

in which Router B has two possible routes to subnet 10.1.1.0 on the left side of the network: a shorter route over a very slow serial link at 1544 Kbps, or a longer route over two Gigabit Ethernet WAN links.

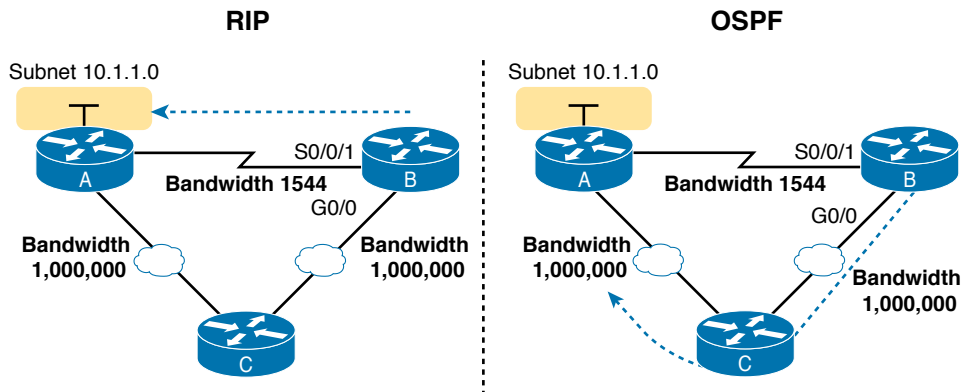


Figure 19-3 *RIP and OSPF Metrics Compared*

The left side of the figure shows the results of RIP in this network. Using hop count, Router B learns of a one-hop route directly to Router A through B's S0/0/1 interface. B also learns of a two-hop route through Router C, through B's G0/0 interface. Router B chooses the lower hop count route, which happens to go over the slow-speed serial link.

The right side of the figure shows the better choice made by OSPF based on its better metric. To cause OSPF to make the right choice, the engineer could use default settings based on the correct interface bandwidth to match the actual link speeds, thereby allowing OSPF to choose the faster route. (The **bandwidth** interface subcommand does not change the actual physical speed of the interface. It just tells IOS what speed to assume the interface is using.)

Other IGP Comparisons

Routing protocols can be compared based on many features, some of which matter to the current CCNA exam, whereas some do not. Table 19-3 introduces a few more points and lists the comparison points mentioned in this book for easier study, with a few supporting comments following the table.

Table 19-3 Interior IP Routing Protocols Compared

Feature	RIPv2	EIGRP	OSPF
Classless/sends mask in updates/supports VLSM	Yes	Yes	Yes
Algorithm (DV, advanced DV, LS)	DV	Advanced DV	LS
Supports manual summarization	Yes	Yes	Yes
Cisco-proprietary	No	Yes ¹	No
Routing updates are sent to a multicast IP address	Yes	Yes	Yes
Convergence	Slow	Fast	Fast

¹ Although Cisco created EIGRP and has kept it as a proprietary protocol for many years, Cisco chose to publish EIGRP as an informational RFC in 2013. This allows other vendors to implement EIGRP, while Cisco retains the rights to the protocol.

Regarding the top row of the table, routing protocols can be considered to be a classless routing protocol or a classful routing protocol. Classless routing protocols support variable-length subnet masks (VLSM) as well as manual route summarization by sending routing protocol messages that include the subnet masks in the message. The older RIPv1 and IGRP routing protocols—both classful routing protocols—do not.

Also, note that the older routing protocols (RIPv1, IGRP) sent routing protocol messages as IP broadcast addresses, while the newer routing protocols in the table all use IP multicast destination addresses. The use of multicasts makes the protocol more efficient and causes less overhead and fewer issues with the devices in the subnet that are not running the routing protocol.

Administrative Distance

Many companies and organizations use a single routing protocol. However, in some cases, a company needs to use multiple routing protocols. For example, if two companies connect their networks so that they can exchange information, they need to exchange some routing information. If one company uses OSPF and the other uses EIGRP on at least one router, both OSPF and EIGRP must be used. Then that router can take routes learned by OSPF and advertise them into EIGRP, and vice versa, through a process called *route redistribution*.

Depending on the network topology, the two routing protocols might learn routes to the same subnets. When a single routing protocol learns multiple routes to the same subnet, the metric tells it which route is best. However, when two different routing protocols learn routes to the same subnet, because each routing protocol’s metric is based on different information, IOS cannot compare the metrics. For example, OSPF might learn a route to subnet 10.1.1.0 with metric 101, and EIGRP might learn a route to 10.1.1.0 with metric 2,195,416, but the EIGRP-learned route might be the better route—or it might not. There is simply no basis for comparison between the two metrics.

When IOS must choose between routes learned using different routing protocols, IOS uses a concept called *administrative distance*. Administrative distance is a number that denotes how believable an entire routing protocol is on a single router. The lower the number, the better, or more believable, the routing protocol. For example, RIP has a default administrative distance of 120, OSPF uses a default of 110, and EIGRP defaults to 90. When using OSPF and EIGRP, the router will believe the EIGRP route instead of the OSPF route (at least by default). The administrative distance values are configured on a single router and are not exchanged with other routers. Table 19-4 lists the various sources of routing information, along with the default administrative distances.

Table 19-4 Default Administrative Distances

Route Type	Administrative Distance
Connected	0
Static	1
BGP (external routes [eBGP])	20
EIGRP (internal routes)	90
IGRP	100
OSPF	110

Route Type	Administrative Distance
IS-IS	115
RIP	120
EIGRP (external routes)	170
BGP (internal routes [iBGP])	200
DHCP default route	254
Unusable	255

NOTE The `show ip route` command lists each route's administrative distance as the first of the two numbers inside the brackets. The second number in brackets is the metric.

The table shows the default administrative distance values, but IOS can be configured to change the administrative distance of a particular routing protocol, a particular route, or even a static route. For example, the command `ip route 10.1.3.0 255.255.255.0 10.1.130.253` defines a static route with a default administrative distance of 1, but the command `ip route 10.1.3.0 255.255.255.0 10.1.130.253 210` defines the same static route with an administrative distance of 210. So, you can actually create a static route that is only used when the routing protocol does not find a route, just by giving the static route a higher administrative distance.

OSPF Concepts and Operation

Routing protocols basically exchange information so routers can learn routes. The routers learn information about subnets, routes to those subnets, and metric information about how good each route is compared to others. The routing protocol can then choose the currently best route to each subnet, building the IP routing table.

Link-state protocols like OSPF take a little different approach to the particulars of what information they exchange and what the routers do with that information once learned. This next (second) major section narrows the focus to only link-state protocols, specifically OSPFv2.

This section begins with an overview of what OSPF does by exchanging data about the network in data structures called *link-state advertisements* (LSA). Then the discussion backs up a bit to provide more details about each of three fundamental parts of how OSPF operates: how OSPF routers use neighbor relationships, how routers exchange LSAs with neighbors, and then how routers calculate the best routes once they learn all the LSAs.

OSPF Overview

Link-state protocols build IP routes with a couple of major steps. First, the routers together build a lot of information about the network: routers, links, IP addresses, status information, and so on. Then the routers flood the information, so all routers know the same information. At that point, each router can calculate routes to all subnets, but from each router's own perspective.

Topology Information and LSAs

Routers using link-state routing protocols need to collectively advertise practically every detail about the internetwork to all the other routers. At the end of the process of *flooding* the information to all routers, every router in the internetwork has the exact same information about the internetwork. Flooding a lot of detailed information to every router sounds like a lot of work, and relative to distance vector routing protocols, it is.

Open Shortest Path First (OSPF), the most popular link-state IP routing protocol, organizes topology information using LSAs and the link-state database (LSDB). Figure 19-4 represents the ideas. Each LSA is a data structure with some specific information about the network topology; the LSDB is simply the collection of all the LSAs known to a router.

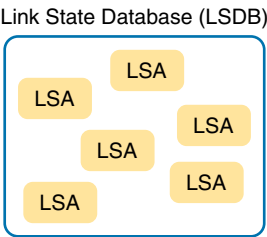


Figure 19-4 LSA and LSDB Relationship

Figure 19-5 shows the general idea of the flooding process, with R8 creating and flooding its *router LSA*. The router LSA for Router R8 describes the router itself, including the existence of subnet 172.16.3.0/24, as seen on the right side of the figure. (Note that Figure 19-5 actually shows only a subset of the information in R8’s router LSA.)

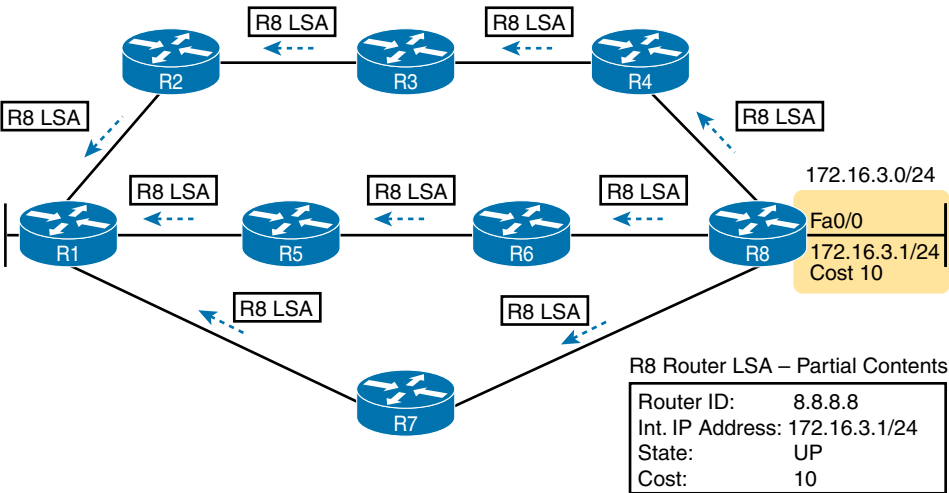


Figure 19-5 Flooding LSAs Using a Link-State Routing Protocol

Figure 19-5 shows the rather basic flooding process, with R8 sending the original LSA for itself, and the other routers flooding the LSA by forwarding it until every router has a copy. The flooding process causes every router to learn the contents of the LSA while preventing

the LSA from being flooded around in circles. Basically, before sending an LSA to yet another neighbor, routers communicate, asking “Do you already have this LSA?,” and then sending the LSA to the next neighbor only if the neighbor has not yet learned about the LSA.

Once flooded, routers do occasionally reflood each LSA. Routers reflood an LSA when some information changes (for example, when a link goes up or comes down). They also reflood each LSA based on each LSA’s separate aging timer (default 30 minutes).

Applying Dijkstra SPF Math to Find the Best Routes

The link-state flooding process results in every router having an identical copy of the LSDB in memory, but the flooding process alone does not cause a router to learn what routes to add to the IP routing table. Although incredibly detailed and useful, the information in the LSDB does not explicitly state each router’s best route to reach a destination.

To build routes, link-state routers have to do some math. Thankfully, you and I do not have to know the math! However, all link-state protocols use a type of math algorithm, called the Dijkstra Shortest Path First (SPF) algorithm, to process the LSDB. That algorithm analyzes (with math) the LSDB and builds the routes that the local router should add to the IP routing table—routes that list a subnet number and mask, an outgoing interface, and a next-hop router IP address.

Now that you have the big ideas down, the next several topics walk through the three main phases of how OSPF routers accomplish the work of exchanging LSAs and calculating routes. Those three phases are

Becoming neighbors: A relationship between two routers that connect to the same data link, created so that the neighboring routers have a means to exchange their LSDBs.

Exchanging databases: The process of sending LSAs to neighbors so that all routers learn the same LSAs.

Adding the best routes: The process of each router independently running SPF, on their local copy of the LSDB, calculating the best routes, and adding those to the IPv4 routing table.

Becoming OSPF Neighbors

Of everything you learn about OSPF in this chapter, OSPF neighbor concepts have the most to do with how you will configure and troubleshoot OSPF in Cisco routers. You configure OSPF to cause routers to run OSPF and become neighbors with other routers. Once that happens, OSPF does the rest of the work to exchange LSAs and calculate routes in the background, with no additional configuration required. This section discusses the fundamental concepts of OSPF neighbors.

The Basics of OSPF Neighbors

OSPF neighbors are routers that both use OSPF and both sit on the same data link. Two routers can become OSPF neighbors if connected to the same VLAN, or same serial link, or same Ethernet WAN link.

Two routers need to do more than simply exist on the same link to become OSPF neighbors; they must send OSPF messages and agree to become OSPF neighbors. To do so, the routers

send OSPF Hello messages, introducing themselves to the potential neighbor. Assuming the two potential neighbors have compatible OSPF parameters, the two form an OSPF neighbor relationship, and would be displayed in the output of the **show ip ospf neighbor** command.

The OSPF neighbor relationship also lets OSPF know when a neighbor might not be a good option for routing packets right now. Imagine R1 and R2 form a neighbor relationship, learn LSAs, and calculate routes that send packets through the other router. Months later, R1 notices that the neighbor relationship with R2 fails. That failed neighbor connection to R2 makes R1 react: R1 refloods LSAs impacted by the failed link, and R1 runs SPF to recalculate its own routes.

Finally, the OSPF neighbor model allows new routers to be dynamically discovered. That means new routers can be added to a network without requiring every router to be reconfigured. Instead, OSPF routers listen for OSPF Hello messages from new routers and react to those messages, attempting to become neighbors and exchange LSDBs.

Meeting Neighbors and Learning Their Router ID

The OSPF Hello process, by which new neighbor relationships are formed, works somewhat like when you move to a new house and meet your various neighbors. When you see each other outside, you might walk over, say hello, and learn each other's name. After talking a bit, you form a first impression, particularly as to whether you think you'll enjoy chatting with this neighbor occasionally, or whether you can just wave and not take the time to talk the next time you see him outside.

Similarly, with OSPF, the process starts with messages called OSPF *Hello* messages. The Hellos in turn list each router's *router ID* (RID), which serves as each router's unique name or identifier for OSPF. Finally, OSPF does several checks of the information in the Hello messages to ensure that the two routers should become neighbors.

OSPF RIDs are 32-bit numbers. As a result, most command output lists these as dotted-decimal numbers (DDN). By default, IOS chooses one of the router's interface IPv4 addresses to use as its OSPF RID. However, the OSPF RID can be directly configured, as covered in the section "Configuring the OSPF Router ID" in Chapter 20, "Implementing OSPF."

As soon as a router has chosen its OSPF RID and some interfaces come up, the router is ready to meet its OSPF neighbors. OSPF routers can become neighbors if they are connected to the same subnet. To discover other OSPF-speaking routers, a router sends multicast OSPF Hello packets to each interface and hopes to receive OSPF Hello packets from other routers connected to those interfaces. Figure 19-6 outlines the basic concept.

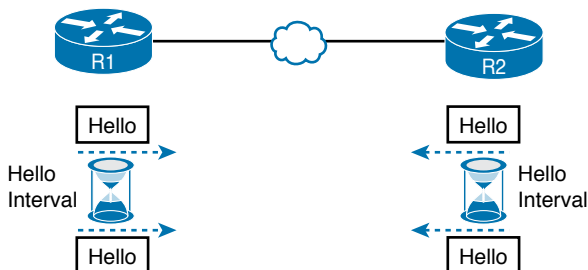


Figure 19-6 OSPF Hello Packets

Routers R1 and R2 both send Hello messages onto the link. They continue to send Hellos at a regular interval based on their Hello timer settings. The Hello messages themselves have the following features:

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

Taking a closer look, Figure 19-7 shows several of the neighbor states used by the early formation of an OSPF neighbor relationship. The figure shows the Hello messages in the center and the resulting neighbor states on the left and right edges of the figure. Each router keeps an OSPF state variable for how it views the neighbor.

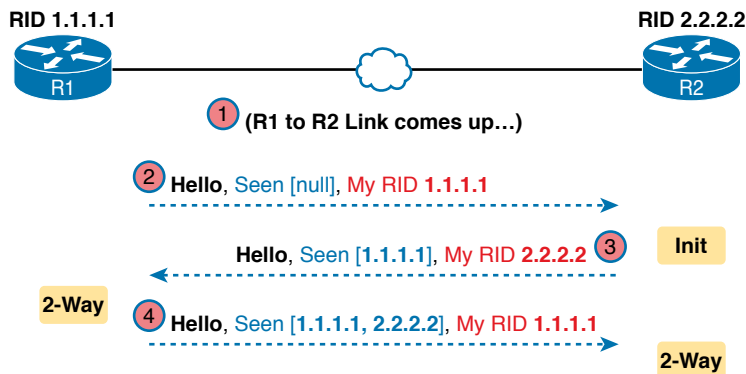


Figure 19-7 Early Neighbor States

Following the steps in the figure, the scenario begins with the link down, so the routers have no knowledge of each other as OSPF neighbors. As a result, they have no state (status) information about each other as neighbors, and they would not list each other in the output of the `show ip ospf neighbor` command. At Step 2, R1 sends the first Hello, so R2 learns of the existence of R1 as an OSPF router. At that point, R2 lists R1 as a neighbor, with an interim beginning state of `init`.

The process continues at Step 3, with R2 sending back a Hello. This message tells R1 that R2 exists, and it allows R1 to move through the `init` state and quickly to a `2-way` state. At Step 4, R2 receives the next Hello from R1, and R2 can also move to a `2-way` state.

The `2-way` state is a particularly important OSPF state. At that point, the following major facts are true:

Key Topic

- The router received a Hello from the neighbor, with that router's own RID listed as being seen by the neighbor.
- The router has checked all the parameters in the Hello received from the neighbor, with no problems. The router is willing to become an OSPF neighbor.
- If both routers reach a `2-way` state with each other, it means that both routers meet all OSPF configuration requirements to become neighbors. Effectively, at that point, they are neighbors and ready to exchange their LSDB with each other.

Exchanging the LSDB Between Neighbors

One purpose of forming OSPF neighbor relationships is to allow the two neighbors to exchange their databases. This next topic works through some of the details of OSPF database exchange.

Fully Exchanging LSAs with Neighbors

The OSPF neighbor state 2-way means that the router is available to exchange its LSDB with the neighbor. In other words, it is ready to begin a 2-way exchange of the LSDB. So, once two routers on a link reach the 2-way state, they can immediately move on to the process of database exchange.

The database exchange process can be quite involved, with several OSPF messages and several interim neighbor states. This chapter is more concerned with a few of the messages and the final state when database exchange has completed: the full state.

After two routers decide to exchange databases, they do not simply send the contents of the entire database. First, they tell each other a list of LSAs in their respective databases—not all the details of the LSAs, just a list. (Think of these lists as checklists.) Then each router can check which LSAs it already has and then ask the other router for only the LSAs that are not known yet.

For instance, R1 might send R2 a checklist that lists 10 LSAs (using an OSPF Database Description, or DD, packet). R2 then checks its LSDB and finds six of those 10 LSAs. So, R2 asks R1 (using a Link-State Request packet) to send the four additional LSAs.

Thankfully, most OSPFv2 work does not require detailed knowledge of these specific protocol steps. However, a few of the terms are used quite a bit and should be remembered. In particular, the OSPF messages that actually send the LSAs between neighbors are called Link-State Update (LSU) packets. That is, the LSU packet holds data structures called link-state advertisements (LSA). The LSAs are not packets, but rather data structures that sit inside the LSDB and describe the topology.

Figure 19-8 pulls some of these terms and processes together, with a general example. The story picks up the example shown in Figure 19-7, with Figure 19-8 showing an example of the database exchange process between Routers R1 and R2. The center shows the protocol messages, and the outer items show the neighbor states at different points in the process. Focus on two items in particular:

- The routers exchange the LSAs inside LSU packets.
- When finished, the routers reach a full state, meaning they have fully exchanged the contents of their LSDBs.

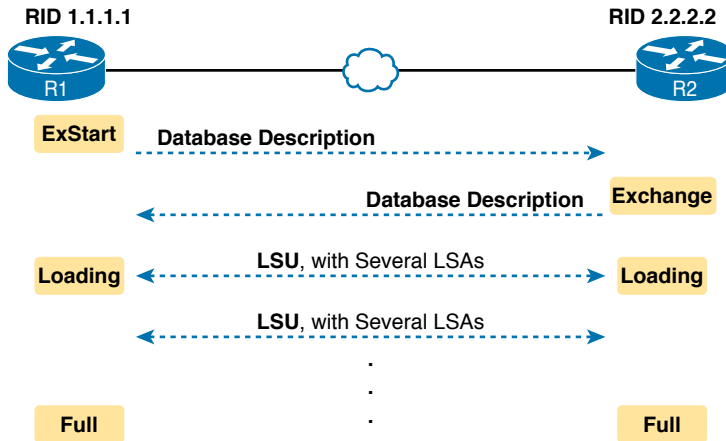


Figure 19-8 Database Exchange Example, Ending in a Full State

Maintaining Neighbors and the LSDB

Once two neighbors reach a full state, they have done all the initial work to exchange OSPF information between them. However, neighbors still have to do some small ongoing tasks to maintain the neighbor relationship.

First, routers monitor each neighbor relationship using Hello messages and two related timers: the *Hello Interval* and the *Dead Interval*. Routers send Hellos every Hello Interval to each neighbor. Each router expects to receive a Hello from each neighbor based on the Hello Interval, so if a neighbor is silent for the length of the Dead Interval (by default, four times as long as the Hello Interval), the loss of Hellos means that the neighbor has failed.

Next, routers must react when the topology changes as well, and neighbors play a key role in that process. When something changes, one or more routers change one or more LSAs. Then the routers must flood the changed LSAs to each neighbor so that the neighbor can change its LSDB.

For example, imagine a LAN switch loses power, so a router's G0/0 interface fails from up/up to down/down. That router updates an LSA that shows the router's G0/0 as being down. That router then sends the LSA to its neighbors, and that neighbor in turn sends it to its neighbors, until all routers again have an identical copy of the LSDB. Each router's LSDB now reflects the fact that the original router's G0/0 interface failed, so each router will then use SPF to recalculate any routes affected by the failed interface.

A third maintenance task done by neighbors is to reflood each LSA occasionally, even when the network is completely stable. By default, each router that creates an LSA also has the responsibility to reflood the LSA every 30 minutes (the default), even if no changes occur. (Note that each LSA has a separate timer, based on when the LSA was created, so there is no single big event where the network is overloaded with flooding LSAs.)

The following list summarizes these three maintenance tasks for easier review:

- Maintain neighbor state by sending Hello messages based on the Hello Interval and listening for Hellos before the Dead Interval expires
- Flood any changed LSAs to each neighbor
- Reflood unchanged LSAs as their lifetime expires (default 30 minutes)

Using Designated Routers on Ethernet Links

OSPF behaves differently on some types of interfaces based on a per-interface setting called the OSPF *network type*. On Ethernet links, OSPF defaults to use a network type of *broadcast*, which causes OSPF to elect one of the routers on the same subnet to act as the *designated router* (DR). The DR plays a key role in how the database exchange process works, with different rules than with point-to-point links.

To see how, consider the example that begins with Figure 19-9. The figure shows five OSPFv2 routers on the same Ethernet VLAN. These five OSPF routers elect one router to act as the DR and one router to be a backup DR (BDR). The figure shows A and B as DR and BDR, for no other reason than the Ethernet must have one of each.

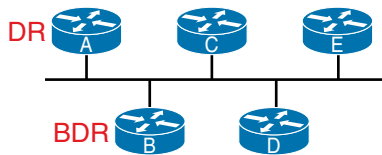


Figure 19-9 Routers A and B Elected as DR and BDR

The database exchange process on an Ethernet link does not happen between every pair of routers on the same VLAN/subnet. Instead, it happens between the DR and each of the other routers, with the DR making sure that all the other routers get a copy of each LSA. In other words, the database exchange happens over the flows shown in Figure 19-10.

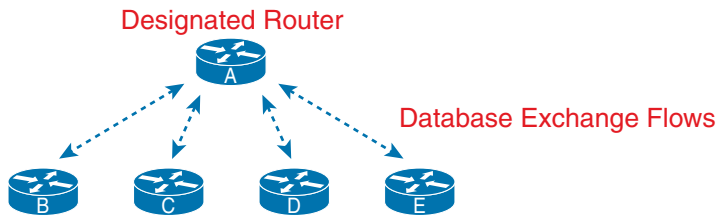


Figure 19-10 Database Exchange to and from the DR on an Ethernet

OSPF uses the BDR concept because the DR is so important to the database exchange process. The BDR watches the status of the DR and takes over for the DR if it fails. (When the DR fails, the BDR takes over, and then a new BDR is elected.)

The use of a DR/BDR, along with the use of multicast IP addresses, makes the exchange of OSPF LSDBs more efficient on networks that allow more than two routers on the same link. The DR can send a packet to all OSPF routers in the subnet by using multicast IP address 224.0.0.5. IANA reserves this address as the “All SPF Routers” multicast address just for this

purpose. For instance, in Figure 19-10, the DR can send one set of messages to all the OSPF routers rather than sending one message to each router.

Similarly, any OSPF router needing to send a message to the DR and also to the BDR (so it remains ready to take over for the DR) can send those messages to the “All SPF DRs” multicast address 224.0.0.6. So, instead of having to send one set of messages to the DR and another set to the BDR, an OSPF router can send one set of messages, making the exchange more efficient.

At this point, you might be getting a little tired of some of the theory, but finally, the theory actually shows something that you may see in **show** commands on a router. Because the DR and BDR both do full database exchange with all the other OSPF routers in the LAN, they reach a full state with all neighbors. However, routers that are neither a DR nor a BDR—called *DROthers* by OSPF—never reach a full state because they do not exchange LSDBs directly with each other. As a result, the **show ip ospf neighbor** command on these DROther routers lists some neighbors in a 2-way state, remaining in that state under normal operation.

For instance, with OSPF working normally on the Ethernet LAN in Figure 19-10, a **show ip ospf neighbor** command on router C (which is a DROther router) would show the following:

- Two neighbors (A and B, the DR and BDR, respectively) with a full state (called *fully adjacent neighbors*)
- Two neighbors (D and E, which are DROthers) with a 2-way state (called *neighbors*)

OSPF requires some terms to describe all neighbors versus the subset of all neighbors that reach the full state. First, all OSPF routers on the same link that reach the 2-way state—that is, they send Hello messages and the parameters match—are called *neighbors*. The subset of neighbors for which the neighbor relationship continues on and reaches the full state are called *adjacent neighbors*. Additionally, OSPFv2 RFC 2328 emphasizes the connection between the full state and the term *adjacent neighbor* by using the synonyms of *fully adjacent* and *fully adjacent neighbor*. Finally, while the terms so far refer to the neighbor, two other terms refer to the relationship: *neighbor relationship* refers to any OSPF neighbor relationship, while the term *adjacency* refers to neighbor relationships that reach a full state. Table 19-5 details the terms.

Key Topic

Table 19-5 Stable OSPF Neighbor States and Their Meanings

Neighbor State	Term for Neighbor	Term for Relationship
2-way	Neighbor	Neighbor Relationship
Full	Adjacent Neighbor Fully Adjacent Neighbor	Adjacency

Calculating the Best Routes with SPF

OSPF LSAs contain useful information, but they do not contain the specific information that a router needs to add to its IPv4 routing table. In other words, a router cannot just copy information from the LSDB into a route in the IPv4 routing table. The LSAs individually are more like pieces of a jigsaw puzzle. So, to know what routes to add to the routing table, each

router must do some SPF math to choose the best routes from that router's perspective. The router then adds each route to its routing table: a route with a subnet number and mask, an outgoing interface, and a next-hop router IP address.

Although engineers do not need to know the details of how SPF does the math, they do need to know how to predict which routes SPF will choose as the best route. The SPF algorithm calculates all the routes for a subnet—that is, all possible routes from the router to the destination subnet. If more than one route exists, the router compares the metrics, picking the best (lowest) metric route to add to the routing table. Although the SPF math can be complex, engineers with a network diagram, router status information, and simple addition can calculate the metric for each route, predicting what SPF will choose.

Once SPF has identified a route, OSPF calculates the metric for a route as follows:

Key
Topic

The sum of the OSPF interface costs for all outgoing interfaces in the route.
Figure 19-11 shows an example with three possible routes from R1 to Subnet X (172.16.3.0/24) at the bottom of the figure.

Key
Topic

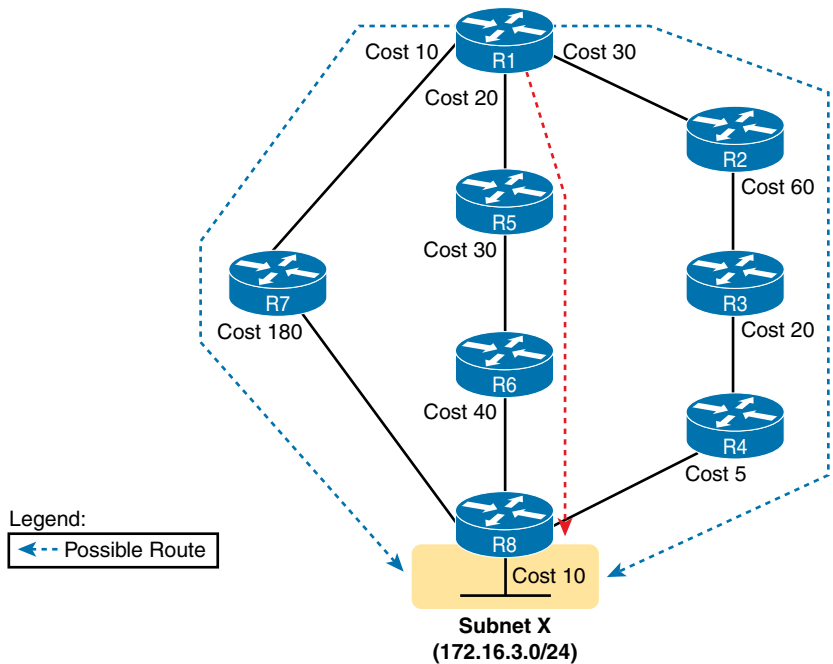


Figure 19-11 SPF Tree to Find R1's Route to 172.16.3.0/24

NOTE OSPF considers the costs of the outgoing interfaces (only) in each route. It does not add the cost for incoming interfaces in the route.

Table 19-6 lists the three routes shown in Figure 19-11, with their cumulative costs, showing that R1's best route to 172.16.3.0/24 starts by going through R5.

Table 19-6 Comparing R1's Three Alternatives for the Route to 172.16.3.0/24

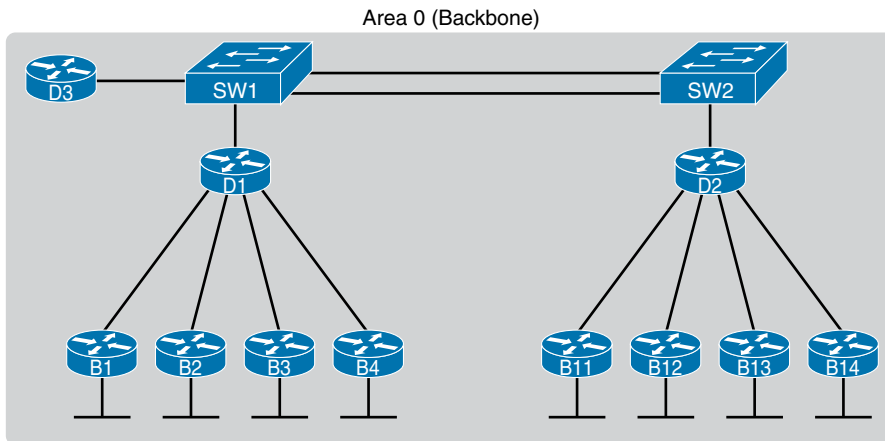
Route	Location in Figure 19-11	Cumulative Cost
R1–R7–R8	Left	$10 + 180 + 10 = 200$
R1–R5–R6–R8	Middle	$20 + 30 + 40 + 10 = 100$
R1–R2–R3–R4–R8	Right	$30 + 60 + 20 + 5 + 10 = 125$

As a result of the SPF algorithm's analysis of the LSDB, R1 adds a route to subnet 172.16.3.0/24 to its routing table, with the next-hop router of R5.

In real OSPF networks, an engineer can do the same process by knowing the OSPF cost for each interface. Armed with a network diagram, the engineer can examine all routes, add the costs, and predict the metric for each route.

OSPF Areas and LSAs

OSPF can be used in some networks with very little thought about design issues. You just turn on OSPF in all the routers, put all interfaces into the same area (usually area 0), and it works! Figure 19-12 shows one such network example, with 11 routers and all interfaces in area 0.

**Figure 19-12** Single-Area OSPF

Larger OSPFv2 networks suffer with a single-area design. For instance, now imagine an enterprise network with 900 routers, rather than only 11, and several thousand subnets. As it turns out, the CPU time to run the SPF algorithm on all that topology data just takes time. As a result, OSPFv2 convergence time—the time required to react to changes in the network—can be slow. The routers might run low on RAM as well. Additional problems with a single area design include the following:

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

The solution is to take the one large LSDB and break it into several smaller LSDBs by using OSPF areas. With areas, each link is placed into one area. SPF does its complicated math on the topology inside the area, and that area's topology only. For instance, an internetwork with 1000 routers and 2000 subnets, broken in 100 areas, would average 10 routers and 20 subnets per area. The SPF calculation on a router would have to only process topology about 10 routers and 20 links, rather than 1000 routers and 2000 links.

So, how large does a network have to be before OSPF needs to use areas? Well, there is no set answer because the behavior of the SPF process depends largely on CPU processing speed, the amount of RAM, the size of the LSDB, and so on. Generally, networks larger than a few dozen routers benefit from areas, and some documents over the years have listed 50 routers as the dividing line at which a network really should use multiple OSPF areas.

The next few pages look at how OSPF area design works, with more reasons as to why areas help make larger OSPF networks work better.

OSPF Areas

OSPF area design follows a couple of basic rules. To apply the rules, start with a clean drawing of the internetwork, with routers, and all interfaces. Then choose the area for each router interface, as follows:



- Put all interfaces connected to the same subnet inside the same area.
- An area should be contiguous.
- Some routers may be internal to an area, with all interfaces assigned to that single area.
- Some routers may be Area Border Routers (ABR) because some interfaces connect to the backbone area, and some connect to nonbackbone areas.
- All nonbackbone areas must have a path to reach the backbone area (area 0) by having at least one ABR connected to both the backbone area and the nonbackbone area.

Figure 19-13 shows one example. An engineer started with a network diagram that showed all 11 routers and their links. On the left, the engineer put four WAN links and the LANs connected to branch routers B1 through B4 into area 1. Similarly, he placed the links to branches B11 through B14 and their LANs in area 2. Both areas need a connection to the backbone area, area 0, so he put the LAN interfaces of D1 and D2 into area 0, along with D3, creating the backbone area.

The figure also shows a few important OSPF area design terms. Table 19-7 summarizes the meaning of these terms, plus some other related terms, but pay closest attention to the terms from the figure.

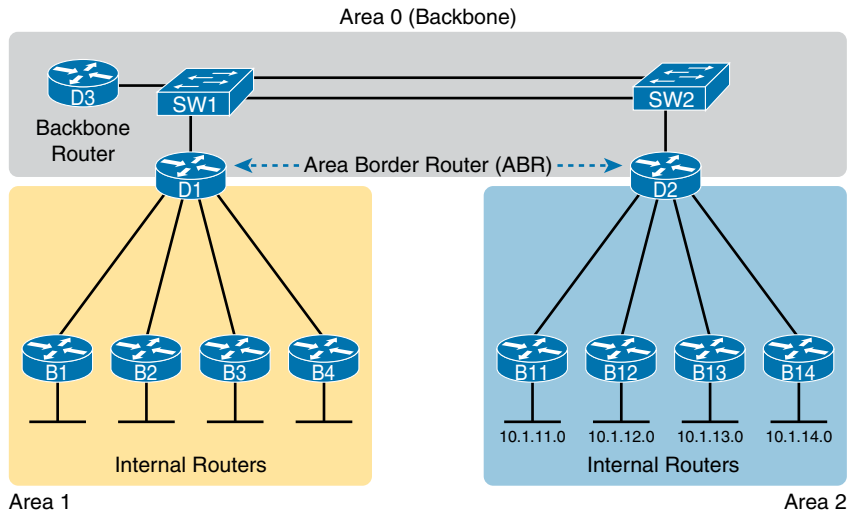
Key
Topic

Figure 19-13 Three-Area OSPF with D1 and D2 as ABRs

Key
Topic

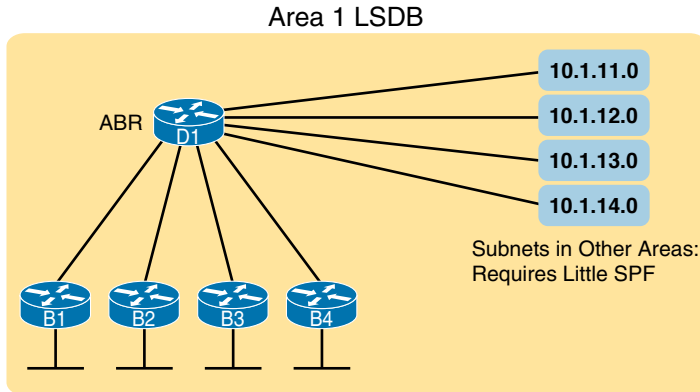
Table 19-7 OSPF Design Terminology

Term	Description
Area Border Router (ABR)	An OSPF router with interfaces connected to the backbone area and to at least one other area
Backbone router	A router connected to the backbone area (includes ABRs)
Internal router	A router in one area (not the backbone area)
Area	A set of routers and links that shares the same detailed LSDB information, but not with routers in other areas, for better efficiency
Backbone area	A special OSPF area to which all other areas must connect—area 0
Intra-area route	A route to a subnet inside the same area as the router
Interarea route	A route to a subnet in an area of which the router is not a part

How Areas Reduce SPF Calculation Time

Figure 19-13 shows a sample area design and some terminology related to areas, but it does not show the power and benefit of the areas. To understand how areas reduce the work SPF has to do, you need to understand what changes about the LSDB inside an area, as a result of the area design.

SPF spends most of its processing time working through all the topology details, namely routers and the links that connect routers. Areas reduce SPF's workload because, for a given area, the LSDB lists only routers and links inside that area, as shown on the left side of Figure 19-14.



Detailed Topology Data (Routers and Links):
Requires Heavy SPF

Figure 19-14 *Smaller Area 1 LSDB Concept*

While the LSDB has less topology information, it still has to have information about all subnets in all areas, so that each router can create IPv4 routes for all subnets. So, with an area design, OSPFv2 uses very brief summary information about the subnets in other areas. These summary LSAs do not include topology information about the other areas; however, each summary LSA *does* list a subnet ID and mask of a subnet in some other area. Summary LSAs do not require much SPF processing at all. Instead, these subnets all appear like subnets connected to the ABR (in Figure 19-14, ABR D1).

Using multiple areas improves OSPF operations in many ways for larger networks. The following list summarizes some of the key points arguing for the use of multiple areas in larger OSPF networks:

- Routers require fewer CPU cycles to process the smaller per-area LSDB with the SPF algorithm, reducing CPU overhead and improving convergence time.
- The smaller per-area LSDB requires less memory.
- Changes in the network (for example, links failing and recovering) require SPF calculations only on routers in the area where the link changed state, reducing the number of routers that must rerun SPF.
- Less information must be advertised between areas, reducing the bandwidth required to send LSAs.

(OSPFv2) Link-State Advertisements

Many people tend to get a little intimidated by OSPF LSAs when first learning about them. Commands that list a summary of the LSDB's contents, like the `show ip ospf database` command, actually list a lot of information. Commands that list the details of the LSDB can list overwhelming amounts of information, and those details appear to be in some kind of code, using lots of numbers. It can seem like a bit of a mess.

However, if you examine LSAs while thinking about OSPF areas and area design, some of the most common LSA types will make a lot more sense. For instance, think about the LSDB

in one area. The topology in one area includes routers and the links between the routers. As it turns out, OSPF defines the first two types of LSAs to define those exact details, as follows:

- One *router LSA* for each router in the area
- One *network LSA* for each network that has a DR plus one neighbor of the DR

Next, think about the subnets in the other areas. The ABR creates summary information about each subnet in one area to advertise into other areas—basically just the subnet IDs and masks—as a third type of LSA:

- One *summary LSA* for each subnet ID that exists in a different area

The next few pages discuss these three LSA types in a little more detail; Table 19-8 lists some information about all three for easier reference and study.

Table 19-8 The Three OSPFv2 LSA Types Seen with a Multiarea OSPF Design

LSA Name	LSA Type	Primary Purpose	Contents of LSA
Router	1	Describe a router	RID, interfaces, IP address/mask, current interface state (status)
Network	2	Describe a network that has a DR	DR and BDR IP addresses, subnet ID, mask
Summary	3	Describe a subnet in another area	Subnet ID, mask, RID of ABR that advertises the LSA

Router LSAs Build Most of the Intra-Area Topology

OSPF needs very detailed topology information inside each area. The routers inside area X need to know all the details about the topology inside area X. And the mechanism to give routers all these details is for the routers to create and flood router (Type 1) and network (Type 2) LSAs about the routers and links in the area.

Router LSAs, also known as Type 1 LSAs, describe the router in detail. Each lists a router's RID, its interfaces, its IPv4 addresses and masks, its interface state, and notes about what neighbors the router knows about via each of its interfaces.

To see a specific instance, first review Figure 19-15. It lists internetwork topology, with subnets listed. Because it's a small internetwork, the engineer chose a single-area design, with all interfaces in backbone area 0.

With the single-area design planned for this small internetwork, the LSDB will contain four router LSAs. Each router creates a router LSA for itself, with its own RID as the LSA identifier. The LSA lists that router's own interfaces, IP address/mask, with pointers to neighbors.

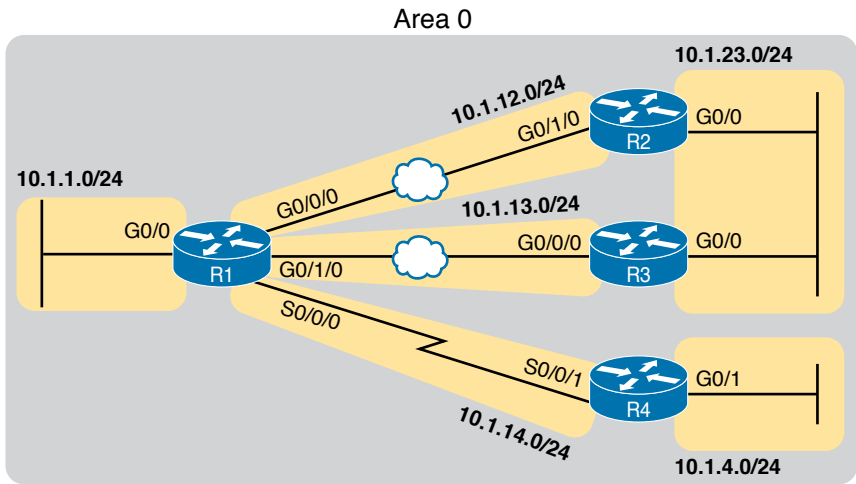


Figure 19-15 Enterprise Network with Seven IPv4 Subnets

Once all four routers have copies of all four router LSAs, SPF can mathematically analyze the LSAs to create a model. The model looks a lot like the concept drawing in Figure 19-16. Note that the drawing shows each router with an obvious RID value. Each router has pointers that represent each of its interfaces, and because the LSAs identify neighbors, SPF can figure out which interfaces connect to which other routers.

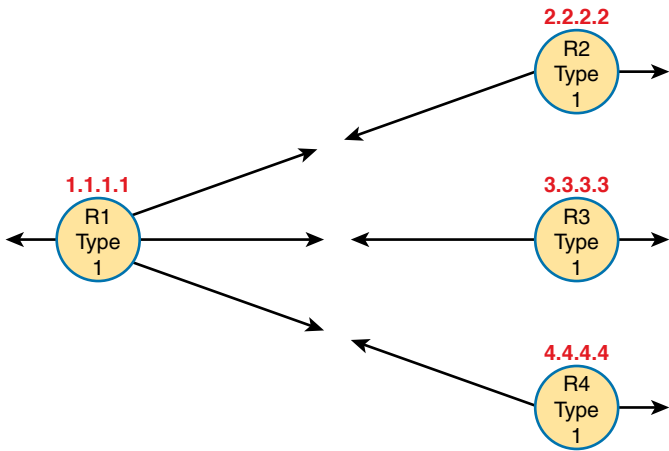


Figure 19-16 Type 1 LSAs, Assuming a Single-Area Design

Network LSAs Complete the Intra-Area Topology

Whereas router LSAs define most of the intra-area topology, network LSAs define the rest. As it turns out, when OSPF elects a DR on some subnet *and* that DR has at least one neighbor, OSPF treats that subnet as another node in its mathematical model of the network. To represent that network, the DR creates and floods a network (Type 2) LSA for that network (subnet).

For instance, back in Figure 19-15, one Ethernet LAN and two Ethernet WANs exist. The Ethernet LAN between R2 and R3 will elect a DR, and the two routers will become neighbors; so, whichever router is the DR will create a network LSA. Similarly, R1 and R2 connect with an Ethernet WAN, so the DR on that link will create a network LSA. Likewise, the DR on the Ethernet WAN link between R1 and R3 will also create a network LSA.

Figure 19-17 shows the completed version of the intra-area LSAs in area 0 with this design. Note that the router LSAs actually point to the network LSAs when they exist, which lets the SPF processes connect the pieces together.

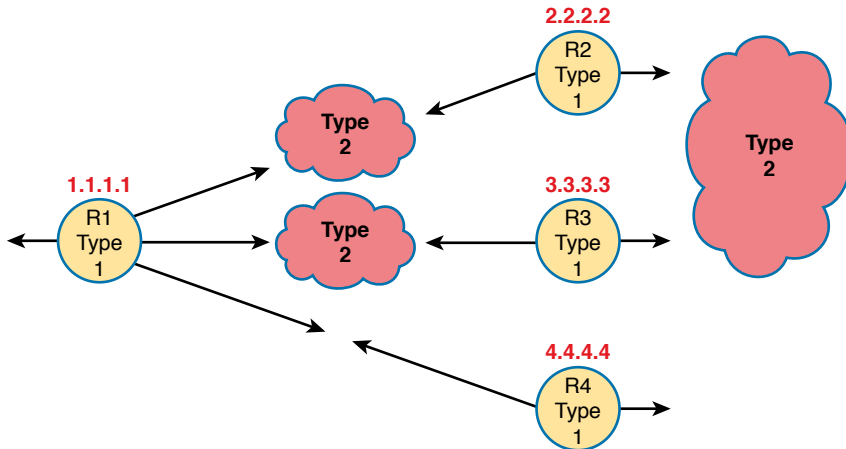


Figure 19-17 Type 1 and Type 2 LSAs in Area 0, Assuming a Single-Area Design

Finally, note that in this single-area design example no summary (Type 3) LSAs exist at all. These LSAs represent subnets in other areas, and there are no other areas. Given that the CCNA 200-301 exam topics refer specifically to single-area OSPF designs, this section stops at showing the details of the intra-area LSAs (Types 1 and 2).

Chapter Review

One key to doing well on the exams is to perform repetitive spaced review sessions. Review this chapter's material using either the tools in the book or interactive tools for the same material found on the book's companion website. Refer to the "Your Study Plan" element for more details. Table 19-9 outlines the key review elements and where you can find them. To better track your study progress, record when you completed these activities in the second column.

Table 19-9 Chapter Review Tracking

Review Element	Review Date(s)	Resource Used:
Review key topics		Book, website
Review key terms		Book, website
Answer DIKTA questions		Book, PTP
Review memory tables		Website

Review All the Key Topics



Table 19-10 Key Topics for Chapter 19

Key Topic Element	Description	Page Number
List	Functions of IP routing protocols	443
List	Definitions of IGP and EGP	444
List	Types of IGP routing protocols	446
Table 19-2	IGP metrics	446
List	Key facts about the OSPF 2-way state	453
Table 19-5	Key OSPF neighbor states	457
Item	Definition of how OSPF calculates the cost for a route	458
Figure 19-11	Example of calculating the cost for multiple competing routes	458
List	OSPF area design rules	460
Figure 19-13	Sample OSPF multiarea design with terminology	461
Table 19-7	OSPF design terms and definitions	461

Key Terms You Should Know

convergence, Shortest Path First (SPF) algorithm, distance vector, Interior Gateway Protocol (IGP), link-state, link-state advertisement (LSA), link-state database (LSDB), metric, 2-way state, full state, Area Border Router (ABR), designated router (DR), backup designated router (BDR), fully adjacent, Hello Interval, Dead Interval, link-state update, neighbor, router ID (RID), topology database, internal router, backbone area

This page intentionally left blank



CHAPTER 20

Implementing OSPF

This chapter covers the following exam topics:

3.0 IP Connectivity

3.2 Determine how a router makes a forwarding decision by default

3.2.b Administrative distance

3.2.c Routing protocol metric

3.4 Configure and verify single area OSPFv2

3.4.a Neighbor adjacencies

3.4.b Point-to-point

3.4.c Broadcast (DR/BR selection)

3.4.d Router ID

OSPFv2 requires only a few configuration commands if you rely on default settings. To use OSPF, all you need to do is enable OSPF on each interface you intend to use in the network, and OSPF uses messages to discover neighbors and learn routes through those neighbors. However, the complexity of OSPFv2 results in a large number of show commands, many of which reveal those default settings. So while you can make OSPFv2 work in a lab with all default settings, to become comfortable working with it, you need to know the most common optional features as well. This chapter begins that process.

The first major section of this chapter focuses on traditional OSPFv2 configuration using the **network** command, along with the large variety of associated show commands. This section teaches you how to make OSPFv2 operate with default settings and convince yourself that it really is working through use of those show commands.

The second major section shows an alternative configuration option called OSPF interface mode, in contrast with the traditional OSPF configuration shown in the first section of the chapter. This mode uses the **ip ospf process-id area area-number** configuration command instead of the **network** command.

The final section then moves on to discuss a variety of optional but popular configuration topics. The features include topics such as how to use passive interfaces, how to change OSPF costs (which influences the routes OSPF chooses), and how to create a default route advertised by OSPF.

“Do I Know This Already?” Quiz

Take the quiz (either here or use the PTP software) if you want to use the score to help you decide how much time to spend on this chapter. The letter answers are listed at the bottom of the page following the quiz. Appendix C, found both at the end of the book as well as on the companion website, includes both the answers and explanations. You can also find both answers and explanations in the PTP testing software.

Table 20-1 “Do I Know This Already?” Foundation Topics Section-to-Question Mapping

Foundation Topics Section	Questions
Implementing Single-Area OSPFv2	1–3
OSPFv2 Interface Configuration	4
Additional OSPFv2 Features	5, 6

- Which of the following **network** commands, following the command **router ospf 1**, tells this router to start using OSPF on interfaces whose IP addresses are 10.1.1.1, 10.1.100.1, and 10.1.120.1?
 - network 10.0.0.0 255.0.0.0 area 0**
 - network 10.0.0.0 0.255.255.255 area 0**
 - network 10.0.0.1 0.0.0.255 area 0**
 - network 10.0.0.1 0.0.255.255 area 0**
- Which of the following **network** commands, following the command **router ospf 1**, tells this router to start using OSPF on interfaces whose IP addresses are 10.1.1.1, 10.1.100.1, and 10.1.120.1?
 - network 10.1.0.0 0.0.255.255 area 0**
 - network 10.0.0.0 0.255.255.0 area 0**
 - network 10.1.1.0 0.x.1x.0 area 0**
 - network 10.1.1.0 255.0.0.0 area 0**
 - network 10.0.0.0 255.0.0.0 area 0**
- Which of the following commands list the OSPF neighbors off interface serial 0/0? (Choose two answers.)
 - show ip ospf neighbor**
 - show ip ospf interface brief**
 - show ip neighbor**
 - show ip interface**
 - show ip ospf neighbor serial 0/0**

4. An engineer migrates from a more traditional OSPFv2 configuration that uses **network** commands in OSPF configuration mode to instead use OSPFv2 interface configuration. Which of the following commands configures the area number assigned to an interface in this new configuration?
 - a. The **area** command in interface configuration mode
 - b. The **ip ospf** command in interface configuration mode
 - c. The **router ospf** command in interface configuration mode
 - d. The **network** command in interface configuration mode
5. Which of the following configuration settings on a router does not influence which IPv4 route a router chooses to add to its IPv4 routing table when using OSPFv2?
 - a. **auto-cost reference-bandwidth**
 - b. **delay**
 - c. **bandwidth**
 - d. **ip ospf cost**
6. OSPF interface configuration uses the **ip ospf process-id area area-number** configuration command. In which modes do you configure the following settings when using this command?
 - a. The router ID is configured explicitly in router mode.
 - b. The router ID is configured explicitly in interface mode.
 - c. An interface's area number is configured in router mode.
 - d. An interface's area number is configured in interface mode.

Foundation Topics

Implementing Single-Area OSPFv2

After an OSPF design has been chosen—a task that can be complex in larger IP internetworks—the configuration can be as simple as enabling OSPF on each router interface and placing that interface in the correct OSPF area. This first major section of the chapter focuses on the required configuration using the traditional OSPFv2 **network** command along with one optional configuration setting: how to set the OSPF router-id. Additionally, this section works through how to show the various lists and tables that confirm how OSPF is working.

For reference and study, the following list outlines the configuration steps covered in this first major section of the chapter:



- Step 1.** Use the **router ospf process-id** global command to enter OSPF configuration mode for a particular OSPF process.
- Step 2.** (Optional) Configure the OSPF router ID by doing the following:
 - A.** Use the **router-id id-value** router subcommand to define the router ID, or
 - B.** Use the **interface loopback number** global command, along with an **ip address address mask** command, to configure an IP address on a loopback interface (chooses the highest IP address of all working loopbacks), or

- C. Rely on an interface IP address (chooses the highest IP address of all working nonloopbacks).

Step 3. Use one or more **network ip-address wildcard-mask area area-id** router subcommands to enable OSPFv2 on any interfaces matched by the configured address and mask, enabling OSPF on the interface for the listed area.

Figure 20-1 shows the relationship between the OSPF configuration commands, with the idea that the configuration creates a routing process in one part of the configuration, and then indirectly enables OSPF on each interface. The configuration does not name the interfaces on which OSPF is enabled, instead requiring IOS to apply some logic by comparing the OSPF **network** command to the interface **ip address** commands. The upcoming example discusses more about this logic.

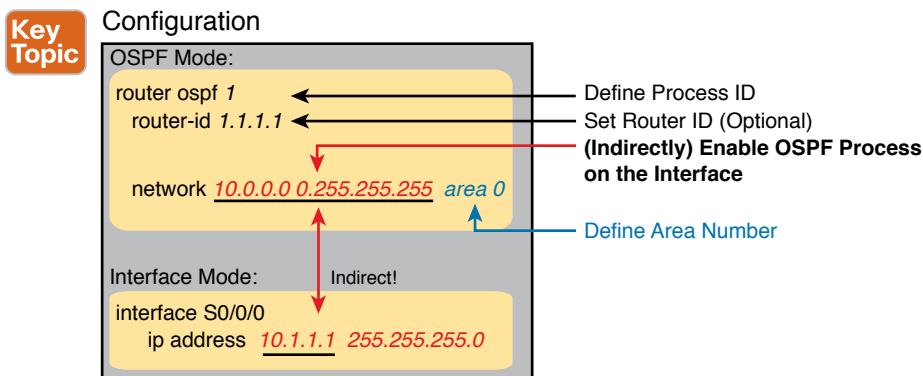


Figure 20-1 Organization of OSPFv2 Configuration with the **network** Command

OSPF Single-Area Configuration

Figure 20-2 shows a sample network that will be used for most examples throughout this chapter. All links reside in area 0, making the area design a single-area design, with four routers. You can think of Router R1 as a router at a central site, with WAN links to each remote site, and using router-on-a-stick (ROAS) to connect to two LAN subnets on the left. Routers R2 and R3 might be at one large remote site that needs two WAN links and two routers for WAN redundancy, with both routers connected to the LAN at that remote site. Router R4 might be a typical smaller remote site with a single router needed for that site.

NOTE The interface numbering on Router R1, with interfaces G0/0 and G0/0/0, may seem a bit strange. However, real routers, like the Cisco 2901 used in the example, use this numbering. That model includes a built-in Gi0/0 and Gi0/1 port. Additionally, if you add one-port Gigabit WAN Interface Cards (WICs), the router numbers them G0/0/0, G0/1/0, and so on. This is just one example of how router hardware may use two-digit interface numbering, or three-digit, or both.

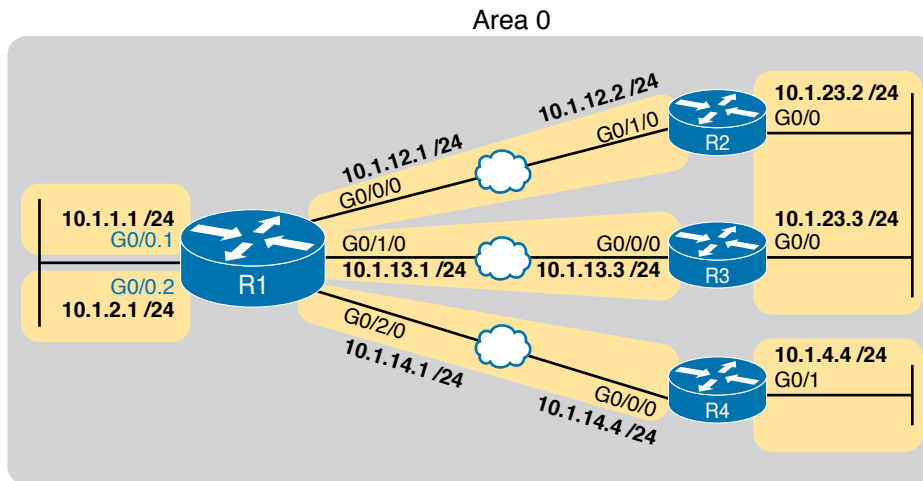


Figure 20-2 Sample Network for OSPF Single-Area Configuration

Example 20-1 shows the IPv4 addressing configuration on Router R1, before getting into the OSPF detail. Note that R1 enables 802.1Q trunking (ROAS) on its G0/0 interface and assigns an IP address to each subinterface.

Example 20-1 IPv4 Address Configuration on R1 (Including VLAN Trunking)

```
interface GigabitEthernet0/0.1
  encapsulation dot1q 1 native
  ip address 10.1.1.1 255.255.255.0
!
interface GigabitEthernet0/0.2
  encapsulation dot1q 2
  ip address 10.1.2.1 255.255.255.0
!
interface GigabitEthernet0/0/0
  ip address 10.1.12.1 255.255.255.0
!
interface GigabitEthernet0/1/0
  ip address 10.1.13.1 255.255.255.0
!
interface GigabitEthernet0/2/0
  ip address 10.1.14.1 255.255.255.0
```

The OSPF configuration begins with the **router ospf process-id** global command, which puts the user in OSPF configuration mode, and sets the OSPF *process-id* value. The *process-id* number just needs to be unique on the local router, allowing the router to support multiple OSPF processes in a single router by using different process IDs. (The

Answers to the “Do I Know This Already?” quiz:

1 B 2 A 3 A, E 4 B 5 B 6 A, D

router command uses the *process-id* to distinguish between the processes.) The *process-id* does not have to match on each router, and it can be any integer between 1 and 65,535.

Second, the configuration needs one or more **network** commands in OSPF mode. These commands tell the router to find its local interfaces that match the first two parameters on the **network** command. Then, for each matched interface, the router enables OSPF on those interfaces, discovers neighbors, creates neighbor relationships, and assigns the interface to the area listed in the **network** command. (Note that the area can be configured as either an integer or a dotted-decimal number, but this book makes a habit of configuring the area number as an integer. The integer area numbers range from 0 through 4,294,967,295.)

Example 20-2 shows an example configuration on router R2 from Figure 20-2. The **router ospf 1** command enables OSPF process 1, and the single **network** command enables OSPF on all interfaces shown in the figure.

Example 20-2 *OSPF Single-Area Configuration on R2 Using One network Command*

```
router ospf 1
network 10.0.0.0 0.255.255.255 area 0
```

For the specific **network** command in Example 20-2, any matched interfaces are assigned to area 0. However, the first two parameters—the *ip_address* and *wildcard_mask* parameter values of 10.0.0.0 and 0.255.255.255—need some explaining. In this case, the command matches both interfaces shown for Router R2; the next topic explains why.

20

Wildcard Matching with the network Command

The key to understanding the traditional OSPFv2 configuration shown in this first example is to understand the OSPF **network** command. The OSPF **network** command compares the first parameter in the command to each interface IP address on the local router, trying to find a match. However, rather than comparing the entire number in the **network** command to the entire IPv4 address on the interface, the router can compare a subset of the octets, based on the wildcard mask, as follows:



Wildcard 0.0.0.0: Compare all four octets. In other words, the numbers must exactly match.

Wildcard 0.0.0.255: Compare the first three octets only. Ignore the last octet when comparing the numbers.

Wildcard 0.0.255.255: Compare the first two octets only. Ignore the last two octets when comparing the numbers.

Wildcard 0.255.255.255: Compare the first octet only. Ignore the last three octets when comparing the numbers.

Wildcard 255.255.255.255: Compare nothing; this wildcard mask means that all addresses will match the **network** command.

Basically, a wildcard mask value of decimal 0 in an octet tells IOS to compare to see if the numbers match, and a value of 255 tells IOS to ignore that octet when comparing the numbers.

The **network** command provides many flexible options because of the wildcard mask. For example, in Router R1, many **network** commands could be used, with some matching all interfaces, and some matching a subset of interfaces. Table 20-2 shows a sampling of options, with notes.

Table 20-2 Example OSPF **network** Commands on R3, with Expected Results

Command	Logic in Command	Matched Interfaces
network 10.1.0.0 0.0.255.255	Match addresses that begin with 10.1	G0/0.1 G0/0.2 G0/0/0 G0/1/0 G0/2/0
network 10.0.0.0 0.255.255.255	Match addresses that begin with 10	G0/0.1 G0/0.2 G0/0/0 G0/1/0 G0/2/0
network 0.0.0.0 255.255.255.255	Match all addresses	G0/0.1 G0/0.2 G0/0/0 G0/1/0 G0/2/0
network 10.1.13.0 0.0.0.255	Match addresses that begin with 10.1.13	G0/1/0
network 10.1.13.1 0.0.0.0	Match one address: 10.1.13.1	G0/1/0

The wildcard mask gives the local router its rules for matching its own interfaces. To show examples of the different options, Example 20-3 shows the configuration on routers R2, R3, and R4, each using different wildcard masks. Note that all three routers (R2, R3, and R4) enable OSPF on all the interfaces shown in Figure 20-2.

Example 20-3 OSPF Configuration on Routers R2, R3, and R4

<pre>! R2 configuration next - one network command enables OSPF on both interfaces interface GigabitEthernet0/0 ip address 10.1.23.2 255.255.255.0 ! interface GigabitEthernet0/1/0 ip address 10.1.12.2 255.255.255.0 ! router ospf 1 network 10.0.0.0 0.255.255.255 area 0</pre>
<pre>! R3 configuration next - One network command per interface interface GigabitEthernet0/0 ip address 10.1.23.3 255.255.255.0</pre>

```

!
interface GigabitEthernet0/0/0
 ip address 10.1.13.3 255.255.255.0
!
router ospf 1
 network 10.1.13.3 0.0.0.0 area 0
 network 10.1.23.3 0.0.0.0 area 0

! R4 configuration next - One network command per interface with wildcard 0.0.0.255
interface GigabitEthernet0/1
 ip address 10.1.4.4 255.255.255.0
!
interface GigabitEthernet0/0/0
 ip address 10.1.14.4 255.255.255.0
!
router ospf 1
 network 10.1.14.0 0.0.0.255 area 0
 network 10.1.4.0 0.0.0.255 area 0

```

Finally, note that OSPF uses the same wildcard mask logic as defined by Cisco IOS access control lists. The section titled “Finding the Right Wildcard Mask to Match a Subnet” section in Chapter 2 of the *CCNA 200-301 Official Cert Guide, Volume 2*, provides more detail about wildcard masks.

NOTE IOS will change a **network** command if it does not follow a particular rule: by convention, if the wildcard mask octet is 255, the matching address octet should be configured as a 0. Interestingly, IOS will actually accept a **network** command that breaks this rule, but then IOS will change that octet of the address to a 0 before putting it into the running configuration file. For example, IOS will change a typed command that begins with **network 1.2.3.4 0.0.255.255** to **network 1.2.0.0 0.0.255.255**.

Verifying OSPF Operation

As mentioned in Chapter 19, “Understanding OSPF Concepts,” OSPF routers use a three-step process to eventually add OSPF-learned routes to the IP routing table. First, they create neighbor relationships. Then they build and flood LSAs between those neighbors so each router in the same area has a copy of the same LSDB. Finally, each router independently computes its own IP routes using the SPF algorithm and adds them to its routing table. This next topic works through how to display the results of each of those steps, which lets you confirm whether OSPF has worked correctly or not.

The **show ip ospf neighbor**, **show ip ospf database**, and **show ip route** commands display information to match each of these three steps, respectively. Figure 20-3 summarizes the commands you can use (and others) when verifying OSPF.

Many engineers begin OSPF verification by looking at the output of the **show ip ospf neighbor** command. For instance, Example 20-4 shows a sample from Router R1, which should have one neighbor relationship each with routers R2, R3, and R4. Example 20-4 shows all three.

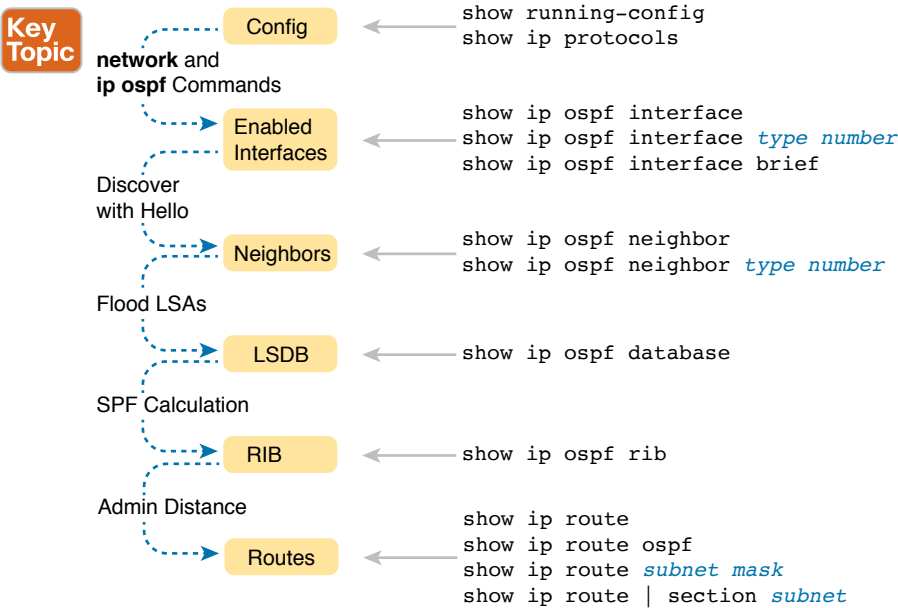


Figure 20-3 OSPF Verification Commands

Key Topic

Example 20-4 OSPF Neighbors on Router R1 from Figure 20-2

R1# show ip ospf neighbor						
Neighbor ID	Pri	State	Dead Time	Address	Interface	
2.2.2.2	1	FULL/DR	00:00:37	10.1.12.2	GigabitEthernet0/0/0	
3.3.3.3	1	FULL/DR	00:00:37	10.1.13.3	GigabitEthernet0/1/0	
4.4.4.4	1	FULL/BDR	00:00:34	10.1.14.4	GigabitEthernet0/2/0	

The detail in the output mentions several important facts, and for most people, working right to left works best in this case. For example, look at the headings:

- Interface:** This is the local router's interface connected to the neighbor. For example, the first neighbor in the list is reachable through R1's G0/0/0 interface.
- Address:** This is the neighbor's IP address on that link. Again, for this first neighbor, which is R1, uses IP address 10.1.13.1.
- State:** While many possible states exist, for the details discussed in this chapter, FULL is the correct and fully working state in this case.
- Neighbor ID:** This is the router ID of the neighbor.

Once OSPF convergence has completed, a router should list each neighbor. On links that use a designated router (DR), the state will also list the role of the neighboring router after the / (DR, BDR, or DROTHER. As a result, the normal working states will be:

**Key
Topic**

FULL/-: The neighbor state is full, with the “-” instead of letters meaning that the link does not use a DR/BDR.

FULL/DR: The neighbor state is full, and the neighbor is the DR.

FULL/BDR: The neighbor state is full, and the neighbor is the backup DR (BDR).

FULL/DROTHER: The neighbor state is full, and the neighbor is neither the DR nor BDR. (It also implies that the local router is a DR or BDR because the state is FULL.)

2WAY/DROTHER: The neighbor state is 2-way, and the neighbor is neither the DR nor BDR—that is, a DROther router. (It also implies that the local router is also a DROther router because otherwise the state would reach a full state.)

Once a router’s OSPF process forms a working neighbor relationship, the routers exchange the contents of their LSDBs, either directly or through the DR on the subnet. Example 20-5 shows the contents of the LSDB on Router R1. Interestingly, with a single-area design, all the routers will have the same LSDB contents once all neighbors are up and all LSAs have been exchanged. So, the **show ip ospf database** command in Example 20-5 should list the same exact information, no matter on which of the four routers it is issued.

Example 20-5 OSPF Database on Router R1 from Figure 20-2

```
R1# show ip ospf database
```

OSPF Router with ID (1.1.1.1) (Process ID 1)

Router Link States (Area 0)					
Link ID	ADV Router	Age	Seq#	Checksum	Link count
1.1.1.1	1.1.1.1	431	0x8000008F	0x00DCCA	5
2.2.2.2	2.2.2.2	1167	0x8000007F	0x009DA1	2
3.3.3.3	3.3.3.3	441	0x80000005	0x002FB1	1
4.4.4.4	4.4.4.4	530	0x80000004	0x007F39	2

Net Link States (Area 0)					
Link ID	ADV Router	Age	Seq#	Checksum	
10.1.12.2	2.2.2.2	1167	0x8000007C	0x00BBD5	
10.1.13.3	3.3.3.3	453	0x80000001	0x00A161	
10.1.14.1	1.1.1.1	745	0x8000007B	0x004449	
10.1.23.3	3.3.3.3	8	0x80000001	0x00658F	

For the purposes of this book, do not be concerned about the specifics in the output of this command. However, for perspective, note that the LSDB should list one “Router Link State” (Type 1 Router LSA) for each of the routers in the same area, so with the design based on Figure 20-2, the output lists four Type 1 LSAs. Also, with all default settings in this design, the routers will create a total of four Type 2 Network LSAs as shown, one each for the subnets that have a DR and contain at least two routers in that subnet.

Next, Example 20-6 shows R4’s IPv4 routing table with the **show ip route** command. As configured, with all links working, the design in Figure 20-2 includes seven subnets. R4 has

connected routes to two of those subnets and should learn OSPF routes to the other five subnets.

Example 20-6 IPv4 Routes Added by OSPF on Router R1 from Figure 20-2

```
R4# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       ! Additional legend lines omitted for brevity

Gateway of last resort is not set

    10.0.0.0/8 is variably subnetted, 9 subnets, 2 masks
O       10.1.1.0/24 [110/2] via 10.1.14.1, 00:27:24, GigabitEthernet0/0/0
O       10.1.2.0/24 [110/2] via 10.1.14.1, 00:27:24, GigabitEthernet0/0/0
C       10.1.4.0/24 is directly connected, Vlan4
L       10.1.4.4/32 is directly connected, Vlan4
O       10.1.12.0/24 [110/2] via 10.1.14.1, 00:27:24, GigabitEthernet0/0/0
O       10.1.13.0/24 [110/2] via 10.1.14.1, 00:25:15, GigabitEthernet0/0/0
C       10.1.14.0/24 is directly connected, GigabitEthernet0/0/0
L       10.1.14.4/32 is directly connected, GigabitEthernet0/0/0
O       10.1.23.0/24 [110/3] via 10.1.14.1, 00:27:24, GigabitEthernet0/0/0
```

Any time you want to check OSPF on a router in a small design like the ones in the book, you can count all the subnets, then count the subnets connected to the local router, and know that OSPF should learn routes to the rest of the subnets. Then just use the **show ip route** command and add up how many connected and OSPF routes exist as a quick check of whether all the routes have been learned or not.

In this case, router R4 has two connected subnets, but seven subnets exist per the figure, so router R4 should learn five OSPF routes. Next look for the code of “O” on the left, which identifies a route as being learned by OSPF. The output lists five such IP routes: two for the LAN subnets off Router R1, one for the LAN subnets connected to both R2 and R3, and one each for the WAN subnets from R1 to R2 and R1 to R3.

Next, take a look at the first route (to subnet 10.1.1.0/24). It lists the subnet ID and mask, identifying the subnet. It also lists two numbers in brackets. The first, 110, is the administrative distance of the route. All the OSPF routes in this example use the default of 110 (see Chapter 19’s Table 19-4 for the list of administrative distance values). The second number, 2, is the OSPF metric for this route. The route also lists the forwarding instructions: the next-hop IP address (10.1.14.1) and R4’s outgoing interface (G0/0/0).

Verifying OSPF Configuration

Once you can configure OSPF with confidence, you will likely verify OSPF focusing on OSPF neighbors and the IP routing table as just discussed. However, if OSPF does not work

immediately, you may need to circle back and check the configuration. To do so, you can use these steps:

- If you have enable mode access, use the **show running-config** command to examine the configuration.
- If you have only user mode access, use the **show ip protocols** command to re-create the OSPF configuration.
- Use the **show ip ospf interface [brief]** command to determine whether the router enabled OSPF on the correct interfaces or not based on the configuration.

NOTE The exam's Sim and Simlet questions can restrict access to enable mode, so knowing how to extract the configuration from show commands other than **show running-config** can be particularly helpful for any configuration topic.

The best way to verify the configuration begins with the **show running-config** command, of course. However, the **show ip protocols** command repeats the details of the OSPFv2 configuration and does not require enable mode access. To see how, consider Example 20-7, which lists the output of the **show ip protocols** command on router R3.

Example 20-7 Router R3 Configuration and the show ip protocols Command

```
! First, a reminder of R3's configuration per Example 20-3:
router ospf 1
  network 10.1.13.3 0.0.0.0 area 0
  network 10.1.23.3 0.0.0.0 area 0
!
! The output from router R3:
R3# show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "ospf 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 3.3.3.3
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    10.1.13.3 0.0.0.0 area 0
    10.1.23.3 0.0.0.0 area 0
  Routing Information Sources:
    Gateway         Distance      Last Update
    1.1.1.1          110          02:05:26
    4.4.4.4          110          02:05:26
    2.2.2.2          110          01:51:16
  Distance: (default is 110)
```

The highlighted output emphasizes some of the configuration. The first highlighted line repeats the parameters on the **router ospf 1** global configuration command. (The second highlighted item points out each router’s router ID, which will be discussed in the next section.) The third set of highlighted lines begins with a heading of “Routing for Networks:” followed by two lines that closely resemble the parameters on the configured **network** commands. In fact, closely compare those last two highlighted lines with the **network** configuration commands at the top of the example, and you will see that they mirror each other, but the **show** command just leaves out the word *network*. For instance:

Configuration: network 10.1.13.3 0.0.0.0 area 0

Show Command: 10.1.13.3 0.0.0.0 area 0

IOS interprets the **network** commands to choose interfaces on which to run OSPF, so it could be that IOS chooses a different set of interfaces than you predicted. To check the list of interfaces chosen by IOS, use the **show ip ospf interface brief** command, which lists all interfaces that have been enabled for OSPF processing. Verifying the interfaces can be a useful step if you have issues with OSPF neighbors because OSPF must first be enabled on an interface before a router will attempt to discover neighbors on that interface. Example 20-8 shows a sample from Router R1.

Example 20-8 Router R1 show ip ospf interface brief Command

R1# show ip ospf interface brief

Interface	PID	Area	IP Address/Mask	Cost	State	Nbrs	F/C
Gi0/0/0	1	0	10.1.12.1/24	1	BDR	1/1	
Gi0/1/0	1	0	10.1.13.1/24	1	BDR	1/1	
Gi0/2/0	1	0	10.1.14.1/24	1	DR	1/1	
Gi0/0.2	1	0	10.1.2.1/24	1	DR	0/0	
Gi0/0.1	1	0	10.1.1.1/24	1	DR	0/0	

First, consider the **show ip ospf interface brief** command shown here. It lists one line per interface, with the list showing all the interfaces on which OSPF has been enabled. Each item in the list identifies the OSPF process ID (per the **router ospf process-id** command), the area, the interface IP address, and the number of neighbors found via each interface.

More generally, note that the **show ip ospf interface** command with the **brief** keyword at the end lists a single line of output per interface, but the **show ip ospf interface** command (without the **brief** keyword) displays about 20 lines of output per interface, with much more information about various OSPF per-interface settings.

Configuring the OSPF Router ID

While OSPF has many other optional features, most enterprise networks that use OSPF choose to configure each router’s OSPF router ID. OSPF-speaking routers must have a router ID (RID) for proper operation. By default, routers will choose an interface IP address to use as the RID. However, many network engineers prefer to choose each router’s router ID, so command output from commands like **show ip ospf neighbor** lists more recognizable router IDs.

To choose its RID, a Cisco router uses the following process when the router reloads and brings up the OSPF process. Note that the router stops looking for a router ID to use once one of the steps identifies a value to use.

Key Topic

1. If the **router-id** *rid* OSPF subcommand is configured, this value is used as the RID.
2. If any loopback interfaces have an IP address configured, and the interface has an interface status of up, the router picks the highest numeric IP address among these loopback interfaces.
3. The router picks the highest numeric IP address from all other interfaces whose interface status code (first status code) is up. (In other words, an interface in up/down state will be included by OSPF when choosing its router ID.)

The first and third criteria should make some sense right away: the RID is either configured or is taken from a working interface's IP address. However, this book has not yet explained the concept of a *loopback interface*, as mentioned in Step 2.

A loopback interface is a virtual interface that can be configured with the **interface loopback interface-number** command, where *interface-number* is an integer. Loopback interfaces are always in an “up and up” state unless administratively placed in a shutdown state. For example, a simple configuration of the command **interface loopback 0**, followed by **ip address 2.2.2.2 255.255.255.0**, would create a loopback interface and assign it an IP address. Because loopback interfaces do not rely on any hardware, these interfaces can be up/up whenever IOS is running, making them good interfaces on which to base an OSPF RID.

Example 20-9 shows the configuration that existed in Routers R1 and R2 before the creation of the **show** command output earlier in this chapter. R1 set its router ID using the direct method, while R2 used a loopback IP address.

Example 20-9 OSPF Router ID Configuration Examples

```
! R1 Configuration first
router ospf 1
  router-id 1.1.1.1
network 10.1.0.0 0.0.255.255 area 0

! R2 Configuration next
!
interface Loopback2
  ip address 2.2.2.2 255.255.255.255
```

Each router chooses its OSPF RID when OSPF is initialized, which happens when the router boots or when a CLI user stops and restarts the OSPF process (with the **clear ip ospf process** command). So, if OSPF comes up, and later the configuration changes in a way that would impact the OSPF RID, OSPF does not change the RID immediately. Instead, IOS waits until the next time the OSPF process is restarted.

Example 20-10 shows the output of the **show ip ospf** command on R1, which identifies the OSPF RID used by R1.

Example 20-10 *Confirming the Current OSPF Router ID*

```
R1# show ip ospf
Routing Process "ospf 1" with ID 1.1.1.1
! lines omitted for brevity
```

Implementing Multiarea OSPF

Even though the current CCNA 200-301 exam blueprint mentions single-area OSPF and does not mention multiarea OSPF, you only need to learn one more idea to know how to configure multiarea OSPF. So, this chapter takes a brief page to show how.

For example, consider a multiarea OSPF design as shown in Figure 20-4. It uses the same routers and IP addresses as shown earlier in Figure 20-2, on which all the examples in this chapter have been based so far. However, the design shows three areas instead of the single-area design shown in Figure 20-2.

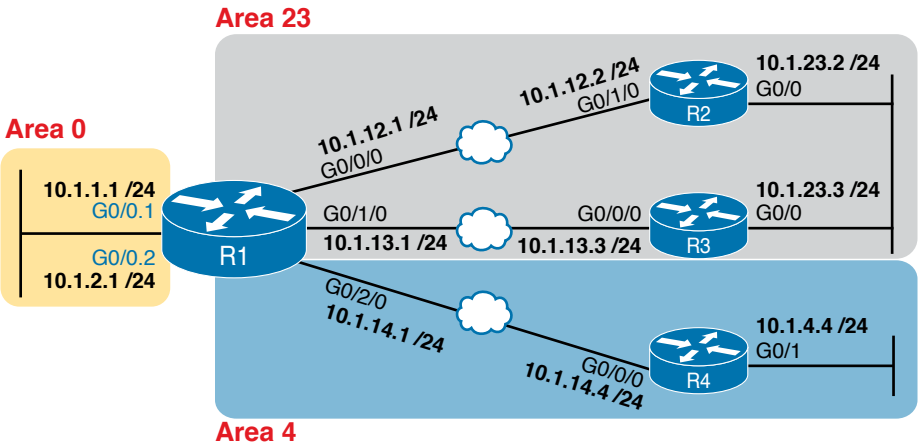


Figure 20-4 *Area Design for an Example Multiarea OSPF Configuration*

Configuring the routers in a multiarea design is almost like configuring OSPFv2 for a single area. To configure multiarea OSPF, all you need is a valid OSPF area design (for instance, like Figure 20-4) and a configuration that places each router interface into the correct area per that design. For example, both of R4’s interfaces connect to links in area 4, making R4 an internal router, so any **network** commands on router R4 will list area 4.

Example 20-11 shows a sample configuration for Router R1. To make the configuration clear, it uses **network** commands with a wildcard mask of 0.0.0.0, meaning each **network** command matches a single interface. Each interface will be placed into either area 0, 23, or 4 to match the figure.

Example 20-11 *OSPF Configuration on R1, Placing Interfaces into Different Areas*

```

router ospf 1
 network 10.1.1.1 0.0.0.0 area 0
 network 10.1.2.1 0.0.0.0 area 0
 network 10.1.12.1 0.0.0.0 area 23
 network 10.1.13.1 0.0.0.0 area 23
 network 10.1.14.1 0.0.0.0 area 4

```

Using OSPFv2 Interface Subcommands

From the earliest days of OSPFv2 support in Cisco routers, the configuration used the OSPF **network** command as discussed in this chapter. However, that configuration style can be confusing, and it does require some interpretation of the **network** commands and interface IP addresses to decide on which interfaces IOS will enable OSPF. As a result, Cisco added another option for OSPFv2 configuration called OSPF interface configuration.

The newer interface-style OSPF configuration still enables OSPF on interfaces, but it does so directly with the **ip ospf** interface subcommand instead of using the **network** command in router configuration mode. Basically, instead of matching interfaces with indirect logic using **network** commands, you directly enable OSPFv2 on interfaces by configuring an interface subcommand on each interface.

OSPF Interface Configuration Example

To show how OSPF interface configuration works, this example basically repeats the example shown earlier in the book using the traditional OSPFv2 configuration with **network** commands. So, before looking at the OSPFv2 interface configuration, take a moment to look back to review traditional OSPFv2 configuration with Figure 20-2 and Examples 20-2 and 20-3.

After reviewing the traditional configuration, consider this checklist, which details how to convert from the old-style configuration in Examples 20-2 and 20-3 to use interface configuration:

Config Checklist

- Step 1.** Use the **no network network-id area area-id** subcommands in OSPF configuration mode to remove the **network** commands.
- Step 2.** Add one **ip ospf process-id area area-id** command in interface configuration mode under each interface on which OSPF should operate, with the correct OSPF process (*process-id*) and the correct OSPF area number.

Figure 20-5 repeats the design for both the original examples in this chapter and for this upcoming interface configuration example.

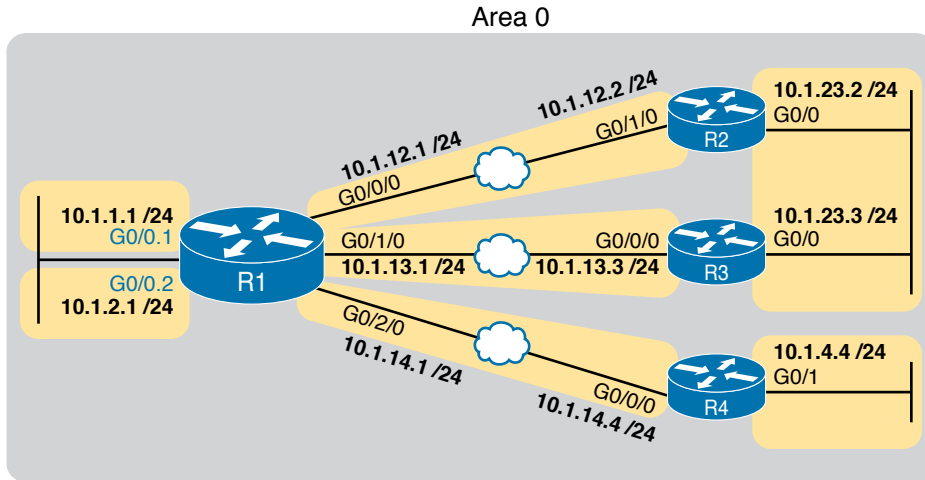


Figure 20-5 Area Design Used in the Upcoming OSPF Interface Config Example

Example 20-2 shows a single **network** command: **network 10.0.0.0 0.255.255.255 area 0**. Example 20-12 follows the steps in the migration checklist, beginning with the removal of the previous configuration using the **no network 10.0.0.0 0.255.255.255 area 0** command. The example then shows the addition of the **ip ospf 1 area 0** command on each of the five interfaces on Router R1, enabling OSPF process 1 on the interface and placing each interface into area 0.

Example 20-12 OSPF Single-Area Configuration on R1 Using One **network** Command

```
R1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# router ospf 1
R1(config-router)# no network 10.0.0.0 0.255.255.255 area 0
R1(config-router)#
*Apr 8 19:35:24.994: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on GigabitEthernet0/0/0
from FULL to DOWN, Neighbor Down: Interface down or detached
*Apr 8 19:35:24.994: %OSPF-5-ADJCHG: Process 1, Nbr 3.3.3.3 on GigabitEthernet0/1/0
from FULL to DOWN, Neighbor Down: Interface down or detached
*Apr 8 19:35:24.994: %OSPF-5-ADJCHG: Process 1, Nbr 4.4.4.4 on GigabitEthernet0/2/0
from FULL to DOWN, Neighbor Down: Interface down or detached
R1(config-router)# interface g0/0.1
R1(config-subif)# ip ospf 1 area 0
R1(config-subif)# interface g0/0.2
R1(config-subif)# ip ospf 1 area 0
R1(config-subif)# interface g0/0/0
R1(config-if)# ip ospf 1 area 0
R1(config-if)#
*Apr 8 19:35:52.970: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on GigabitEthernet0/0/0
from LOADING to FULL, Loading Done
R1(config-if)# interface g0/1/0
R1(config-if)# ip ospf 1 area 0
```

```

R1(config-if)#
*Apr  8 19:36:13.362: %OSPF-5-ADJCHG: Process 1, Nbr 3.3.3.3 on GigabitEthernet0/1/0
  from LOADING to FULL, Loading Done
R1(config-if)# interface g0/2/0
R1(config-if)# ip ospf 1 area 0
R1(config-if)#
*Apr  8 19:37:05.398: %OSPF-5-ADJCHG: Process 1, Nbr 4.4.4.4 on GigabitEthernet0/2/0
  from LOADING to FULL, Loading Done
R1(config-if)#

```

When reading the example, read from top to bottom, and also consider the details about the failed and recovered neighbor relationships shown in the log messages. Removing the **network** command disabled OSPF on all interfaces on Router R1, causing all three neighbor relationships to fail. The example then shows the addition of the **ip ospf 1 area 0** command on the two LAN subinterfaces, which enables OSPF. Then the example shows the same command added to each of the WAN links in succession, and in each case, the OSPF neighbor available over that WAN link comes up (as noted in the log messages.)

Verifying OSPF Interface Configuration

OSPF operates the same way whether you use the new style or old style of configuration. The OSPF area design works the same, neighbor relationships form the same way, routers negotiate to become the DR and BDR the same way, and so on. However, you can see a few small differences in show command output when using the newer OSPFv2 configuration if you look closely.

The **show ip protocols** command relists most of the routing protocol configuration, so it does list some different details if you use interface configuration versus the **network** command. With the newer-style configuration, the output lists the phrase “Interfaces Configured Explicitly,” with the list of interfaces configured with the new **ip ospf process-id area area-id** commands, as highlighted in Example 20-13. The example first shows the relevant parts of the **show ip protocols** command when using interface configuration on Router R1, and then lists the same portions of the command from when R1 used **network** commands.

Example 20-13 Differences in show ip protocols Output: Old- and New-Style OSPFv2 Configuration

```

! First, with the new interface configuration
R1# show ip protocols
! ... beginning lines omitted for brevity
Routing for Networks:
  Routing on Interfaces Configured Explicitly (Area 0):
    GigabitEthernet0/2/0
    GigabitEthernet0/1/0
    GigabitEthernet0/0/0
    GigabitEthernet0/0.2
    GigabitEthernet0/0.1
Routing Information Sources:
  Gateway          Distance      Last Update

```

4.4.4.4	110	00:09:30
2.2.2.2	110	00:10:49
3.3.3.3	110	05:20:07
Distance: (default is 110)		
! For comparison, the old results with the use of the OSPF network command		
R1# show ip protocols		
! ... beginning lines omitted for brevity		
Routing for Networks:		
10.1.0.0 0.0.255.255 area 0		
! ... ending line omitted for brevity		

Another small piece of different output exists in the **show ip ospf interface** *[interface]* command. The command lists details about OSPF settings for the interface(s) on which OSPF is enabled. The output also makes a subtle reference to whether that interface was enabled for OSPF with the old or new configuration style. Example 20-14 also begins with output based on interface configuration on Router R1, followed by the output that would exist if R1 still used the old-style **network** command.

Key Topic

Example 20-14 Differences in show ip ospf interface Output with OSPFv2 Interface Configuration

! First, with the new interface configuration	
R1# show ip ospf interface g0/0/0	
GigabitEthernet0/0/0 is up, line protocol is up	
Internet Address 10.1.12.1/24, Area 0, Attached via Interface Enable	
! Lines omitted for brevity	
! For comparison, the old results with the use of the OSPF network command	
R1# show ip ospf interface g0/0/0	
GigabitEthernet0/0/0 is up, line protocol is up	
Internet Address 10.1.12.1/24, Area 0, Attached via Network Statement	
! ... ending line omitted for brevity	

Other than these small differences in a few show commands, the rest of the commands show nothing different depending on the style of configuration. For instance, the **show ip ospf interface brief** command does not change depending on the configuration style, nor do the **show ip ospf database**, **show ip ospf neighbor**, or **show ip route** commands.

Additional OSPFv2 Features

This final major section of the chapter discusses some very popular but optional OSPFv2 configuration features, as listed here in their order of appearance:

- Passive interfaces
- Default routes
- Metrics
- Load balancing

OSPF Passive Interfaces

Once OSPF has been enabled on an interface, the router tries to discover neighboring OSPF routers and form a neighbor relationship. To do so, the router sends OSPF Hello messages on a regular time interval (called the Hello Interval). The router also listens for incoming Hello messages from potential neighbors.

Sometimes, a router does not need to form neighbor relationships with neighbors on an interface. Often, no other routers exist on a particular link, so the router has no need to keep sending those repetitive OSPF Hello messages. In such cases, an engineer can make the interface passive, which means

Key Topic

- OSPF continues to advertise about the subnet that is connected to the interface.
- OSPF no longer sends OSPF Hellos on the interface.
- OSPF no longer processes any received Hellos on the interface.

The result of enabling OSPF on an interface but then making it passive is that OSPF still advertises about the connected subnet, but OSPF also does not form neighbor relationships over the interface.

To configure an interface as passive, two options exist. First, you can add the following command to the configuration of the OSPF process, in router configuration mode:

```
passive-interface type number
```

Alternately, the configuration can change the default setting so that all interfaces are passive by default and then add a **no passive-interface** command for all interfaces that need to not be passive:

```
passive-interface default
```

```
no passive-interface type number
```

For example, in the sample internetwork in Figure 20-2 (and in Figure 20-5), Router R1, on the left side of the figure, has a LAN interface configured for VLAN trunking. The only router connected to both VLANs is Router R1, so R1 will never discover an OSPF neighbor on these subnets. Example 20-15 shows two alternative configurations to make the two LAN subinterfaces passive to OSPF.

Example 20-15 Configuring Passive Interfaces on R1 from Figure 20-5

```
! First, make each subinterface passive directly
router ospf 1
  passive-interface GigabitEthernet0/0.1
  passive-interface GigabitEthernet0/0.2

! Or, change the default to passive, and make the other interfaces not be passive
router ospf 1
  passive-interface default
  no passive-interface GigabitEthernet0/0/0
  no passive-interface GigabitEthernet0/1/0
  no passive-interface GigabitEthernet0/2/0
```

In real internetworks, the choice of configuration style reduces to which option requires the least number of commands. For example, a router with 20 interfaces, 18 of which are passive to OSPF, has far fewer configuration commands when using the **passive-interface default** command to change the default to passive. If only two of those 20 interfaces need to be passive, use the default setting, in which all interfaces are not passive, to keep the configuration shorter.

Interestingly, OSPF makes it a bit of a challenge to use **show** commands to find whether or not an interface is passive. The **show running-config** command lists the configuration directly, but if you cannot get into enable mode to use this command, note these two facts:

The **show ip ospf interface brief** command lists all interfaces on which OSPF is enabled, *including passive interfaces*.

The **show ip ospf interface** command lists a single line that mentions that the interface is passive.

Example 20-16 shows these two commands on Router R1, based on the configuration shown in the top of Example 20-15. Note that subinterfaces G0/0.1 and G0/0.2 both show up in the output of **show ip ospf interface brief**.

Example 20-16 *Displaying Passive Interfaces*

```
R1# show ip ospf interface brief
Interface                IP-Address      OK? Method Status          Protocol
GigabitEthernet0/0       unassigned      YES manual up              up
GigabitEthernet0/0.1     10.1.1.1        YES manual up              up
GigabitEthernet0/0.2     10.1.2.1        YES manual up              up
GigabitEthernet0/1       unassigned      YES manual administratively down down
GigabitEthernet0/0/0     10.1.12.1       YES manual up              up
GigabitEthernet0/1/0     10.1.13.1       YES manual up              up
GigabitEthernet0/2/0     10.1.14.1       YES manual up              up

R1# show ip ospf interface g0/0.1
GigabitEthernet0/0.1 is up, line protocol is up
  Internet Address 10.1.1.1/24, Area 0, Attached via Network Statement
  Process ID 1, Router ID 1.1.1.1, Network Type BROADCAST, Cost: 1
  Topology-MTID      Cost      Disabled   Shutdown   Topology Name
    0                1         no         no         Base
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 1.1.1.1, Interface address 10.1.1.1
  No backup designated router on this network
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
  No Hellos (Passive interface)
! Lines omitted for brevity
```

OSPF Default Routes

Chapter 16, “Configuring IPv4 Addressing and Static Routes,” showed some of the uses and benefits of default routes, with examples of static default routes. For those exact same reasons, networks can use OSPF to advertise default routes.

The most classic case for using a routing protocol to advertise a default route has to do with an enterprise’s connection to the Internet. As a strategy, the enterprise engineer uses these design goals:

- All routers learn specific (nondefault) routes for subnets inside the company; a default route is not needed when forwarding packets to these destinations.
- One router connects to the Internet, and it has a default route that points toward the Internet.
- All routers should dynamically learn a default route, used for all traffic going to the Internet, so that all packets destined to locations in the Internet go to the one router connected to the Internet.

Figure 20-6 shows the idea of how OSPF advertises the default route, with the specific OSPF configuration. In this case, a company connects to an ISP with its Router R1. That router has a static default route (destination 0.0.0.0, mask 0.0.0.0) with a next-hop address of the ISP router. Then the use of the OSPF **default-information originate** command (Step 2) makes the router advertise a default route using OSPF to the remote routers (B1 and B2).

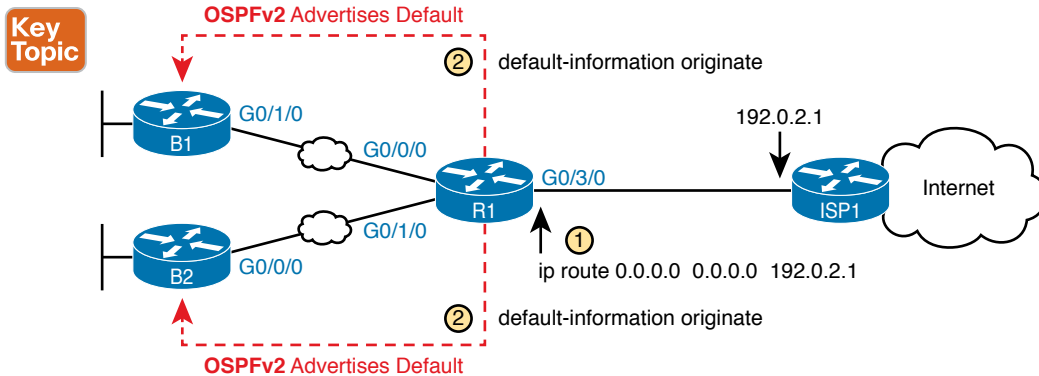


Figure 20-6 Using OSPF to Create and Flood a Default Route

Figure 20-7 shows the default routes that result from OSPF’s advertisements in Figure 20-6. On the far left, the branch routers all have OSPF-learned default routes, pointing to R1. R1 itself also needs a default route, pointing to the ISP router, so that R1 can forward all Internet-bound traffic to the ISP.

Finally, this feature gives the engineer control over when the router originates this default route. First, R1 needs a default route, either defined as a static default route, learned from the ISP with DHCP or learned from the ISP with a routing protocol like eBGP. The OSPF subcommand **default-information originate** then tells OSPF on R1 to advertise a default route when its own default route is working and to advertise the default route as down when its own default route fails.

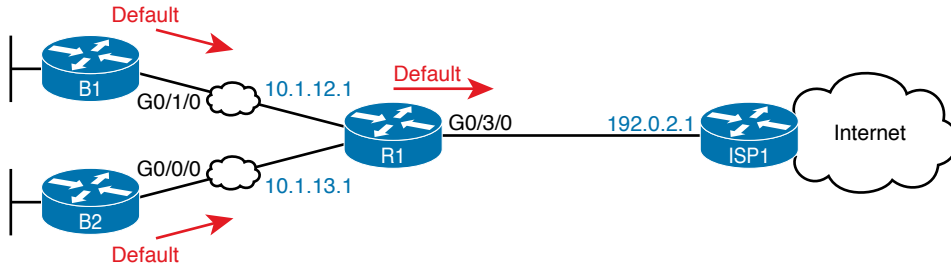


Figure 20-7 *Default Routes Resulting from the default-information originate Command*

NOTE Interestingly, the **default-information originate** always router subcommand tells the router to always advertise the default route, no matter whether the router's default route is working or not.

Example 20-17 shows details of the default route on both R1 and branch router B1 from Figure 20-7. R1 then creates a static default route with the ISP router's IP address of 192.0.2.1 as the next-hop address, as highlighted in the output of the **show ip route static** command output.

Example 20-17 *Default Routes on Routers R1 and B1*

```
! The next command is from Router R1. Note the static code for the default route
R1# show ip route static
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
! Rest of the legend omitted for brevity

Gateway of last resort is 192.0.2.1 to network 0.0.0.0

S*    0.0.0.0/0 [254/0] via 192.0.2.1

! The next command is from router B01; notice the External route code for the default
B1# show ip route ospf
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
! Rest of the legend omitted for brevity

Gateway of last resort is 10.1.12.1 to network 0.0.0.0

O*E2   0.0.0.0/0 [110/1] via 10.1.12.1, 00:20:51, GigabitEthernet0/1/0
        10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
O       10.1.3.0/24 [110/3] via 10.1.12.1, 00:20:51, GigabitEthernet0/1/0
O       10.1.13.0/24 [110/2] via 10.1.12.1, 00:20:51, GigabitEthernet0/1/0
```

Keeping the focus on the command on Router R1, note that R1 indeed has a default route—that is, a route to 0.0.0.0/0. The “Gateway of last resort,” which refers to the default route currently used by the router, points to next-hop IP address 192.0.2.1, which is the ISP router’s IP address. (Refer to Figure 20-7 for the particulars.)

Next look to the bottom half of the example and router B1’s OSPF-learned default route. B1 lists a route for 0.0.0.0/0 as well. The next-hop router in this case is 10.1.12.1, which is Router R1’s IP address on the WAN link. The code on the far left is O*E2, meaning an OSPF-learned route, which is a default route, and is specifically an external OSPF route. Finally, B1’s gateway of last resort setting uses that one OSPF-learned default route, with next-hop router 10.1.12.1.

OSPF Metrics (Cost)

The section “Calculating the Best Routes with SPF” in Chapter 19 discussed how SPF calculates the metric for each route, choosing the route with the best metric for each destination subnet. OSPF routers can influence that choice by changing the OSPF interface cost on any and all interfaces.

Cisco routers allow three different ways to change the OSPF interface cost:

- Directly, using the interface subcommand **ip ospf cost x**.
- Using the default calculation per interface, and changing the *interface bandwidth* setting, which changes the calculated value.
- Using the default calculation per interface, and changing the OSPF *reference bandwidth* setting, which changes the calculated value.

20

Setting the Cost Directly

Setting the cost directly requires a simple configuration command, as shown in Example 20-18. The example sets the cost of two interfaces on Router R1. (This example uses the Figure 20-2 design, as configured in Examples 20-2 and 20-3.) The **show ip ospf interface brief** command that follows details the cost of each interface. Note that the show command confirms the cost settings.

Example 20-18 Confirming OSPF Interface Costs

```
R1# conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# interface g0/0/0
R1(config-if)# ip ospf cost 4
R1(config-if)# interface g0/1/0
R1(config-if)# ip ospf cost 5
R1(config-if)# ^Z
R1#
R1# show ip ospf interface brief
```

Interface	PID	Area	IP Address/Mask	Cost	State	Nbrs	F/C
Gi0/0.2	1	0	10.1.2.1/24	1	DR	0/0	
Gi0/0.1	1	0	10.1.1.1/24	1	DR	0/0	

Gi0/0/0	1	0	10.1.12.1/24	4	DR	1/1
Gi0/1/0	1	0	10.1.13.1/24	5	BDR	1/1
Gi0/2/0	1	0	10.1.14.1/24	1	DR	1/1

The output also shows a cost value of 1 for the other Gigabit interfaces, which is the default OSPF cost for any interface faster than 100 Mbps. The next topic discusses how IOS determines the default cost values.

Setting the Cost Based on Interface and Reference Bandwidth

Routers use a per-interface bandwidth setting to describe the speed of the interface. Note that the interface bandwidth setting does not influence the actual transmission speed. Instead, the interface bandwidth acts as a configurable setting to represent the speed of the interface, with the option to configure the bandwidth to match the actual transmission speed...or not. To support this logic, IOS sets a default interface bandwidth value that matches the physical transmission speed when possible, but also allows the configuration of the interface bandwidth using **bandwidth speed** interface subcommand.

OSPF (as well as other IOS features) uses the interface bandwidth to make decisions, with OSPF using the interface bandwidth in its calculation of the default OSPF cost for each interface. IOS uses the following formula to choose an interface’s OSPF cost if the cost for cases in which the **ip ospf cost** command is not configured on the interface. IOS puts the interface’s bandwidth in the denominator and an OSPF setting called the *reference bandwidth* in the numerator:

$$\text{Reference_bandwidth} / \text{Interface_bandwidth}$$

Note that while you can change both the interface bandwidth and reference bandwidth via configuration, because several IOS features make use of the bandwidth setting, you should avoid changing the interface bandwidth as a means to influence the default OSPF cost.

That being said, many enterprises do use default cost settings while influencing the default by changing the OSPF reference bandwidth while leaving the interface bandwidth as an accurate representation of link speed. Cisco chose the IOS default reference bandwidth setting decades ago in an era with much slower links. As a result, any interface with an interface bandwidth of 100 Mbps or faster ties with a calculated OSPF cost of 1 when using the default reference bandwidth. So, when relying on the default OSPF cost calculation, it helps to configure the reference bandwidth to another value.

To see the issue, consider Table 20-3, which lists several types of interfaces, the default interface bandwidth on those interfaces, and the OSPF cost calculated with the default OSPF reference bandwidth of 100 MBps (that is, 100,000 Kbps). (OSPF rounds up for these calculations, resulting in a lowest possible OSPF interface cost of 1.)

Table 20-3 Faster Interfaces with Equal OSPF Costs

Interface	Interface Default Bandwidth (Kbps)	Formula (Kbps)	OSPF Cost
Serial	1544 Kbps	100,000 / 1544	64
Ethernet	10,000 Kbps	100,000 / 10,000	10
Fast Ethernet	100,000 Kbps	100,000/100,000	1
Gigabit Ethernet	1,000,000 Kbps	100,000/1,000,000	1
10 Gigabit Ethernet	10,000,000 Kbps	100,000/10,000,000	1
100 Gigabit Ethernet	100,000,000 Kbps	100,000/100,000,000	1

As you can see from the table, with a default reference bandwidth, all interfaces from Fast Ethernet's 100 Mbps and faster tie with their default OSPF cost. As a result, OSPF would treat a 100-Mbps link as having the same cost as a 10- or 100-Gbps link, which is probably not the right basis for choosing routes.

You can still use OSPF's default cost calculation (and many do) just by changing the reference bandwidth with the **auto-cost reference-bandwidth speed** OSPF mode subcommand. This command sets a value in a unit of megabits per second (Mbps). Set the reference bandwidth value to a value at least as much as the fastest link speed in the network, but preferably higher, in anticipation of adding even faster links in the future.

For instance, in an enterprise whose fastest links are 10 Gbps (10,000 Mbps), you could set all routers to use **auto-cost reference-bandwidth 10000**, meaning 10,000 Mbps or 10 Gbps. In that case, by default, a 10-Gbps link would have an OSPF cost of 1, while a 1-Gbps link would have a cost of 10, and a 100-Mbps link a cost of 100.

Better still, in that same enterprise, use a reference bandwidth of a faster speed than the fastest interface in the network, to allow room for higher speeds. For instance, in that same enterprise, whose fastest link is 10 Gbps, set the reference bandwidth to 40 Gbps or even 100 Gbps to be ready for future upgrades to use 40-Gbps links, or even 100-Gbps links. (For example, use the **auto-cost reference-bandwidth 100000** command, meaning 100,000 Mbps or 100 Gbps.) That causes 100-Gbps links to have an OSPF cost of 1, 40-Gbps links to have a cost of 4, 10-Gbps links to have a cost of 10, and 1-Gbps links to have a cost of 100.

NOTE Cisco recommends making the OSPF reference bandwidth setting the same on all OSPF routers in an enterprise network.

For convenient study, the following list summarizes the rules for how a router sets its OSPF interface costs:



1. Set the cost explicitly, using the **ip ospf cost x** interface subcommand, to a value between 1 and 65,535, inclusive.
2. Although it should be avoided, change the interface bandwidth with the **bandwidth speed** command, with *speed* being a number in kilobits per second (Kbps).
3. Change the reference bandwidth, using router OSPF subcommand **auto-cost reference-bandwidth ref-bw**, with a unit of megabits per second (Mbps).

OSPF Load Balancing

When a router uses SPF to calculate the metric for each of several routes to reach one subnet, one route may have the lowest metric, so OSPF puts that route in the routing table. However, when the metrics tie for multiple routes to the same subnet, the router can put multiple equal-cost routes in the routing table (the default is four different routes) based on the setting of the **maximum-paths *number*** router subcommand. For example, if an internet-work has six possible paths between some parts of the network, and the engineer wants all routes to be used, the routers can be configured with the **maximum-paths 6** subcommand under **router ospf**.

The more challenging concept relates to how the routers use those multiple routes. A router could load balance the packets on a per-packet basis. For example, if the router has three equal-cost OSPF routes for the same subnet in the routing table, the router could send the one packet over the first route, the next packet over the second route, the next packet over the third route, and then start over with the first route for the next packet. Note that per-packet load balancing is generally a poor choice because it causes the most overhead work on the router. Alternatively, using the default (and better) method, the load balancing could be on a per-destination IP address basis.

Note that the default setting of **maximum-paths** varies by router platform.

Chapter Review

One key to doing well on the exams is to perform repetitive spaced review sessions. Review this chapter’s material using either the tools in the book or interactive tools for the same material found on the book’s companion website. Refer to the “Your Study Plan” element for more details. Table 20-4 outlines the key review elements and where you can find them. To better track your study progress, record when you completed these activities in the second column.

Table 20-4 Chapter Review Tracking

Review Element	Review Date(s)	Resource Used:
Review key topics		Book, website
Review key terms		Book, website
Answer DIKTA questions		Book, PTP
Review Config Checklists		Book, website
Review command tables		Book
Do labs		Blog

Review All the Key Topics

Key Topic

Table 20-5 Key Topics for Chapter 20

Key Topic Element	Description	Page Number
Figure 20-1	Organization of OSPFv2 configuration with the network command	471
List	Example OSPF wildcard masks and their meaning	473
Figure 20-3	OSPF verification commands	476
Example 20-4	Example of the show ip ospf neighbor command	476
List	Neighbor states and their meanings	477
List	Rules for setting the router ID	481
Example 20-14	Differences in show ip ospf interface output with OSPF interface configuration	486
List	Actions IOS takes when an OSPF interface is passive	487
Figure 20-6	Actions taken by the OSPF default-information originate command	489
List	Rules for setting OSPF interface cost	493

Key Terms You Should Know

reference bandwidth, interface bandwidth, maximum paths

Command References

Tables 20-6 and 20-7 list configuration and verification commands used in this chapter. As an easy review exercise, cover the left column in a table, read the right column, and try to recall the command without looking. Then repeat the exercise, covering the right column, and try to recall what the command does.

Table 20-6 Chapter 20 Configuration Command Reference

Command	Description
router ospf <i>process-id</i>	Router subcommand that enters OSPF configuration mode for the listed process.
network <i>ip-address wildcard-mask</i> <i>area area-id</i>	Router subcommand that enables OSPF on interfaces matching the address/wildcard combination and sets the OSPF area.
ip ospf <i>process-id</i> area <i>area-number</i>	Interface subcommand to enable OSPF on the interface and to assign the interface to a specific OSPF area.

Command	Description
ip ospf cost <i>interface-cost</i>	Interface subcommand that sets the OSPF cost associated with the interface.
bandwidth <i>bandwidth</i>	Interface subcommand that directly sets the interface bandwidth (Kbps).
auto-cost reference-bandwidth <i>number</i>	Router subcommand that tells OSPF the numerator in the Reference_bandwidth / Interface_bandwidth formula used to calculate the OSPF cost based on the interface bandwidth.
router-id <i>id</i>	OSPF command that statically sets the router ID.
interface loopback <i>number</i>	Global command to create a loopback interface and to navigate to interface configuration mode for that interface.
maximum-paths <i>number-of-paths</i>	Router subcommand that defines the maximum number of equal-cost routes that can be added to the routing table.
passive-interface <i>type number</i>	Router subcommand that makes the interface passive to OSPF, meaning that the OSPF process will not form neighbor relationships with neighbors reachable on that interface.
passive-interface <i>default</i>	OSPF subcommand that changes the OSPF default for interfaces to be passive instead of active (not passive).
no passive-interface <i>type number</i>	OSPF subcommand that tells OSPF to be active (not passive) on that interface or subinterface.
default-information originate [always]	OSPF subcommand to tell OSPF to create and advertise an OSPF default route, as long as the router has some default route (or to always advertise a default, if the always option is configured).

Table 20-7 Chapter 20 EXEC Command Reference

Command	Description
show ip ospf	Lists information about the OSPF process running on the router, including the OSPF router ID, areas to which the router connects, and the number of interfaces in each area.
show ip ospf interface brief	Lists the interfaces on which the OSPF protocol is enabled (based on the network commands), including passive interfaces.
show ip ospf interface [<i>type number</i>]	Lists a long section of settings, status, and counters for OSPF operation on all interfaces, or on the listed interface, including the Hello and Dead Timers.
show ip protocols	Shows routing protocol parameters and current timer values.

Command	Description
show ip ospf neighbor [<i>type number</i>]	Lists brief output about neighbors, identified by neighbor router ID, including current state, with one line per neighbor; optionally, limits the output to neighbors on the listed interface.
show ip ospf neighbor <i>neighbor-ID</i>	Lists the same output as the show ip ospf neighbor detail command, but only for the listed neighbor (by neighbor RID).
show ip ospf database	Lists a summary of the LSAs in the database, with one line of output per LSA. It is organized by LSA type (first type 1, then type 2, and so on).
show ip route	Lists all IPv4 routes.
show ip route ospf	Lists routes in the routing table learned by OSPF.
show ip route <i>ip-address mask</i>	Shows a detailed description of the route for the listed subnet/mask.
clear ip ospf process	Resets the OSPF process, resetting all neighbor relationships and also causing the process to make a choice of OSPF RID.



CHAPTER 21

OSPF Network Types and Neighbors

This chapter covers the following exam topics:

3.0 IP Connectivity

3.4 Configure and verify single area OSPFv2

3.4.a Neighbor adjacencies

3.4.b Point-to-point

3.4.c Broadcast (DR/BDR selection)

3.4.d Router ID

Chapter 20, “Implementing OSPF,” discussed the required and most common optional OSPF configuration settings, along with the many verification commands to show how OSPF works with those settings. This chapter continues with more OSPF implementation topics, both to round out the discussion of OSPF and to focus even more on the specific CCNA 200-301 exam topics.

The first of two major sections of this chapter focuses on OSPF network types, specifically types point-to-point and broadcast. The CCNA 200-301 exam topics mention those by name. Chapter 20 showed how OSPF operates on Ethernet interfaces when using their default network type (broadcast). This first section of the chapter discusses the meaning of OSPF network types, default settings, how to configure to use other settings, and how OSPF works differently with different settings.

The second major section then focuses on neighbors and neighbor adjacencies as mentioned in yet another of the OSPF exam topics. OSPF routers cannot exchange LSAs with another router unless they first become neighbors. This second section discusses the various OSPF features that can prevent OSPF routers from becoming neighbors and how you can go about discovering if those bad conditions exist—even if you do not have access to the running configuration.

“Do I Know This Already?” Quiz

Take the quiz (either here or use the PTP software) if you want to use the score to help you decide how much time to spend on this chapter. The letter answers are listed at the bottom of the page following the quiz. Appendix C, found both at the end of the book as well as on the companion website, includes both the answers and explanations. You can also find both answers and explanations in the PTP testing software.

Table 21-1 “Do I Know This Already?” Foundation Topics Section-to-Question Mapping

Foundation Topics Section	Questions
OSPF Network Types	1–3
OSPF Neighbor Relationships	4–6

1. Routers R1 and R2, with router IDs 1.1.1.1 and 2.2.2.2, connect over an Ethernet WAN link. If using all default OSPF settings, if the WAN link initializes for both routers at the same time, which of the following answers are true? (Choose two answers.)
 - a. Router R1 will become the DR.
 - b. Router R1 will dynamically discover the existence of router R2.
 - c. Router R2 will be neither the DR nor the BDR.
 - d. Router R1’s **show ip ospf neighbor** command will list R2 with a state of “FULL/DR.”
2. Routers R1 and R2, with router IDs 1.1.1.1 and 2.2.2.2, connect over an Ethernet WAN link. The configuration uses all defaults, except giving R1 an interface priority of 11 and changing both routers to use OSPF network type point-to-point. If the WAN link initializes for both routers at the same time, which of the following answers are true? (Choose two answers.)
 - a. Router R1 will become the DR.
 - b. Router R1 will dynamically discover the existence of router R2.
 - c. Router R2 will be neither the DR nor the BDR.
 - d. Router R2’s **show ip ospf neighbor** command will list R1 with a state of “FULL/DR.”
3. Per the command output, with how many routers is router R9 full adjacent over its Gi0/0 interface?

R9# **show ip ospf interface brief**

Interface	PID	Area	IP Address/Mask	Cost	State	Nbrs	F/C
Gi0/0	1	0	10.1.1.1/24	1	DROTH	2/5	

- a. 7
- b. 0
- c. 5
- d. 2

4. An engineer connects routers R11 and R12 to the same Ethernet LAN and configures them to use OSPFv2. Which answers describe a combination of settings that would prevent the two routers from becoming OSPF neighbors? (Choose two answers.)
 - a. R11's interface uses area 11 while R12's interface uses area 12.
 - b. R11's OSPF process uses process ID 11 while R12 uses process ID 12.
 - c. R11's interface uses OSPF priority 11 while R12's uses OSPF priority 12.
 - d. R11's interface uses an OSPF Hello timer value of 11 while R12's uses 12.
5. An engineer connects routers R13 and R14 to the same Ethernet LAN and configures them to use OSPFv2. Which answers describe a combination of settings that would prevent the two routers from becoming OSPF neighbors?
 - a. Both routers' interface IP addresses reside in the same subnet.
 - b. Both routers' OSPF process uses process ID 13.
 - c. Both routers' OSPF process uses router ID 13.13.13.13.
 - d. Both routers' interfaces use an OSPF Dead interval of 40.
6. Router R15 has been a working part of a network that uses OSPFv2. An engineer then issues the **shutdown** command in OSPF configuration mode on R15. Which of the following occurs?
 - a. R15 empties its IP routing table of all OSPF routes but keeps its LSDB intact.
 - b. R15 empties its LSDB but keeps OSPF neighbor relationships active.
 - c. R15 keeps OSPF neighbors open but does not accept new OSPF neighbors.
 - d. R15 keeps all OSPF configuration but ceases all OSPF activities (routes, LSDB, neighbors).

Foundation Topics

OSPF Network Types

Two CCNA 200-301 exam topics might be completely misunderstood without taking a closer look at yet more default OSPF settings. In particular, the following exam topics refer to a specific per-interface OSPF setting called the *network type*—even listing the keywords used to configure the setting in the exam topics:

3.4.b: **point-to-point**

3.4.c: **broadcast** (DR/BDR selection)

OSPF includes a small number of network types as a setting on each OSPF-enabled interface. The setting tells the router whether or not to dynamically discover OSPF neighbors (versus requiring the static configuration of the neighboring router's IP address) and whether or not the router should attempt to use a designated router (DR) and backup DR (BDR) in the subnet. Of the two OSPF network types included in the CCNA exam topics, both cause routers to dynamically discover neighbors, but one calls for the use of a DR while the other does not. Table 21-2 summarizes the features of the two OSPF network types mentioned in the exam topics.

Key Topic

Table 21-2 Two OSPF Network Types and Key Behaviors

Network Type Keyword	Dynamically Discovers Neighbors	Uses a DR/BDR
broadcast	Yes	Yes
point-to-point	Yes	No

The rest of this first major section of the chapter explores each type.

The OSPF Broadcast Network Type

OSPF defaults to use a *broadcast* network type on all types of Ethernet interfaces. Note that all the Ethernet interfaces in examples in Chapter 20 relied on that default setting.

To see all the details of how the OSPF broadcast network type works, this chapter begins with a different design than the examples in Chapter 20, instead using a single area design that connects four routers to the same subnet, as shown in Figure 21-1. All links reside in area 0, making the design a single area design.

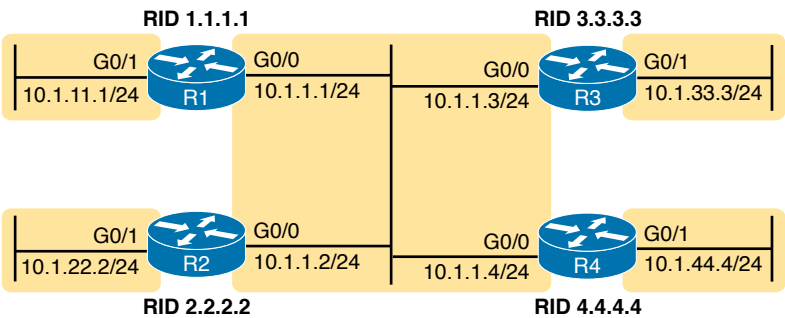


Figure 21-1 The Single Area Design Used in This Chapter

To get a sense for how OSPF operates with the broadcast network type, imagine that all four routers use a straightforward OSPF interface configuration like the router R1 configuration shown in Example 21-1. Both GigabitEthernet interfaces on all four routers default to use network type broadcast. Note that the configuration on routers R2, R3, and R4 mirrors R1's configuration except that they use router IDs 2.2.2.2, 3.3.3.3, and 4.4.4.4, respectively, and they use the IP addresses shown in the figure.

Example 21-1 R1 OSPF Configuration to Match Figure 21-1

```
router ospf 1
  router-id 1.1.1.1
  !
interface gigabitEthernet0/0
  ip ospf 1 area 0
  !
interface gigabitEthernet0/1
  ip ospf 1 area 0
```

This simple design gives us a great backdrop from which to observe the results of the broadcast network type on each router. Both interfaces (G0/0 and G0/1) on each router use the broadcast network type and perform the following actions:

- Attempt to discover neighbors by sending OSPF Hellos to the 224.0.0.5 multicast address (an address reserved for sending packets to all OSPF routers in the subnet)
- Attempt to elect a DR and BDR on each subnet
- On the interface with no other routers on the subnet (G0/1), become the DR
- On the interface with three other routers on the subnet (G0/0), be either DR, BDR, or a DROther router
- When sending OSPF messages to the DR or BDR, send the messages to the all-OSPF-DRs multicast address 224.0.0.6

Example 21-2 shows some of the results using the **show ip ospf neighbor** command. Note that R1 lists R2, R3, and R4 as neighbors (based on their 2.2.2.2, 3.3.3.3, and 4.4.4.4 router IDs), confirming that R1 dynamically discovered the other routers. Also, note that the output lists 4.4.4.4 as the DR and 3.3.3.3 as the BDR.

Example 21-2 R1's List of Neighbors

R1# show ip ospf neighbor

Neighbor ID	Pri	State	Dead Time	Address	Interface
2.2.2.2	1	2WAY/DROTHER	00:00:35	10.1.1.2	GigabitEthernet0/0
3.3.3.3	1	FULL/BDR	00:00:33	10.1.1.3	GigabitEthernet0/0
4.4.4.4	1	FULL/DR	00:00:35	10.1.1.4	GigabitEthernet0/0

Verifying Operations with Network Type Broadcast

As discussed in the section “Using Designated Routers on Ethernet Links” in Chapter 19, “Understanding OSPF Concepts,” all discovered routers on the link should become neighbors and at least reach the *2-way* state. For all neighbor relationships that include the DR and/or BDR, the neighbor relationship should further reach the *full* state. That section defined the term *fully adjacent* as a special term that refers to neighbors that reach this full state.

The design in Figure 21-1, with four routers on the same LAN, provides just enough routers so that one neighbor relationship will remain in a 2-way state and not reach the full state, as a perfectly normal way for OSPF to operate. Figure 21-2 shows the current conditions when the **show** commands in this chapter were gathered, with R4 as the DR, R3 as the BDR, and with R1 and R2 as DROther routers.

Now consider router R1's neighbors as listed in Example 21-2. R1 has three neighbors, all reachable out its G0/0 interface. However, R1's **show ip ospf neighbor** command refers to the state of R1's relationship with the neighbor: 2-way with router 2.2.2.2. Because both R1 and R2 currently serve as DROther routers—that is, they wait ready to become the BDR if either the DR or BDR fails—their neighbor relationship remains in a 2-way state.

Answers to the “Do I Know This Already?” quiz:

1 B, D 2 B, C 3 D 4 A, D 5 C 6 D

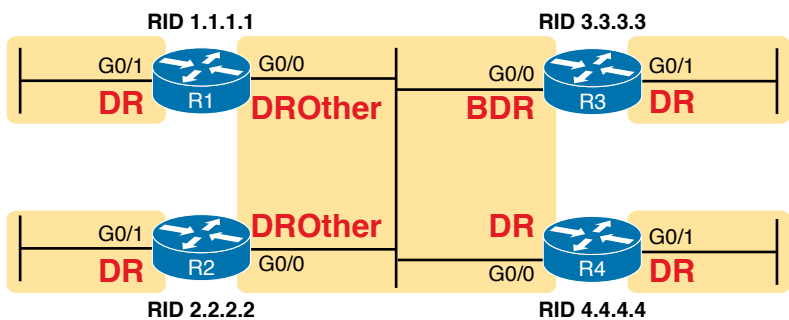


Figure 21-2 OSPF DR/BDR/DROther Roles in the Network

Examining Example 21-2 one last time, R1, as a DROther router itself, has two neighbor relationships that reach a full state: R1's neighbor adjacency with DR R4 and R1's neighbor adjacency with BDR R3. But R1 has a total of three neighbors, all reachable off R1's G0/0 interface.

The idea that R1 has three neighbors off its G0/0 interface, with two being fully adjacent, is reflected on the far right of the output of the **show ip ospf interface brief** command output in Example 21-3. It shows "2/3," meaning two neighbors in the full state off port G0/0, with three total neighbors on that interface. Also, note that this command's "State" column differs from the **show ip ospf neighbor** commands, in that the **show ip ospf interface brief** command lists the local router's role on the interface (as shown in Figure 21-2), with R1's G0/1 acting as DR and R1's G0/0 acting as a DROther router.

Key Topic

Example 21-3 Router R1 OSPF Interfaces: Local Role and Neighbor Counts

R1# **show ip ospf interface brief**

Interface	PID	Area	IP Address/Mask	Cost	State	Nbrs	F/C
Gi0/1	1	0	10.1.11.1/24	1	DR	0/0	
Gi0/0	1	0	10.1.1.1/24	1	DROTH	2/3	

So far, this topic has described the effect of the OSPF broadcast network type by taking advantage of the default setting on Ethernet interfaces. To see the setting, use the **show ip ospf interface** command, as shown in Example 21-4. The first highlighted item identifies the network type. However, this command's output restates many of the facts seen in both the **show ip ospf neighbor** and **show ip ospf interface brief** commands in Examples 21-2 and 21-3, so take the time to browse through all of Example 21-4 and focus on the additional highlights to see those familiar items.

Example 21-4 Displaying OSPF Network Type Broadcast

R1# **show ip ospf interface g0/0**

```
GigabitEthernet0/0 is up, line protocol is up
  Internet Address 10.1.1.1/24, Area 0, Attached via Interface Enable
  Process ID 1, Router ID 1.1.1.1, Network Type BROADCAST, Cost: 1
  Topology-MTID      Cost      Disabled      Shutdown      Topology Name
    0                  1          no            no            Base
  Enabled by interface config, including secondary ip addresses
```

```

Transmit Delay is 1 sec, State DROTHER, Priority 1
Designated Router (ID) 4.4.4.4, Interface address 10.1.1.4
Backup Designated router (ID) 3.3.3.3, Interface address 10.1.1.3
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
  oob-resync timeout 40
  Hello due in 00:00:00
Supports Link-local Signaling (LLS)
Cisco NSF helper support enabled
IETF NSF helper support enabled
Index 1/1/1, flood queue length 0
Next 0x0(0)/0x0(0)/0x0(0)
Last flood scan length is 0, maximum is 1
Last flood scan time is 0 msec, maximum is 0 msec
Neighbor Count is 3, Adjacent neighbor count is 2
  Adjacent with neighbor 3.3.3.3 (Backup Designated Router)
  Adjacent with neighbor 4.4.4.4 (Designated Router)
Suppress hello for 0 neighbor(s)

```

Although you would not need to configure an Ethernet interface to use the broadcast network type, some older types of interfaces over the years have used different defaults and with the option to use the broadcast network type. In those cases, the **ip ospf network broadcast** interface subcommand would configure the setting.

Configuring to Influence the DR/BDR Election

In some cases, you may want to influence the OSPF DR election. However, before deciding that makes sense in every case, note that OSPF DR/BDR election rules will not result in a specific router always being the DR, and another always being the BDR, assuming that each is up and working. In short, here are the rules once a DR and BDR have been elected:

- If the DR fails, the BDR becomes the DR, and a new BDR is elected.
- When a better router enters the subnet, no preemption of the existing DR or BDR occurs.

As a result of these rules, while you can configure a router to be the best (highest priority) router to become the DR in an election, doing so only increases that router's statistical chances of being the DR at a given point in time. If the router fails, other routers will become DR and BDR, and the best router will not be DR again until the current DR and BDR fail.

NOTE If you have begun to think about STP elections, note that the rules are similar, but with two key differences. STP uses a lowest-is-best approach and allows new switches to preempt the existing root switch to become the root. OSPF uses a highest-is-best approach and does not preempt the DR as just noted.

In some cases, you may want to influence the DR/BDR election with two configurable settings, listed here in order of precedence:

**Key
Topic**

- **The highest OSPF interface priority:** The highest value wins during an election, with values ranging from 0 to 255.
- **The highest OSPF Router ID:** If the priority ties, the election chooses the router with the highest OSPF RID.

For example, imagine all four routers in the design shown in Figure 21-1 trying to elect the DR and BDR at the same time—for instance, after a power hit in which all four routers power off and back on again. They all participate in the election. They all tie with default priority values of 1 (see Example 21-4 for R1's priority in the **show ip ospf interface** command output.) In this case, R4, with the numerically highest RID of 4.4.4.4, wins the election, and R3, with the next highest RID of 3.3.3.3, becomes the BDR.

To influence the election, you could set the various RIDs with your preferred router with the highest RID value. However, many networks choose OSPF router IDs to help identify the router easily. Instead, using the OSPF priority setting makes better sense. For instance, if an engineer preferred that R1 be the DR, the engineer could add the configuration in Example 21-5 to set R1's interface priority to 99.

Example 21-5 *Influencing DR/BDR Election Using OSPF Priority*

```
R1# configure terminal
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# interface g0/0
R1(config-if)# ip ospf priority 99
R1(config-if)# ^Z
R1#
R1# show ip ospf interface g0/0 | include Priority
    Transmit Delay is 1 sec, State DROTHER, Priority 99

R1# show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
2.2.2.2	1	2WAY/DROTHER	00:00:36	10.1.1.2	GigabitEthernet0/0
3.3.3.3	1	FULL/BDR	00:00:30	10.1.1.3	GigabitEthernet0/0
4.4.4.4	1	FULL/DR	00:00:37	10.1.1.4	GigabitEthernet0/0

```
R1# show ip ospf interface brief
```

Interface	PID	Area	IP Address/Mask	Cost	State	Nbrs	F/C
Gi0/1	1	0	10.1.11.1/24	1	DR	0/0	
Gi0/0	1	0	10.1.1.1/24	1	DROTH	2/3	

The configuration shows R1's interface priority value now as 99, and the **show ip ospf interface G0/0** command that follows confirms the setting. However, the last two commands in the example seem to show that the DR and BDR have not changed at all—and that output is indeed correct. In the example, note that the **show ip ospf neighbor** command still lists R4's state as DR, meaning R4 still acts as the DR, while the **show ip ospf interface brief** command lists R1's State (role) as DROTH.

Just to complete the process, Example 21-6 shows the results after forcing a free election (by failing the LAN switch that sits between the four routers). As expected, R1 wins and becomes DR due to its higher priority, with the other three routers tying based on priority. R4 wins between R2, R3, and R4 due to its higher RID to become the BDR.

Example 21-6 Results of a Completely New DR/BDR Election

! Not shown: LAN fails, and then recovers, causing a new OSPF Election

```
R1# show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
2.2.2.2	1	FULL/DROTHER	00:00:37	10.1.1.2	GigabitEthernet0/0
3.3.3.3	1	FULL/DROTHER	00:00:38	10.1.1.3	GigabitEthernet0/0
4.4.4.4	1	FULL/BDR	00:00:38	10.1.1.4	GigabitEthernet0/0


```
R1# show ip ospf interface brief
```

Interface	PID	Area	IP Address/Mask	Cost	State	Nbrs	F/C
Gi0/1	1	0	10.1.11.1/24	1	DR	0/0	
Gi0/0	1	0	10.1.1.1/24	1	DR	3/3	

The OSPF Point-to-Point Network Type

The other OSPF network type mentioned in the CCNA 200-301 exam topics, point-to-point, works well for data links that by their nature have just two routers on the link. For example, consider the topology in Figure 21-3, which shows router R1 with three WAN links—two Ethernet WAN links and one serial link.

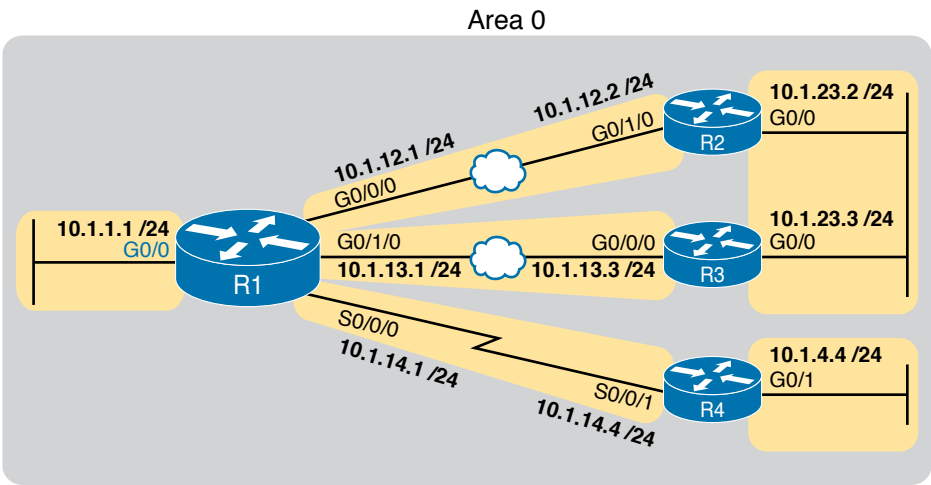


Figure 21-3 Sample OSPF Design with Serial and Ethernet WAN

First, focus on the serial link itself. To review, the jagged line represents a physical link that can at most have two devices using the link, specifically R1 and R4 in this case. The link does not support the ability to add a third router to the link. As you might guess, the data-link protocols to control a link with at most two devices can work differently than Ethernet.

For instance, the data-link protocols most often used on the link (HDLC and PPP) do not support data-link broadcasts.

Next, consider the OSPF point-to-point network type: it exists for serial links and other links that use a point-to-point topology. These links often do not support data-link broadcasts. Additionally, with only two devices on the link, using a DR/BDR is not a help, and it actually adds a little extra convergence time. Using a network type of point-to-point tells the router to not use a DR/BDR on the link.

While you may see some serial links in networks today, the CCNA and CCNP Enterprise exams make no specific mention of serial technology at this point. However, you will see other point-to-point links—like some Ethernet WAN links.

To connect the thoughts, note that all the Ethernet WAN links used in this book happen to use a point-to-point Ethernet WAN service called an Ethernet Private Wire Service or simply an Ethernet Line (E-Line). For that service, the service provider will send Ethernet frames between two devices (routers) connected to the service, but only those two devices. In other words, an E-line is a point-to-point service in concept. So while the Ethernet data-link protocol supports broadcast frames, only two devices can exist on the link, and there is no advantage to using a DR/BDR. As a result, many engineers prefer to instead use an OSPF point-to-point network type on Ethernet WAN links that in effect act as a point-to-point link.

Example 21-7 shows the configuration of router R1's G0/0/0 interface in Figure 21-3 to use OSPF network type point-to-point. R2, on the other end of the WAN link, would need the same configuration command on its matching interface.

Example 21-7 *OSPF Network Type Point-to-Point on an Ethernet WAN Interface on R1*

```
R1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# interface g0/0/0
R1(config-if)# ip ospf network point-to-point
R1(config-if)#

R1# show ip ospf interface g0/0/0
GigabitEthernet0/0/0 is up, line protocol is up
  Internet Address 10.1.12.1/24, Area 0, Attached via Interface Enable
  Process ID 1, Router ID 1.1.1.1, Network Type POINT_TO_POINT, Cost: 1
  Topology-MTID      Cost      Disabled      Shutdown      Topology Name
           0           4          no           no           Base
  Enabled by interface config, including secondary ip addresses
  Transmit Delay is 1 sec, State POINT_TO_POINT
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:01
  Supports Link-local Signaling (LLS)
  Cisco NSF helper support enabled
  IETF NSF helper support enabled
  Index 1/3/3, flood queue length 0
  Next 0x0(0)/0x0(0)/0x0(0)
```

```
Last flood scan length is 1, maximum is 3
Last flood scan time is 0 msec, maximum is 0 msec
Neighbor Count is 1, Adjacent neighbor count is 1
  Adjacent with neighbor 2.2.2.2
Suppress hello for 0 neighbor(s)
```

Note the highlighted portions of the **show** command in Example 21-6. The first two highlights note the network type. The final highlight with two lines notes that R1 has one neighbor on the interface, a neighbor with which it has become fully adjacent per the output.

Example 21-8 closes this section with a confirmation of some of those facts with two more commands. Note that the **show ip ospf neighbor** command on R1 lists router R2 (RID 2.2.2.2) with a full state, but with no DR nor BDR designation, instead listing a -. The - acts as a reminder that the link does not use a DR/BDR. The second command, **show ip ospf interface brief**, shows the state (the local router's role) as P2P, which is short for point-to-point, with a counter of 1 for the number of fully adjacent neighbors and total number of neighbors.



Example 21-8 OSPF Network Type Point-to-Point on an Ethernet WAN Interface on R1

```
R1# show ip ospf neighbor

Neighbor ID      Pri   State           Dead Time   Address      Interface
2.2.2.2          0     FULL/-          00:00:39    10.1.12.2    GigabitEthernet0/0/0
! lines omitted for brevity

R1# show ip ospf interface brief

Interface  PID   Area      IP Address/Mask    Cost   State Nbrs F/C
Gi0/0/0    1     0         10.1.12.1/24       4      P2P   1/1
! lines omitted for brevity
```

When using Ethernet WAN links that behave as a point-to-point link, consider using OSPF network type point-to-point rather than using the default broadcast type.

OSPF Neighbor Relationships

A router's OSPF configuration enables OSPF on a set of interfaces. IOS then attempts to discover other neighbors on those interfaces by sending and listening for OSPF Hello messages. However, once discovered, two routers may not become neighbors. They must have compatible values for several settings as listed in the Hellos exchanged between the two routers. This second major section of the chapter examines those reasons.

OSPF Neighbor Requirements

After an OSPF router hears a Hello from a new neighbor, the routing protocol examines the information in the Hello and compares that information with the local router's own settings. If the settings match, great. If not, the routers do not become neighbors. Because there is no formal term for all these items that a routing protocol considers, this book just calls them *neighbor requirements*. Table 21-3 lists the neighbor requirements for OSPF, with some comments about the various issues following the table.

**Table 21-3** Neighbor Requirements for OSPF

Requirement	Required for OSPF	Neighbor Missing if Incorrect
Interfaces must be in an up/up state.	Yes	Yes
Access control lists (ACL) must not filter routing protocol messages.	Yes	Yes
Interfaces must be in the same subnet.	Yes	Yes
They must pass routing protocol neighbor authentication (if configured).	Yes	Yes
Hello and hold/dead timers must match.	Yes	Yes
Router IDs (RID) must be unique.	Yes	Yes
They must be in the same area.	Yes	Yes
OSPF process must not be shut down.	Yes	Yes
Neighboring interfaces must use same MTU setting.	Yes	No
Neighboring interfaces must use same OSPF network type.	Yes	No

First, consider the meaning of the two rightmost columns. The column labeled “Required for OSPF” means that the item must be working correctly for the neighbor relationship to work correctly. Note that all the items in this column list a “yes,” meaning that all must be correct for the neighbor relationship to work correctly. The last column heading states “Neighbor Missing if Incorrect.” For items listing a “yes” in this column, if that item is configured incorrectly, the neighbor will not appear in lists of OSPF neighbors—for instance, with the **show ip ospf neighbor** command.

Next, focus on the shaded items at the top of the table. The symptom that occurs if either of these is a problem is that the **show ip ospf neighbor** command would not list the other router. For instance, the first item states that the router interfaces must be up and working. If the router interface is not working, the router cannot send any OSPF messages and discover any OSPF neighbors on that interface.

The middle section of the table (the unshaded rows) focuses on some OSPF settings. These items must be correct, but if not, they also result in the neighbor not being listed in the output of the **show ip ospf neighbor** command.

As you can see, using the **show ip ospf neighbor** command can give you a good starting point to troubleshoot OSPF on the exam and in real life. If you see the neighbor you expect to see, great! If not, the table gives you a good list to use for items to investigate.

Finally, the last section (shaded) lists a couple of OSPF settings that give a different symptom when incorrect. Again, those two items must be correct for OSPF neighbors to work. However, for these two items, when incorrect, a router can list the other router as a neighbor, but the neighbor relationship does not work properly in that the routers do not exchange LSAs as they should.

For reference, Table 21-4 relists some of the requirements from Table 21-3, along with the most useful commands with which to find the answers.

Key Topic

Table 21-4 OSPF Neighbor Requirements and the Best show/debug Commands

Requirement	Best show Command
Hello and dead timers must match.	show ip ospf interface
They must be in the same area.	show ip ospf interface brief
RIDs must be unique.	show ip ospf
They must pass any neighbor authentication.	show ip ospf interface
OSPF process must not be shut down.	show ip ospf, show ip ospf interface

The rest of this section looks at some of the items from Table 21-3 in a little more detail.

NOTE One configuration choice that people sometimes think is an issue, but is not, is the process ID as defined by the `router ospf process-id` command. Neighboring routers can use the same process ID values, or different process ID values, with no impact on whether two routers become OSPF neighbors.

Issues That Prevent Neighbor Adjacencies

The next few pages look at three of the topics from Table 21-3 for which, if a problem exists, the router does not become a neighbor (that is, the unshaded parts of the table.). To show the issues, this section uses the same topology shown earlier in Figure 21-1 but now with some incorrect configuration introduced. In other words, the configuration matches Example 21-1 that began this chapter, but with the following errors introduced:

- R2 has been configured with both LAN interfaces in area 1, whereas the other three routers' G0/0 interfaces are assigned to area 0.
- R3 is using the same RID (1.1.1.1) as R1.
- R4 has been configured with a Hello/dead timer of 5/20 on its G0/0 interface, instead of the 10/40 used (by default) on R1, R2, and R3.

Figure 21-4 shows these same problems for reference.

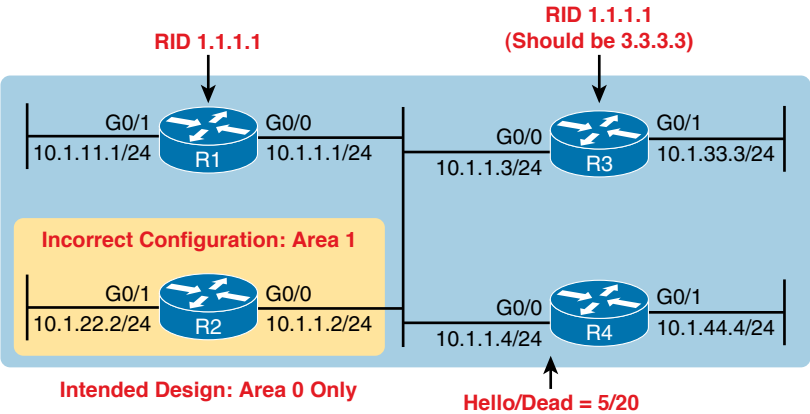


Figure 21-4 Summary of Problems That Prevent OSPF Neighbors on the Central LAN

Finding Area Mismatches

To create an area mismatch, the configuration on some router must place the interface into the wrong area per the design. As shown in Figure 21-4, router R2 was configured incorrectly, placing both its interfaces into area 1 instead of area 0. Example 21-9 shows the configuration, which uses the correct syntax (and is therefore accepted by the router) but sets the wrong area number.

Example 21-9 *Setting Area 1 on R2's Interfaces, When They Should Be in Area 0*

```
router ospf 1
  router-id 2.2.2.2
  !
  interface gigabitEthernet0/0
    ip ospf 1 area 1
  !
  interface gigabitEthernet0/1
    ip ospf 1 area 1
```

With an area mismatch error, the **show ip ospf neighbor** command will not list the neighbor. Because you see nothing in the OSPF neighbor table, to troubleshoot this problem, you need to find the area configuration on each interface on potentially neighboring routers. To do so:

- Check the output of **show running-config** to look for
 - **ip ospf process-id area area-number** interface subcommands
 - **network** commands in OSPF configuration mode
- Use the **show ip ospf interface [brief]** command to list the area number

21

Finding Duplicate OSPF Router IDs

Next, Example 21-10 shows R1 and R3 both trying to use RID 1.1.1.1. Interestingly, both routers automatically generate a log message for the duplicate OSPF RID problem between R1 and R3; the end of Example 21-10 shows one such message. For the exams, just use the **show ip ospf** commands on both R3 and R1 to easily list the RID on each router, noting that they both use the same value.

Example 21-10 *Comparing OSPF Router IDs on R1 and R3*

```
! Next, on R3: R3 lists the RID of 1.1.1.1
!
R3# show ip ospf
Routing Process "ospf 1" with ID 1.1.1.1
Start time: 00:00:37.136, Time elapsed: 02:20:37.200
! lines omitted for brevity

! Back to R1: R1 also uses RID 1.1.1.1
R1# show ip ospf
Routing Process "ospf 1" with ID 1.1.1.1
```

```

Start time: 00:01:51.864, Time elapsed: 12:13:50.904
! lines omitted for brevity

*May 29 00:01:25.679: %OSPF-4-DUP_RTRID_NBR: OSPF detected duplicate router-id
1.1.1.1 from 10.1.1.3 on interface GigabitEthernet0/0

```

First, focus on the problem: the duplicate RIDs. The first line of the **show ip ospf** command on the two routers quickly shows the duplicate use of 1.1.1.1. To solve the problem, assuming R1 should use 1.1.1.1 and R3 should use another RID (maybe 3.3.3.3), change the RID on R3 and restart the OSPF process. To do so, use the **router-id 3.3.3.3** OSPF subcommand and use the EXEC mode command **clear ip ospf process**. (OSPF will not begin using a new RID value until the process restarts, either via command or reload.)

Finding OSPF Hello and Dead Timer Mismatches

First, as a reminder from chapters past:

- **Hello interval/timer:** The per-interface timer that tells a router how often to send OSPF Hello messages on an interface.
- **Dead interval/timer:** The per-interface timer that tells the router how long to wait without having received a Hello from a neighbor before believing that neighbor has failed. (Defaults to four times the Hello timer.)

Next, consider the problem created on R4, with the configuration of a different Hello timer and dead timer (5 and 20, respectively) as compared with the default settings on R1, R2, and R3 (10 and 40, respectively). A Hello or Dead interval mismatch prevents R4 from becoming neighbors with any of the other three OSPF routers. Routers list their Hello and Dead interval settings in their Hello messages and choose to not become neighbors if the values do not match. As a result, none of the routers become neighbors with router R4 in this case.

Example 21-11 shows the easiest way to find the mismatch using the **show ip ospf interface** command on both R1 and R4. This command lists the Hello and dead timers for each interface, as highlighted in the example. Note that R1 uses 10 and 40 (Hello and dead), whereas R4 uses 5 and 20.

Example 21-11 Finding Mismatched Hello/Dead Timers

```

R1# show ip ospf interface G0/0
GigabitEthernet0/0 is up, line protocol is up
  Internet Address 10.1.1.1/24, Area 0, Attached via Network Statement
  Process ID 1, Router ID 1.1.1.1, Network Type BROADCAST, Cost: 1
  Topology-MTID   Cost   Disabled   Shutdown   Topology Name
                0       1         no         no         Base
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 1.1.1.1, Interface address 10.1.1.1
  No backup designated router on this network

```

```

Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
! lines omitted for brevity
! Moving on to R4 next
!
R4# show ip ospf interface Gi0/0
GigabitEthernet0/0 is up, line protocol is up
  Internet Address 10.1.1.4/24, Area 0, Attached via Network Statement
  Process ID 4, Router ID 10.1.44.4, Network Type BROADCAST, Cost: 1
  Topology-MTID  Cost  Disabled  Shutdown  Topology Name
                0      1         no       no       Base
  Transmit Delay is 1 sec, State DR, Priority 1
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 10.1.44.4, Interface address 10.1.1.4
  No backup designated router on this network
  Timer intervals configured, Hello 5, Dead 20, Wait 20, Retransmit 5
! lines omitted for brevity

```

Shutting Down the OSPF Process

Similar to administratively disabling and enabling an interface, IOS also allows the OSPFv2 routing protocol process to be disabled and enabled with the **shutdown** and **no shutdown** router mode subcommands, respectively. When a routing protocol process is shut down, IOS does the following:

- Brings down all neighbor relationships and clears the OSPF neighbor table
- Clears the LSDB
- Clears the IP routing table of any OSPF-learned routes

At the same time, shutting down OSPF does retain some important details about OSPF, in particular:

- IOS retains all OSPF configuration.
- IOS still lists all OSPF-enabled interfaces in the OSPF interface list (**show ip ospf interface**) but in a DOWN state.

Basically, shutting down the OSPF routing protocol process gives the network engineer a way to stop using the routing protocol on that router without having to remove all the configuration. Once shut down, the **show ip ospf interface [brief]** command should still list some output, as will the **show ip ospf** command, but the rest of the commands will list nothing.

Example 21-12 shows an example on Router R5, as shown in Figure 21-5. R5 is a different router than the one used in earlier examples, but it begins the example with two OSPF neighbors, R2 and R3, with router IDs 2.2.2.2 and 3.3.3.3. The example shows the OSPF process being shut down, the neighbors failing, and those two key OSPF **show** commands: **show ip ospf neighbor** and **show ip ospf interface brief**.

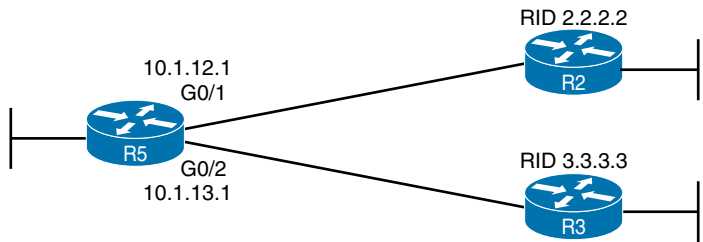


Figure 21-5 Example Network to Demonstrate OSPF Process Shutdown

Example 21-12 Shutting Down an OSPF Process, and the Resulting Neighbor States

```
R5# show ip ospf neighbor
Neighbor ID    Pri   State       Dead Time   Address        Interface
2.2.2.2        1     FULL/DR     00:00:35    10.1.12.2      GigabitEthernet0/1
3.3.3.3        1     FULL/DR     00:00:33    10.1.13.3      GigabitEthernet0/2

R5# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R5(config)# router ospf 1
R5(config-router)# shutdown
R5(config-router)# ^Z

*Mar 23 12:43:30.634: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on GigabitEthernet0/1
  from FULL to DOWN, Neighbor Down: Interface down or detached
*Mar 23 12:43:30.635: %OSPF-5-ADJCHG: Process 1, Nbr 3.3.3.3 on GigabitEthernet0/2
  from FULL to DOWN, Neighbor Down: Interface down or detached

R5# show ip ospf interface brief
Interface      PID     Area        IP Address/Mask    Cost   State Nbrs F/C
Gi0/1          1       0           10.1.12.1/24       1      DOWN  0/0
Gi0/2          1       0           10.1.13.1/24       1      DOWN  0/0

R5# show ip ospf
Routing Process "ospf 1" with ID 5.5.5.5
Start time: 5d23h, Time elapsed: 1d04h
Routing Process is shutdown
! lines omitted for brevity

R5# show ip ospf neighbor
R5#

R5# show ip ospf database
OSPF Router with ID (3.3.3.3) (Process ID 1)
R5#
```

First, before the **shutdown**, the **show ip ospf neighbor** command lists two neighbors. After the **shutdown**, the same command lists no neighbors at all. Second, the **show ip ospf interface brief** command does list the interfaces on which OSPF is enabled, on the local router's own IP addresses. However, it lists a state of **DOWN**, which is a reference to the local router's state. Also, note that the **show ip ospf** command positively states that the

OSPF process is in a shutdown state, while the **show ip ospf database** command output lists only a heading line, with no LSAs.

Issues That Allow Adjacencies but Prevent IP Routes

The last two issues to discuss in this section have a symptom in which the **show ip ospf neighbor** command does list a neighbor, but some other problem exists that prevents the eventual addition of OSPF routes to the routing table. The two issues: a mismatched MTU setting and a mismatched OSPF network type.

Mismatched MTU Settings

The MTU size defines a per-interface setting used by the router for its Layer 3 forwarding logic, defining the largest network layer packet that the router will forward out each interface. For instance, the IPv4 MTU size of an interface defines the maximum size IPv4 packet that the router can forward out an interface.

Routers often use a default MTU size of 1500 bytes, with the ability to set the value as well. The **ip mtu size** interface subcommand defines the IPv4 MTU setting, and the **ipv6 mtu size** command sets the equivalent for IPv6 packets.

In an odd twist, two OSPFv2 routers can actually become OSPF neighbors, be listed in the output of the **show ip ospf neighbor** command, and reach 2-way state, even if they happen to use different IPv4 MTU settings on their interfaces. However, they fail to exchange their LSDBs. Eventually, after trying and failing to exchange their LSDBs, the neighbor relationship also fails. So also keep a watch for MTU mismatches, although they may be unusual and obscure, by looking at the running-config and by using the **show interfaces** command (which lists the IP MTU).

Mismatched OSPF Network Types

Earlier in this chapter you read about the OSPF broadcast network type, which uses a DR/BDR, and the OSPF point-to-point network type, which does not. Interestingly, if you misconfigure network type settings such that one router uses broadcast, and the other uses point-to-point, the following occurs:



- The two routers become fully adjacent neighbors (that is, they reach a full state).
- They exchange their LSDBs.
- They do not add IP routes to the IP routing table.

The reason for not adding the routes has to do with the details of LSAs and how the use of a DR (or not) changes those LSAs. Basically, the two routers expect different details in the LSAs, and the SPF algorithm notices those differences and cannot trust the LSAs because of those differences.

For instance, earlier in Example 21-7, the configuration showed router R1 using network type point-to-point on its G0/0/0 interface, with the expectation that router R2 would also use point-to-point on its matching G0/1/0 interface. Example 21-13 shows some of the results if the engineer neglected to configure R2, leaving it with the default setting of broadcast.

Example 21-13 *Shutting Down an OSPF Process, and the Resulting Neighbor States*

*Apr 10 16:31:01.951: %OSPF-4-NET_TYPE_MISMATCH: Received Hello from 2.2.2.2 on GigabitEthernet0/0/0 indicating a potential network type mismatch					
R1# show ip ospf neighbor					
Neighbor ID	Pri	State	Dead Time	Address	Interface
2.2.2.2	0	FULL/ -	00:00:38	10.1.12.2	GigabitEthernet0/0/0
R1#					
R2# show ip ospf neighbor					
Neighbor ID	Pri	State	Dead Time	Address	Interface
1.1.1.1	1	FULL/BDR	00:00:30	10.1.12.1	GigabitEthernet0/1/0

As you can see, both routers list the other as an OSPF neighbor in the full state. However, R1, with network type point-to-point, does not list a DR or BDR role in the output, while R2 does, which is one clue for this type of problem. The other comes with noticing that the expected routes are not in the IP routing table.

Chapter Review

One key to doing well on the exams is to perform repetitive spaced review sessions. Review this chapter’s material using either the tools in the book or interactive tools for the same material found on the book’s companion website. Refer to the “Your Study Plan” element for more details. Table 21-5 outlines the key review elements and where you can find them. To better track your study progress, record when you completed these activities in the second column.

Table 21-5 Chapter Review Tracking

Review Element	Review Date(s)	Resource Used:
Review key topics		Book, website
Review command tables		Book
Review memory tables		Website
Watch video		Website

Review All the Key Topics



Table 21-6 Key Topics for Chapter 21

Key Topic Element	Description	Page Number
Table 21-2	Two OSPF Network Types and Key Behaviors	501
Example 21-3	OSPF interfaces, local roles, and neighbor counts	503
List	Rules for electing an OSPF DR/BDR	505
Example 21-8	Evidences of OSPF network type point-to-point	508

Key Topic Element	Description	Page Number
Table 21-3	Neighbor requirements for OSPF	509
Table 21-4	Useful commands to discover OSPF neighbor issues	510
List	Symptoms of an OSPF network type mismatch	515

Command References

Tables 21-7 and 21-8 list configuration and verification commands used in this chapter. As an easy review exercise, cover the left column in a table, read the right column, and try to recall the command without looking. Then repeat the exercise, covering the right column, and try to recall what the command does.

Table 21-7 Chapter 21 Configuration Command Reference

Command	Description
ip ospf hello-interval <i>seconds</i>	Interface subcommand that sets the interval for periodic Hellos
ip ospf dead-interval <i>number</i>	Interface subcommand that sets the OSPF dead timer
passive-interface <i>type number</i>	Router subcommand, for both OSPF and EIGRP, that tells the routing protocol to stop sending Hellos and stop trying to discover neighbors on that interface
ip ospf priority <i>value</i>	Interface subcommand that sets the OSPF priority, used when electing a new DR or BDR
ip ospf network {broadcast point-to-point}	Interface subcommand used to set the OSPF network type on the interface
[no] shutdown	An OSPF configuration mode command to disable (shutdown) or enable (no shutdown) the OSPF process

Table 21-8 Chapter 21 **show** Command Reference

Command	Description
show ip protocols	Shows routing protocol parameters and current timer values, including an effective copy of the routing protocols' network commands and a list of passive interfaces
show ip ospf interface brief	Lists the interfaces on which the OSPF protocol is enabled (based on the network commands), including passive interfaces
show ip ospf interface [<i>type number</i>]	Lists detailed OSPF settings for all interfaces, or the listed interface, including Hello and dead timers and OSPF area
show ip ospf neighbor	Lists neighbors and current status with neighbors, per interface
show ip ospf	Lists a group of messages about the OSPF process itself, listing the OSPF Router ID in the first line
show interfaces	Lists a long set of messages, per interface, that lists configuration, state, and counter information



Part VI Review

Keep track of your part review progress with the checklist in Table P6-1. Details about each task follow the table.

Table P6-1 Part VI Part Review Checklist

Activity	1st Date Completed	2nd Date Completed
Repeat All DIKTA Questions		
Answer Part Review Questions		
Review Key Topics		
Do Labs		
Watch Videos		

Repeat All DIKTA Questions

For this task, answer the “Do I Know This Already?” questions again for the chapters in this part of the book using the PTP software. See the section “How to View Only DIKTA Questions by Chapter or Part” in the Introduction to this book to learn how to make the PTP software show you DIKTA questions for this part only.

Answer Part Review Questions

For this task, answer the Part Review questions for this part of the book using the PTP software. See the section “How to View Part Review Questions” in the Introduction to this book to learn how to make the PTP software show you Part Review questions for this part only.

Review Key Topics

Review all Key Topics in all chapters in this part, either by browsing the chapters or by using the Key Topics application on the companion website.

Do Labs

Depending on your chosen lab tool, here are some suggestions for what to do in lab:

Pearson Network Simulator: If you use the full Pearson ICND1 or CCNA simulator, focus more on the configuration scenario and troubleshooting scenario labs associated with the topics in this part of the book. These types of labs include a larger set of topics and work well as Part Review activities. (See the Introduction for some details about how to find which labs are about topics in this part of the book.)

Blog: Config Labs: The author’s blog includes a series of configuration-focused labs that you can do on paper, each in 10–15 minutes. Review and perform the labs for this part of the book, as found at <http://blog.certskills.com>. Then navigate to the Hands-on Config labs.

Other: If using other lab tools, here are a few suggestions: Make sure to experiment heavily with VLAN configuration and VLAN trunking configuration.

Watch Videos

Chapter 21 recommends one video from the companion website about troubleshooting OSPF neighbors. Take a few minutes to watch the video if you haven't done so already.



So far, this book has mostly ignored IP version 6 (IPv6). This part reverses the trend, collecting all the specific IPv6 topics into four chapters.

The chapters in Part VII walk you through the same topics discussed throughout this book for IPv4, often using IPv4 as a point of comparison. Certainly, many details differ when comparing IPv4 and IPv6. However, many core concepts about IP addressing, subnetting, routing, and routing protocols remain the same. The chapters in this part build on those foundational concepts, adding the specific details about how IPv6 forwards IPv6 packets from one host to another.

Part VII

IP Version 6

Chapter 22: Fundamentals of IP Version 6

Chapter 23: IPv6 Addressing and Subnetting

Chapter 24: Implementing IPv6 Addressing on Routers

Chapter 25: Implementing IPv6 Routing

Part VII Review

Fundamentals of IP Version 6

This chapter covers the following exam topics:

1.0 Network Fundamentals

1.8 Configure and verify IPv6 addressing and prefix

IPv4 has been a solid and highly useful part of the growth of TCP/IP and the Internet. For most of the long history of the Internet, and for most corporate networks that use TCP/IP, IPv4 is the core protocol that defines addressing and routing. However, even though IPv4 has many great qualities, it does have some shortcomings, creating the need for a replacement protocol: IP version 6 (IPv6).

IPv6 defines the same general functions as IPv4, but with different methods of implementing those functions. For example, both IPv4 and IPv6 define addressing, the concepts of subnetting larger groups of addresses into smaller groups, headers used to create an IPv4 or IPv6 packet, and the rules for routing those packets. At the same time, IPv6 handles the details differently; for example, using a 128-bit IPv6 address rather than the 32-bit IPv4 address.

This chapter focuses on the core network layer functions of addressing and routing. The first section of this chapter looks at the big concepts, while the second section looks at the specifics of how to write and type IPv6 addresses.

“Do I Know This Already?” Quiz

Take the quiz (either here or use the PTP software) if you want to use the score to help you decide how much time to spend on this chapter. The letter answers are listed at the bottom of the page following the quiz. Appendix C, found both at the end of the book as well as on the companion website, includes both the answers and explanations. You can also find both answers and explanations in the PTP testing software.

Table 22-1 “Do I Know This Already?” Foundation Topics Section-to-Question Mapping

Foundation Topics Section	Questions
Introduction to IPv6	1–2
IPv6 Addressing Formats and Conventions	3–6

1. Which of the following was a short-term solution to the IPv4 address exhaustion problem?
 - a. IP version 6
 - b. IP version 5
 - c. NAT/PAT
 - d. ARP

2. A router receives an Ethernet frame that holds an IPv6 packet. The router then makes a decision to route the packet out a serial link. Which of the following statements is true about how a router forwards an IPv6 packet?
- a. The router discards the Ethernet data-link header and trailer of the received frame.
 - b. The router makes the forwarding decision based on the packet's source IPv6 address.
 - c. The router keeps the Ethernet header, encapsulating the entire frame inside a new IPv6 packet before sending it over the serial link.
 - d. The router uses the IPv4 routing table when choosing where to forward the packet.
3. Which of the following is the shortest valid abbreviation for FE80:0000:0000:0100:0000:0000:0000:0123?
- a. FE80::100::123
 - b. FE8::1::123
 - c. FE80:100:0:0:0:123:4567
 - d. FE80:0:0:100::123
4. Which of the following is the shortest valid abbreviation for 2000:0300:0040:0005:6000:0700:0080:0009?
- a. 2:3:4:5:6:7:8:9
 - b. 2000:300:40:5:6000:700:80:9
 - c. 2000:300:4:5:6000:700:8:9
 - d. 2000:3:4:5:6:7:8:9
5. Which of the following is the unabbreviated version of IPv6 address 2001:DB8::200:28?
- a. 2001:0DB8:0000:0000:0000:0000:0200:0028
 - b. 2001:0DB8::0200:0028
 - c. 2001:0DB8:0:0:0:0:0200:0028
 - d. 2001:0DB8:0000:0000:0000:0000:200:0028
6. Which of the following is the prefix for address 2000:0000:0000:0005:6000:0700:0080:0009, assuming a mask of /64?
- a. 2000::5::/64
 - b. 2000::5:0:0:0:0/64
 - c. 2000:0:0:5::/64
 - d. 2000:0:0:5:0:0:0:0/64

Foundation Topics

Introduction to IPv6

IP version 6 (IPv6) serves as the replacement protocol for IP version 4 (IPv4).

Unfortunately, that one bold statement creates more questions than it answers. Why does IPv4 need to be replaced? If IPv4 needs to be replaced, when will that happen—and will it happen quickly? What exactly happens when a company or the Internet replaces IPv4 with IPv6? And the list goes on.

While this introductory chapter cannot get into every detail of why IPv4 needs to eventually be replaced by IPv6, the clearest and most obvious reason for migrating TCP/IP networks to use IPv6 is growth. IPv4 uses a 32-bit address, which totals to a few billion addresses. Interestingly, that seemingly large number of addresses is too small. IPv6 increases the address to 128 bits in length. For perspective, IPv6 supplies more than 10,000,000,000,000,000,000,000,000 times as many addresses as IPv4.

The fact that IPv6 uses a different size address field, with some different addressing rules, means that many other protocols and functions change as well. For example, IPv4 routing—in other words, the packet-forwarding process—relies on an understanding of IPv4 addresses. To support IPv6 routing, routers must understand IPv6 addresses and routing. To dynamically learn routes for IPv6 subnets, routing protocols must support these different IPv6 addressing rules, including rules about how IPv6 creates subnets. As a result, the migration from IPv4 to IPv6 is much more than changing one protocol (IP), but it impacts many protocols.

This first section of the chapter discusses some of the reasons for the change from IPv4 to IPv6, along with the protocols that must change as a result.

The Historical Reasons for IPv6

In the last 40+ years, the Internet has gone from its infancy to being a huge influence in the world. It first grew through research at universities, from the ARPANET beginnings of the Internet in the late 1960s into the 1970s. The Internet kept growing fast in the 1980s, with the Internet's fast growth still primarily driven by research and the universities that joined in that research. By the early 1990s, the Internet began to transform to allow commerce, allowing people to sell services and products over the Internet, which drove yet another steep spike upward in the growth of the Internet. Eventually, fixed Internet access (primarily through dial, digital subscriber line [DSL], and cable) became common, followed by the pervasive use of the Internet from mobile devices like smartphones. Figure 22-1 shows some of these major milestones with general dates.

The incredible growth of the Internet over a fairly long time created a big problem for public IPv4 addresses: the world was running out of addresses. For instance, in 2011, IANA allocated the final /8 address blocks (the same size as a Class A network), allocating one final /8 block to each of the five Regional Internet Registries (RIR). At that point, RIRs could no

Answers to the “Do I Know This Already?” quiz:

1 C 2 A 3 D 4 B 5 A 6 C

longer receive new allocations of public addresses from IANA to then turn around and assign smaller address blocks to companies or ISPs.

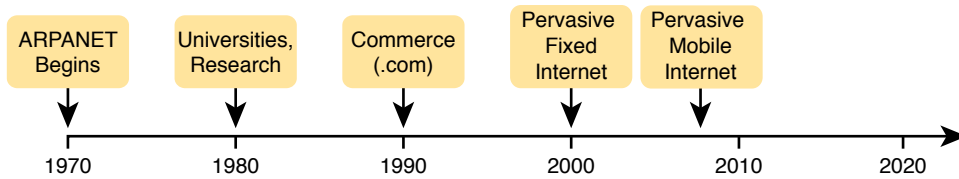


Figure 22-1 *Some Major Events in the Growth of the Internet*

At that point in 2011, each of the five RIRs still had public addresses to allocate or assign. However, that same year, APNIC (Asia Pacific) became the first RIR to exhaust its available IPv4 address allocation. In late 2015, ARIN (North America) announced that it had exhausted its supply. When we were revising this chapter in 2019, IANA considered all RIRs except AFRINIC to have exhausted their supply of IPv4 addresses, with AFRINIC expected to run out of IPv4 address during the year 2019.

These events are significant in that the day has finally come in which new companies can attempt to connect to the Internet, but they can no longer simply use IPv4, ignoring IPv6. Their only option will be IPv6 because IPv4 has no public addresses left.

NOTE You can track ARIN's progress through this interesting transition in the history of the Internet at its IPv4 address depletion site: <http://teamarin.net/category/ipv4-depletion/>. You can also see a summary report at <http://ipv4.potaroo.net>.

Even though the press has rightfully made a big deal about running out of IPv4 addresses, those who care about the Internet knew about this potential problem since the late 1980s. The problem, generally called the *IPv4 address exhaustion* problem, could literally have caused the huge growth of the Internet in the 1990s to have come to a screeching halt! Something had to be done.

The IETF came up with several short-term solutions to make IPv4 addresses last longer, and one long-term solution: IPv6. However, several other tools like Network Address Translation (NAT) and classless interdomain routing (CIDR) helped extend IPv4's life another couple of decades. IPv6 creates a more permanent and long-lasting solution, replacing IPv4, with a new IPv6 header and new IPv6 addresses. The address size supports a huge number of addresses, solving the address shortage problem for generations (we hope). Figure 22-2 shows some of the major address exhaustion timing.

The rest of this first section examines IPv6, comparing it to IPv4, focusing on the common features of the two protocols. In particular, this section compares the protocols (including addresses), routing, routing protocols, and miscellaneous other related topics.

NOTE You might wonder why the next version of IP is not called IP version 5. There was an earlier effort to create a new version of IP, and it was numbered version 5. IPv5 did not progress to the standards stage. However, to prevent any issues, because version 5 had been used in some documents, the next effort to update IP was numbered as version 6.

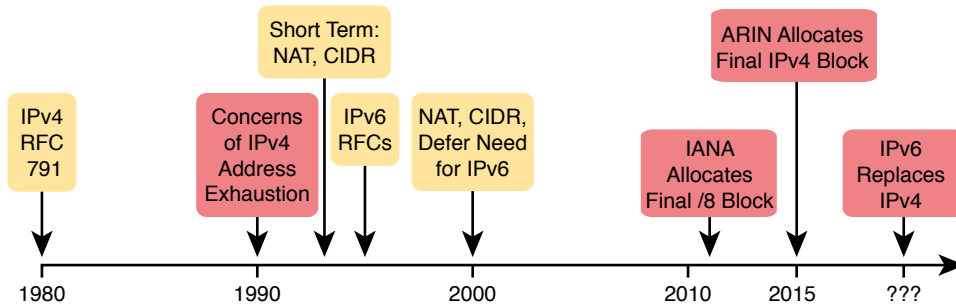


Figure 22-2 *Timeline for IPv4 Address Exhaustion and Short-/Long-Term Solutions*

The IPv6 Protocols

The primary purpose of the core IPv6 protocol mirrors the same purpose of the IPv4 protocol. That core IPv6 protocol, as defined in RFC 2460, defines a packet concept, addresses for those packets, and the role of hosts and routers. These rules allow the devices to forward packets sourced by hosts, through multiple routers, so that they arrive at the correct destination host. (IPv4 defines those same concepts for IPv4 back in RFC 791.)

However, because IPv6 impacts so many other functions in a TCP/IP network, many more RFCs must define details of IPv6. Some other RFCs define how to migrate from IPv4 to IPv6. Others define new versions of familiar protocols or replace old protocols with new ones. For example:

Older OSPF Version 2 Upgraded to OSPF Version 3: The older Open Shortest Path First (OSPF) version 2 works for IPv4, but not for IPv6, so a newer version, OSPF version 3, was created to support IPv6. (Note: OSPFv3 was later upgraded to support advertising both IPv4 and IPv6 routes.)

ICMP Upgraded to ICMP Version 6: Internet Control Message Protocol (ICMP) worked well with IPv4 but needed to be changed to support IPv6. The new name is ICMPv6.

ARP Replaced by Neighbor Discovery Protocol: For IPv4, Address Resolution Protocol (ARP) discovers the MAC address used by neighbors. IPv6 replaces ARP with a more general Neighbor Discovery Protocol (NDP).

NOTE If you go to any website that lists the RFCs, like <http://www.rfc-editor.org>, you can find almost 300 RFCs that have IPv6 in the title.

Although the term IPv6, when used broadly, includes many protocols, the one specific protocol called IPv6 defines the new 128-bit IPv6 address. Of course, writing these addresses in binary would be a problem—they probably would not even fit on the width of a piece of paper! IPv6 defines a shorter hexadecimal format, requiring at most 32 hexadecimal digits (one hex digit per 4 bits), with methods to abbreviate the hexadecimal addresses as well.

For example, all of the following are IPv6 addresses, each with 32 or fewer hex digits:

```
2345:1111:2222:3333:4444:5555:6666:AAAA
```

```
2000:1:2:3:4:5:6:A
```

```
FE80::1
```


The upcoming section “IPv6 Addressing Formats and Conventions” discusses the specifics of how to represent IPv6 addresses, including how to legally abbreviate the hex address values.

Like IPv4, IPv6 defines a header, with places to hold both the source and destination address fields. Compared to IPv4, the IPv6 header does make some other changes besides simply making the address fields larger. However, even though the IPv6 header is larger than an IPv4 header, the IPv6 header is actually simpler (on purpose), to reduce the work done each time a router must route an IPv6 packet. Figure 22-3 shows the required 40-byte part of the IPv6 header.

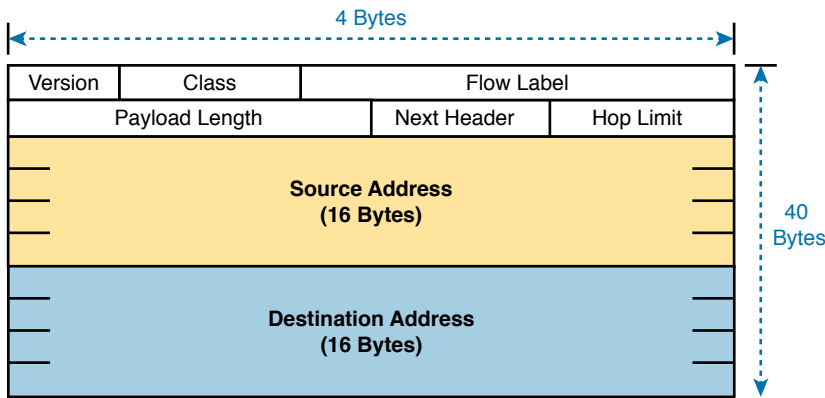


Figure 22-3 *IPv6 Header*

IPv6 Routing

As with many functions of IPv6, IPv6 routing looks just like IPv4 routing from a general perspective, with the differences being clear only once you look at the specifics. Keeping the discussion general for now, IPv6 uses these ideas the same way as IPv4:

Key Topic

- To be able to build and send IPv6 packets out an interface, end-user devices need an IPv6 address on that interface.
- End-user hosts need to know the IPv6 address of a default router, to which the host sends IPv6 packets if the host is in a different subnet.
- IPv6 routers de-encapsulate and re-encapsulate each IPv6 packet when routing the packet.
- IPv6 routers make routing decisions by comparing the IPv6 packet's destination address to the router's IPv6 routing table; the matched route lists directions of where to send the IPv6 packet next.

NOTE You could take the preceding list and replace every instance of IPv6 with IPv4, and all the statements would be true of IPv4 as well.

While the list shows some concepts that should be familiar from IPv4, the next few figures show the concepts with an example. First, Figure 22-4 shows a few settings on a host.

The host (PC1) has an address of 2345::1. PC1 also knows its default gateway of 2345::2. (Both values are valid abbreviations for real IPv6 addresses.) To send an IPv6 packet to host PC2, on another IPv6 subnet, PC1 creates an IPv6 packet and sends it to R1, PC1's default gateway.

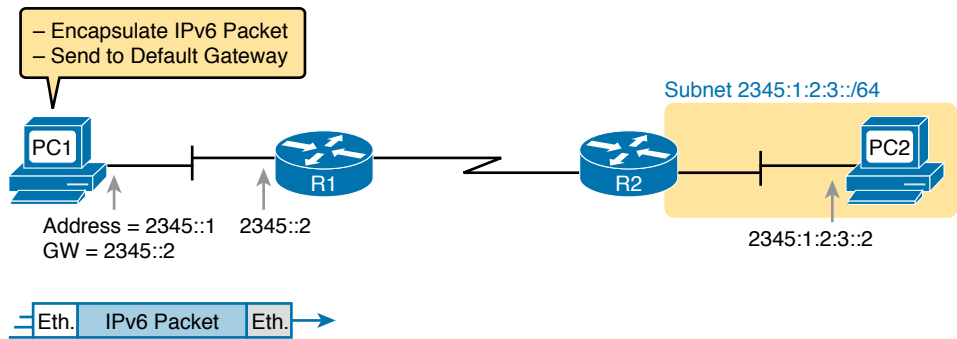


Figure 22-4 IPv6 Host Building and Sending an IPv6 Packet

The router (R1) has many small tasks to do when forwarding this IPv6 packet, but for now, focus on the work R1 does related to encapsulation. As seen in Step 1 of Figure 22-5, R1 receives the incoming data-link frame and extracts (de-encapsulates) the IPv6 packet from inside the frame, discarding the original data-link header and trailer. At Step 2, once R1 knows to forward the IPv6 packet to R2, R1 adds a correct outgoing data-link header and trailer to the IPv6 packet, encapsulating the IPv6 packet.

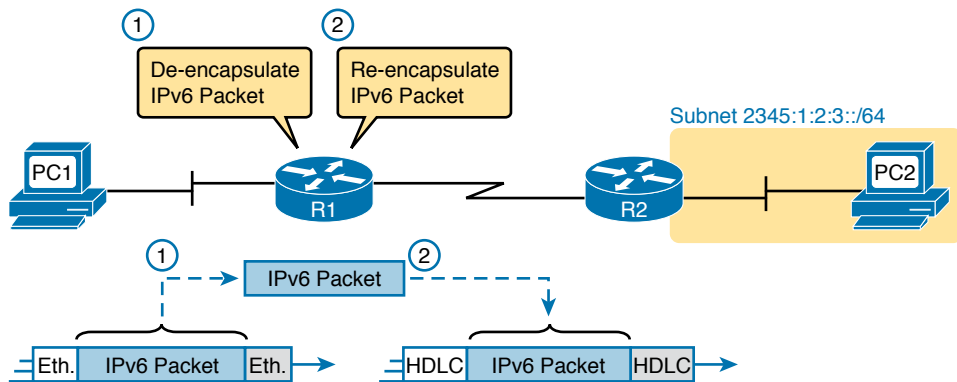


Figure 22-5 IPv6 Router Performing Routine Encapsulation Tasks When Routing IPv6

When a router like R1 de-encapsulates the packet from the data-link frame, it must also decide what type of packet sits inside the frame. To do so, the router must look at a protocol type field in the data-link header, which identifies the type of packet inside the data-link frame. Today, most data-link frames carry either an IPv4 packet or an IPv6 packet.

To route an IPv6 packet, a router must use its IPv6 routing table instead of the IPv4 routing table. The router must look at the packet's destination IPv6 address and compare that address to the router's current IPv6 routing table. The router uses the forwarding instructions in the matched IPv6 route to forward the IPv6 packet. Figure 22-6 shows the overall process.

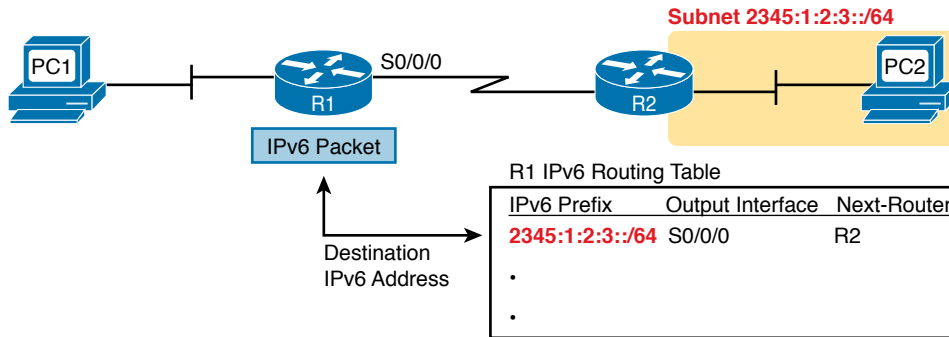


Figure 22-6 Comparing an IPv6 Packet to R1's IPv6 Routing Table

Note that again, the process works like IPv4, except that the IPv6 packet lists IPv6 addresses, and the IPv6 routing table lists routing information for IPv6 subnets (called prefixes).

Finally, in most enterprise networks, the routers will route both IPv4 and IPv6 packets at the same time. That is, your company will not decide to adopt IPv6, and then late one week-end night turn off all IPv4 and enable IPv6 on every device. Instead, IPv6 allows for a slow migration, during which some or all routers forward both IPv4 and IPv6 packets. (The migration strategy of running both IPv4 and IPv6 is called *dual stack*.) All you have to do is configure the router to route IPv6 packets, in addition to the existing configuration for routing IPv4 packets.

IPv6 Routing Protocols

IPv6 routers need to learn routes for all the possible IPv6 prefixes (subnets). Just like with IPv4, IPv6 routers use routing protocols, with familiar names, and generally speaking, with familiar functions.

None of the IPv4 routing protocols could be used to advertise IPv6 routes originally. They all required some kind of update to add messages, protocols, and rules to support IPv6. Over time, Routing Information Protocol (RIP), Open Shortest Path First (OSPF), Enhanced Interior Gateway Routing Protocol (EIGRP), and Border Gateway Protocol (BGP) were all updated to support IPv6. Table 22-2 lists the names of these routing protocols, with a few comments.

Table 22-2 IPv6 Routing Protocols

Routing Protocol	Defined By	Notes
RIPng (RIP next generation)	RFC	The “next generation” is a reference to a TV series, <i>Star Trek: the Next Generation</i> .
OSPFv3 (OSPF version 3)	RFC	The OSPF you have worked with for IPv4 is actually OSPF version 2, so the new version for IPv6 is OSPFv3.
EIGRPv6 (EIGRP for IPv6)	Cisco	Cisco owns the rights to the EIGRP protocol, but Cisco also now publishes EIGRP as an informational RFC.
MP BGP-4 (Multiprotocol BGP version 4)	RFC	BGP version 4 was created to be highly extendable; IPv6 support was added to BGP version 4 through one such enhancement, MP BGP-4.

In addition, these routing protocols also follow the same interior gateway protocol (IGP) and exterior gateway protocol (EGP) conventions as their IPv4 cousins. RIPng, EIGRPv6, and OSPFv3 act as interior gateway protocols, advertising IPv6 routes inside an enterprise.

As you can see from this introduction, IPv6 uses many of the same big ideas as IPv4. Both define headers with a source and destination address. Both define the routing of packets, with the routing process discarding old data-link headers and trailers when forwarding the packets. And routers use the same general process to make a routing decision, comparing the packet's destination IP address to the routing table.

The big differences between IPv4 and IPv6 revolve around the bigger IPv6 addresses. The next topic begins looking at the specifics of these IPv6 addresses.

IPv6 Addressing Formats and Conventions

The CCNA exam requires some fundamental skills in working with IPv4 addresses. For example, you need to be able to interpret IPv4 addresses, like 172.21.73.14. You need to be able to work with prefix-style masks, like /25, and interpret what that means when used with a particular IPv4 address. And you need to be able to take an address and mask, like 172.21.73.14/25, and find the subnet ID.

This second major section of this chapter discusses these same ideas for IPv6 addresses. In particular, this section looks at

- How to write and interpret unabbreviated 32-digit IPv6 addresses
- How to abbreviate IPv6 addresses and how to interpret abbreviated addresses
- How to interpret the IPv6 prefix length mask
- How to find the IPv6 prefix (subnet ID), based on an address and prefix length mask

The biggest challenge with these tasks lies in the sheer size of the numbers. Thankfully, the math to find the subnet ID—often a challenge for IPv4—is easier for IPv6, at least to the depth discussed in this book.

Representing Full (Unabbreviated) IPv6 Addresses

IPv6 uses a convenient hexadecimal (hex) format for addresses. To make it more readable, IPv6 uses a format with eight sets of four hex digits, with each set of four digits separated by a colon. For example:

```
2340:1111:AAAA:0001:1234:5678:9ABC:1234
```

NOTE For convenience, the author uses the term *quartet* for one set of four hex digits, with eight quartets in each IPv6 address. Note that the IPv6 RFCs do not use the term *quartet*.

IPv6 addresses also have a binary format as well, but thankfully, most of the time you do not need to look at the binary version of the addresses. However, in those cases, converting from hex to binary is relatively easy. Just change each hex digit to the equivalent 4-bit value listed in Table 22-3.

Table 22-3 Hexadecimal/Binary Conversion Chart

Hex	Binary	Hex	Binary
0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	F	1111

Abbreviating and Expanding IPv6 Addresses

IPv6 also defines ways to abbreviate or shorten how you write or type an IPv6 address. Why? Although using a 32-digit hex number works much better than working with a 128-bit binary number, 32 hex digits are still a lot of digits to remember, recognize in command output, and type on a command line. The IPv6 address abbreviation rules let you shorten these numbers.

Computers and routers typically use the shortest abbreviation, even if you type all 32 hex digits of the address. So even if you would prefer to use the longer unabbreviated version of the IPv6 address, you need to be ready to interpret the meaning of an abbreviated IPv6 address as listed by a router or host. This section first looks at abbreviating addresses and then at expanding addresses.

Abbreviating IPv6 Addresses

Two basic rules let you, or any computer, shorten or abbreviate an IPv6 address:

Key Topic

1. Inside each quartet of four hex digits, remove the leading 0s (0s on the left side of the quartet) in the three positions on the left. (Note: at this step, a quartet of 0000 will leave a single 0.)
2. Find any string of two or more consecutive quartets of all hex 0s, and replace that set of quartets with a double colon (::). The :: means “two or more quartets of all 0s.” However, you can use :: only once in a single address because otherwise the exact IPv6 might not be clear.

For example, consider the following IPv6 address. The bold digits represent digits in which the address could be abbreviated.

FE00:0000:0000:0001:0000:0000:0000:0056

Applying the first rule, you would look at all eight quartets independently. In each, remove all the leading 0s. Note that five of the quartets have four 0s, so for these, remove only three 0s, leaving the following value:

FE00:0:0:1:0:0:0:56

While this abbreviation is valid, the address can be abbreviated more, using the second rule. In this case, two instances exist where more than one quartet in a row has only a 0. Pick the longest such sequence, and replace it with ::, giving you the shortest legal abbreviation:

```
FE00:0:0:1::56
```

While FE00:0:0:1::56 is indeed the shortest abbreviation, this example happens to make it easier to see the two most common mistakes when abbreviating IPv6 addresses. First, never remove trailing 0s in a quartet (0s on the right side of the quartet). In this case, the first quartet of FE00 cannot be shortened at all because the two 0s trail. So, the following address, which begins now with only FE in the first quartet, is not a correct abbreviation of the original IPv6 address:

```
FE:0:0:1::56
```

The second common mistake is to replace all series of all 0 quartets with a double colon. For example, the following abbreviation would be incorrect for the original IPv6 address listed in this topic:

```
FE00::1::56
```

The reason this abbreviation is incorrect is that now you do not know how many quartets of all 0s to substitute into each :: to find the original unabbreviated address.

Expanding Abbreviated IPv6 Addresses

To expand an IPv6 address back into its full unabbreviated 32-digit number, use two similar rules. The rules basically reverse the logic of the previous two rules:



- 1. In each quartet, add leading 0s as needed until the quartet has four hex digits.
- 2. If a double colon (::) exists, count the quartets currently shown; the total should be less than 8. Replace the :: with multiple quartets of 0000 so that eight total quartets exist.

The best way to get comfortable with these addresses and abbreviations is to do some yourself. Table 22-4 lists some practice problems, with the full 32-digit IPv6 address on the left and the best abbreviation on the right. The table gives you either the expanded or abbreviated address, and you need to supply the opposite value. The answers sit at the end of the chapter, in the section “Answers to Earlier Practice Problems.”

Table 22-4 IPv6 Address Abbreviation and Expansion Practice

Full	Abbreviation
2340:0000:0010:0100:1000:ABCD:0101:1010	
	30A0:ABCD:EF12:3456:ABC:B0B0:9999:9009
2222:3333:4444:5555:0000:0000:6060:0707	
	3210::
210F:0000:0000:0000:CCCC:0000:0000:000D	
	34BA:B:B::20
FE80:0000:0000:0000:DEAD:BEFF:FEEF:CAFE	
	FE80::FACE:BAFF:FEFE:CAFE

Representing the Prefix Length of an Address

IPv6 uses a mask concept, called the *prefix length*, similar to IPv4 subnet masks. Similar to the IPv4 prefix-style mask, the IPv6 prefix length is written as a /, followed by a decimal number. The prefix length defines how many bits of the IPv6 address define the IPv6 prefix, which is basically the same concept as the IPv4 subnet ID.

When writing an IPv6 address and prefix length in documentation, you can choose to leave a space before the /, or not, as shown in the next two examples.

```
2222:1111:0:1:A:B:C:D/64
2222:1111:0:1:A:B:C:D /64
```

Finally, note that the prefix length is a number of bits, so with IPv6, the legal value range is from 0 through 128, inclusive.

Calculating the IPv6 Prefix (Subnet ID)

With IPv4, you can take an IP address and the associated subnet mask, and calculate the subnet ID. With IPv6 subnetting, you can take an IPv6 address and the associated prefix length, and calculate the IPv6 equivalent of the subnet ID: an *IPv6 prefix*.

Like with different IPv4 subnet masks, some IPv6 prefix lengths make for an easy math problem to find the IPv6 prefix, while some prefix lengths make the math more difficult. This section looks at the easier cases, mainly because the size of the IPv6 address space lets us all choose to use IPv6 prefix lengths that make the math much easier.

Finding the IPv6 Prefix

In IPv6, a prefix represents a group of IPv6 addresses. For now, this section focuses on the math, and only the math, for finding the number that represents that prefix. Chapter 23, “IPv6 Addressing and Subnetting,” then starts putting more meaning behind the actual numbers.

Each IPv6 prefix, or subnet if you prefer, has a number that represents the group. Per the IPv6 RFCs, the number itself is also called the *prefix*, but many people just call it a subnet number or subnet ID, using the same terms as IPv4.

As with IPv4, you can start with an IPv6 address and prefix length, and find the prefix, with the same general rules that you use in IPv4. If the prefix length is /P, use these rules:



1. Copy the first P bits.
2. Change the rest of the bits to 0.

When using a prefix length that happens to be a multiple of 4, you do not have to think in terms of bits, but in terms of hex digits. A prefix length that is a multiple of 4 means that each hex digit is either copied or changed to hex 0. Just for completeness, if the prefix length is indeed a multiple of 4, the process becomes

1. Identify the number of hex digits in the prefix by dividing the prefix length (which is in bits) by 4.
2. Copy the hex digits determined to be in the prefix per the first step.
3. Change the rest of the hex digits to 0.

Figure 22-7 shows an example, with a prefix length of 64. In this case, Step 1 looks at the /64 prefix length and calculates that the prefix has 16 hex digits. Step 2 copies the first 16 digits of the IPv6 address, while Step 3 records hex 0s for the rest of the digits.

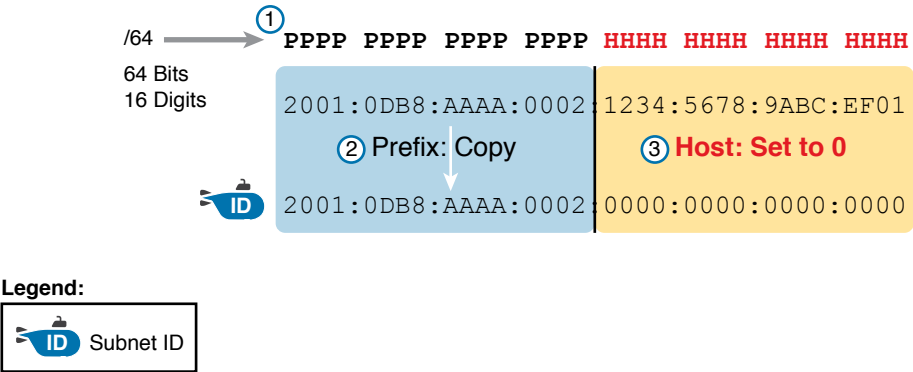


Figure 22-7 *Creating the IPv6 Prefix from an Address/Length*

After you find the IPv6 prefix, you should also be ready to abbreviate the IPv6 prefix using the same rules you use to abbreviate IPv6 addresses. However, you should pay extra attention to the end of the prefix because it often has several octets of all 0 values. As a result, the abbreviation typically ends with two colons (::).

For example, consider the following IPv6 address that is assigned to a host on a LAN:

2000:1234:5678:9ABC:1234:5678:9ABC:1111/64

This example shows an IPv6 address that itself cannot be abbreviated. After you calculate the prefix for the subnet in which the address resides, by zeroing out the last 64 bits (16 digits) of the address, you find the following prefix value:

2000:1234:5678:9ABC:0000:0000:0000:0000/64

This value can be abbreviated, with four quartets of all 0s at the end, as follows:

2000:1234:5678:9ABC::/64

To get better at the math, take some time to work through finding the prefix for several practice problems, as listed in Table 22-5. The answers sit at the end of the chapter, in the section “Answers to Earlier Practice Problems.”

Table 22-5 Finding the IPv6 Prefix from an Address/Length Value

Address/Length	Prefix
2340:0:10:100:1000:ABCD:101:1010/64	
30A0:ABCD:EF12:3456:ABC:B0B0:9999:9009/64	
2222:3333:4444:5555::6060:707/64	
3210::ABCD:101:1010/64	

Address/Length	Prefix
210F::CCCC:B0B0:9999:9009/64	
34BA:B:B:0:5555:0:6060:707/64	
3124::DEAD:CAFE:FF:FE00:1/64	
2BCD::FACE:BEFF:FEFE:CAFE/64	

Working with More-Difficult IPv6 Prefix Lengths

Some prefix lengths make the math to find the prefix very easy, some mostly easy, and some require you to work in binary. If the prefix length is a multiple of 16, the process of copying part of the address copies entire quartets. If the prefix length is not a multiple of 16 but is a multiple of 4, at least the boundary sits at the edge of a hex digit, so you can avoid working in binary.

Although the /64 prefix length is by far the most common prefix length, you should be ready to find the prefix when using a prefix length that is any multiple of 4. For example, consider the following IPv6 address and prefix length:

2000:1234:5678:9ABC:1234:5678:9ABC:1111/56

Because this example uses a /56 prefix length, the prefix includes the first 56 bits, or first 14 complete hex digits, of the address. The rest of the hex digits will be 0, resulting in the following prefix:

2000:1234:5678:9A00:0000:0000:0000:0000/56

This value can be abbreviated, with four quartets of all 0s at the end, as follows:

2000:1234:5678:9A00::/56

This example shows an easy place to make a mistake. Sometimes, people look at the /56 and think of that as the first 14 hex digits, which is correct. However, they then copy the first 14 hex digits and add a double colon, showing the following:

2000:1234:5678:9A::/56

This abbreviation is not correct because it removed the trailing “00” at the end of the fourth quartet. If you expanded the abbreviated value, it would begin with 2000:1234:5678:009A, not 2000:1234:5678:9A00. So, be careful when abbreviating when the boundary is not at the edge of a quartet.

Once again, some extra practice can help. Table 22-6 uses examples that have a prefix length that is a multiple of 4, but is not on a quartet boundary, just to get some extra practice. The answers sit at the end of the chapter, in the section “Answers to Earlier Practice Problems.”

Table 22-6 Finding the IPv6 Prefix from an Address/Length Value

Address/Length	Prefix
34BA:B:B:0:5555:0:6060:707/80	
3124::DEAD:CAFE:FF:FE00:1/80	
2BCD::FACE:BEFF:FEBE:CAFE/48	
3FED:F:E0:D00:FACE:BAFF:FE00:0/48	
210F:A:B:C:CCCC:B0B0:9999:9009/40	
34BA:B:B:0:5555:0:6060:707/36	
3124::DEAD:CAFE:FF:FE00:1/60	
2BCD::FACE:1:BEFF:FEBE:CAFE/56	

Chapter Review

One key to doing well on the exams is to perform repetitive spaced review sessions. Review this chapter’s material using either the tools in the book or interactive tools for the same material found on the book’s companion website. Refer to the “Your Study Plan” element for more details. Table 22-7 outlines the key review elements and where you can find them. To better track your study progress, record when you completed these activities in the second column.

Table 22-7 Chapter Review Tracking

Review Element	Review Date(s)	Resource Used
Review key topics		Book, website
Review key terms		Book, website
Repeat DIKTA questions		Book, PTP
Review command tables		Book
Review memory table		Book, website

Review All the Key Topics



Table 22-8 Key Topics for Chapter 22

Key Topic Element	Description	Page Number
List	Similarities between IPv4 and IPv6	527
List	Rules for abbreviating IPv6 addresses	531
List	Rules for expanding an abbreviated IPv6 address	532
List	Process steps to find an IPv6 prefix, based on the IPv6 address and prefix length	533

Key Terms You Should Know

IPv4 address exhaustion, IP version 6 (IPv6), OSPF version 3 (OSPFv3), EIGRP version 6 (EIGRPv6), prefix, prefix length, quartet

Additional Practice for This Chapter's Processes

For additional practice with IPv6 abbreviations, you may do the same set of practice problems based on Appendix G, “Practice for Chapter 22: Fundamentals of IP Version 6.” You have two options to use:

PDF: Navigate to the companion website and open the PDF for Appendix G.

Application: Navigate to the companion website and use these applications:

“Practice Exercise: Abbreviating and Expanding Addresses”

“Practice Exercise: Calculating the IPv6 Prefix”

“Practice Exercise: Calculating the IPv6 Prefix Round 2”

Answers to Earlier Practice Problems

This chapter includes practice problems spread around different locations in the chapter. The answers are located in Tables 22-9, 22-10, and 22-11.

Table 22-9 Answers to Questions in the Earlier Table 22-4

Full	Abbreviation
2340:0000:0010:0100:1000:ABCD:0101:1010	2340:0:10:100:1000:ABCD:101:1010
30A0:ABCD:EF12:3456:0ABC:B0B0:9999:9009	30A0:ABCD:EF12:3456:ABC:B0B0:9999:9009
2222:3333:4444:5555:0000:0000:6060:0707	2222:3333:4444:5555::6060:707
3210:0000:0000:0000:0000:0000:0000	3210::
210F:0000:0000:0000:CCCC:0000:0000:000D	210F::CCCC:0:0:D
34BA:000B:000B:0000:0000:0000:0000:0020	34BA:B:B::20
FE80:0000:0000:0000:DEAD:BEFF:FEEF:CAFE	FE80::DEAD:BEFF:FEEF:CAFE
FE80:0000:0000:0000:FACE:BAFF:FEFE:CAFE	FE80::FACE:BAFF:FEFE:CAFE

Table 22-10 Answers to Questions in the Earlier Table 22-5

Address/Length	Prefix
2340:0:10:100:1000:ABCD:101:1010/64	2340:0:10:100::/64
30A0:ABCD:EF12:3456:ABC:B0B0:9999:9009/64	30A0:ABCD:EF12:3456::/64
2222:3333:4444:5555::6060:707/64	2222:3333:4444:5555::/64
3210::ABCD:101:1010/64	3210::/64
210F::CCCC:B0B0:9999:9009/64	210F::/64
34BA:B:B:0:5555:0:6060:707/64	34BA:B:B::/64
3124::DEAD:CAFE:FF:FE00:1/64	3124:0:0:DEAD::/64
2BCD::FACE:BEFF:FEFE:CAFE/64	2BCD::/64

Table 22-11 Answers to Questions in the Earlier Table 22-6

Address/Length	Prefix
34BA:B:B:0:5555:0:6060:707/80	34BA:B:B:0:5555::/80
3124::DEAD:CAFE:FF:FE00:1/80	3124:0:0:DEAD:CAFE::/80
2BCD::FACE:BEFF:FEBE:CAFE/48	2BCD::/48
3FED:F:E0:D00:FACE:BAFF:FE00:0/48	3FED:F:E0::/48
210F:A:B:C:CCCC:B0B0:9999:9009/40	210F:A::/40
34BA:B:B:0:5555:0:6060:707/36	34BA:B::/36
3124::DEAD:CAFE:FF:FE00:1/60	3124:0:0:DEA0::/60
2BCD::FACE:1:BEFF:FEBE:CAFE/56	2BCD:0:0:FA00::/56

This page intentionally left blank



CHAPTER 23

IPv6 Addressing and Subnetting

This chapter covers the following exam topics:

1.0 Network Fundamentals

1.8 Configure and verify IPv6 addressing and prefix

1.9 Compare and contrast IPv6 address types

1.9.a Global unicast

1.9.b Unique local

IPv4 organizes the address space in a couple of ways. First, IPv4 splits addresses by class, with Classes A, B, and C defining unicast IPv4 addresses. (The term *unicast* refers to the fact that each address is used by only one interface.) Then, within the Class A, B, and C address range, the Internet Assigned Numbers Authority (IANA) and the Internet Corporation for Assigned Names and Numbers (ICANN) reserve most of the addresses as public IPv4 addresses, with a few reserved as private IPv4 addresses.

IPv6 does not use any concept like the classful network concept used by IPv4. However, IANA does still reserve some IPv6 address ranges for specific purposes, even with some address ranges that serve as both public IPv6 addresses and private IPv6 addresses. IANA also attempts to take a practical approach to reserving ranges of the entire IPv6 address space for different purposes, using the wisdom gained from several decades of fast growth in the IPv4 Internet.

This chapter has two major sections. The first examines *global unicast addresses*, which serve as public IPv6 addresses. The second major section looks at *unique local addresses*, which serve as private IPv6 addresses.

“Do I Know This Already?” Quiz

Take the quiz (either here or use the PTP software) if you want to use the score to help you decide how much time to spend on this chapter. The letter answers are listed at the bottom of the page following the quiz. Appendix C, found both at the end of the book as well as on the companion website, includes both the answers and explanations. You can also find both answers and explanations in the PTP testing software.

Table 23-1 “Do I Know This Already?” Foundation Topics Section-to-Question Mapping

Foundation Topics Section	Questions
Global Unicast Addressing Concepts	1–4
Unique Local Unicast Addresses	5

1. Which of the following IPv6 addresses appears to be a unique local unicast address, based on its first few hex digits?
 - a. 3123:1:3:5::1
 - b. FE80::1234:56FF:FE78:9ABC
 - c. FDAD::1
 - d. FF00::5
2. Which of the following IPv6 addresses appears to be a global unicast address, based on its first few hex digits?
 - a. 3123:1:3:5::1
 - b. FE80::1234:56FF:FE78:9ABC
 - c. FDAD::1
 - d. FF00::5
3. When subnetting an IPv6 address block, an engineer shows a drawing that breaks the address structure into three pieces. Comparing this concept to a three-part IPv4 address structure, which part of the IPv6 address structure is most like the IPv4 network part of the address?
 - a. Subnet
 - b. Interface ID
 - c. Network
 - d. Global routing prefix
 - e. Subnet router anycast
4. When subnetting an IPv6 address block, an engineer shows a drawing that breaks the address structure into three pieces. Assuming that all subnets use the same prefix length, which of the following answers lists the name of the field on the far right side of the address?
 - a. Subnet
 - b. Interface ID
 - c. Network
 - d. Global routing prefix
 - e. Subnet router anycast
5. For the IPv6 address FD00:1234:5678:9ABC:DEF1:2345:6789:ABCD, which part of the address is considered the global ID of the unique local address?
 - a. None; this address has no global ID.
 - b. 00:1234:5678:9ABC
 - c. DEF1:2345:6789:ABCD
 - d. 00:1234:5678
 - e. FD00

Foundation Topics

Global Unicast Addressing Concepts

This first major section of the chapter focuses on one type of unicast IPv6 addresses: global unicast addresses. As it turns out, many of the general concepts and processes behind these global unicast IPv6 addresses follow the original intent for public IPv4 addresses. So, this section begins with a review of some IPv4 concepts, followed by the details of how a company can use global unicast addresses.

This first section also discusses IPv6 subnetting and the entire process of taking a block of global unicast addresses and creating subnets for one company. This process takes a globally unique global routing prefix, creates IPv6 subnets, and assigns IPv6 addresses from within each subnet, much like with IPv4.

Public and Private IPv6 Addresses

In the history of IPv4 addressing, the world started out with a plan that gave every single host a globally unique public IPv4 address. However, as discussed in several places already, the IPv4 address space had too few addresses. So, in the 1990s, companies started using addresses from the private IPv4 address range, as defined in RFC 1918. These companies either simply did not connect to the Internet, or to connect to the Internet, they used Network Address Translation (NAT), sharing a few public globally unique IPv4 addresses for all host connections into the Internet.

IPv6 allows two similar options of public and private unicast addressing, beginning with *global unicast* addresses as the public IPv6 address space. Similar to public IPv4 addresses, IPv6 global unicast addresses rely on an administrative process that assigns each company a unique IPv6 address block. Each company then subnets this IPv6 address block and only uses addresses from within that block. The result: that company uses addresses that are unique across the globe as well.

The second IPv6 option uses *unique local* IPv6 addresses, which work more like the IPv4 private addresses. Companies that do not plan to connect to the Internet and companies that plan to use IPv6 NAT can use these private unique local addresses. The process also works similarly to IPv4: The engineer can read the details in an RFC, pick some numbers, and start assigning IPv6 addresses without having to register with IANA or any other authority.

The following lists summarizes the comparisons between global unicast addresses and unique local addresses:

Key Topic

Global unicast: Addresses that work like public IPv4 addresses. The organization that needs IPv6 addresses asks for a registered IPv6 address block, which is assigned as a global routing prefix. After that, only that organization uses the addresses inside that block of addresses—that is, the addresses that begin with the assigned prefix.

Unique local: Works somewhat like private IPv4 addresses, with the possibility that multiple organizations use the exact same addresses, and with no requirement for registering with any numbering authority.

Answers to the “Do I Know This Already?” quiz:

1 C 2 A 3 D 4 B 5 D

The rest of this first major section of the chapter examines global unicast addresses in more detail, while the second major section of the chapter examines unique local addresses.

The IPv6 Global Routing Prefix

IPv6 global unicast addresses allow IPv6 to work more like the original design of the IPv4 Internet. Each organization asks for a block of IPv6 addresses, which no one else can use. That organization further subdivides the address block into smaller chunks, called *subnets*. Finally, to choose what IPv6 address to use for any host, the engineer chooses an address from the right subnet.

That reserved block of IPv6 addresses—a set of addresses that only one company can use—is called a *global routing prefix*. Each organization that wants to connect to the Internet and use IPv6 global unicast addresses should ask for and receive a global routing prefix. Very generally, you can think of the global routing prefix like an IPv4 Class A, B, or C network number from the range of public IPv4 addresses.

The term *global routing prefix* might not make you think of a block of IPv6 addresses at first. The term actually refers to the idea that Internet routers can have one route that refers to all the addresses inside the address block, without a need to have routes for smaller parts of that block. For example, Figure 23-1 shows three companies, with three different IPv6 global routing prefixes; the router on the right (R4) has one IPv6 route for each global routing prefix.

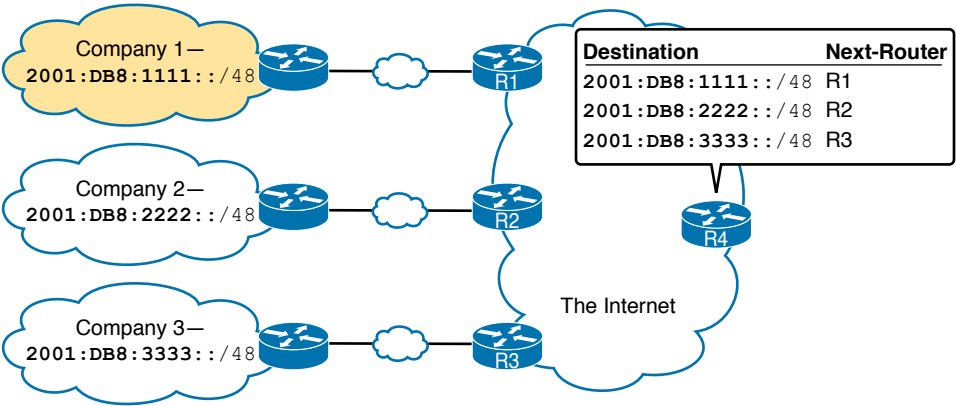


Figure 23-1 Three Global Routing Prefixes, with One Route per Prefix

The global routing prefix sets those IPv6 addresses apart for use by that one company, just like a public IPv4 network or CIDR address block does in IPv4. All IPv6 addresses inside that company should begin with that global routing prefix, to avoid using other companies' IPv6 addresses. No other companies should use IPv6 addresses with that same prefix. And thankfully, IPv6 has plenty of space to allow all companies to have a global routing prefix, with plenty of addresses.

Both the IPv6 and IPv4 address assignment processes rely on the same organizations: IANA (along with ICANN), the Regional Internet Registries (RIR), and ISPs. For example,

an imaginary company, Company1, received the assignment of a global routing prefix. The prefix means “All addresses whose first 12 hex digits are 2001:0DB8:1111,” as represented by prefix 2001:0DB8:1111::/48. To receive that assignment, the process shown in Figure 23-2 happened.

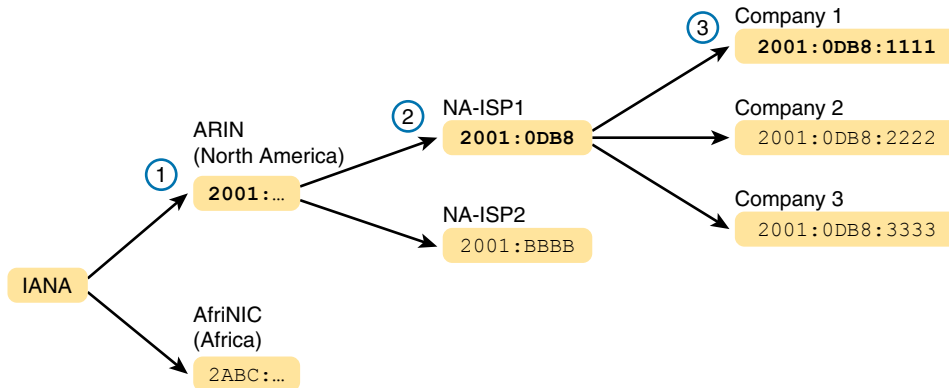


Figure 23-2 Prefix Assignment with IANA, RIRs, and ISPs

The event timeline in the figure uses a left-to-right flow; in other words, the event on the far left must happen first. Following the flow from left to right in the figure:

1. **IANA allocates ARIN prefix 2001::/16:** ARIN (the RIR for North America) asks IANA for the allocation of a large block of addresses. In this imaginary example, IANA gives ARIN a prefix of “all addresses that begin 2001,” or 2001::/16.
2. **ARIN allocates NA-ISP1 prefix 2001:0DB8::/32:** NA-ISP1, an imaginary ISP based in North America, asks ARIN for a new IPv6 prefix. ARIN takes a subset of its 2001::/16 prefix, specifically all addresses that begin with the 32 bits (8 hex digits) 2001:0DB8, and allocates it to the ISP.
3. **NA-ISP1 assigns Company 1 2001:0DB8:1111::/48:** Company 1 decides to start supporting IPv6, so it goes to its ISP, NA-ISP1, to ask for a block of global unicast addresses. NA-ISP1 assigns Company 1 a “small” piece of NA-ISP1’s address block, in this case the addresses that begin with the 48 bits (12 hex digits) of 2001:0DB8:1111 (2001:0DB8:1111::/48).

NOTE If you do not plan to connect to the Internet using IPv6 for a while and just want to experiment, you do not need to ask for an IPv6 global routing prefix to be assigned. Just make up IPv6 addresses and configure your devices, or use unique local addresses as discussed toward the end of this chapter.

Address Ranges for Global Unicast Addresses

Global unicast addresses make up the majority of the IPv6 address space. However, unlike IPv4, the rules for which IPv6 addresses fall into which category are purposefully more flexible than they were with IPv4 and the rules for IPv4 Classes A, B, C, D, and E.

Originally, IANA reserved all IPv6 addresses that begin with hex 2 or 3 as global unicast addresses. (This address range can be written succinctly as prefix 2000::/3.)

Later IANA made the global unicast address range wider, basically to include all IPv6 addresses not otherwise allocated for other purposes. For example, the unique local unicast addresses, discussed later in this chapter, all start with hex FD. So, while global unicast addresses would not include any addresses that begin with FD, any address ranges that are not specifically reserved, for now, are considered to be global unicast addresses.

Finally, just because an amazingly enormous number of addresses sit within the global unicast address range, IANA does not assign prefixes from all over the address range. IPv4 has survived well for more than 30 years with an admittedly too-small address size because IANA has adopted good practices to conserve the IPv4 address space. By making smart and practical choices in assigning IPv6 addresses, the IPv6 address space could last much longer than IPv4.

Table 23-2 lists the address prefixes discussed in this book and their purpose.



Table 23-2 Some Types of IPv6 Addresses and Their First Hex Digit(s)

Address Type	First Hex Digits
Global unicast	2 or 3 (originally); all not otherwise reserved (today)
Unique local	FD
Multicast	FF
Link local	FE80

IPv6 Subnetting Using Global Unicast Addresses

After an enterprise has a block of reserved global unicast addresses—in other words, a global routing prefix—the company needs to subdivide that large address block into subnets.

Subnetting IPv6 addresses works generally like IPv4, but with mostly simpler math (hoorah!). Because of the absolutely large number of addresses available, most everyone uses the easiest possible IPv6 prefix length: /64. Using /64 as the prefix length for all subnets makes the IPv6 subnetting math just as easy as using a /24 mask for all IPv4 subnets. In addition, the dynamic IPv6 address assignment process works better with a /64 prefix length as well; so in practice, and in this book, expect IPv6 designs to use a /64 prefix length.

This section does walk you through the different parts of IPv6 subnetting, while mostly using examples that use a /64 prefix length. The discussion defines the rules about which addresses should be in the same subnet and which addresses need to be in different subnets. Plus this section looks at how to analyze the global routing prefix and associated prefix length to find all the IPv6 prefixes (subnet IDs) and the addresses in each subnet.

NOTE If the IPv4 subnetting concepts are a little vague, you might want to reread Chapter 11, “Perspectives on IPv4 Subnetting,” which discusses the subnetting concepts for IPv4.

Deciding Where IPv6 Subnets Are Needed

First, IPv6 and IPv4 both use the same concepts about where a subnet is needed: one for each VLAN and one for each point-to-point WAN connection (serial and Ethernet). Figure 23-3 shows an example of the idea, using the small enterprise internetwork of Company 1. Company 1 has two LANs, with a point-to-point serial link connecting the sites. It also has an Ethernet WAN link connected to an ISP. Using the same logic you would use for IPv4, Company 1 needs four IPv6 subnets.

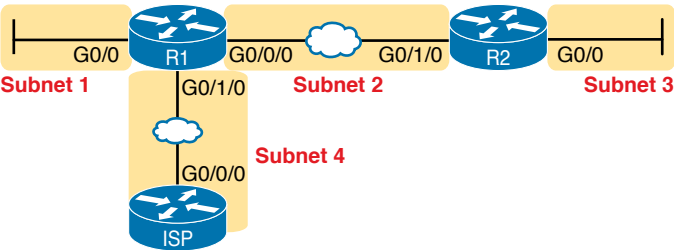


Figure 23-3 Locations for IPv6 Subnets

The Mechanics of Subnetting IPv6 Global Unicast Addresses

To understand how to subnet your one large block of IPv6 addresses, you need to understand some of the theory and mechanisms IPv6 uses. To learn those details, it can help to compare IPv6 with some similar concepts from IPv4.

With IPv4, without subnetting, an address has two parts: a network part and a host part. Class A, B, and C rules define the length of the network part, with the host part making up the rest of the 32-bit IPv4 address, as shown in Figure 23-4.

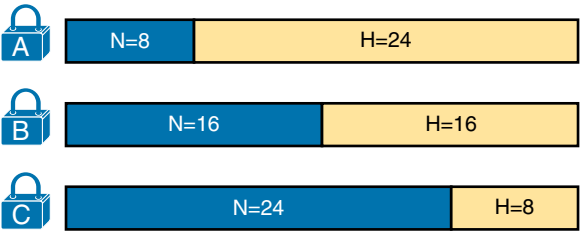


Figure 23-4 Classful View of Unsubnetted IPv4 Networks

To subnet an IPv4 Class A, B, or C network, the network engineer for the enterprise makes some choices. Conceptually, the engineer creates a three-part view of the addresses, adding a subnet field in the center while shortening the host field. (Many people call this “borrowing host bits.”) The size of the network part stays locked per the Class A, B, and C rules, with the line between the subnet and host part being flexible, based on the choice of subnet mask. Figure 23-5 shows the idea for a subnetted Class B network.