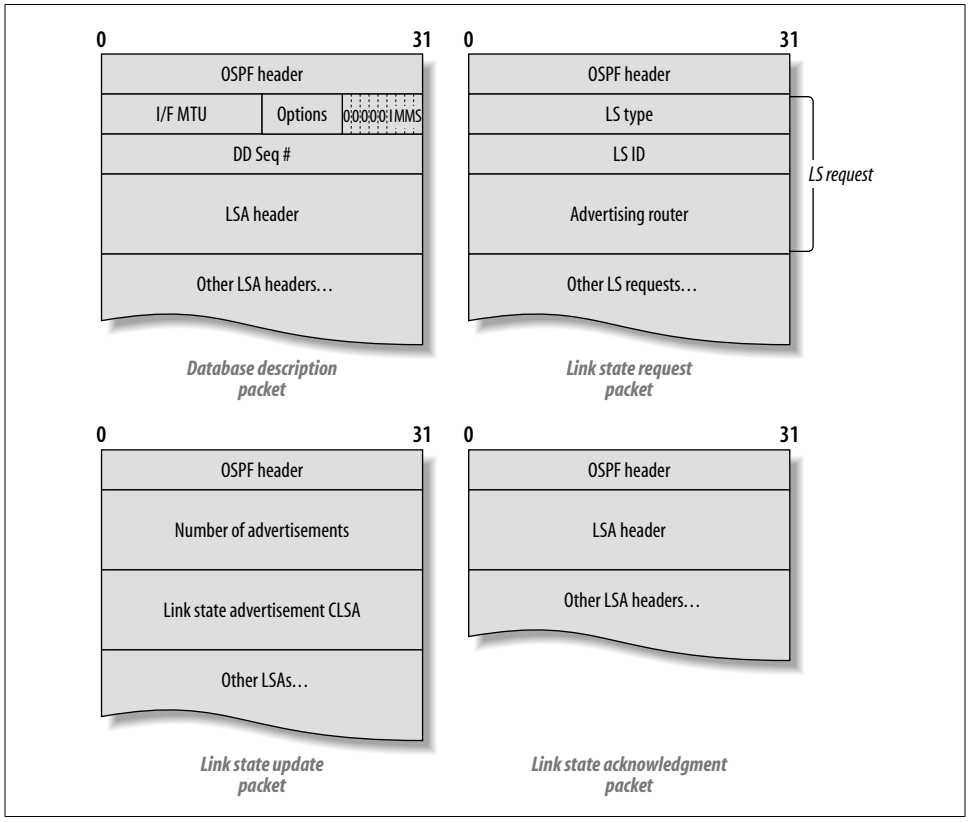

Figure 6-9. Database description, link state request, link state update, and link state acknowledgment packets

There are six types of LSA records, each representing a different piece of the network topology. We'll use TraderMary's network with a French extension (Figure 6-10) to take a closer look at the various LSA types.

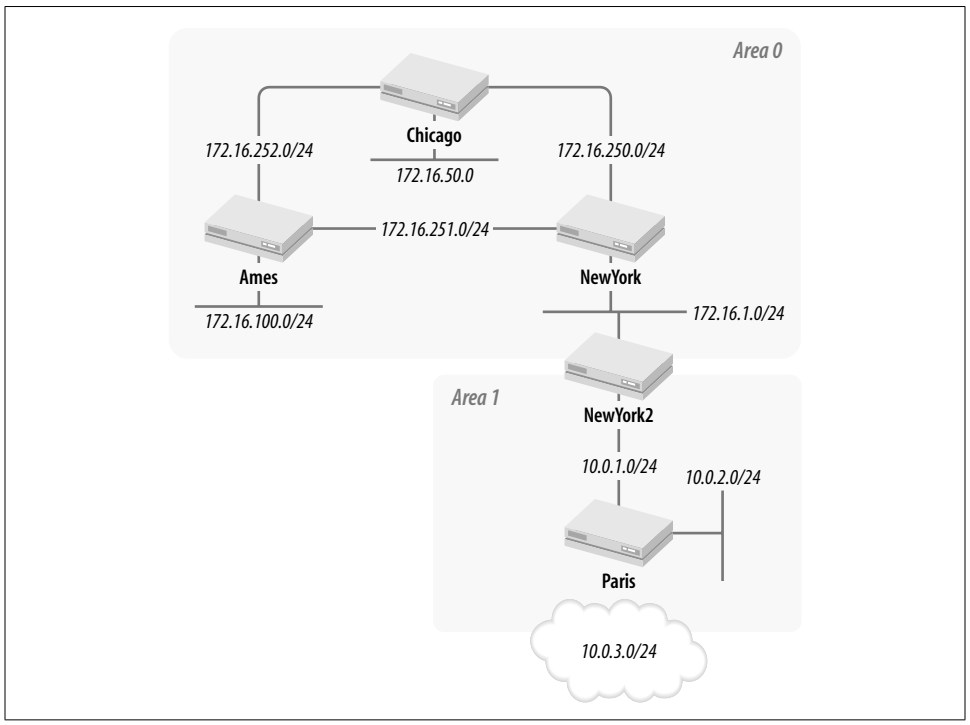

Figure 6-10. TraderMary's network with a French extension

TraderMary's network in New York is configured as follows. *NewYork2* is an ABR with a serial link in area 1 to router *Paris* (line 15).

```
hostname NewYork2
!
interface Loopback0
 ip address 192.168.1.4 255.255.255.0
!
interface Ethernet0
 ip address 172.16.1.2 255.255.255.0
 ip pim sparse-mode
!
interface Serial1
 description Paris link
 ip address 10.0.1.2 255.255.255.0
 bandwidth 56
!
router ospf 10
 network 172.16.0.0 0.0.255.255 area 0
15  network 10.0.0.0 0.255.255.255 area 1
```

Paris is an ASBR redistributing RIP routes from a legacy network into OSPF (line 16):

```
hostname Paris
!
interface Loopback0
 ip address 192.168.1.5 255.255.255.255
!
interface Ethernet0
 ip address 10.0.2.1 255.255.255.0
!
interface Serial1
 description link to NewYork2
 ip address 10.0.1.1 255.255.255.0
!
router ospf 10
16  redistribute rip metric 100 subnets
 network 10.0.0.0 0.255.255.255 area 1
!
router rip
 network 10.0.0.0
```

The 10.0.0.0 subnets—10.0.1.0, 10.0.2.0, and 10.0.3.0—are known to both the OSPF and RIP processes on router *Paris*. Let's see how *NewYork* learns these subnets. Here is *NewYork*'s routing table:

```
NewYork#sh ip route
...
     10.0.0.0/24 is subnetted, 3 subnets
17 O IA    10.0.2.0 [110/1805] via 172.16.1.2, 00:07:45, Ethernet0
18 O E2    10.0.3.0 [110/100] via 172.16.1.2, 00:07:46, Ethernet0
19 O IA    10.0.1.0 [110/1795] via 172.16.1.2, 00:07:46, Ethernet0
     192.168.1.0/32 is subnetted, 1 subnets
C       192.168.1.1 is directly connected, Loopback0
     172.16.0.0/24 is subnetted, 6 subnets
O       172.16.252.0 [110/128] via 172.16.250.2, 00:07:46, Serial0
C       172.16.250.0 is directly connected, Serial0
C       172.16.251.0 is directly connected, Serial1
O       172.16.50.0 [110/74] via 172.16.250.2, 00:07:46, Serial0
C       172.16.1.0 is directly connected, Ethernet0
O       172.16.100.0 [110/192] via 172.16.250.2, 00:07:46, Serial0
```

Note that the routing table shows that *NewYork* learns 10.0.3.0 as an external route whereas 10.0.1.0 and 10.0.2.0 are learned as inter-area routes (lines 17–19)—this is because inter-area routes are preferred over external routes. The OSPF order of route preference, from most preferred to least preferred, is as follows: intra-area, inter-area, type 1 external, type 2 external.

Router LSA (type 1)

A router LSA describes the advertising router's directly connected links. Routers *Chicago*, *Ames*, *NewYork*, and *NewYork2* advertise router LSAs that are flooded throughout area 0. *NewYork*'s LS database holds router LSAs from all these routers, but for the sake of brevity I'll show only the contents of the LSA from *NewYork2*.

The number of links (as in line 20 in the upcoming code block) described in the LSA is 1. Although *NewYork2* has two directly connected links—an Ethernet segment and a serial link—only the Ethernet segment is described in the LSA to *NewYork*. This is because the serial link is in area 1 and router LSAs do not cross OSPF area boundaries.

The link described is a *transit network* (line 21), implying that there are multiple routers on the link. Other link types are point-to-point (for serial links), stub network (for a network with only one router), and virtual link (for OSPF virtual links).

The value of the link ID field depends on the type of link being described, as shown in Table 6-4.

Table 6-4. Link type and link ID

Link type	Link ID
Point-to-point	Neighbor's router ID
Transit network	DR's IP address on network
Stub network	IP network number or subnet number
Virtual link	Neighbor's router ID

In our example, the DR is *NewYork*, so the link ID (in line 22) contains *NewYork*'s IP address.

The contents of the link data field also depend on the link type, as shown in Table 6-5.

Table 6-5. Link type and link data

Link type	Link data
Point-to-point	IP address of network interface
Transit network	IP address of network interface
Stub network	IP network number or subnet number
Virtual link	MIB II ifIndex for the router's interface

In our example, the link data field (in line 23) specifies the IP address of *NewYork2*:

```
NewYork#sh ip ospf database router

       OSPF Router with ID (192.168.1.1) (Process ID 10)

Routing Bit Set on this LSA

  LS age: 209

  Options: (No TOS-capability, DC)
  LS Type: Router Links
  Link State ID: 192.168.1.4

  Advertising Router: 192.168.1.4
```

```
      LS Seq Number: 800000FF
      Checksum: 0x2BA1
      Length: 36
      Area Border Router
      AS Boundary Router
20     Number of Links: 1

21      Link connected to: a Transit Network
22       (Link ID) Designated Router address: 172.16.1.1
23       (Link Data) Router Interface address: 172.16.1.2
          Number of TOS metrics: 0
           TOS 0 Metrics: 10
```

Network LSA (type 2)

A network LSA describes broadcast/NBMA networks. The network LSA is originated by the DR and describes all attached routers.

The LSA in the following example is self-originated, as seen in the advertising router field (line 24), which shows *NewYork*'s own router ID. The network LSA describes the mask on the multi-access network (line 25) and the IP addresses of the routers on the multi-access network (lines 26 and 27).

```
NewYork#sh ip ospf database network

       OSPF Router with ID (192.168.1.1) (Process ID 10)


           Net Link States (Area 0)

      Routing Bit Set on this LSA
      LS age: 1728
      Options: (No TOS-capability, DC)
      LS Type: Network Links
      Link State ID: 172.16.1.1 (address of Designated Router)
24    Advertising Router: 192.168.1.1
      LS Seq Number: 800000F4
      Checksum: 0x172B
      Length: 32
25    Network Mask: /24
26     Attached Router: 192.168.1.1
27     Attached Router: 192.168.1.4
```

Summary LSA (type 3)

A summary LSA is advertised by an ABR and describes inter-area routes.

The summary LSAs in the following example are originated by *NewYork2* (`192.168.1.4`) and describe routes to `10.0.1.0` and `10.0.2.0`, respectively. The link state ID describes the summary network number (lines 28 and 31). Note that each LSA describes just one summary network number.

```
NewYork#sh ip ospf database summary

       OSPF Router with ID (192.168.1.1) (Process ID 10)


            Summary Net Link States (Area 0)

  Routing Bit Set on this LSA
  LS age: 214
  Options: (No TOS-capability, DC)
  LS Type: Summary Links(Network)
28 Link State ID: 10.0.1.0 (summary Network Number)
29 Advertising Router: 192.168.1.4
  LS Seq Number: 80000062
  Checksum: 0x85A
  Length: 28
30 Network Mask: /24
    TOS: 0     Metric: 1785

  Routing Bit Set on this LSA
  LS age: 214
  Options: (No TOS-capability, DC)
  LS Type: Summary Links(Network)
31 Link State ID: 10.0.2.0 (summary Network Number)
32 Advertising Router: 192.168.1.4
  LS Seq Number: 80000061
  Checksum: 0x62F5
  Length: 28
33 Network Mask: /24
    TOS: 0     Metric: 1795
```

ASBR summary LSA (type 4)

An ASBR summary LSA describes the route to the ASBR. The mask associated with a type 4 LSA is 32 bits long because the route advertised is to a host—the host being the ASBR. ASBR summary LSAs are originated by ABRs.

The link state ID (line 34) in this example describes the router ID of *Paris*, which is the ASBR redistributing RIP into OSPF. The advertising router is the ABR—*NewYork2* (line 35).

```
NewYork#sh ip ospf database asbr-summary

       OSPF Router with ID (192.168.1.1) (Process ID 10)


           Summary ASB Link States (Area 0)

  Routing Bit Set on this LSA
  LS age: 115
  Options: (No TOS-capability, DC)
  LS Type: Summary Links(AS Boundary Router)
```

```
34    Link State ID: 192.168.1.5 (AS Boundary Router address)
35    Advertising Router: 192.168.1.4
      LS Seq Number: 80000061
      Checksum: 0x9A63
      Length: 28
      Network Mask: /0
        TOS: 0     Metric: 1785
```

External LSA (type 5)

External LSAs originate at ASBRs and describe routes external to the OSPF process. External LSAs are flooded throughout the OSPF network, with the exception of stub areas.

Network 10.0.1.0 is learned via RIP from *NewYork2*, which floods an external LSA with a link state ID of 10.0.1.0. Interestingly, 10.0.1.0 is also known as an inter-area route (see the section "Summary LSA (type 3)"). Router *NewYork* prefers the IA route (see line 19) but will keep the external LSA in its topological database. The advertising router (line 37) is *Paris*, the ASBR, which redistributes RIP into OSPF. The forwarding address (in line 39) is 0.0.0.0, indicating that the destination for 10.0.1.0 is the ASBR. The LSA (in line 40) specifies an external route tag of 0, which indicates a type 1 external route; a value of 1 would indicate a type 2 external route.

```
NewYork#sh ip ospf database external

       OSPF Router with ID (192.168.1.1) (Process ID 10)


            Type-5 AS External Link States

      LS age: 875
      Options: (No TOS-capability, No DC)
      LS Type: AS External Link
36    Link State ID: 10.0.1.0 (External Network Number )
37    Advertising Router: 192.168.1.5
      LS Seq Number: 80000060
      Checksum: 0x6F27
      Length: 36
38    Network Mask: /24
       Metric Type: 2 (Larger than any link state path)
       TOS: 0
       Metric: 100
39     Forward Address: 0.0.0.0
40     External Route Tag: 0
...
```

Note that *NewYork*'s external database contains two other LSAs—with link state IDs of 10.0.2.0 and 10.0.3.0—which were not shown here.

NSSA external LSA (type 7)

NSSA external LSAs describe routes external to the OSPF process. However, unlike type 5 external LSAs, NSSA external LSAs are flooded only within the NSSA.

There are no type 7 LSAs in this network. In fact, there aren't even any NSSAs in this network:

```
NewYork#sh ip  ospf database nssa-external

       OSPF Router with ID (192.168.1.1) (Process ID 10)
```

The format of the NSSA external LSA is identical to that of the AS external LSA, except for the forwarding address field. The forwarding address field in an NSSA external LSA always indicates the address to which traffic should be forwarded.

Flooding of LSAs

LSAs are generated every 30 minutes, or sooner if there is a change in the state of a link. LSAs are exchanged between routers that have established *adjacency*, as was described earlier.

The rules for the flooding of LSAs are governed by the hierarchical structure of OSPF, as given in Table 6-6.

Table 6-6. Rules for the flooding of LSAs

LSA type	Originating router	Area in which flooded
Router LSA (type 1)	Every router	Router's local area.
Network LSA (type 2)	DR	Router's local area.
Summary LSA (type 3)	ABR	Nonlocal area.
ASBR summary LSA (type 4)	ASBR	All areas except stub area, totally stubby area, or NSSA.
External LSA (type 5)	ASBR	All areas except stub area, totally stubby area, or NSSA.
NSSA external LSA (type 7)	ASBR	Router's local area. NSSA external LSA may be forwarded by ABR as a type 5 LSA.

Route Summarization

RIP-1 and IGRP automatically summarize subnets into a major network number when crossing a network-number boundary. OSPF does not automatically summarize routes. Route summarization in OSPF must be manually configured on an ABR or an ASBR. Further, OSPF allows route summarization on any bit boundary (unlike RIP and IGRP, which summarize only classful network numbers).

Summarizing routes keeps the routing tables smaller and easier to troubleshoot. However, route summarization in OSPF is not just a nice thing to do—it is necessary

to reduce the size of the OSPF topology database, especially in a large network. A large topology database requires a large amount of router memory, which slows down all processes, including SPF calculations.

To allow summarization at ABRs and ASBRs, IP addresses must be carefully assigned. First, allocate enough addresses to each area to allow for expansion. Then set a bit boundary on which to summarize routes. This is easier said than done. Most network engineers inherit a network with a haphazard mess of addresses and changing requirements.

Summarizing at the ABR (Inter-Area Summarization)

Consider TraderMary's network in Figure 6-10. Network `10.0.0.0` exists in area 1, and network `172.16.0.0` exists in area 0. Let's see how we can summarize on these area boundaries.

The command to summarize at an ABR is:

```
area area-id range address mask
```

where *area-id* is the area whose routes are to be summarized, *address* is a network number, and *mask* specifies the number of bits in *address* to summarize.

The OSPF configuration on *NewYork2* can now be modified to summarize `172.16.0.0` routes into area 1 (line 41) and `10.0.0.0` routes into area 0 (line 42).

```
hostname NewYork2
...
router ospf 10
 redistribute static metric 10
 network 172.16.0.0 0.0.255.255 area 0
 network 10.0.0.0 0.255.255.255 area 1
41  area 0 range 172.16.0.0 255.255.0.0
42  area 1 range 10.0.0.0 255.0.0.0
```

The routing table in *Paris* is now as follows. Note that *Paris* has only one summary route for `172.16.0.0/16` (line 43).

```
Paris#show ip route
...
10.0.0.0/24 is subnetted, 2 subnets
C       10.0.2.0 is directly connected, Ethernet0
C       10.0.1.0 is directly connected, Serial1
     192.168.1.0/32 is subnetted, 1 subnets
C       192.168.1.5 is directly connected, Loopback0
43  O IA 172.16.0.0/16 [110/74] via 10.0.1.2, 1d23h, Serial1
```

The routing table for *NewYork* is now as follows. Note that *NewYork* has only one summarized route for `10.0.0.0/8` (line 44).

```
NewYork#sh ip route
...
```

```
O IA 10.0.0.0/8 [110/1795] via 172.16.1.2, 1d23h, Ethernet0
     192.168.1.0/32 is subnetted, 1 subnets
C       192.168.1.1 is directly connected, Loopback0
     172.16.0.0/24 is subnetted, 6 subnets
O       172.16.252.0 [110/128] via 172.16.250.2, 1d23h, Serial0
C       172.16.250.0 is directly connected, Serial0
C       172.16.251.0 is directly connected, Serial1
O       172.16.50.0 [110/74] via 172.16.250.2, 1d23h, Serial0
C       172.16.1.0 is directly connected, Ethernet0
O       172.16.100.0 [110/192] via 172.16.250.2, 1d23h, Serial0
```

When an EIGRP router summarizes, it automatically builds a route to *null0* for the summarized route. (This is explained in detail in the section "Route Summarization" in Chapter 4). The router to *null0* prevents packets that do not match a specific entry in the routing table from following a default route. (The route to *null0* causes the packet to be dropped). However, as you saw earlier, OSPF does not build a null route. You may want to manually add a static route to *null0* on the ABR.

Summarizing at the ASBR (or External Route Summarization)

In the configuration in Figure 6-10, *Paris* is the ASBR redistributing RIP into OSPF. Note from the figure that the RIP network contains routes in the network `10.3.0.0/24` (the RIP subnets may be `10.3.1.0/24`, `10.3.2.0/24`, `10.3.3.0/24`, ... `10.3.255.0/24`). It is desirable to summarize `10.3.0.0/16` into the OSPF network rather than carrying the individual subnets.

The routes being redistributed into OSPF can be summarized at the ASBR (which is *Paris* in the previous example) using the following command:

```
summary-address address mask
```

where *address* defines a summary IP address and *mask* describes the range of addresses.

Router *Paris* may thus be configured as follows to summarize `10.3.0.0/16` into the OSPF network:

```
hostname Paris
!
interface Loopback0
 ip address 192.168.1.5 255.255.255.255
!
interface Ethernet0
 ip address 10.0.2.1 255.255.255.0
!
interface Serial1
 ip address 10.0.1.1 255.255.255.0
!
router ospf 10
 summary-address 10.3.0.0 255.255.252.0
```

```
 redistribute rip metric 100 subnets
 network 10.0.0.0 0.255.255.255 area 1
!
router rip
 network 10.0.0.0
```

The LS database will now contain a single external LSA with a link state ID of `10.3.0.0` advertised by *Paris*.

Default Routes

Earlier chapters showed how a default route could be used for branch office connectivity. A default route can also be used when connecting to the Internet to represent *all* the routes in the Internet. Let's say that TraderMary established a connection from *NewYork2*, *Serial2* (line 45) to an Internet service provider (ISP). A static default route is also installed on *NewYork2* (line 47), pointing to the ISP.

NewYork2 is configured as in line 46 to source a default route. The keyword *always* implies that the default route must be originated whether or not the default route is up. *metric-value* is the metric to associate with the default route (the default for this field is 10). Note that this redistribution of a default route into OSPF makes *NewYork2* an ASBR. The keyword *metric-type* can be set to 1 or 2 to specify whether the default route is external type 1 or 2 (the default is 2).

```
   hostname NewYork2
   !
45 interface Serial2
   description Connection to the ISP
   ip address 146.146.1.1 255.255.255.0
   !
   router ospf 10
   network 172.16.0.0 0.0.255.255 area 0
46 default-information originate always metric-value 20 metric-type 1
   !
47 ip route 0.0.0.0 0.0.0.0 interface serial2
```

Since the keyword *always* was specified, the default route will not disappear from the OSPF routing table if *Serial2* (the link to the ISP) is down. If TraderMary has two (or more) routers connecting to ISPs and each router announced a default route into OSPF, do not use the *always* keyword—if one ISP connection is lost, traffic will find its way to the other ISP connection.

To ensure that the default route is always announced (even if *Serial2* goes down) choose the *always* option.

A default route of type 1 includes the internal cost of reaching the ASBR. If TraderMary has multiple Internet connections, announcing a default route from each with a metric type of 1 would have the advantage that any router in the network would find the closest ASBR.

Virtual Links

TraderMary is planning to establish a new office in Paris with an area ID of 2. The first router in area 2 will be called *Paris2*. A direct circuit needs to be established from *NewYork2* (the ABR) to *Paris2*, since all OSPF areas must connect directly to the backbone (area 0). This international circuit has a long installation time. And, since a physical path is already available to area 2 via area 1, you may ask if OSPF provides some mechanism to activate area 2 before the *NewYork2* → *Paris2* circuit can be installed. The answer is yes. OSPF defines virtual links (VLs) which can extend the backbone area. Area 2 will directly attach to the backbone via the VL. A VL may be viewed as a point-to-point link belonging to area 0. The endpoints of a VL must be ABRs.

In our example in Figure 6-11, a virtual link may be defined from *NewYork2* to *Paris2* through area 1.

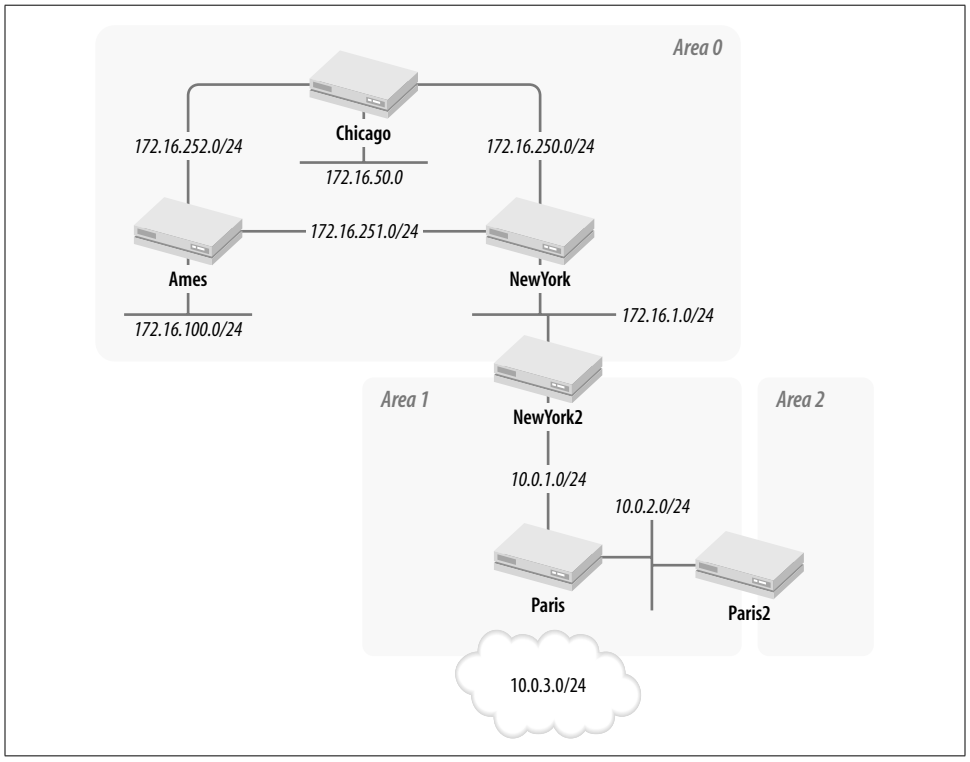

Figure 6-11. Virtual link to area 2

The syntax for configuring a virtual link is as follows:

```
area area-id virtual-link router-id [hello-interval seconds] [retransmit-interval
seconds] [transmit-delay seconds] [dead-interval seconds] [[authentication-key key] |
[message-digest-key keyid md5 key]]
```

where *area-id* specifies the transit area and *router-id* specifies the ABR with which the neighbor relationship is to be established. The four timers refer to the time between hello packets (default is 10 s), the time between LSA retransmissions (default is 5 s), the time by which LSAs are aged when they transmit this interface (default is 1 s), and the router dead-interval (default is four times the hello-interval). The parameter *key* is a string of characters up to 8 bytes long, *keyid* is in the range 1–255, and *key* is an alphanumeric string up to 16 characters in length.

Remember that a virtual link can be created only between ABRs and can traverse only one area. *Paris2* is an ABR because it has connectivity to areas 1 and 2. *NewYork2* is an ABR with connectivity to areas 0 and 1. Thus, a virtual link may be configured between *Paris2* and *NewYork2* traversing area 1:

```
hostname Paris2
!
interface Loopback1
 ip address 192.168.1.6 255.255.255.255
!
interface Loopback2
 ip address 192.168.2.1 255.255.255.0
!
interface Ethernet0
 ip address 10.0.2.2 255.255.255.0
!
router ospf 10
 network 10.0.0.0 0.255.255.255 area 1
 network 192.168.2.0 0.0.0.255 area 2
 area 1 virtual-link 192.168.1.4
```

```
hostname NewYork2
!
interface Loopback0
 ip address 192.168.1.4 255.255.255.255
!
interface Ethernet0
 ip address 172.16.1.2 255.255.255.0
!
interface Serial1
 ip address 10.0.1.2 255.255.255.0
 bandwidth 56
!
router ospf 10
 redistribute static metric 10
 network 172.16.0.0 0.0.255.255 area 0
 network 10.0.0.0 0.255.255.255 area 1
 area 1 virtual-link 192.168.1.6
```

The status of the virtual link can be verified as follows:

```
Paris2#sh ip ospf virtual-link
Virtual Link to router 192.168.1.4 is up
```

```
  Transit area 1, via interface Ethernet0, Cost of using 74
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 0:00:00
    Adjacency State FULL


NewYork2#show ip ospf virtual-link
Virtual Link OSPF_VL0 to router 192.168.1.6 is up
  Run as demand circuit
  DoNotAge LSA not allowed (Number of DCbitless LSA is 8).
  Transit area 1, via interface Serial1, Cost of using 1795
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:05
    Adjacency State FULL
```

VLs cannot traverse stub areas (or totally stubby areas or NSSAs). This is because VLs belong to area 0, and in order for area 0 to route correctly it must have the complete topology database. Stub areas do not contain the complete topology database.

VLs find one other use in OSPF—they may be used to repair the network in the event that an area loses its link to the backbone. For example, in Figure 6-4, the loss of the link *R1* → *R4* will isolate area 2 from the rest of the network. Until the *R1* → *R4* link is repaired, a VL may be defined between *R4* and *R5* to join area 2 to the backbone.

Demand Circuits

The cost of a demand circuit, such as an ISDN link or a dial-up line, is dependent on its usage. It is desirable to use a demand circuit only for user traffic and not for overhead such as OSPF hellos or periodic LSAs. RFC 1793 describes modifications to OSPF that allow the support of demand circuits. This is an optional capability in OSPF; a router will set the DC bit in the options field if it supports the capability. Routers that support the capability will also set the high bit of the LS age field to 1 to indicate that the LSA should not be aged. This bit is also referred to as the do-not-age bit. OSPF demand circuits suppress periodic hellos and LSAs, but a topology change will still activate the demand circuit since LSA updates are required to keep the LS database accurate. Since any large network is likely to experience frequent topology changes, it may be prudent to define demand circuits in stub areas. Stub areas have a limited topology database and hence are shielded from frequent topology changes.

If a demand circuit is created in a stub area, all routers in the stub area must support the DC option—routers that do not support demand circuits will misinterpret the age field (as the high bit is set). An LSA with the DC bit set to 1 is flooded into an area only if all LSAs in the database have their DC bits set to 1.

To configure an interface as a demand circuit, enter the following command in interface configuration mode on one end of the demand circuit:

```
ip ospf demand-circuit
```

LSA updates will bring up the demand circuit only if there is a change in topology.

Stub, Totally Stubby, and Not So Stubby Areas

External LSAs are flooded through the OSPF backbone as well as through all regular areas. Let's test this using TraderMary's network of Figure 6-10. A static route for `192.168.3.0` is defined (pointing to *null0*) on *Chicago* and redistributed into OSPF. Router *Chicago* then advertises an external LSA with a link state ID of `192.168.3.0`:

```
hostname Chicago
!
router ospf 10
 redistribute static metric 100 metric-type 1 subnets
 network 172.16.0.0 0.0.255.255 area 0
!
ip route 192.168.3.0 255.255.255.0 Null0
```

The LSA is flooded to all routers in the network. Let's check *Paris* as an instance:

```
Paris#sh ip ospf database external

       OSPF Router with ID (192.168.1.5) (Process ID 10)


           AS External Link States

  Routing Bit Set on this LSA
  LS age: 158
  Options: (No TOS-capability)
  LS Type: AS External Link
  Link State ID: 192.168.3.0 (External Network Number )
  Advertising Router: 192.168.1.3
  LS Seq Number: 80000001
  Checksum: 0x8F67
  Length: 36
  Network Mask: /24
    Metric Type: 1 (Comparable directly to link state metric)
    TOS: 0
    Metric: 100
    Forward Address: 0.0.0.0
    External Route Tag: 0
```

The route to `192.168.3.0` also appears in the routing table:

```
Paris#sh ip route
...
```

```
Gateway of last resort is not set
...
O E1 192.168.3.0/24 [110/302] via 10.0.1.2, 00:02:08, Serial1
...
```

Flooding external LSAs throughout an OSPF network may be a waste of resources. Stub areas block the flooding of external LSAs, as we will see in the next section.

Stub Areas

Referring to Figure 6-1, the router in area 1 that connects to the RIP network floods external LSAs into the network. It appears that nothing is gained by importing external LSAs into areas 2 and 3, which can point all external routes to their ABRs using default routes. Representing every external LSA in areas 2 and 3 would be a waste of resources. With this in mind, OSPF defines *stub areas*. When an area is defined as a stub area, all external LSAs are blocked at the ABRs, and, in place, the ABRs source a single default route into the stub area.

All routers in a stub area must be configured as stub routers. Stub routers form adjacencies only with other stub routers and do not propagate external LSAs. (How does a router know if its neighbor is a stub router? The E bit in the hello packet is turned to zero if the router is a stub router).

Area 1 in TraderMary's network can be made stubby via the following configuration changes:

```
hostname NewYork2
...
router ospf 10
network 172.16.0.0 0.0.255.255 area 0
network 10.0.0.0 0.255.255.255 area 1
area 1 stub


hostname Paris
...
router ospf 10
 redistribute rip
 network 10.0.0.0 0.255.255.255 area 1
 area 1 stub
```

The routing table for *Paris* now shows a default route pointing to the ABR (*NewYork2*) but does not show the external route to `192.168.3.0` (sourced by *Chicago*):

```
Paris#sh ip route
...
Gateway of last resort is 10.0.1.2 to network 0.0.0.0
...
O*IA 0.0.0.0/0 [110/65] via 10.0.1.2, 00:00:35, Serial1
O IA 172.16.0.0/16 [110/74] via 10.0.1.2, 1d23h, Serial1
...
```

After making this change, however, we will find that the network has lost connectivity to 10.0.3.0, which represents the RIP external network connecting to router *Paris*. The reason for this is rather obvious: stub areas do not propagate external LSAs. In other words, an ASBR cannot belong to a stub area.

The other major restriction with stub areas is that they cannot support virtual links, because they don't have the complete routing table. An area that needs to support a VL cannot be a stub area.

Any area that does not contain an ASBR (i.e., does not support a connection to an external network) and is not a candidate for supporting a virtual link should be made a stub area.

There is one major disadvantage to configuring an area as a stub area. When multiple ABRs source a default route, the routers in the stub area may fail to recognize the shortest path to the destination network. This may help determine whether you choose to implement an area as a regular area or as a stub area.

Totally Stubby Areas

Totally stubby areas carry the concept of stub areas further by blocking all summary LSAs in addition to external LSAs.

In the configuration in the previous section, where *Paris* is configured as a stub area, the LS database for *Paris* will not show external LSAs but will still show all summary LSAs, so *Paris*'s routing table still shows the summarized inter-area route to 172.16.0.0/16. If *NewYork2* did not summarize the 172.16.0.0 subnets, *Paris* would show all six 172.16.0.0 subnets: 172.16.1.0/24, 172.16.50.0/24, 172.16.100.0/24, 172.16.250.0/24, 172.16.251.0/24, and 172.16.252.0/24. Totally stubby areas, unlike stub areas, replace all inter-area routes (in addition to external routes) with a default route.

Area 1 can be configured as a totally stubby area by modifying the configuration of *NewYork2* as follows. No change is required to router *Paris*.

```
hostname NewYork2
!
router ospf 10
 redistribute static metric 10
 network 172.16.0.0 0.0.255.255 area 0
 network 10.0.0.0 0.255.255.255 area 1
 area 1 stub no-summary
```

Paris's routing table now does not contain any IA routes (other than the default sourced by *NewYork2*):

```
Paris#sh ip route
...
Gateway of last resort is 10.0.1.2 to network 0.0.0.0

     10.0.0.0/24 is subnetted, 2 subnets
```

```
C       10.0.2.0 is directly connected, Ethernet0
C       10.0.1.0 is directly connected, Serial1
     192.168.1.0/32 is subnetted, 1 subnets
C       192.168.1.5 is directly connected, Loopback0
O*IA 0.0.0.0/0 [110/65] via 10.0.1.2, 00:00:23, Serial1
```

Totally stubby areas have the same restrictions as stub areas—no ASBRs (no external LSAs) and no virtual links. Also, like stub areas, totally stubby areas see all ABRs as equidistant to all destinations that match the default route. When multiple ABRs source a default route, the routers in the totally stubby area may not recognize the shortest path to the destination network.

NSSAs

What if a stub area needs to learn routes from another routing protocol? For example, *Paris*—in area 1—may need to learn some RIP routes from a legacy network. NSSAs—as specified in RFC 1587—allow external routes to be imported into an area without losing the character of a stub area (i.e., without importing any external routes from the backbone area).

NSSAs import external routes through an ASBR in type 7 LSAs. Type 7 LSAs are flooded within the NSSA. Type 7 LSAs may optionally be flooded into the entire OSPF domain as a type 5 LSAs by the ABR(s) or be blocked at the ABR(s). As with any stub area, NSSAs do not import type 5 LSAs from the ABR.

The option (of whether or not to translate a type 7 LSA into a type 5 LSA at the NSSA ABR) is indicated in the P bit (in the options field) of the type 7 LSA. If this bit is set to 1, the LSA is translated by the ABR into a type 5 LSA to be flooded throughout the OSPF domain. If this bit is set to 0, the LSA is not advertised outside the NSSA area.

All routers in the NSSA must be configured with the *nssa* keyword (line 48):

```
hostname NewYork2
!
router ospf 10
 redistribute static metric 10
 network 172.16.0.0 0.0.255.255 area 0
 network 10.0.0.0 0.255.255.255 area 1
48  area 1 nssa
```

There are three optional keywords for NSSA configuration:

```
area 1 nssa ?
49    default-information-originate
50    no-redistribution
51    no-summary
```

When configured on the NSSA ABR, the *default-information-originate* keyword (line 49) causes the ABR to source a default route into the NSSA.

The *no-redistribution* keyword (line 50) is useful on NSSA ABRs that are also ASBRs. The *no-redistribution* keyword stops the redistribution of external LSAs (from the other AS) into the NSSA.

The *no-summary* keyword (line 51) gives you another oxymoron—it makes the NSSA a totally stubby NSSA, so no type 3 or 4 LSAs are sent into the area.

NSSAs are thus a variant of stub areas with one less restriction—external connections are allowed. In all other respects, NSSAs are just stub areas.

NBMA Networks

Remember how a DR is elected—basic to DR election is the broadcast or multicast capability of the underlying network. NBMA networks such as Frame Relay or X.25 have no inherent broadcast or multicast capability, but they can simulate a broadcast network if fully meshed. However, a fully meshed network with n nodes requires $n\times(n–1)/2$ virtual circuits. The cost of $n\times(n–1)/2$ virtual circuits may be unpalatable, and besides, the failure of a single virtual circuit would disrupt this full mesh.

One option around a fully meshed network is to (statically) configure the DR for the network. The DR will then advertise the NBMA network as a multi-access network using a single IP subnet in a network LSA.

Another option is to configure the network as a set of point-to-point networks. This is simpler to configure, manage, and understand. However, each point-to-point network wastes an IP subnet. So what? You can use VLSM in OSPF, with a two-bit subnet for each point-to-point network. That is a good argument. However, the trade-off is the processing overhead of an LSA for each point-to-point network.

Let's look at examples of each of these options.

NewYork2 is set up with a serial interface to support Frame Relay PVCs to offices in Miami and New Orleans, as shown in Figure 6-12.

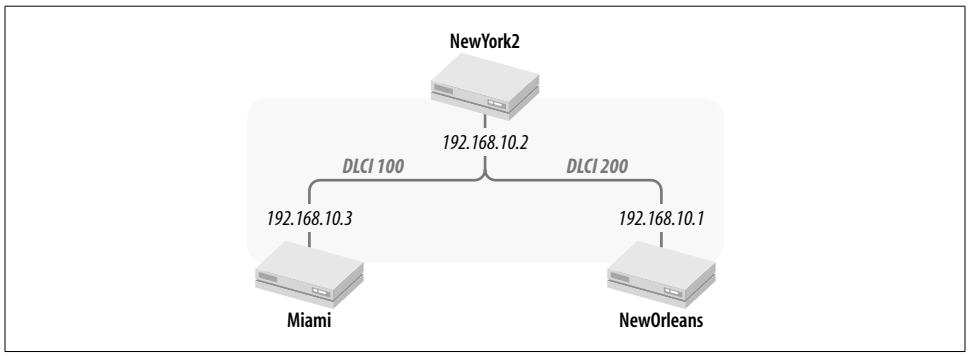

Figure 6-12. TraderMary's Frame Relay network

The command *ip ospf network broadcast* (lines 52, 53, and 55) makes OSPF believe that the attached network is multi-access, like an Ethernet segment. However, since the network has no true broadcast capability, the priorities on *NewYork2*, *Miami*, and *NewOrleans* must be specified to force *NewYork2* to be the DR on the NBMA network. *NewYork2* will become the DR while the state of the interface on *Miami* and *NewOrleans* will be DRother (implying that the interface has not been elected the DR). *NewYork2* uses the default priority of 1. *Miami* and *NewOrleans* are configured with a priority value of 0 (lines 54 and 56), which makes them ineligible for DR election.

```
    hostname NewYork2
    !
    interface Serial3
     ip address 192.168.10.2 255.255.255.0
     encapsulation frame-relay
52   ip ospf network broadcast
     ip ospf hello-interval 30
     keepalive 15
     frame-relay lmi-type ansi
    !
    router ospf 10
    network 192.168.10.0 0.0.0.255 area 0


    hostname Miami
    !
    interface Serial0
     no ip address
     encapsulation frame-relay
     keepalive 15
     frame-relay lmi-type ansi
    !
    interface Serial0.1 point-to-point
     ip address 192.168.10.3 255.255.255.0
53   ip ospf network broadcast
     ip ospf hello-interval 30
54   ip ospf priority 0
     frame-relay interface-dlci 100
    !
    router ospf 10
     network 192.168.10.0 0.0.0.255 area 0


    hostname NewOrleans
    !
    interface Serial0
     no ip address
     encapsulation frame-relay
     bandwidth 1544
     keepalive 15
     lat enabled
     frame-relay lmi-type ansi
```

```
!
interface Serial0.1 point-to-point
 ip address 192.168.10.1 255.255.255.0
55  ip ospf network broadcast
 ip ospf hello-interval 30
56  ip ospf priority 0
 frame-relay interface-dlci 200
!
router ospf 10
network 192.168.10.0 0.0.0.255 area 0
```

IOS releases prior to 10.0 did not support the command *ip ospf network broadcast* and required the static configuration of neighbors and their priorities:

```
neighbor ip-address [priority number] [poll-interval seconds]
```

where *ip-address* is the IP address of the neighbor, *number* is the neighbor's priority (0–255), and *seconds* is the dead router poll interval.

The NBMA network may be modeled as a collection of point-to-point networks. Configure the routers the same way, but configure the interfaces as point-to-multipoint instead of broadcast and do not specify the OSPF priority, since a point-to-multipoint network does not elect a DR (the hello protocol is used to find neighbors):

```
ip ospf network point-to-multipoint
```

The point-to-multipoint network consumes only one IP subnet but creates multiple host routes.

You can also use subinterfaces to model the NBMA network as a collection of point-to-point networks. Routers at the ends of a point-to-point subinterface always form adjacency, much like routers at the ends of a serial interface. No DR election takes place. Since OSPF supports VLSM, one cannot argue that this will waste IP address space. However, using point-to-point subinterfaces in lieu of a single broadcast network generates LSAs for every subinterface, which adds to the processing overhead.

OSPF Design Heuristics

The following sections provide a partial and ad hoc checklist to use when executing an OSPF design. As with any other discipline, the engineer will do best if he spends time understanding the details of OSPF and then designs his network as simply as possible.

OSPF Hierarchy

Building a large, unstructured OSPF network is courting disaster. The design of the OSPF network must be clearly defined: all changes in the OSPF environment must bear the imprint of the OSPF architecture. For example, when adding a new router, the network engineer must answer the following questions:

- Will the router be an area router, a stub router, or an ABR?
- If the router is an ABR or an ASBR, what routes should the router summarize?
- What impact would the failure of the router have on OSPF routing?
- Will this router be a DR/BDR?
- How will this router affect the performance of other OSPF routers?

IP Addressing

IP addresses must be allocated in blocks that allow route summarization at ABRs. The address blocks must take into account the number of users in the area, leaving room for growth. VLSM should be considered when planning IP address allocation.

Router ID

Use loopback addresses to assign router IDs. Choose the router IDs carefully—the router ID will impact DR/BDR election on all attached multi-access networks. Keep handy a list of router IDs and router names. This will make it easier to troubleshoot the network.

DR/BDR

Routers with low processor/memory/bandwidth resources should be made DR-ineligible. A router that becomes the DR/BDR on multiple networks may see high memory/CPU utilization.

Backbone Area

Since all inter-area traffic will traverse the backbone, ensure that there is adequate bandwidth on the backbone links. The backbone area will typically be composed of the highest-bandwidth links in the network, with multiple paths between routers.

The backbone should have multiple paths between any pair of nonbackbone areas. A partitioned backbone will disrupt inter-area traffic—ensure that there is adequate redundancy in the backbone.

Use the backbone solely for inter-area traffic—do not place users or servers on the backbone.

Number of Routers in an Area

The maximum number of routers in an area depends on a number of factors—number of networks, router CPU, router memory, etc.—but Cisco documentation suggests that between 40 and 50 is a reasonable number. However, it is not uncommon

to have a couple of hundred routers in an area, although problems such as flaky links may overload the CPU of the routers in the area. As a corollary of the previous argument, if you think that the total number of routers in your network will not exceed 50, all the routers can be in area 0.

Number of Neighbors

If the number of routers on a multi-access network exceeds 12 to 15 and the DR/BDR is having performance problems, look into a higher-horsepower router for the DR/BDR. Note that having up to 50 routers on a broadcast network is not uncommon. The total number of neighbors on all networks should not exceed 50 or so.

Route Summarization

To summarize the routes:

- Allocate address blocks for each area based on bit boundaries. As areas grow, keep in mind that the area may ultimately need to be split into two. If possible, allocate addresses within an area in contiguous blocks to allow summarization at the time of the split.
- Summarize into the backbone at the ABR (as opposed to summarizing into the nonbackbone area). This reduces the sizes of the LS database in the backbone area and the LS databases in the nonbackbone areas.
- Route summarization has the advantage that a route-flap in a subnet (that has been summarized) does not trigger an LSA to be flooded, reducing the OSPF processing overhead.
- If an area has multiple ABRs and one ABR announces more specific routes, all the traffic will flow to that router. This is good if this is the desired effect. Otherwise, if you intend to use all ABRs equally, all ABRs must have identical summary statements.
- Summarize external routes at the ASBR.
- Golden rule: summarize, summarize, summarize.

VLSM

OSPF LSA records carry subnet masks; the use of VLSM is encouraged to conserve the available IP address space.

Stub Areas

An area with only one ABR is an ideal candidate for a stub area. Changing the area into a stub area will reduce the size of the LS database without the loss of any useful routing information. Remember that stub areas cannot support VLs or type 5 LSAs.

Virtual Links

Design the network so that virtual links are not required. VLs should be used only as emergency fixes, not as a part of the design.

OSPF Timers

In an all-Cisco network environment, the OSPF timers (hello-interval, dead-interval, etc.) can be left to their default values; in a multivendor environment, however, the network engineer may need to adjust the timers to make sure they match.

Troubleshooting OSPF

OSPF is a complex organism and hence can be difficult to troubleshoot. However, since the operation of OSPF has been described in great detail by the standards bodies, the network engineer would do well to become familiar with its internal workings. The following sections describe some of the more common OSPF troubles.

OSPF Area IDs

When you're using multiple network area statements under the OSPF configuration, the order of the statements is critical. Check that the networks have been assigned the desired area IDs by checking the output of the *show ip ospf interface* command.

OSPF Does Not Start

The OSPF process cannot start on a router if a router ID cannot be established. Check the output of *show ip ospf* to see if a router ID has been established. If a router ID has not been established, check to see if the router has an active interface (preferably a loopback interface) with an IP address.

Verifying Neighbor Relationships

Once a router has been able to start OSPF, it will establish an interface data structure for each interface configured to run OSPF. Check the output of *show ip ospf interface* to ensure that OSPF is active on the intended interfaces. If OSPF is active, check for the parameters described in the section "How OSPF Works." Many OSPF problems may be traced to an incorrectly configured interface.

```
NewYork#sh ip ospf interface
...
Ethernet0 is up, line protocol is up
  Internet Address 172.16.1.1/24, Area 0
  Process ID 10, Router ID 172.16.251.1, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State DR, Priority 1
```

```
    Designated Router (ID) 172.16.251.1, Interface address 172.16.1.1
    No backup designated router on this network
    Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
      Hello due in 00:00:02
    Neighbor Count is 0, Adjacent neighbor count is 0
    Suppress hello for 0 neighbor(s)
  Serial0 is up, line protocol is up
59  Internet Address 172.16.250.1/24, Area 0
60  Process ID 10, Router ID 172.16.251.1, Network Type POINT_TO_POINT, Cost: 64
    Transmit Delay is 1 sec, State POINT_TO_POINT,
    Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
      Hello due in 00:00:01
    Neighbor Count is 1, Adjacent neighbor count is 1
      Adjacent with neighbor 69.1.1.1
    Suppress hello for 0 neighbor(s)
```

Remember that two routers will not form a neighbor relationship unless the parameters specified in the hello protocol match.

```
NewYork#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.1.2       1   FULL/  -        00:00:31    172.16.250.2    Serial0
192.168.1.3       1   FULL/  -        00:00:32    172.16.251.2    Serial1
```

If two routers have not been able to establish a neighbor relationship and both are active on the multi-access network (i.e., they are able to ping each other), it is likely that their hello parameters do not match. Use the *debug ip ospf adjacency* command to get details on hello parameter mismatches.

Route Summarization

If an area has multiple ABRs and one ABR announces more specific routes than the others, all the traffic will flow to that router. This is good if this is the desired effect. Otherwise, if you intend to use all ABRs equally, all ABRs must have identical summary statements.

Overloaded Routers

The design engineer should be familiar with OSPF—ABRs do more work than internal routers, and DRs/BDRs do more work than other routers. A router that becomes the DR/BDR on multiple networks does even more work. Routers in stub areas and NSSA areas do less work.

SPF Overrun

To check the number of times the SPF algorithm has executed, use the command *show ip ospf*. A flapping interface may result in frequent executions of the SPF algorithm that, in turn, may take CPU time away from other critical router processes.

```
NewYork#sh ip ospf
61   Routing Process "ospf 10" with ID 172.16.251.1
     Supports only single TOS(TOS0) routes
62   SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
     Number of DoNotAge external LSA 0
     Number of areas in this router is 1. 1 normal 0 stub 0 nssa
        Area BACKBONE(0)
        Number of interfaces in this area is 3
        Area has no authentication
63      SPF algorithm executed 24 times
        Area ranges are
        Link State Update Interval is 00:30:00 and due in 00:11:48
        Link State Age Interval is 00:20:00 and due in 00:11:48
        Number of DCbitless LSA 1
        Number of indication LSA 0
        Number of DoNotAge LSA 0
```

In this example, the SPF algorithm has been executed 24 times since the router was rebooted (line 63). Note that SPF is scheduled to delay its execution for 5 seconds after the receipt of an LSA update and the minimum time between SPF executions is set to 10 seconds (line 61). This keeps SPF from using up all the processor resources in the event that an interface is flapping.

To change these timers, use the following command under the OSPF configuration:

```
timers spf <schedule delay in seconds> <hold-time in seconds>
```

Using the LS Database

Since the LS database is the input to the SPF algorithm, you can analyze it to troubleshoot missing routes. Analyzing the LS database can be particularly useful when you're working with stub areas, totally stubby areas, or NSSAs, since these areas block certain LSAs.

The output of *show ip ospf database database-summary* is a useful indicator of the size of the LS database and its components. The command *show ip ospf database* shows the header information from each LSA.

Network Logs

The output of the command *show log* contains useful historical data and may be used to analyze a network outage.

Debug Commands

The most useful *debug* commands are *debug ip ospf adjacency* and *debug ip ospf events*. These commands are useful in troubleshooting neighbor relationships. Other *debug* commands available are *debug ip ospf flood*, *debug ip ospf lsa-generation*, *debug ip ospf packet*, *debug ip ospf retransmission*, *debug ip ospf spf*, and *debug ip ospf tree*.

Summing Up

OSPF can support very large networks—the OSPF hierarchy allows almost unlimited growth because new areas can be added to the network without impacting other areas. Dijkstra's SPF algorithm leads to radical improvements in convergence time, and OSPF does not suffer from the routing loop issues that DV protocols manifest.

OSPF exhibits all the advantages of a classless routing protocol. Variable Length Subnet Masks permit efficient use of IP addresses. Discontiguous networks can be supported since LSAs carry subnet mask information, and routes can be summarized at arbitrary bit boundaries. Summarization reduces routing protocol overhead and simplifies network management.

Furthermore, OSPF does not tie up network bandwidth and CPU resources in periodic routing updates. Only small hello packets are transmitted on a regular basis.

These OSPF benefits come at a price:

- OSPF is a complex protocol requiring a structured topology. A haphazard environment, without a plan for network addresses, route summarization, LS database sizes, and router performance, will yield a real mess.
- A highly trained staff is required to engineer and operate a large OSPF network.
- OSPF maintains an LS database that requires sizeable memory, and the SPF algorithm can hog CPU resources if the size of the topology database has grown out of bounds. Splitting an area to reduce the size of the LS database may not be straightforward, depending on the topology of the area.
- OSPF assumes a hierarchical network topology—migrating a network from another protocol to OSPF requires extensive planning.