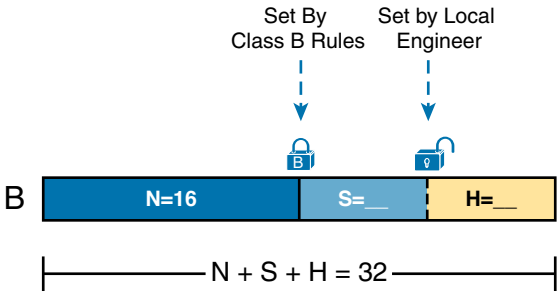


Figure 23-5 *Classful View of Subnetted IPv4 Networks*

IPv6 uses a similar concept, with the details in Figure 23-6. The structure shows three major parts, beginning with the global routing prefix, which is the initial value that must be the same in all IPv6 addresses inside the enterprise. The address ends with the interface ID, which acts like the IPv4 host field. The subnet field sits between the two other fields, used as a way to number and identify subnets, much like the subnet field in IPv4 addresses.

Key Topic

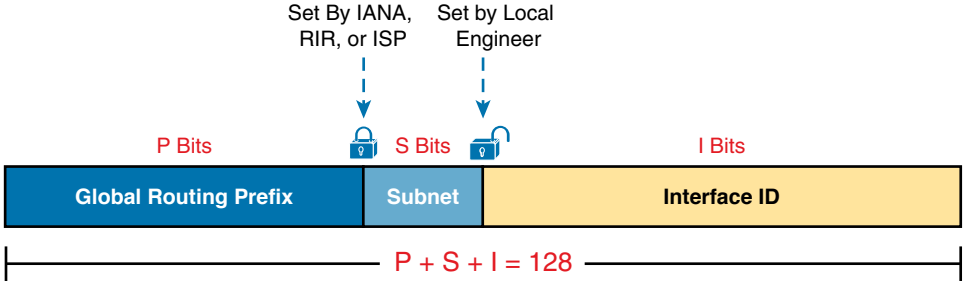


Figure 23-6 *Structure of Subnetted IPv6 Global Unicast Addresses*

First, just think about the general idea with IPv6, comparing Figure 23-6 to Figure 23-5. The IPv6 global routing prefix (the prefix/length assigned by the RIR or ISP) acts like the IPv4 network part of the address structure. The IPv6 subnet part acts like the IPv4 subnet part. And the right side of the IPv6, formally called the *interface ID* (short for interface identifier), acts like the IPv4 host field.

Now focus on the IPv6 global routing prefix and its prefix length. Unlike IPv4, IPv6 has no concept of address classes, so no preset rules determine the prefix length of the global routing prefix. However, when a company applies to an ISP, RIR, or any other organization that can assign a global routing prefix, that assignment includes both the prefix and the prefix length. After a company receives a global routing prefix and that prefix length, the length of the prefix typically does not change over time and is basically locked. (Note that the prefix length of the global routing prefix is often between /32 and /48, or possibly as long as /56.)

Next, look to the right side of Figure 23-6 to the interface ID field. For several reasons that become more obvious the more you learn about IPv6, this field is often 64 bits long. Does it have to be 64 bits long? No. However, using a 64-bit interface ID field works well in real networks, and there are no reasons to avoid using a 64-bit interface ID field.

Finally, look to the subnet field in the center of Figure 23-6. Similar to IPv4, this field creates a place with which to number IPv6 subnets. The length of the subnet field is based on the other two facts: the length of the global routing prefix and the length of the interface ID. And with the commonly used 64-bit interface ID field, the subnet field is typically 64–P bits, with P being the length of the global routing prefix.

Next, consider the structure of a specific global unicast IPv6 address, 2001:0DB8:1111:0001:0000:0000:0000:0001, as seen in Figure 23-7. In this case:

Key Topic

- The company was assigned prefix 2001:0DB8:1111, with prefix length /48.
- The company uses the usual 64-bit interface ID.
- The company has a subnet field of 16 bits, allowing for 2^{16} IPv6 subnets.

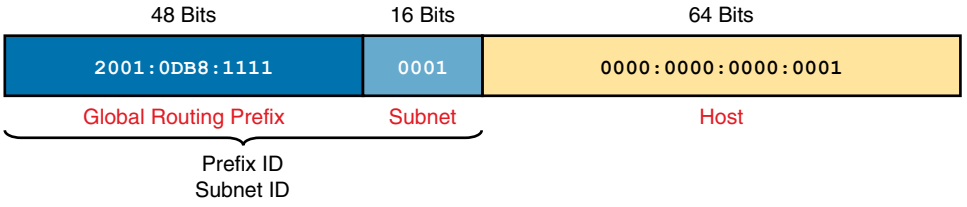


Figure 23-7 *Address Structure for Company 1 Example*

The example in Figure 23-7, along with a little math, shows one reason why so many companies use a /64 prefix length for all subnets. With this structure, Company 1 can support 2^{16} possible subnets (65,536). Few companies need that many subnets. Then, each subnet supports over 10^{18} addresses per subnet (2^{64}, minus some reserved values). So, for both subnets and hosts, the address structure supports far more than are needed. Plus, the /64 prefix length for all subnets makes the math simple because it cuts the 128-bit IPv6 address in half.

Listing the IPv6 Subnet Identifier

Like with IPv4, IPv6 needs to identify each IPv6 subnet with some kind of a subnet identifier, or subnet ID. Figure 23-7 lists the informal names for this number (subnet ID) and the more formal name (prefix ID). Routers then list the IPv6 subnet ID in routing tables, along with the prefix length.

Chapter 22, "Fundamentals of IP Version 6," already discussed how to find the subnet ID, given an IPv6 address and prefix length. The math works the same way when working with global unicast addresses, as well as the unique local addresses discussed later in the chapter. Chapter 28, "Securing Wireless Networks," has already discussed the math, but for completeness, note that the subnet ID shown in Figure 23-7 would be

```
2001:DB8:1111:1::/64
```

List All IPv6 Subnets

With IPv4, if you choose to use a single subnet mask for all subnets, you can sit and write down all the subnets of a Class A, B, or C network using that one subnet mask. With IPv6,

the same ideas apply. If you plan to use a single prefix length for all subnets, you can start with the global routing prefix and write down all the IPv6 subnet IDs as well.

To find all the subnet IDs, you simply need to find all the unique values that will fit inside the subnet part of the IPv6 address, basically following these rules:

- All subnet IDs begin with the global routing prefix.
- Use a different value in the subnet field to identify each different subnet.
- All subnet IDs have all 0s in the interface ID.

As an example, take the IPv6 design shown in Figure 23-7, and think about all the subnet IDs. First, all subnets will use the commonly used /64 prefix length. This company uses a global routing prefix of 2001:0DB8:1111::/48, which defines the first 12 hex digits of all the subnet IDs. To find all the possible IPv6 subnet IDs, think of all the combinations of unique values in the fourth quartet and then represent the last four quartets of all 0s with a :: symbol. Figure 23-8 shows the beginning of just such a list.

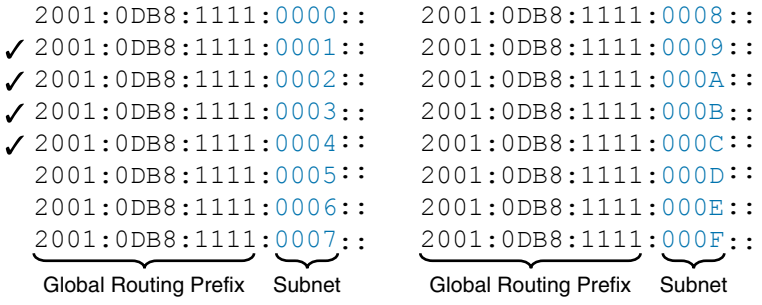


Figure 23-8 *First 16 Possible Subnets with a 16-bit Subnet Field in This Example*

The example allows for 65,536 subnets, so clearly the example will not list all the possible subnets. However, in that fourth quartet, all combinations of hex values would be allowed.

NOTE The IPv6 subnet ID, more formally called the *subnet router anycast address*, is reserved and should not be used as an IPv6 address for any host.

Assign Subnets to the Internetwork Topology

After an engineer lists all the possible subnet IDs (based on the subnet design), the next step is to choose which subnet ID to use for each link that needs an IPv6 subnet. Just like with IPv4, each VLAN, each serial link, each Ethernet WAN link, and many other data-link instances need an IPv6 subnet.

Figure 23-9 shows an example using Company 1 again. The figure uses the four subnets from Figure 23-8 that have check marks beside them. The check marks are just a reminder to not use those four subnets in other locations.

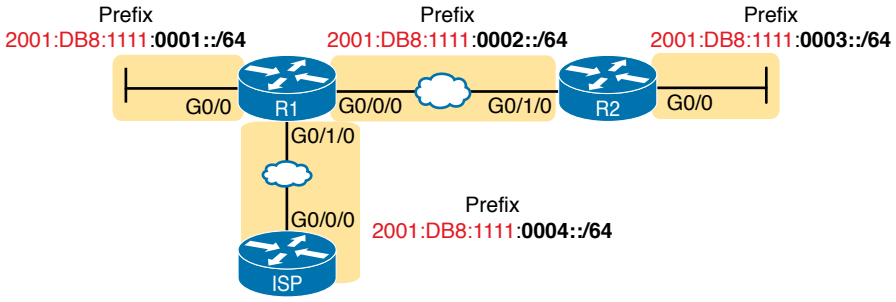


Figure 23-9 *Subnets in Company 1, with Global Routing Prefix of 2001:0DB8:1111::/48*

Assigning Addresses to Hosts in a Subnet

Now that the engineer has planned which IPv6 subnet will be used in each location, the individual IPv6 addressing can be planned and implemented. Each address must be unique, in that no other host interface uses the same IPv6 address. Also, the hosts cannot use the subnet ID itself.

The process of assigning IPv6 addresses to interfaces works similarly to IPv4. Addresses can be configured statically, along with the prefix length, default router, and Domain Name System (DNS) IPv6 addresses. Alternatively, hosts can learn these same settings dynamically, using either Dynamic Host Configuration Protocol (DHCP) or a built-in IPv6 mechanism called Stateless Address Autoconfiguration (SLAAC).

For example, Figure 23-10 shows some static IP addresses that could be chosen for the router interfaces based on the subnet choices shown in Figure 23-9. In each case, the router interfaces use an interface ID that is a relatively low number, easily remembered.

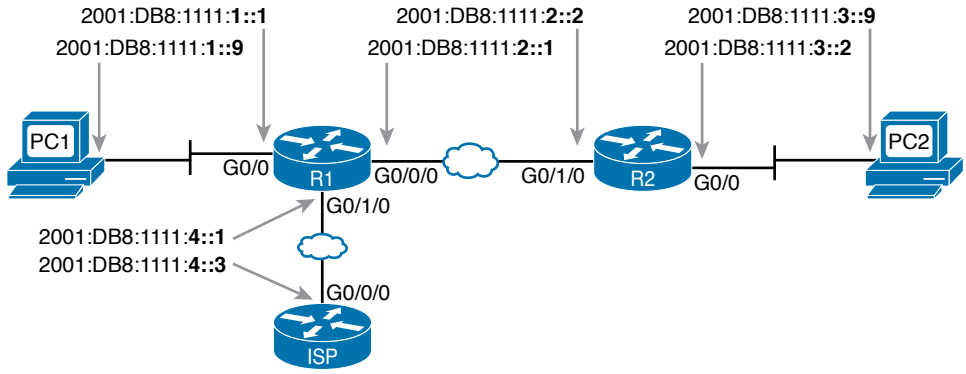


Figure 23-10 *Example Static IPv6 Addresses Based on the Subnet Design of Figure 23-9*

This chapter puts off the details of how to configure the IPv6 addresses until Chapter 24, "Implementing IPv6 Addressing on Routers."

Unique Local Unicast Addresses

Unique local unicast addresses act as private IPv6 addresses. These addresses have many similarities with global unicast addresses, particularly in how to subnet. The biggest difference lies in the literal number (unique local addresses begin with hex FD) and with the administrative process: the unique local prefixes are not registered with any numbering authority and can be used by multiple organizations.

Although the network engineer creates unique local addresses without any registration or assignment process, the addresses still need to follow some rules, as follows:

Key Topic

- Use FD as the first two hex digits.
- Choose a unique 40-bit global ID.
- Append the global ID to FD to create a 48-bit prefix, used as the prefix for all your addresses.
- Use the next 16 bits as a subnet field.
- Note that the structure leaves a convenient 64-bit interface ID field.

Figure 23-11 shows the format of these unique local unicast addresses.

Key Topic

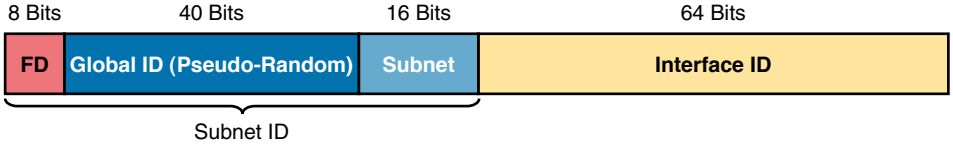


Figure 23-11 *IPv6 Unique Local Unicast Address Format*

NOTE Just to be completely exact, IANA actually reserves prefix FC00::/7, and not FD00::/8, for these addresses. FC00::/7 includes all addresses that begin with hex FC and FD. However, an RFC (4193) requires the eighth bit of these addresses to be set to 1, which means that in practice today, the unique local addresses all begin with their first two digits as FD.

Subnetting with Unique Local IPv6 Addresses

Subnetting using unique local addresses works just like subnetting with global unicast addresses with a 48-bit global routing prefix. The only difference is that with global unicasts, you start by asking for a global routing prefix to be assigned to your company, and that global routing prefix might or might not have a /48 prefix length. With unique local, you create that prefix locally, and the prefix begins with /48, with the first 8 bits set and the next 40 bits randomly chosen.

The process can be as simple as choosing a 40-bit value as your global ID. These 40 bits require 10 hex digits, so you can even avoid thinking in binary and just make up a unique 10-hex-digit value and add hex FD to the front. For example, imagine you chose a 10-hex-digit value of hex 00 0001 0001, prepend a hex FD, making the entire prefix be FD00:0001:0001::/48, or FD00:1:1::/48 when abbreviated.

To create subnets, just as you did in the earlier examples with a 48-bit global routing prefix, treat the entire fourth quartet as a subnet field, as shown in Figure 23-11.

Figure 23-12 shows an example subnetting plan using unique local addresses. The example repeats the same topology shown earlier in Figure 23-9; that figure showed subnetting with a global unicast prefix. This example uses the exact same numbers for the fourth quartet's subnet field, simply replacing the 48-bit global unicast prefix with this new local unique prefix of FD00:1:1.

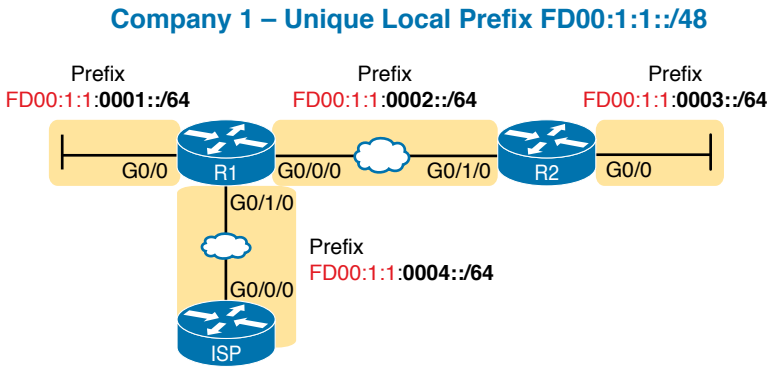


Figure 23-12 *Subnetting Using Unique Local Addresses*

The Need for Globally Unique Local Addresses

The example in Figure 23-12 shows an easy-to-remember prefix of FD00:1:1::/48. Clearly, I made up the easy-to-remember global ID in this example. What global ID would you choose for your company? Would you pick a number that you could not abbreviate and make it shorter? If you had to pick the IPv6 prefix for your unique local addresses from the options in the following list, which would you pick for your company?

- `FDE9:81BE:A059::/48`
- `FDF0:E1D2:C3B4::/48`
- `FD00:1:1::/48`

Given freedom to choose, most people would pick an easy-to-remember, short-to-type prefix, like FD00:1:1::/48. And in a lab or other small network used for testing, making up an easy-to-use number is reasonable. However, for use in real corporate networks, you should not just make up any global ID you like; you should try to follow the unique local address rules that strive to help make your addresses unique in the universe—even without registering a prefix with an ISP or RIR.

RFC 4193 defines unique local addresses, and that RFC stresses the importance of choosing your global ID in a way to make it statistically unlikely to be used by other companies. What is the result of unique global IDs at every company? Making all these unique local addresses unique across the globe. So, if you do plan on using unique local addresses in a real network, plan on using the random number generator logic listed in RFC 4193 to create your prefix.

One of the big reasons to attempt to use a unique prefix, rather than everyone using the same easy-to-remember prefixes, is to be ready for the day that your company merges with

or buys another company. Today, with IPv4, a high percentage of companies use private IPv4 network 10.0.0.0. When they merge their networks, the fact that both use network 10.0.0.0 makes the network merger more painful than if the companies had used different private IPv4 networks. With IPv6 unique local addresses, if both companies did the right thing and randomly chose a prefix, they will most likely be using completely different prefixes, making the merger much simpler. However, companies that take the seemingly easy way out and choose an easy-to-remember prefix like FD00:1:1 greatly increase their risk of requiring extra effort when merging with another company that also chose to use that same prefix.

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 23-3 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 23-3 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 table		Website

Review All the Key Topics

Key Topic

Table 23-4 Key Topics for Chapter 23

Key Topic Element	Description	Page Number
List	Two types of IPv6 unicast addresses	542
Table 23-2	Values of the initial hex digits of IPv6 addresses, and the address type implied by each	545
Figure 23-6	Subnetting concepts for IPv6 global unicast addresses	547
List	Rules for how to find all IPv6 subnet IDs, given the global routing prefix, and prefix length used for all subnets	548
List	Rules for building unique local unicast addresses	551
Figure 23-11	Subnetting concepts for IPv6 unique local addresses	551

Key Terms You Should Know

global unicast address, global routing prefix, unique local address, subnet ID (prefix ID), subnet router anycast address

CHAPTER 24

Implementing IPv6 Addressing on Routers

This chapter covers the following exam topics:

1.0 Network Fundamentals

1.9 Compare and contrast IPv6 address types

1.9.a Global unicast

1.9.b Unique local

1.9.c Link local

1.9.d Anycast

1.9.e Multicast

1.9.f Modified EUI 64

With IPv4 addressing, some devices, like servers and routers, typically use static predefined IPv4 addresses. End-user devices do not mind if their address changes from time to time, and they typically learn an IPv4 address dynamically using DHCP. IPv6 uses the same approach, with servers, routers, and other devices in the control of the IT group often using predefined IPv6 addresses, and with end-user devices using dynamically learned IPv6 addresses.

This chapter focuses on IPv6 address configuration on routers. The chapter begins with the more obvious IPv6 addressing configuration, with features that mirror IPv4 features, showing how to configure interfaces with IPv6 addresses and view that configuration with **show** commands. The second half of the chapter introduces new IPv6 addressing concepts, showing some other addresses used by routers when doing different tasks.

"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 24-1 "Do I Know This Already?" Foundation Topics Section-to-Question Mapping

Foundation Topics Section	Questions
Implementing Unicast IPv6 Addresses on Routers	1–3
Special Addresses Used by Routers	4–5

1. Router R1 has an interface named Gigabit Ethernet 0/1, whose MAC address has been set to 0200.0001.000A. Which of the following commands, added in R1's Gigabit Ethernet 0/1 configuration mode, gives this router's G0/1 interface a unicast IPv6 address of 2001:1:1:1:1:200:1:A, with a /64 prefix length?
   a. **ipv6 address 2001:1:1:1:1:200:1:A/64**
   b. **ipv6 address 2001:1:1:1:1:200:1:A/64 eui-64**
   c. **ipv6 address 2001:1:1:1:1:200:1:A /64 eui-64**
   d. **ipv6 address 2001:1:1:1:1:200:1:A /64**
   e. None of the other answers are correct.
2. Router R1 has an interface named Gigabit Ethernet 0/1, whose MAC address has been set to 5055.4444.3333. This interface has been configured with the **ipv6 address 2000:1:1:1::/64 eui-64** subcommand. What unicast address will this interface use?
   a. 2000:1:1:1:52FF:FE55:4444:3333
   b. 2000:1:1:1:5255:44FF:FE44:3333
   c. 2000:1:1:1:5255:4444:33FF:FE33
   d. 2000:1:1:1:200:FF:FE00:0
3. Router R1 currently supports IPv4, routing packets in and out all its interfaces. R1's configuration needs to be migrated to support dual-stack operation, routing both IPv4 and IPv6. Which of the following tasks must be performed before the router can also support routing IPv6 packets? (Choose two answers.)
   a. Enable IPv6 on each interface using an **ipv6 address** interface subcommand.
   b. Enable support for both versions with the **ip versions 4 6** global command.
   c. Additionally enable IPv6 routing using the **ipv6 unicast-routing** global command.
   d. Migrate to dual-stack routing using the **ip routing dual-stack** global command.
4. Router R1 has an interface named Gigabit Ethernet 0/1, whose MAC address has been set to 0200.0001.000A. The interface is then configured with the **ipv6 address 2001:1:1:1:200:FF:FE01:B/64** interface subcommand; no other **ipv6 address** commands are configured on the interface. Which of the following answers lists the link-local address used on the interface?
   a. FE80::FF:FE01:A
   b. FE80::FF:FE01:B
   c. FE80::200:FF:FE01:A
   d. FE80::200:FF:FE01:B

5. Which of the following multicast addresses is defined as the address for sending packets to only the IPv6 routers on the local link?

   a. FF02::1
   b. FF02::2
   c. FF02::5
   d. FF02::A

Foundation Topics

Implementing Unicast IPv6 Addresses on Routers

Every company bases its enterprise network on one or more protocol models, or protocol stacks. In the earlier days of networking, enterprise networks used one or more protocol stacks from different vendors, as shown on the left of Figure 24-1. Over time, companies added TCP/IP (based on IPv4) to the mix. Eventually, companies migrated fully to TCP/IP as the only protocol stack in use.

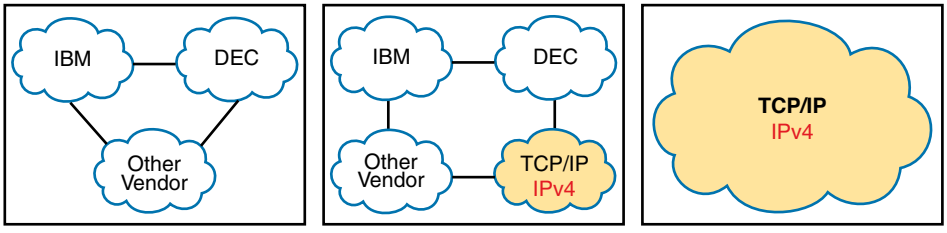


Figure 24-1 *Migration of Enterprise Networks to Use TCP/IP Stack Only, IPv4*

The emergence of IPv6 requires that IPv6 be implemented in end-user hosts, servers, routers, and other devices. However, corporations cannot just migrate all devices from IPv4 to IPv6 over one weekend. Instead, what will likely occur is some kind of long-term migration and coexistence, in which for a large number of years, most corporate networks again use multiple protocol stacks—one based on IPv4 and one based on IPv6.

Eventually, over time, we might all see the day when enterprise networks run only IPv6, without any IPv4 remaining, but that day might take awhile. Figure 24-2 shows the progression, just to make the point, but who knows how long it will take?

One way to add IPv6 support to an established IPv4-based enterprise internetwork is to implement a *dual-stack* strategy. To do so, the routers can be configured to route IPv6 packets, with IPv6 addresses on their interfaces, with a similar model to how routers support IPv4. Then hosts can implement IPv6 when ready, running both IPv4 and IPv6 (dual stacks). The first major section of this chapter shows how to configure and verify unicast IPv6 addresses on routers.

Answers to the "Do I Know This Already?" quiz:

1 A **2** B **3** A, C **4** A **5** B

Key Topic

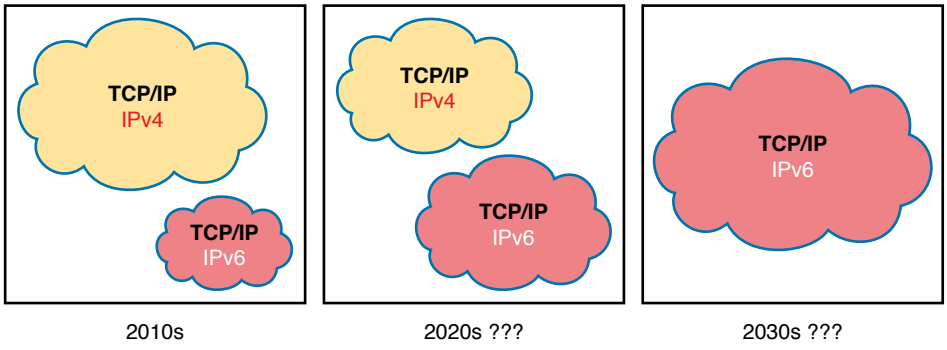


Figure 24-2 *Possible Path Through Dual-Stack (IPv4 and IPv6) over a Long Period*

Static Unicast Address Configuration

Cisco routers give us two options for static configuration of IPv6 addresses. In one case, you configure the full 128-bit address, while in the other, you configure a 64-bit prefix and let the router derive the second half of the address (the interface ID). The next few pages show how to configure both options and how the router chooses the second half of the IPv6 address.

Configuring the Full 128-Bit Address

To statically configure the full 128-bit unicast address—either global unicast or unique local—the router needs an **ipv6 address** *address/prefix-length* interface subcommand on each interface. The address can be an abbreviated IPv6 address or the full 32-digit hex address. The command includes the prefix length value, at the end, with no space between the address and prefix length.

The configuration of the router interface IPv6 address really is that simple. Figure 24-3, along with Examples 24-1 and 24-2, shows a basic example. The figure shows the global unicast IPv6 address used by two different routers, on two interfaces each. As usual, all subnets use a /64 prefix length.

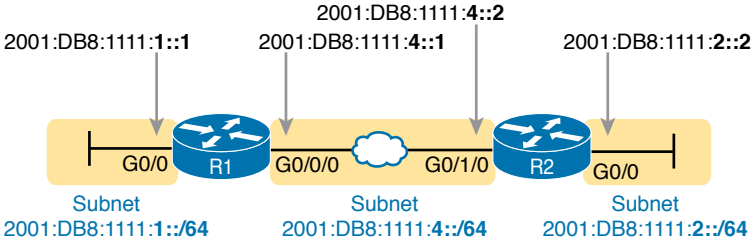


Figure 24-3 *Sample 128-bit IPv6 Addresses to Be Configured on Cisco Router Interfaces*

Example 24-1 *Configuring Static IPv6 Addresses on R1*

```
ipv6 unicast-routing
!
interface GigabitEthernet0/0
 ipv6 address 2001:DB8:1111:1::1/64
!
interface GigabitEthernet0/0/0
 ipv6 address 2001:0db8:1111:0004:0000:0000:0000:0001/64
```

Example 24-2 *Configuring Static IPv6 Addresses on R2*

```
ipv6 unicast-routing
!
interface GigabitEthernet0/0
 ipv6 address 2001:DB8:1111:2::2/64
!
interface GigabitEthernet0/1/0
 ipv6 address 2001:db8:1111:4::2/64
```

NOTE The configuration on R1 in Example 24-1 uses both abbreviated and unabbreviated addresses, and both lowercase and uppercase hex digits, showing that all are allowed. Router **show** commands list the abbreviated value with uppercase hex digits.

Enabling IPv6 Routing

While the configurations shown in Examples 24-1 and 24-2 focus on the IPv6 address configuration, they also include an important but often overlooked step when configuring IPv6 on Cisco routers: IPv6 routing needs to be enabled. On Cisco routers, IPv4 routing is enabled by default, but IPv6 routing is not enabled by default. The solution takes only a single command—**ipv6 unicast-routing**—which enables IPv6 routing on the router.

A router must enable IPv6 globally (**ipv6 unicast-routing**) and enable IPv6 on the interface (**ipv6 address**) before the router will attempt to route IPv6 packets in and out an interface. If you omit the **ipv6 unicast-routing** command but configure interface IPv6 addresses, the router will not route any received IPv6 packets, but the router will act as an IPv6 host. If you include the **ipv6 unicast-routing** command but omit all the interface IPv6 addresses, the router will be ready to route IPv6 packets but have no interfaces that have IPv6 enabled, effectively disabling IPv6 routing.

Verifying the IPv6 Address Configuration

IPv6 uses many **show** commands that mimic the syntax of IPv4 **show** commands. For example:

- The **show ipv6 interface brief** command gives you interface IPv6 address info, but not prefix length info, similar to the IPv4 **show ip interface brief** command.
- The **show ipv6 interface** command gives the details of IPv6 interface settings, much like the **show ip interface** command does for IPv4.

The one notable difference in the most common commands is that the **show interfaces** command still lists the IPv4 address and mask but tells us nothing about IPv6. So, to see IPv6 interface addresses, use commands that begin with **show ipv6**. Example 24-3 lists a few samples from Router R1, with the explanations following.

Example 24-3 *Verifying Static IPv6 Addresses on Router R1*

```
! The first interface is in subnet 1
R1# show ipv6 interface GigabitEthernet 0/0
GigabitEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::1:AAFF:FE00:1
  No Virtual link-local address(es):
  Global unicast address(es):
    2001:DB8:1111:1::1, subnet is 2001:DB8:1111:1::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF00:1
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachables are sent
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds (using 30000)
  ND advertised reachable time is 0 (unspecified)
  ND advertised retransmit interval is 0 (unspecified)
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  ND advertised default router preference is Medium
  Hosts use stateless autoconfig for addresses.

R1# show ipv6 interface brief
GigabitEthernet0/0      [up/up]
    FE80::1:AAFF:FE00:1
    2001:DB8:1111:1::1
GigabitEthernet0/1      [administratively down/down]
    unassigned
GigabitEthernet0/0/0    [up/up]
    FE80::32F7:DFF:FE29:8568
    2001:DB8:1111:4::1
GigabitEthernet0/1/0    [administratively down/down]
    unassigned
```

First, focus on the output of the two **show ipv6 interface** commands at the top of the example, which lists interface G0/0, showing output about that interface only. Note that the output lists the configured IPv6 address and prefix length, as well as the IPv6 subnet (2001:DB8:1111:1::/64), which the router calculated based on the IPv6 address.

The end of the example lists the output of the **show ipv6 interface brief** command. Similar to the IPv4-focused **show ip interface brief** command, this command lists IPv6 addresses, but not the prefix length or prefixes. This command also lists all interfaces on the router, whether or not IPv6 is enabled on the interfaces. For example, in this case, the only two interfaces on R1 that have an IPv6 address are G0/0 and G0/0/0, as configured earlier in Example 24-1.

Beyond the IPv6 addresses on the interfaces, the router also adds IPv6 connected routes to the IPv6 routing table off each interface. Just as with IPv4, the router keeps these connected routes in the IPv6 routing table only when the interface is in a working (up/up) state. But if the interface has an IPv6 unicast address configured, and the interface is working, the router adds the connected routes. Example 24-4 shows the connected IPv6 on Router R1 from Figure 24-3.

Example 24-4 *Displaying Connected IPv6 Routes on Router R1*

```
R1# show ipv6 route connected
IPv6 Routing Table - default - 5 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, HA - Home Agent, MR - Mobile Router, R - RIP
       H - NHRP, I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
       IS - ISIS summary, D - EIGRP, EX - EIGRP external, NM - NEMO
       ND - ND Default, NDp - ND Prefix, DCE - Destination, NDr - Redirect
       RL - RPL, O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1
       OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
       la - LISP alt, lr - LISP site-registrations, ld - LISP dyn-eid
       lA - LISP away, a - Application
C   2001:DB8:1111:1::/64 [0/0]
     via GigabitEthernet0/0, directly connected
C   2001:DB8:1111:4::/64 [0/0]
     via GigabitEthernet0/0/0, directly connected
```

Generating a Unique Interface ID Using Modified EUI-64

IPv6 follows the same general model as IPv4 regarding which types of devices typically use static, predefined addresses and which use dynamically learned address. For example, routers inside an enterprise use static IPv4 addresses, while end-user devices typically learn their IPv4 address using DHCP. With IPv6, routers also typically use static IPv6 addresses, while user devices use DHCP or Stateless Address Auto Configuration (SLAAC) to dynamically learn their IPv6 address.

Even though engineers typically choose to use stable and predictable IPv6 interface addresses, IOS supports two different methods to configure a stable address. One method uses the **ipv6 address** command to define the entire 128-bit address, as shown in Examples 24-1 and 24-2. The other method uses this same **ipv6 address** command, but the command configures only the 64-bit IPv6 prefix for the interface and lets the router automatically generate a unique interface ID.

This second method uses rules called *modified EUI-64* (extended unique identifier). Often, in the context of IPv6 addressing, people refer to modified EUI-64 as just EUI-64; there is no other term or concept about EUI-64 that you need to know for IPv6. The configuration that uses EUI-64 includes a keyword to tell the router to use EUI-64 rules, along with the 64-bit prefix. The router then uses EUI-64 rules to create the interface ID part of the address, as follows:

Key Topic

1. Split the 6-byte (12-hex-digit) MAC address in two halves (6 hex digits each).
2. Insert FFFE in between the two, making the interface ID now have a total of 16 hex digits (64 bits).
3. Invert the seventh bit of the interface ID.

Figure 24-4 shows the major pieces of how the address is formed.

Key Topic

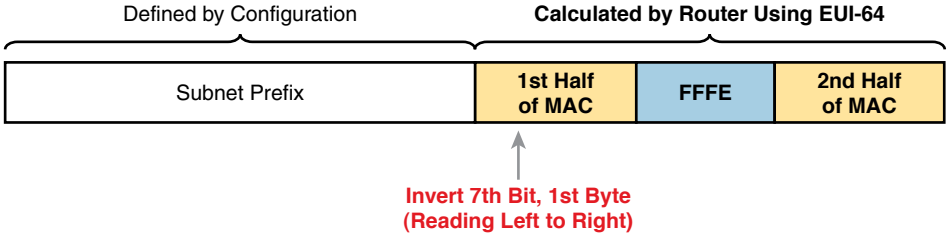


Figure 24-4 *IPv6 Address Format with Interface ID and EUI-64*

NOTE You can find a video about the EUI-64 process on the companion website, in the Chapter Review section for this chapter.

Although this process might seem a bit convoluted, it works. Also, with a little practice, you can look at an IPv6 address and quickly notice the FFFE in the middle of the interface ID and then easily find the two halves of the corresponding interface's MAC address. But you need to be ready to do the same math, in this case to predict the EUI-64 formatted IPv6 address on an interface.

For example, if you ignore the final step of inverting the seventh bit, the rest of the steps just require that you move the pieces around. Figure 24-5 shows two examples, just so you see the process.

Key Topic

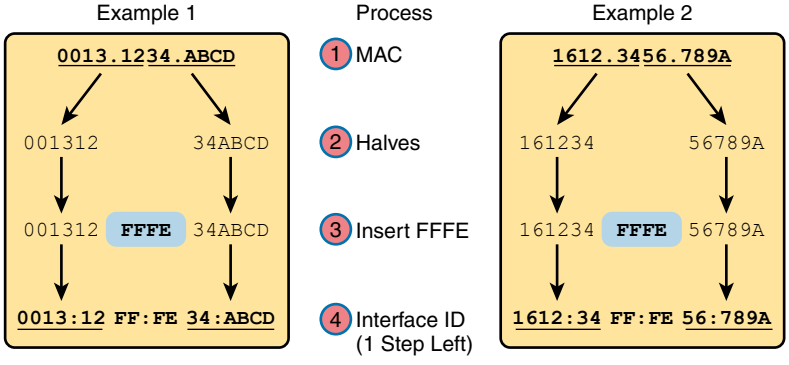


Figure 24-5 *Two Examples of Most of the EUI-64 Interface ID Process*

Both examples follow the same process. Each starts with the MAC address, breaking it into two halves (Step 2). The third step inserts FFFE in the middle, and the fourth step inserts a colon every four hex digits, keeping with IPv6 conventions.

While the examples in Figure 24-5 show most of the steps, they omit the final step. The final step requires that you convert the first byte (first two hex digits) from hex to binary, invert the seventh of the 8 bits, and convert the bits back to hex. Inverting a bit means that if the bit is a 0, make it a 1; if it is a 1, make it a 0. Most of the time, with IPv6 addresses, the original bit will be 0 and will be inverted to a 1.

For example, Figure 24-6 completes the two examples from Figure 24-5, focusing only on the first two hex digits. The examples show each pair of hex digits (Step 1) and the binary equivalent (Step 2). Step 3 shows a copy of those same 8 bits, except the seventh bit is inverted; the example on the left inverts from 0 to 1, and the example on the right inverts from 1 to 0. Finally, the bits are converted back to hex at Step 4.

Key Topic

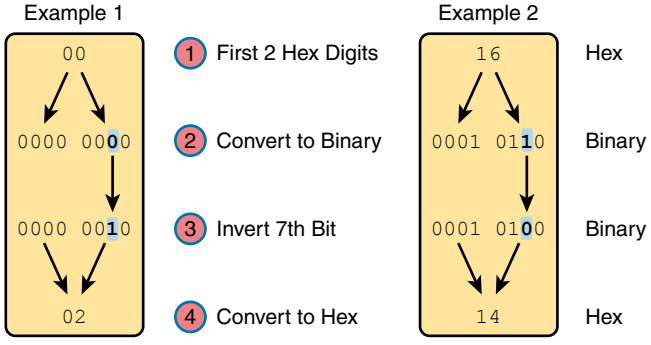


Figure 24-6 *Inverting the Seventh Bit of an EUI-64 Interface ID Field*

NOTE If you do not remember how to do hex-to-binary conversions, take a few moments to review the process. If you memorize the 16 hex values for digits 0 through F, with the corresponding binary values, the conversion can be easy. If you do not have those handy in your memory, take a few moments to look at Table A-2 in Appendix A, "Numeric Reference Tables."

For those of you who prefer the decimal shortcuts, with a little memorization you can do the bit-flip math without doing any hex-binary conversions. First, note that the process to invert the seventh bit, when working with a hexadecimal IPv6 address, flips the third of 4 bits in a single hex digit. With only 16 single hex digits, you could memorize what each hex digit becomes if its third bit is inverted, and you can easily memorize those values with a visual process.

If you want to try to memorize the values, it helps to work through the following process a few times, so grab a piece of scratch paper. Then write the 16 single hex digits as shown on the left side of Figure 24-7. That is, write them in eight rows of two numbers each, with the spacing as directed in the figure.

Next, start at the top of the lists and draw arrow lines between two numbers in the same column on the top left (0 and 2). Then move down the left-side column, connecting the next two digits (4 and 6) with an arrow line, then 8 and A, and then C and E. Repeat the process on the right, re-creating the right side of Figure 24-7.

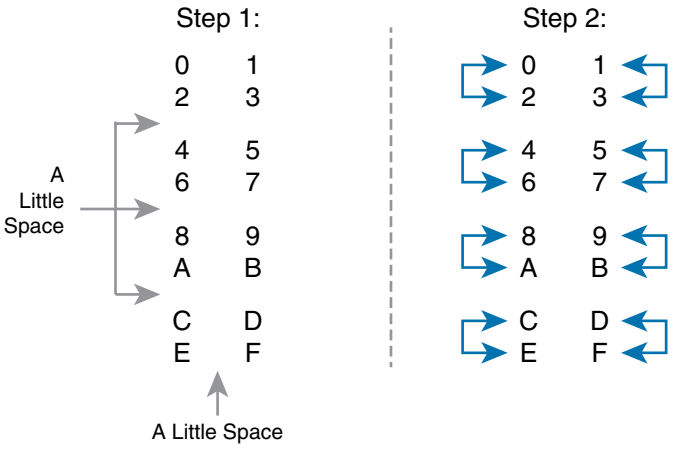


Figure 24-7 *A Mnemonic Device to Help Memorize Bit Inversion Shortcut*

The figure you drew (and the right side of Figure 24-7) shows the hex digits which, when you invert their third bit, convert to the other. That is, 0 converts to 2; 2 converts to 0; 1 converts to 3; 3 converts to 1; 4 converts to 6; 6 converts to 4; and so on. So, on the exam, if you can remember the pattern to redraw Figure 24-7, you could avoid doing binary/hexadecimal conversion. Use whichever approach makes you more comfortable.

As usual, the best way to get comfortable with forming these EUI-64 interface IDs is to calculate some yourself. Table 24-2 lists some practice problems, with an IPv6 64-bit prefix in the first column and the MAC address in the second column. Your job is to calculate the full (unabbreviated) IPv6 address using EUI-64 rules. The answers are at the end of the chapter, in the section "Answers to Earlier Practice Problems."

Table 24-2 IPv6 EUI-64 Address Creation Practice

Prefix	MAC Address	Unabbreviated IPv6 Address
2001:DB8:1:1::/64	0013.ABAB.1001	
2001:DB8:1:1::/64	AA13.ABAB.1001	
2001:DB8:1:1::/64	000C.BEEF.CAFE	
2001:DB8:1:1::/64	B80C.BEEF.CAFE	
2001:DB8:FE:FE::/64	0C0C.ABAC.CABA	
2001:DB8:FE:FE::/64	0A0C.ABAC.CABA	

Configuring a router interface to use the EUI-64 format uses the **ipv6 address** *address/prefix-length* **eui-64** interface subcommand. The **eui-64** keyword tells the router to find the interface MAC address and do the EUI-64 conversion math to find the interface ID.

Example 24-5 shows a revised configuration on Router R1, as compared to the earlier Example 24-1. In this case, R1 uses EUI-64 formatting for its IPv6 addresses.

Example 24-5 *Configuring R1's IPv6 Interfaces Using EUI-64*

```
ipv6 unicast-routing
!
! The ipv6 address command now lists a prefix, not the full address
interface GigabitEthernet0/0
 mac-address 0201.aa00.0001
 ipv6 address 2001:DB8:1111:1::/64 eui-64
!
interface GigabitEthernet0/0/0
 ipv6 address 2001:DB8:1111:4::/64 eui-64

R1# show ipv6 interface brief
GigabitEthernet0/0      [up/up]
    FE80::1:AAFF:FE00:1
    2001:DB8:1111:1:1:AAFF:FE00:1
GigabitEthernet0/1      [administratively down/down]
    unassigned
GigabitEthernet0/0/0    [up/up]
    FE80::32F7:DFF:FE29:8568
    2001:DB8:1111:4:32F7:DFF:FE29:8568
GigabitEthernet0/0/1    [administratively down/down]
    unassigned
```

The example uses only Ethernet interfaces, all of which have a universal MAC address to use to create their EUI-64 interface IDs. However, in this case, the configuration includes the **mac-address** command under R1's G0/0 interface, which causes IOS to use the configured MAC address instead of the universal (burned-in) MAC address. Interface G0/0/0 defaults to use its universal MAC address. Following that math:

G0/0 – MAC `0201.AA00.0001` – Interface ID `0001.AAFF.FE00.0001`

G0/0 – MAC `30F7.0D29.8568` – Interface ID `32F7.0DFF.FE29.8568`

Also, be aware that for interfaces that do not have a MAC address, like serial interfaces, the router uses the MAC of the lowest-numbered router interface that does have a MAC.

NOTE When you use EUI-64, the address value in the **ipv6 address** command should be the prefix, not the full 128-bit IPv6 address. However, if you mistakenly type the full address and still use the **eui-64** keyword, IOS accepts the command and converts the address to the matching prefix before putting the command into the running config file. For example, IOS converts **ipv6 address 2000:1:1:1::1/64 eui-64** to **ipv6 address 2000:1:1:1::/64 eui-64**.

Dynamic Unicast Address Configuration

In most cases, network engineers will configure the IPv6 addresses of router interfaces so that the addresses do not change until the engineer changes the router configuration. However, routers can be configured to use dynamically learned IPv6 addresses. These can be

useful for routers connecting to the Internet through some types of Internet access technologies, like DSL and cable modems.

Cisco routers support two ways for the router interface to dynamically learn an IPv6 address to use:

- Stateful DHCP
- Stateless Address Autoconfiguration (SLAAC)

Both methods use the familiar **ipv6 address** command. Of course, neither option configures the actual IPv6 address; instead, the commands configure a keyword that tells the router which method to use to learn its IPv6 address. Example 24-6 shows the configuration, with one interface using stateful DHCP and one using SLAAC.

Example 24-6 *Router Configuration to Learn IPv6 Addresses with DHCP and SLAAC*

```
! This interface uses DHCP to learn its IPv6 address
interface FastEthernet0/0
 ipv6 address dhcp
!
! This interface uses SLAAC to learn its IPv6 address
interface FastEthernet0/1
 ipv6 address autoconfig
```

Special Addresses Used by Routers

IPv6 configuration on a router begins with the simple steps discussed in the first part of this chapter. After you configure the **ipv6 unicast-routing** global configuration command, to enable the function of IPv6 routing, the addition of a unicast IPv6 address on an interface causes the router to do the following:

Key Topic

- Gives the interface a unicast IPv6 address
- Enables the routing of IPv6 packets in/out that interface
- Defines the IPv6 prefix (subnet) that exists off that interface
- Tells the router to add a connected IPv6 route for that prefix, to the IPv6 routing table, when that interface is up/up

NOTE In fact, if you pause and look at the list again, the same ideas happen for IPv4 when you configure an IPv4 address on a router interface.

While all the IPv6 features in this list work much like similar features in IPv4, IPv6 also has a number of additional functions not seen in IPv4. Often, these additional functions use other IPv6 addresses, many of which are multicast addresses. This second major section of the chapter examines the additional IPv6 addresses seen on routers, with a brief description of how they are used.

Link-Local Addresses

IPv6 uses link-local addresses as a special kind of unicast IPv6 address. These addresses are not used for normal IPv6 packet flows that contain data for applications. Instead, these addresses are used by some overhead protocols and for routing. This next topic first looks at how IPv6 uses link-local addresses and then how routers create link-local addresses.

Link-Local Address Concepts

IPv6 defines rules so that packets sent to any link-local address should not be forwarded by any router to another subnet. As a result, several IPv6 protocols make use of link-local addresses when the protocol's messages need to stay within the local LAN. For example, Neighbor Discovery Protocol (NDP), which replaces the functions of IPv4's ARP, uses link-local addresses.

Routers also use link-local addresses as the next-hop IP addresses in IPv6 routes, as shown in Figure 24-8. IPv6 hosts also use a default router (default gateway) concept, like IPv4, but instead of the router address being in the same subnet, hosts refer to the router's link-local address. The **show ipv6 route** command lists the link-local address of the neighboring router, rather than the global unicast or unique local unicast address.

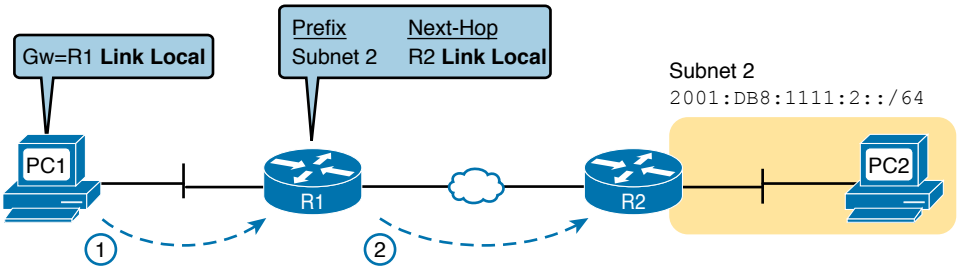


Figure 24-8 *IPv6 Using Link-Local Addresses as the Next-Hop Address*

Following are some key facts about link-local addresses:

Key Topic

Unicast (not multicast): Link-local addresses represent a single host, and packets sent to a link-local address should be processed by only that one IPv6 host.

Forwarding scope is the local link only: Packets sent to a link-local address do not leave the local data link because routers do not forward packets with link-local destination addresses.

Automatically generated: Every IPv6 host interface (and router interface) can create its own link-local address automatically, solving some initialization problems for hosts before they learn a dynamically learned global unicast address.

Common uses: Link-local addresses are used for some overhead protocols that stay local to one subnet and as the next-hop address for IPv6 routes.

Creating Link-Local Addresses on Routers

IPv6 hosts and routers can calculate their own link-local address, for each interface, using some basic rules. First, all link-local addresses start with the same prefix, as shown on the left side of Figure 24-9. By definition, the first 10 bits must match prefix FE80::/10, meaning

that the first three hex digits will be either FE8, FE9, FEA, or FEB. However, when following the RFC, the next 54 bits should be binary 0, so the link-local address should always start with FE80:0000:0000:0000 as the first four unabbreviated quartets.

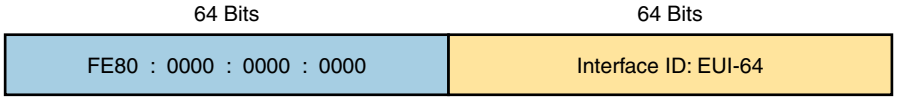


Figure 24-9 *Link-Local Address Format*

The second half of the link-local address, in practice, can be formed using EUI-64 rules, can be randomly generated, or even configured. Cisco routers use the EUI-64 format to create the interface ID (see the earlier section "Generating a Unique Interface ID Using Modified EUI-64"). As a result, a router's complete link-local address should be unique because the MAC address that feeds into the EUI-64 process should be unique.

Alternately, some OSs create their link-local addresses by randomly generating the interface ID. For example, Microsoft OSs use a somewhat random process to choose the interface ID and change it over time in an attempt to prevent some forms of attacks.

IOS creates a link-local address for any interface that has configured at least one other unicast address using the **ipv6 address** command (global unicast or unique local). To see the link-local address, just use the usual commands that also list the unicast IPv6 address: **show ipv6 interface** and **show ipv6 interface brief**. Note that Example 24-7 shows an example from Router R1 just after it was configured as shown in Example 24-5 (with the **eui-64** keyword on the **ipv6 address** commands).

Example 24-7 *Comparing Link-Local Addresses with EUI-Generated Unicast Addresses*

```
R1# show ipv6 interface brief
GigabitEthernet0/0      [up/up]
    FE80::1:AAFF:FE00:1
    2001:DB8:1111:1:1:AAFF:FE00:1
GigabitEthernet0/1      [administratively down/down]
    unassigned
GigabitEthernet0/0/0    [up/up]
    FE80::32F7:DFF:FE29:8568
    2001:DB8:1111:4:32F7:DFF:FE29:8568
GigabitEthernet0/0/1    [administratively down/down]
    unassigned
```


First, examine the two pairs of highlighted entries in the example. For each of the two interfaces that have a global unicast address (G0/0 and G0/0/0), the output lists the global unicast, which happens to begin with 2001 in this case. At the same time, the output also lists the link-local address for each interface, beginning with FE80.

Next, focus on the two addresses listed under interface G0/0. If you look closely at the second half of the two addresses listed for interface G0/0, you will see that both addresses have the same interface ID value. The global unicast address was configured in this case with the

ipv6 address 2001:DB8:1111:1::/64 eui-64 command, so the router used EUI-64 logic to form both the global unicast address and the link-local address. The interface MAC address in this case is 0201.AA00.0001, so the router calculates an interface ID portion of both addresses as 0001:AAFF:FE00:0001 (unabbreviated). After abbreviation, Router R1's link-local address on interface G0/0 becomes FE80::AAFF:FE00:1.

IOS can either automatically create the link-local address, or it can be configured. IOS chooses the link-local address for the interface based on the following rules:

- If configured, the router uses the value in the **ipv6 address** *address* **link-local** interface subcommand. Note that the configured link-local address must be from the correct address range for link-local addresses; that is, an address from prefix FE80::/10. In other words, the address must begin with FE8, FE9, FEA, or FEB.
- If not configured, the IOS calculates the link-local address using EUI-64 rules, as discussed and demonstrated in and around Example 24-7. The calculation uses EUI-64 rules even if the interface unicast address does not use EUI-64.

Routing IPv6 with Only Link-Local Addresses on an Interface

This chapter has shown four variations on the **ipv6 address** command so far. To review:

ipv6 address *address/prefix-length*: Static configuration of a specific address

ipv6 address *prefix/prefix-length* **eui-64:** Static configuration of a specific prefix and prefix length, with the router calculating the interface ID using EUI-64 rules

ipv6 address dhcp: Dynamic learning on the address and prefix length using DHCP

ipv6 address autoconfig: Dynamic learning of the prefix and prefix length, with the router calculating the interface ID using EUI-64 rules (SLAAC)

This next short topic completes the list with the following command:

ipv6 enable: Enables IPv6 processing and adds a link-local address, but adds no other unicast IPv6 addresses.

The purpose of the **ipv6 enable** command will not make sense until you realize that some links, particularly WAN links, do not need a global unicast address. Using the backdrop of Figure 24-10, think about the destination of packets sent by hosts like PC1 and PC2. When PC1 sends PC2 an IPv6 packet, the packet holds PC1's and PC2's IPv6 addresses and never contains the WAN link's IPv6 addresses. PC1 and PC2 may need to know the routers' LAN IPv6 addresses, to use as their default gateway, but the hosts do not need to know the routers' WAN interface addresses.

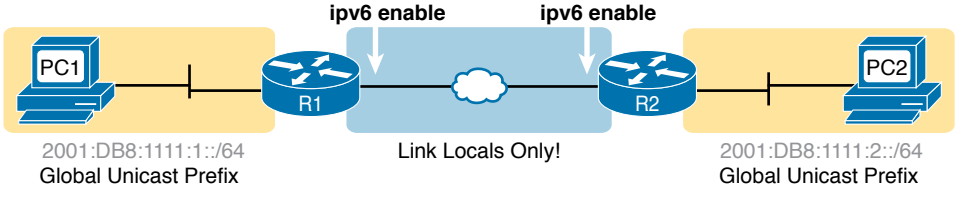


Figure 24-10 *Typical Use of the* **ipv6 enable** *Command*

Additionally, the routers do not need to have global unicast (or unique local) addresses on the WAN links for routing to work. IPv6 routing protocols use link-local addresses as the next-hop address when dynamically building IPv6 routes. Additionally, static routes, as discussed in Chapter 25, "Implementing IPv6 Routing," can use link-local addresses for the next-hop address.

In short, creating a WAN link with no global unicast (or unique local) addresses works. As a result, you would not even need to assign an IPv6 subnet to each WAN link. Then to configure the WAN interfaces, use the **ipv6 enable** command, enabling IPv6 and giving each interface a generated link-local IPv6 address.

To use the command, just configure the **ipv6 enable** command on the interfaces on both ends of the WAN link.

IPv6 Multicast Addresses

IPv6 uses multicast IPv6 addresses for several purposes. Like IPv4, IPv6 includes a range of multicast addresses that can be used by multicast applications, with many of the same fundamental concepts as IPv4 multicasts. For instance, IANA defines the range FF30::/12 (all IPv6 addresses that begin with FF3) as the range of addresses to be used for some types of multicast applications.

Additionally, different IPv6 RFCs reserve multicast addresses for specific purposes. For instance, OSPFv3 uses FF02::5 and FF02::6 as the all-OSPF-routers and all-DR-Routers multicast addresses, respectively, similar to how OSPFv2 uses IPv4 addresses 224.0.0.5 and 224.0.0.6 for the equivalent purposes.

This next section focuses on IPv6 multicast addresses reserved for use with different protocols. The first, link-local multicast addresses, are multicast addresses useful for communicating over a single link. The other type is a special overhead multicast address calculated for each host, called the solicited-node multicast address.

Reserved Multicast Addresses

Stop for a moment and think about some of the control plane protocols discussed throughout this book so far. Some of those IPv4 control plane protocols used IPv4 broadcasts, meaning that the packet destination address was either 255.255.255.255 (the address for all hosts in the local LAN) or the subnet broadcast address (the address for all hosts in that specific subnet). Those broadcast packets were then sent as Ethernet broadcast frames, destined to the Ethernet broadcast address of FFFF.FFFF.FFFF.

While useful, the IPv4 approach of IPv4 broadcast and LAN broadcast requires every host in the VLAN to process the broadcast frame, even if only one other device needed to think about the message. Also, each host has to process the frame, then packet, read the type of message, and so on, before ignoring the task. For example, an IPv4 ARP Request—an IPv4 and LAN broadcast—requires a host to process the Ethernet, IP, and ARP details of the message before deciding whether to reply or not.

IPv6, instead of using Layer 3 and Layer 2 broadcasts, instead uses Layer 3 multicast addresses, which in turn cause Ethernet frames to use Ethernet multicast addresses. As a result:

- All the hosts that should receive the message receive the message, which is necessary for the protocols to work. However…
- …Hosts that do not need to process the message can make that choice with much less processing as compared to IPv4.

For instance, OSPFv3 uses IPv6 multicast addresses FF02::5 and FF02::6. In a subnet, the OSPFv3 routers will listen for packets sent to those addresses. However, all the endpoint hosts do not use OSPFv3 and should ignore those OSPFv3 messages. If a host receives a packet with FF02::5 as the destination IPv6 address, the host can ignore the packet because the host knows it does not care about packets sent to that multicast address. That check takes much less time than the equivalent checks with IPv4.

Table 24-3 lists the most common reserved IPv6 multicast addresses.

Table 24-3 Key IPv6 Local-Scope Multicast Addresses

Short Name	Multicast Address	Meaning	IPv4 Equivalent
All-nodes	FF02::1	All-nodes (all interfaces that use IPv6 that are on the link)	224.0.0.1
All-routers	FF02::2	All-routers (all IPv6 router interfaces on the link)	224.0.0.2
All-OSPF, All-OSPF-DR	FF02::5, FF02::6	All OSPF routers and all OSPF-designated routers, respectively	224.0.0.5, 224.0.0.6
RIPng Routers	FF02::9	All RIPng routers	224.0.0.9
EIGRPv6 Routers	FF02::A	All routers using EIGRP for IPv6 (EIGRPv6)	224.0.0.10
DHCP Relay Agent	FF02::1:2	All routers acting as a DHCPv6 relay agent	None

NOTE An Internet search of "IPv6 Multicast Address Space Registry" will show the IANA page that lists all the reserved values and the RFC that defines the use of each address.

Example 24-8 repeats the output of the **show ipv6 interface** command to show the multicast addresses used by Router R1 on its G0/0 interface. In this case, the highlighted lines show the all-nodes address (FF02::1), all-routers (FF02::2), and two for OSPFv3 (FF02::5 and FF02::6). Note that the IPv6 multicast addresses that the router interface is listening for and processing are listed under the heading "Joined group address(es):" at the top of the highlighted section of the output.

Example 24-8 *Verifying Static IPv6 Addresses on Router R1*

```
R1# show ipv6 interface GigabitEthernet 0/0
GigabitEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::1
  No Virtual link-local address(es):
  Global unicast address(es):
    2001:DB8:1111:1::1, subnet is 2001:DB8:1111:1::/64 [EUI]
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::5
    FF02::6
    FF02::1:FF00:1
 ! Lines omitted for brevity
```

Multicast Address Scopes

IPv6 RFC 4291 defines IPv6 addressing including the ideas of IPv6 address scope. Each scope defines a different set of rules about whether routers should or should not forward a packet, and how far routers should forward packets, based on those scopes.

For instance, you read earlier in this chapter about the link-local address on an interface—a unicast IPv6 address—but with a link-local scope. The scope definition called "link-local" dictates that packets sent to a link-local unicast address should remain on the link and not be forwarded by any router.

Most of the scope discussion in RFC 4291 applies to multicast addresses, using the term *multicast scope*. Per that RFC, the fourth digit of the multicast address identifies the scope, as noted in Table 24-4.

Key Topic

Table 24-4 IPv6 Multicast Scope Terms

Scope Name	First Quartet	Scope Defined by...	Meaning
Interface-Local	FF01	Derived by Device	Packet remains within the device. Useful for internally sending packets to services running on that same host.
Link-Local	FF02	Derived by Device	Host that creates the packet can send it onto the link, but no routers forward the packet.
Site-Local	FF05	Configuration on Routers	Intended to be more than Link-Local, so routers forward, but must be less than Organization-Local; generally meant to limit packets so they do not cross WAN links.
Organization-Local	FF08	Configuration on Routers	Intended to be broad, probably for an entire company or organization. Must be broader than Site-Local.
Global	FF0E	No Boundaries	No boundaries.

Breaking down the concepts a little further, packets sent to a multicast address with a link-local scope should stay on the local link, that is, the local subnet. Hosts know they can process a link-local packet if received, as do routers. However, routers know to not route the packet to other subnets because of the scope. Packets with an organization-local scope should be routed inside the organization but not out to the Internet or over a link to another company. (Note that routers can predict the boundaries of some scopes, like link-local, but they need configuration to know the boundaries of other scopes, for instance, organization-local.)

Comparing a few of the scopes in terms of where the packets can flow, the higher the value in the fourth hex digit, the further away from the sending host the scope allows the packet to be forwarded. Table 24-4 shows that progression top to bottom, while Figure 24-11 shows an example with three scopes: link-local, site-local, and organization-local. In the figure, site-local messages do not cross the WAN, and organization-local messages do not leave the organization over the link to the Internet.

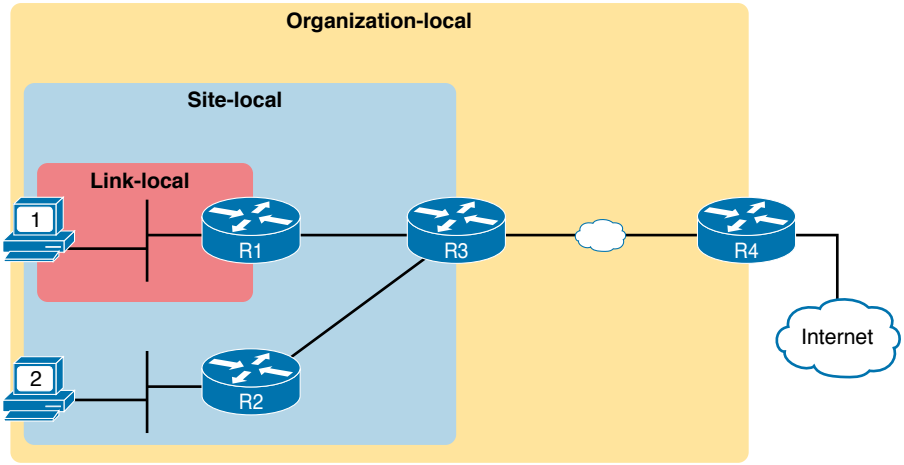


Figure 24-11 *IPv6 Multicast Scopes*

Finally, the term *link-local* has a couple of common uses in IPv6 and can be confusing as a result. The following descriptions should clarify the different uses of the term:

Key Topic

Link-local address: An IPv6 address that begins FE80. This serves as a unicast address for an interface to which devices apply a link-local scope. Devices often create their own link-local addresses using EUI-64 rules. A more complete term for comparison would be *link-local unicast address*.

Link-local multicast address: An IPv6 address that begins with FF02. This serves as a reserved multicast address to which devices apply a link-local scope.

Link-local scope: A reference to the scope itself, rather than an address. This scope defines that routers should not forward packets sent to an address in this scope.

Solicited-Node Multicast Addresses

IPv6 Neighbor Discovery Protocol (NDP) replaces IPv4 ARP, as discussed in Chapter 25. NDP improves the MAC-discovery process by sending IPv6 multicast packets that can be processed by the correct host but discarded with less processing by the rest of the hosts in the subnet. The process uses the solicited-node multicast address associated with the unicast IPv6 address.

Figure 24-12 shows how to determine the solicited node multicast address associated with a unicast address. Start with the predefined /104 prefix (26 hex digits) shown in Figure 24-12. In other words, all the solicited-node multicast addresses begin with the abbreviated FF02::1:FF. In the last 24 bits (6 hex digits), copy the last 6 hex digits of the unicast address into the solicited-node address.

Key Topic

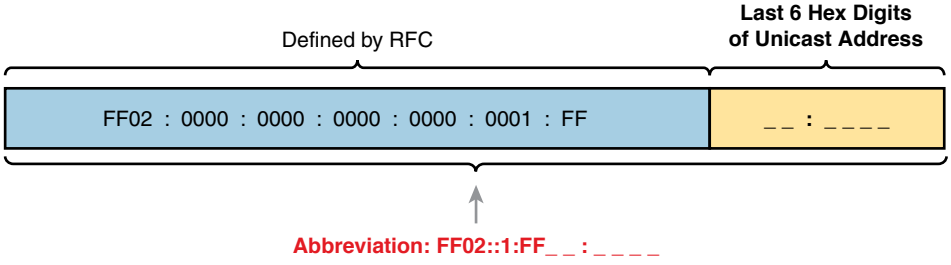


Figure 24-12 *Solicited-Node Multicast Address Format*

Note that a host or router calculates a matching solicited node multicast address for every unicast address on an interface. Example 24-9 shows an example, in which the router interface has a unicast address of 2001:DB8:1111:1::1/64, and a link-local address of FE80::AA:AAAA. As a result, the interface has two solicited node multicast addresses, shown at the end of the output.

Example 24-9 *Verifying Static IPv6 Addresses on Router R1*

```
R1# show ipv6 interface GigabitEthernet 0/0
GigabitEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::AA:AAAA
  No Virtual link-local address(es):
  Global unicast address(es):
    2001:DB8:1111:1::1, subnet is 2001:DB8:1111:1::/64 [TEN]
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::5
    FF02::1:FF00:1
    FF02::1:FFAA:AAAA
 ! Lines omitted for brevity
```

Note that in this case, R1's global unicast address ends with 00:0001 (unabbreviated), resulting in an unabbreviated solicited node multicast address of FF02:0000:0000:0000:0000:0001:FF00:00001. This value begins with the 26-hex-digit prefix shown in Figure 24-12, followed by 00:0001. The solicited node multicast address corresponding to link-local address FE80::AA:AAAA ends in AA:AAAA and is shown in the last line of the example.

Miscellaneous IPv6 Addresses

Together, this chapter and the preceding chapter have introduced most of the IPv6 addressing concepts included in this book. This short topic mentions a few remaining IPv6 addressing ideas and summarizes the topics for easy study.

First, all IPv6 hosts can use two additional special addresses:

Key Topic

- The unknown (unspecified) IPv6 address, ::, or all 0s
- The loopback IPv6 address, ::1, or 127 binary 0s with a single 1

A host can use the unknown address (::) when its own IPv6 address is not yet known or when the host wonders if its own IPv6 address might have problems. For example, hosts use the unknown address during the early stages of dynamically discovering their IPv6 address. When a host does not yet know what IPv6 address to use, it can use the :: address as its source IPv6 address.

The IPv6 loopback address gives each IPv6 host a way to test its own protocol stack. Just like the IPv4 127.0.0.1 loopback address, packets sent to ::1 do not leave the host but are instead simply delivered down the stack to IPv6 and back up the stack to the application on the local host.

Anycast Addresses

Imagine that routers collectively need to implement some service. Rather than have one router supply that service, that service works best when implemented on several routers. But the hosts that use the service need to contact only the nearest such service, and the network wants to hide all these details from the hosts. Hosts can send just one packet to an IPv6 address, and the routers will forward the packet to the nearest router that supports that service by virtue of supporting that destination IPv6 address.

IPv6 anycast addresses provide that exact function. The *any* part of the name refers to the fact that any instances of the service can be used. Figure 24-13 shows this big concept, with two major steps:

Step 1. Two routers configure the exact same IPv6 address, designated as an anycast address, to support some service.

Step 2. In the future, when any router receives a packet for that anycast address, the other routers simply route the packet to the nearest router that supports the address.

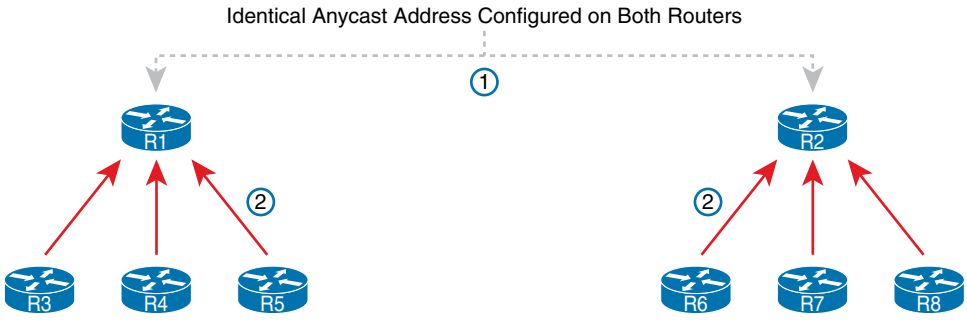


Figure 24-13 *IPv6 Anycast Addresses*

To make this anycast process work, the routers implementing the anycast address must be configured and then advertise a route for the anycast address. The addresses do not come from a special reserved range of addresses; instead, they are from the unicast address range. Often, the address is configured with a /128 prefix so that the routers advertise a host route for that one anycast address. At that point, the routing protocol advertises the route just like any other IPv6 route; the other routers cannot tell the difference.

Example 24-10 shows a sample configuration on a router. Note that the actual address (2001:1:1:2::99) looks like any other unicast address; the value can be chosen like any other IPv6 unicast addresses. However, note the different **anycast** keyword on the **ipv6 address** command, telling the local router that the address has a special purpose as an anycast address. Finally, note that the **show ipv6 interface** command does identify the address as an anycast address, but the **show ipv6 interface brief** command does not.

Example 24-10 *Configuring and Verifying IPv6 Anycast Addresses*

```
R1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# interface gigabitEthernet 0/0
R1(config-if)# ipv6 address 2001:1:1:1::1/64
R1(config-if)# ipv6 address 2001:1:1:2::99/128 anycast
R1(config-if)# ^Z
R1#
R1# show ipv6 interface g0/0
GigabitEthernet0/0 is up, line protocol is up
    IPv6 is enabled, link-local address is FE80::11FF:FE11:1111
    No Virtual link-local address(es):
    Global unicast address(es):
       2001:1:1:1::1, subnet is 2001:1:1:1::/64
       2001:1:1:2::99, subnet is 2001:1:1:2::99/128 [ANY]
 ! Lines omitted for brevity
R1# show ipv6 interface brief g0/0
GigabitEthernet0/0 [up/up]
     FE80::11FF:FE11:1111
     2001:1:1:1::1
     2001:1:1:2::99
```

NOTE The *subnet router anycast address* is one special anycast address in each subnet. It is reserved for use by routers as a way to send a packet to any router on the subnet. The address's value in each subnet is the same number as the subnet ID; that is, the address has the same prefix value as the other addresses and all binary 0s in the interface ID.

IPv6 Addressing Configuration Summary

This chapter completes the discussion of various IPv6 address types, while showing how to enable IPv6 on interfaces. Many implementations will use the **ipv6 address** command on each router LAN interface, and either that same command or the **ipv6 enable** command on the WAN interfaces. For exam prep, Table 24-5 summarizes the various commands and the automatically generated IPv6 addresses in one place for review and study.

Key Topic

Table 24-5 Summary of IPv6 Address Types and the Commands That Create Them

Type	Prefix/Address Notes	Enabled with What Interface Subcommand
Global unicast	Many prefixes	**ipv6 address** *address/prefix-length* **ipv6 address** *prefix/prefix-length* **eui-64**
Unique Local	FD00::/8	**ipv6 address** *prefix/prefix-length* **eui-64**
Link local	FE80::/10	**ipv6 address** *address* **link-local** Autogenerated by all **ipv6 address** commands Autogenerated by the **ipv6 enable** command
All hosts multicast	FF02::1	Autogenerated by all **ipv6 address** commands
All routers multicast	FF02::2	Autogenerated by all **ipv6 address** commands
Routing protocol multicasts	Various	Added to the interface when the corresponding routing protocol is enabled on the interface
Solicited-node multicast	FF02::1:FF /104	Autogenerated by all **ipv6 address** commands

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 24-6 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 24-6 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 command tables		Book
Review memory tables		Website
Do labs		Blog
Watch video		Website

Review All the Key Topics

Key Topic

Table 24-7 Key Topics for Chapter 24

Key Topic Element	Description	Page Number
Figure 24-2	Conceptual drawing about the need for dual stacks for the foreseeable future	557
List	Rules for creating an IPv6 address using EUI-64 rules	561
Figure 24-4	IPv6 EUI-64 Address Format and Rules	561
Figure 24-5	Conceptual drawing of how to create an IPv6 address using EUI-64 rules	561
Figure 24-6	Example of performing the bit inversion when using EUI-64	562
List	Functions IOS enables when an IPv6 is configured on a working interface	565
List	Key facts about IPv6 link-local addresses	566
Table 24-4	Link-local scope terms and meanings	571
List	Comparisons of the use of the term *link-local*	572
Figure 24-12	Conceptual drawing of how to make a solicited-node multicast address	573
List	Other special IPv6 addresses	574
Table 24-5	IPv6 address summary with the commands that enable each address type	576

Key Terms You Should Know

anycast address, dual stacks, EUI-64, link-local address, link-local scope, link-local multicast address, site-local scope, organization-local scope, interface-local scope, IPv6 address scope, solicited-node multicast address, all-nodes multicast address, all-routers multicast address, subnet-router anycast address

Additional Practice for This Chapter's Processes

For additional practice with IPv6 abbreviations, you may do the same set of practice problems using your choice of tools:

For additional practice with calculating IPv6 address using EUI-64 rules and finding the solicited-node multicast address based on a unicast address, use the exercises in Appendix H, "Practice for Chapter 24: Implementing IPv6 Addressing on Routers." You have two options to use:

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

Application: Navigate to the companion website and open the application "Practice Exercise: EUI-64 and Solicited Node Multicast Problems"

Additionally, you can create your own problems using any real router or simulator: Get into the router CLI, into configuration mode, and configure the **mac-address** *address* and **ipv6 address** *prefix*/**64 eui-64** command. Then predict the IPv6 unicast address, link-local address, and solicited-node multicast address; finally, check your predictions against the **show ipv6 interface** command.

Command References

Tables 24-8 and 24-9 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 24-8 Chapter 24 Configuration Command Reference

Command	Description
ipv6 unicast-routing	Global command that enables IPv6 routing on the router.
ipv6 address *ipv6-address*/*prefix-length* [**eui-64**]	Interface subcommand that manually configures either the entire interface IP address or a /64 prefix with the router building the EUI-64 format interface ID automatically.
ipv6 address *ipv6-address*/*prefix-length* [**anycast**]	Interface subcommand that manually configures an address to be used as an anycast address.
ipv6 enable	Command that enables IPv6 on an interface and generates a link-local address.
ipv6 address dhcp	Interface subcommand that enables IPv6 on an interface, causes the router to use DHCP client processes to try to lease an IPv6 address, and creates a link-local address for the interface.

Table 24-9 Chapter 24 EXEC Command Reference

Command	Description
show ipv6 route [connected] [local]	Lists IPv6 routes, or just the connected routes, or just the local routes.
show ipv6 interface [*type number*]	Lists IPv6 settings on an interface, including link-local and other unicast IP addresses (or for the listed interface).
show ipv6 interface brief [*type number*]	Lists interface status and IPv6 addresses for each interface (or for the listed interface).

Answers to Earlier Practice Problems

Table 24-2, earlier in this chapter, listed several practice problems in which you needed to calculate the IPv6 address based on EUI-64 rules. Table 24-10 lists the answers to those problems.

Table 24-10 Answers to IPv6 EUI-64 Address Creation Practice

Prefix	MAC Address	Unabbreviated IPv6 Address
2001:DB8:1:1::/64	0013.ABAB.1001	2001:DB8:1:1:0213:ABFF:FEAB:1001
2001:DB8:1:1::/64	AA13.ABAB.1001	2001:DB8:1:1:A813:ABFF:FEAB:1001
2001:DB8:1:1::/64	000C.BEEF.CAFE	2001:DB8:1:1:020C:BEFF:FEEF:CAFE
2001:DB8:1:1::/64	B80C.BEEF.CAFE	2001:DB8:1:1:BA0C:BEFF:FEEF:CAFE
2001:DB8:FE:FE::/64	0C0C.ABAC.CABA	2001:DB8:FE:FE:0E0C:ABFF:FEAC:CABA
2001:DB8:FE:FE::/64	0A0C.ABAC.CABA	2001:DB8:FE:FE:080C:ABFF:FEAC:CABA

CHAPTER 25

Implementing IPv6 Routing

3.0 IP Connectivity

3.3 Configure and verify IPv4 and IPv6 static routing

3.3.a Default route

3.3.b Network route

3.3.c Host route

3.3.d Floating static

This last chapter in Part VII of the book completes the materials about IPv6 by examining three major topics. The first section examines IPv6 connected and local routes, similar to IPv4, showing how a router adds both connected and local routes based on each interface IPv6 address. The second major section of this chapter then looks at how to configure static IPv6 routes by typing in commands, in this case using the **ipv6 route** command instead of IPv4's **ip route** command. The final major section examines the Neighbor Discovery Protocol (NDP).

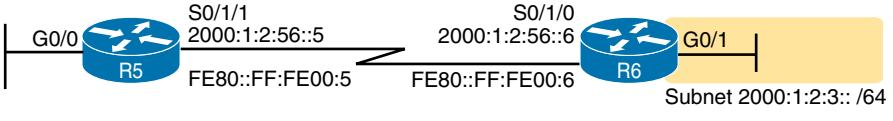
"Do I Know This Already?" Quiz

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 25-1 "Do I Know This Already?" Foundation Topics Section-to-Question Mapping

Foundation Topics Section	Questions
Connected and Local IPv6 Routes	1–2
Static IPv6 Routes	3–6
The Neighbor Discovery Protocol	7–8

Refer to the following figure for questions 1, 3, and 4.



1. A router has been configured with the **ipv6 address 2000:1:2:3::1/64** command on its G0/1 interface as shown in the figure. The router creates a link-local address of FE80::FF:FE00:1 as well. The interface is working. Which of the following routes will the router add to its IPv6 routing table? (Choose two answers.)
   a. A route for 2000:1:2:3::/64
   b. A route for FE80::FF:FE00:1/64
   c. A route for 2000:1:2:3::1/128
   d. A route for FE80::FF:FE00:1/128
2. A router has been configured with the **ipv6 address 3111:1:1:1::1/64** command on its G0/1 interface and **ipv6 address 3222:2:2:2::1/64** on its G0/2 interface. Both interfaces are working. Which of the following routes would you expect to see in the output of the **show ipv6 route connected** command? (Choose two answers.)
   a. A route for 3111:1:1:1::/64
   b. A route for 3111:1:1:1::1/64
   c. A route for 3222:2:2:2::/64
   d. A route for 3222:2:2:2::2/128
3. An engineer needs to add a static IPv6 route for prefix 2000:1:2:3::/64 to Router R5's configuration, in the figure shown with question 1. Which of the following answers shows a valid static IPv6 route for that subnet, on Router R5?
   a. **ipv6 route 2000:1:2:3::/64 S0/1/1**
   b. **ipv6 route 2000:1:2:3::/64 S0/1/0**
   c. **ip route 2000:1:2:3::/64 S0/1/1**
   d. **ip route 2000:1:2:3::/64 S0/1/0**
4. An engineer needs to add a static IPv6 route for prefix 2000:1:2:3::/64 to Router R5 in the figure shown with question 1. Which of the following answers shows a valid static IPv6 route for that subnet on Router R5?
   a. **ipv6 route 2000:1:2:3::/64 2000:1:2:56::5**
   b. **ipv6 route 2000:1:2:3::/64 2000:1:2:56::6**
   c. **ipv6 route 2000:1:2:3::/64 FE80::FF:FE00:5**
   d. **ipv6 route 2000:1:2:3::/64 FE80::FF:FE00:6**

5. An engineer types the command **ipv6 route 2001:DB8:8:8::/64 2001:DB8:9:9::9 129** in configuration mode of Router R1 and presses **Enter**. Later, a **show ipv6 route** command does not list any route for subnet 2001:DB8:8:8::/64. Which of the following could have caused the route to not be in the IPv6 routing table?

   a. The command should be using a next-hop link-local address instead of a global unicast.

   b. The command is missing an outgoing interface parameter, so IOS rejected the **ipv6 route** command.

   c. The router has no routes that match 2001:DB8:9:9::9.

   d. A route for 2001:DB8:8:8::/64 with administrative distance 110 already exists.

6. The command output shows two routes from the longer output of the **show ipv6 route** command. Which answers are true about the output? (Choose two answers.)

```
R1# show ipv6 route static
! Legend omitted for brevity
S 2001:DB8:2:2::/64 [1/0]
 via 2001:DB8:4:4::4
S ::/0 [1/0]
 via Serial0/0/1, directly connected
```

   a. The route to ::/0 is added because of an **ipv6 route** global command.

   b. The administrative distance of the route to 2001:DB8:2:2::/64 is 1.

   c. The route to ::/0 is added because of an **ipv6 address** interface subcommand.

   d. The route to 2001:DB8:2:2::/64 is added because of an IPv6 routing protocol.

7. PC1, PC2, and Router R1 all connect to the same VLAN and IPv6 subnet. PC1 wants to send its first IPv6 packet to PC2. What protocol or message will PC1 use to discover the MAC address to which PC1 should send the Ethernet frame that encapsulates this IPv6 packet?

   a. ARP

   b. NDP NS

   c. NDP RS

   d. SLAAC

8. Which of the following pieces of information does a router supply in an NDP Router Advertisement (RA) message? (Choose two answers.)

   a. Router IPv6 address

   b. Host name of the router

   c. IPv6 prefix(es) on the link

   d. IPv6 address of DHCP server

Foundation Topics

Connected and Local IPv6 Routes

A Cisco router adds IPv6 routes to its IPv6 routing table for several reasons. Many of you could predict those reasons at this point in your reading, in part because the logic mirrors the logic routers use for IPv4. Specifically, a router adds IPv6 routes based on the following:

Key Topic

- The configuration of IPv6 addresses on working interfaces (connected and local routes)
- The direct configuration of a static route (static routes)
- The configuration of a routing protocol, like OSPFv3, on routers that share the same data link (dynamic routes)

The first two sections of this chapter examine the first of these two topics, with discussions of IPv6 routing protocols now residing in the CCNP Enterprise exams.

Rules for Connected and Local Routes

Routers add and remove connected routes and local routes, based on the interface configuration and the interface state. First, the router looks for any configured unicast addresses on any interfaces by looking for the **ipv6 address** command. Then, if the interface is working—if the interface has a "line status is up, protocol status is up" notice in the output of the **show interfaces** command—the router adds both a connected and local route.

NOTE Routers do not create connected or local IPv6 routes for link-local addresses.

The connected and local routes themselves follow the same general logic as with IPv4. The connected route represents the subnet connected to the interface, whereas the local route is a host route for only the specific IPv6 address configured on the interface.

As an example, consider a router, with a working interface, configured with the **ipv6 address 2000:1:1:1::1/64** command. The router will calculate the subnet ID based on this address and prefix length, and it will place a connected route for that subnet (2000:1:1:1::/64) into the routing table. The router also takes the listed IPv6 address and creates a host route for that address, with a /128 prefix length. (With IPv4, host routes have a /32 prefix length, while IPv6 uses a /128 prefix length, meaning "exactly this one address.")

The following list summarizes the rules about how routers create routes based on the configuration of an interface IPv6 unicast address, for easier review and study:

Key Topic

1. Routers create IPv6 routes based on each unicast IPv6 address on an interface, as configured with the **ipv6 address** command, as follows:
   - **A.** The router creates a route for the subnet (a connected route).
   - **B.** The router creates a host route (/128 prefix length) for the router IPv6 address (a local route).
2. Routers do not create routes based on the link-local addresses associated with the interface.
3. Routers remove the connected and local routes for an interface if the interface fails, and they re-add these routes when the interface is again in a working (up/up) state.

Example of Connected IPv6 Routes

While the concept of connected and local IPv6 routes works much like IPv4 routes, seeing a few examples can certainly help. To show some sample routes, Figure 25-1 gives the details of one sample internetwork used in this chapter. The figure shows the IPv6 subnet IDs. The upcoming examples focus on the connected and local routes on Router R1.

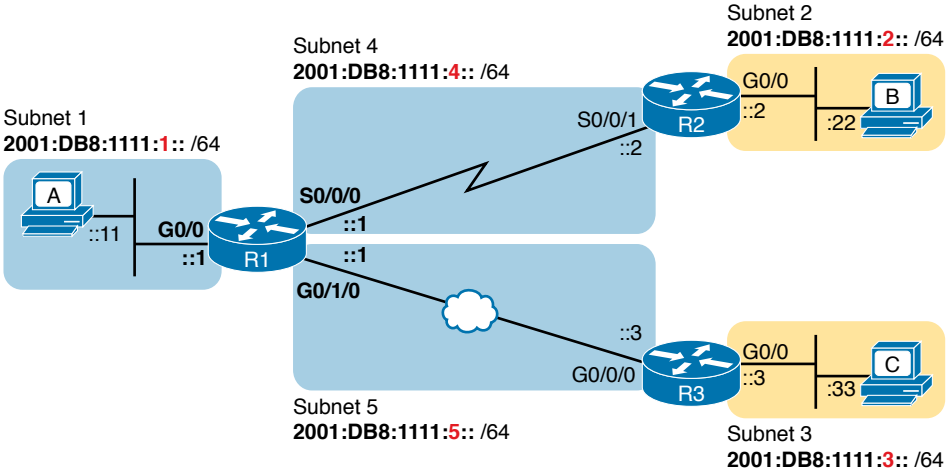


Figure 25-1 *Sample Network Used to Show Connected and Local Routes*

To clarify the notes in Figure 25-1, note that the figure shows IPv6 prefixes (subnets), with a shorthand notation for the interface IPv6 addresses. The figure shows only the abbreviated interface ID portion of each interface address near each interface. For example, R1's G0/0 interface address would begin with subnet ID value 2001:DB8:1111:1, added to ::1, for 2001:DB8:1111:1::1.

Now on to the example of connected routes. To begin, consider the configuration of Router R1 from Figure 25-1, as shown in Example 25-1. The excerpt from the **show running-config** command on R1 shows three interfaces, all of which are working. Also note that no static route or routing protocol configuration exists.

Example 25-1 *IPv6 Addressing Configuration on Router R1*

```
ipv6 unicast-routing
!
interface GigabitEthernet0/0
 ipv6 address 2001:DB8:1111:1::1/64
!
interface Serial0/0/0
 ipv6 address 2001:db8:1111:4::1/64
!
interface GigabitEthernet0/1/0
 ipv6 address 2001:db8:1111:5::1/64
```

Answers to the "Do I Know This Already?" quiz:

1 A, C **2** A, C **3** A **4** B **5** C **6** A, B **7** B **8** A, C

Based on Figure 25-1 and Example 25-1, R1 should have three connected IPv6 routes, as highlighted in Example 25-2.

Example 25-2 *Routes on Router R1 Before Adding Static Routes or Routing Protocols*

```
R1# show ipv6 route
IPv6 Routing Table - default - 7 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, HA - Home Agent, MR - Mobile Router, R - RIP
       H - NHRP, I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
       IS - ISIS summary, D - EIGRP, EX - EIGRP external, NM - NEMO
       ND - ND Default, NDp - ND Prefix, DCE - Destination, NDr - Redirect
       RL - RPL, O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1
       OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
       la - LISP alt, lr - LISP site-registrations, ld - LISP dyn-eid
       lA - LISP away, a - Application
C   2001:DB8:1111:1::/64 [0/0]
     via GigabitEthernet0/0, directly connected
L   2001:DB8:1111:1::1/128 [0/0]
     via GigabitEthernet0/0, receive
C   2001:DB8:1111:4::/64 [0/0]
     via Serial0/0/0, directly connected
L   2001:DB8:1111:4::1/128 [0/0]
     via GigabitEthernet0/0/0, receive
C   2001:DB8:1111:5::/64 [0/0]
     via GigabitEthernet0/1/0, directly connected
L   2001:DB8:1111:5::1/128 [0/0]
     via GigabitEthernet0/1/0, receive
L   FF00::/8 [0/0]
     via Null0, receive
```

All three highlighted routes show the same basic kinds of information, so for discussion, focus on the first pair of highlighted lines, which detail the connected route for subnet 2001:DB8:1111:1::/64. The first pair of highlighted lines state: The route is a "directly connected" route; the interface ID is GigabitEthernet0/0; and the prefix/length is 2001:DB8:1111:1::/64. At the far left, the code letter "C" identifies the route as a connected route (per the legend above). Also note that the numbers in brackets mirror the same ideas as IPv4's **show ip route** command: The first number represents the administrative distance, and the second is the metric.

Examples of Local IPv6 Routes

Continuing this same example, three local routes should exist on R1 for the same three interfaces as the connected routes. Indeed, that is the case, with one extra local route for other purposes. Example 25-3 shows only the local routes, as listed by the **show ipv6 route local** command, with highlights of one particular local route for discussion.

Example 25-3 *Local IPv6 Routes on Router R1*

```
R1# show ipv6 route local
! Legend omitted for brevity

L  2001:DB8:1111:1::1/128 [0/0]
     via GigabitEthernet0/0, receive
L  2001:DB8:1111:4::1/128 [0/0]
     via Serial0/0/0, receive
L  2001:DB8:1111:5::1/128 [0/0]
     via GigabitEthernet0/1/0, receive
L  FF00::/8 [0/0]
     via Null0, receive
```

For the highlighted local route, look for a couple of quick facts. First, look back to R1's configuration in Example 25-1, and note R1's IPv6 address on its G0/0 interface. This local route lists the exact same address. Also note the /128 prefix length, meaning this route matches packets sent to that address (2001:DB8:1111:1::1), and only that address.

NOTE While the **show ipv6 route local** command shows all local IPv6 routes, the **show ipv6 route connected** command shows all connected routes.

Static IPv6 Routes

While routers automatically add connected and local routes based on the interface configuration, static routes require direct configuration with the **ipv6 route** command. Simply put, someone configures the command, and the router places the details from the command into a route in the IPv6 routing table.

The **ipv6 route** command follows the same general logic as does IPv4's **ip route** command, as discussed in Chapter 16, "Configuring IPv4 Addressing and Static Routes." For IPv4, the **ip route** command starts by listing the subnet ID and mask, so for IPv6, the **ipv6 route** command begins with the prefix and prefix length. Then the respective commands list the directions of how this router should forward packets toward that destination subnet or prefix by listing the outgoing interface or the address of the next-hop router.

Figure 25-2 shows the concepts behind a single **ipv6 route** command, demonstrating the concepts behind a static route on Router R1 for the subnet on the right (subnet 2, or 2001:DB8:1111:2::/64). A static route on R1, for this subnet, will begin with **ipv6 route 2001:DB8:1111:2::/64**, followed by either the outgoing interface (S0/0/0) or the next-hop IPv6 address, or both.

Now that you understand the big ideas with IPv6 static routes, the next few pages walk you through a series of examples. In particular, the examples look at configuring static routes with an outgoing interface, then with a next-hop global unicast address, and then with a next-hop link-local address. This section ends with a discussion of static IPv6 default routes.

Key Topic

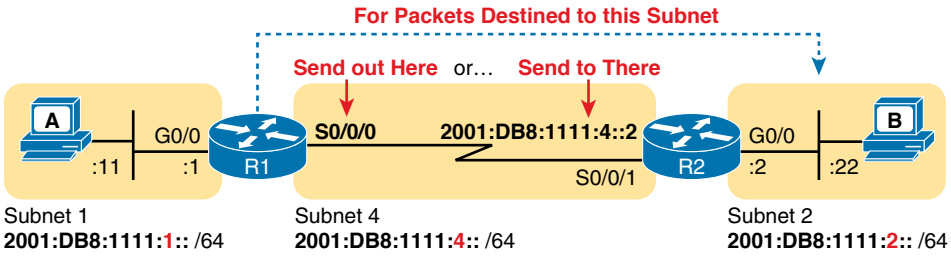


Figure 25-2 *Logic Behind IPv6 Static Route Commands (IPv6 Route)*

Static Routes Using the Outgoing Interface

This first IPv6 static route example uses the outgoing interface option. As a reminder, for both IPv4 and IPv6 static routes, when the command references an interface, the interface is a local interface. That is, it is an interface on the router where the command is added. In this case, as shown in Figure 25-2, R1's **ipv6 route** command would use interface S0/0/0, as shown in Example 25-4.

Example 25-4 *Static IPv6 Routes on Router R1*

```
! Static route on router R1
R1(config)# ipv6 route 2001:db8:1111:2::/64 S0/0/0
```

While Example 25-4 shows the correct syntax of the route, if using static routes throughout this internetwork, more static routes are needed. For example, to support traffic between hosts A and B, R1 is now prepared. Host A will forward all its IPv6 packets to its default router (R1), and R1 can now route those packets out S0/0/0 to R2 next. However, Router R2 does not yet have a route back to host A's subnet, subnet 1 (2001:DB8:1111:1::/64), so a complete solution requires more routes.

Example 25-5 solves this problem by giving Router R2 a static route for subnet 1 (2001:DB8:1111:1::/64). After this route is added, hosts A and B should be able to ping each other.

Example 25-5 *Static IPv6 Routes on Router R2*

```
! Static route on router R2
R2(config)# ipv6 route 2001:db8:1111:1::/64 s0/0/1
```

Many options exist for verifying the existence of the static route and testing whether hosts can use the route. **ping** and **traceroute** can test connectivity. From the router command line, the **show ipv6 route** command will list all the IPv6 routes. The shorter output of the **show ipv6 route static** command, which lists only static routes, could also be used; Example 25-6 shows that output, with the legend omitted.

Example 25-6 *Verification of Static Routes Only on R1*

```
R1# show ipv6 route static
! Legend omitted for brevity
S   2001:DB8:1111:2::/64 [1/0]
     via Serial0/0/0, directly connected
```

This command lists many facts about the one static route on R1. First, the code "S" in the left column does identify the route as a static route. (However, the later phrase "directly connected" might mislead you to think this is a connected route; trust the "S" code.) Note that the prefix (2001:DB8:1111:2::/64) matches the configuration (in Example 25-4), as does the outgoing interface (S0/0/0).

While this command lists basic information about each static route, it does not state whether this route would be used when forwarding packets to a particular destination. For example, if host A sent an IPv6 packet to host B (2001:DB8:1111:2::22), would R1 use this static route? As it turns out, R1 would use that route, as confirmed by the **show ipv6 route 2001:DB8:1111:2::22** command. This command asks the router to list the route that the router would use when forwarding packets to that particular address. Example 25-7 shows an example.

Example 25-7 *Displaying the Route R1 Uses to Forward to Host B*

```
R1# show ipv6 route 2001:db8:1111:2::22
Routing entry for 2001:DB8:1111:2::/64
  Known via "static", distance 1, metric 0
  Route count is 1/1, share count 0
  Routing paths:
    directly connected via Serial0/0/0
      Last updated 00:01:29 ago
```

Static Routes Using Next-Hop IPv6 Address

The previous example used a serial WAN link on purpose. With a point-to-point WAN link, the **ipv6 route** command can use the outgoing interface style of configuration Static IPv6 routes that refer to a next-hop address have two options: the unicast address on the neighboring router (global unicast or unique local) or the link-local address of that same neighboring router. Figure 25-3 spells out those two options with an updated version of Figure 25-2, this time showing Router R2's global unicast as well as R2's link-local address.

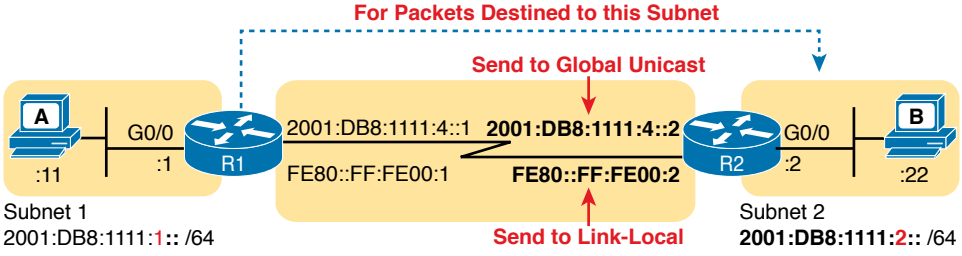


Figure 25-3 *Using Unicast or Link-Local as the Next-Hop Address for Static Routes*

The next few pages walk you through examples, first with a global unicast as a next-hop and then with a link-local as a next-hop.

Example Static Route with a Global Unicast Next-Hop Address

This example uses the internetwork shown in Figure 25-3, but with the earlier static routes removed. That is, both routers have only connected and local routes to begin the example.

In Example 25-8, both R1 and R2 add static routes that refer to the neighbor's global unicast address. R1 adds a route for subnet 2 (on the right), while R2 adds a route for subnet 1 (on the left). Note that the example shows routes in both directions so that the two hosts can send packets to each other.

Example 25-8 *Static IPv6 Routes Using Global Unicast Addresses*

```
! The first command is on router R1, listing R2's global unicast address
R1(config)# ipv6 route 2001:db8:1111:2::/64 2001:DB8:1111:4::2
```

```
! The next command is on router R2, listing R1's global unicast address
R2(config)# ipv6 route 2001:db8:1111:1::/64 2001:db8:1111:4::1
```

The **ipv6 route** command itself is relatively straightforward. Focus on R1's route, which matches the logic shown in Figure 25-3. The command lists subnet 2 (2001:DB8:1111:2::/64). It then lists R2's global unicast address (ending in 4::2).

The verification commands on R1, as shown in Example 25-9, list the usual information. Example 25-9 shows two commands, first listing R1's only static route (the one configured in Example 25-8). The end of the example lists the **show ipv6 route 2001:DB8:1111:2::22** command, which lists the route R1 uses when forwarding packets to Host B, proving that R1 uses this new static route when forwarding packets to that host.

Example 25-9 *Verification of Static Routes to a Next-Hop Global Unicast Address*

```
R1# show ipv6 route static
! Legend omitted for brevity
S   2001:DB8:1111:2::/64 [1/0]
     via 2001:DB8:1111:4::2

R1# show ipv6 route 2001:db8:1111:2::22/64
Routing entry for 2001:DB8:1111:2::/64
  Known via "static", distance 1, metric 0
  Backup from "ospf 1 [110]"
  Route count is 1/1, share count 0
  Routing paths:
    2001:DB8:1111:4::2
      Last updated 00:07:43 ago
```


Example Static Route with a Link-Local Next-Hop Address

Static routes that refer to a neighbor's link-local address work a little like both of the preceding two styles of static routes. First, the **ipv6 route** command refers to a next-hop address,

namely a link-local address. However, the command must also refer to the router's local outgoing interface. Why both? The **ipv6 route** command cannot simply refer to a link-local next-hop address by itself because the link-local address does not, by itself, tell the local router which outgoing interface to use.

Interestingly, when the **ipv6 route** command refers to a global unicast next-hop address, the router can deduce the outgoing interface. For example, the earlier example on R1, as shown in Example 25-8, shows R1 with a static IPv6 route with a next-hop IPv6 address of 2001:DB8:1111:4::2. R1 can look at its IPv6 routing table, see its connected route that includes this 2001:DB8:1111:4::2 address, and see a connected route off R1's S0/0/0. As a result, with a next-hop global unicast address, R1 can deduce the correct outgoing interface (R1's S0/0/0).

With a link-local next-hop address, a router cannot work through this same logic, so the outgoing interface must also be configured. Example 25-10 shows the configuration of static routes on R1 and R2, replacements for the two routes previously configured in Example 25-8.

Example 25-10 *Static IPv6 Routes Using Link-Local Neighbor Addresses*

```
! The first command is on router R1, listing R2's link-local address
R1(config)# ipv6 route 2001:db8:1111:2::/64 S0/0/0 FE80::FF:FE00:2

! The next command is on router R2, listing R1's link-local address
R2(config)# ipv6 route 2001:db8:1111:1::/64 S0/0/1 FE80::FF:FE00:1
```

Example 25-11 verifies the configuration in Example 25-10 by repeating the **show ipv6 route static** and **show ipv6 route 2001:DB8:1111:2::22** commands used in Example 25-9. Note that the output from both commands differs slightly in regard to the forwarding details. Because the new commands list both the next-hop address and outgoing interface, the **show** commands also list both the next-hop (link-local) address and the outgoing interface. If you refer back to Example 25-9, you will see only a next-hop address listed.

Example 25-11 *Verification of Static Routes to a Next-Hop Link-Local Address*

```
R1# show ipv6 route static
! Legend omitted for brevity

S   2001:DB8:1111:2::/64 [1/0]
      via FE80::FF:FE00:2, Serial0/0/0

R1# show ipv6 route 2001:db8:1111:2::22
Routing entry for 2001:DB8:1111:2::/64
  Known via "static", distance 1, metric 0
  Backup from "ospf 1 [110]"
  Route count is 1/1, share count 0
  Routing paths:
    FE80::FF:FE00:2, Serial0/0/0
      Last updated 00:08:10 ago
```

Static Routes over Ethernet Links

You might have wondered why the chapter shows examples with a serial link, knowing that most networks use fewer and fewer serial links today. Using serial links in the examples avoids one complication when defining static routes that use Ethernet interfaces (LAN or WAN). The next example discusses the issues and shows configuration options for static routes when the outgoing interface is an Ethernet interface.

To configure a static route that uses an Ethernet interface, the **ipv6 route** command's forwarding parameters should always include a next-hop IPv6 address. IOS allows you to configure the **ipv6 route** command using only the outgoing-interface parameter, without listing a next-hop address. The router will accept the command; however, if that outgoing interface happens to be an Ethernet interface, the router cannot successfully forward IPv6 packets using the route.

To configure the **ipv6 route** correctly when directing packets out an Ethernet interface, the configuration should use one of these styles:

- Refer to the next-hop global unicast address (or unique local address) only
- Refer to both the outgoing interface and next-hop global unicast address (or unique local address)
- Refer to both the outgoing interface and next-hop link-local address

Example 25-12 shows a sample configuration from routers R1 and R3 in Figure 25-4. The top part of the figure shows the details for R1's route to the subnet on the right side of the figure, with the details labeled with an "A." The bottom half shows the details for R3's route to the LAN subnet on the left of the figure, labeled with a "B."

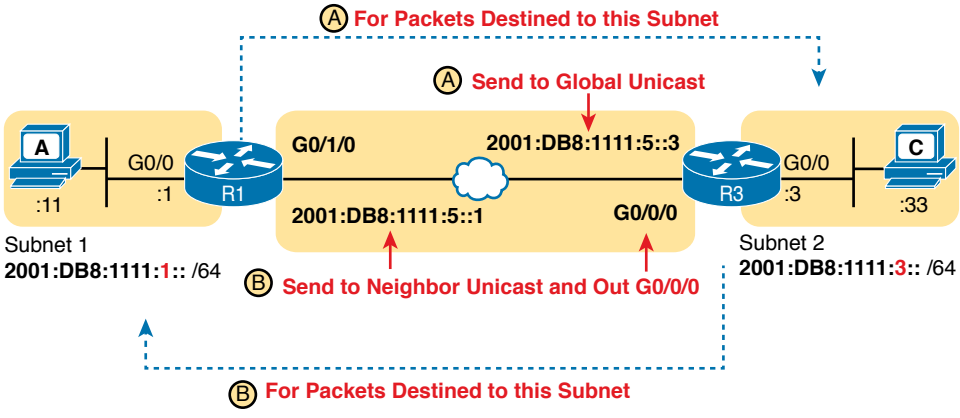


Figure 25-4 *Network Details for IPv6 Static Routes on an Ethernet Interface*

Example 25-12 *Static IPv6 Routes with an Ethernet WAN Interface*

```
! The first command is on router R1, listing R3's global unicast address
R1(config)# ipv6 route 2001:db8:1111:3::/64 2001:db8:1111:5::3
```

```
! The next command is on router R2, listing R1's link-local address
R2(config)# ipv6 route 2001:db8:1111:1::/64 G0/0/0 2001:db8:1111:5::1
```

Static Default Routes

IPv6 supports a default route concept, similar to IPv4. The default route tells the router what to do with an IPv6 packet when the packet matches no other IPv6 route. The logic is pretty basic:

- With no default route, the router discards the IPv6 packet.
- With a default route, the router forwards the IPv6 packet based on the default route.

Default routes can be particularly useful in a couple of network design cases. For example, with an enterprise network design that uses a single router at each branch office, with one WAN link to each branch, the branch routers have only one possible path over which to forward packets. In a large network, when using a routing protocol, the branch router could learn thousands of routes—all of which point back toward the core of the network over that one WAN link.

Branch routers could use default routes instead of a routing protocol. The branch router would forward all traffic to the core of the network. Figure 25-5 shows just such an example, with two sample branch routers on the right and a core site router on the left.

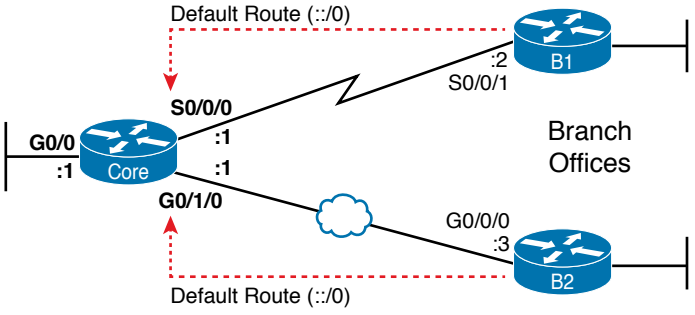


Figure 25-5 *Using Static Default Routes at Branches to Forward Back to the Core*

To configure a static default route, use the same rules already discussed in this section of the chapter, but use a specific value to note the route as a default route: ::/0. Taken literally, the double colon (::) is the IPv6 abbreviation for all 0s, and the /0 means the prefix length is 0. This idea mirrors the IPv4 convention to refer to the default route as 0.0.0.0/0. Otherwise, just configure the **ipv6 route** command as normal.

Example 25-13 shows one such sample static default route on Router B1 from Figure 25-5. This example uses the outgoing interface option.

Example 25-13 *Static Default Route for Branch Router B1*

```
!Forward out B1's S0/0/1 local interface...
B1(config)# ipv6 route ::/0 S0/0/1
```

With IPv6, the router displays the default a little more cleanly than with IPv4. The **show ipv6 route** command simply includes the route in the output of the command, along with the other routes. Example 25-14 shows an example, with "::/0" listed to denote this route as the default route.

Example 25-14 *Router B1's Static Default Route (Using Outgoing Interface)*

```
B1# show ipv6 route static
IPv6 Routing Table - default - 10 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, R - RIP, I1 - ISIS L1, I2 - ISIS L2
       IA - ISIS interarea, IS - ISIS summary, D - EIGRP, EX - EIGRP external
       ND - ND Default, NDp - ND Prefix, DCE - Destination, NDr - Redirect
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
S   ::/0 [1/0]
     via Serial0/0/1, directly connected
```

Static IPv6 Host Routes

Both IPv4 and IPv6 allow the definition of static host routes—that is, a route to a single host IP address. With IPv4, those routes use a /32 mask, which identifies a single IPv4 address in the **ip route** command; with IPv6, a /128 mask identifies that single host in the **ipv6 route** command.

A host route follows the same rules as a route for any other IPv6 subnet. For instance, if you refer back to Figure 25-3, host B sits on the right side of the figure. Earlier examples showed R1's static routes for the subnet in which host B resides—for example, the routes for Router R1 in Examples 25-8 and 25-10. To create a host route on R1, referring to host B's specific IPv6 address, just change those commands to refer to host B's entire IPv6 address (2001:DB8:1111:2::22), with prefix length /128.

Example 25-15 shows two sample host routes on Router R1. Both define a host route to host B's IPv6 address as seen in Figure 25-3. One route uses Router R2's link-local address as the next-hop address, and one route uses R2's global unicast address as the next-hop address.

Example 25-15 *Static Host IPv6 Routes on R1, for Host B*

```
! The first command lists host B's address, prefix length /128,
! with R2's link-local address as next-hop, with an outgoing interface.
R1(config)# ipv6 route 2001:db8:1111:2::22/128 S0/0/0 FE80::FF:FE00:2
R1(config)#
! The next command also lists host B's address, prefix length /128,
! but with R2's global unicast address as next-hop, and no outgoing interface.
R1(config)# ipv6 route 2001:db8:1111:2::22/128 2001:DB8:1111:4::2
```

Floating Static IPv6 Routes

Next, consider the case in which a static route competes with other static routes or routes learned by a routing protocol. For example, consider the topology shown in Figure 25-6, which shows a branch office with two WAN links: one very fast Gigabit Ethernet link and one rather slow (but cheap) T1. In this design, the network uses OSPFv3 to learn IPv6 routes over the primary link, learning a route for subnet 2001:DB8:1111:7::/64. R1 also defines a

static route over the backup link to that exact same subnet, so R1 must choose whether to use the static route or the OSPF-learned route.

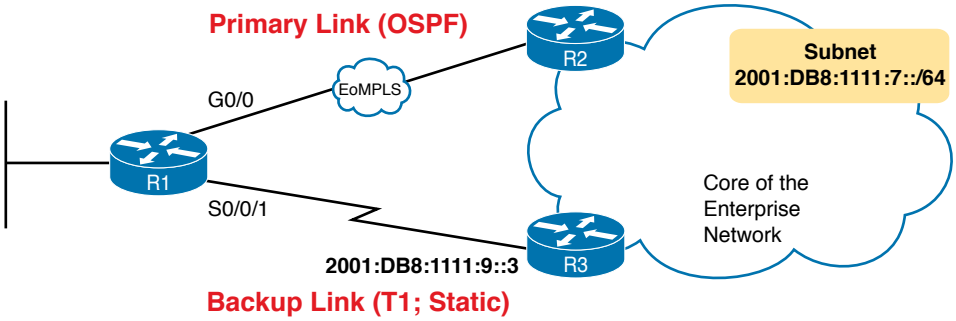


Figure 25-6 *Using a Floating Static Route to Key Subnet 2001:DB8:1111:7::/64*

IOS considers static routes better than OSPF-learned routes by default due to administrative distance. IOS uses the same administrative distance concept and default values for IPv6 as it does for IPv4. As a result, a static IPv6 route over the lower path would be given an administrative distance of 1, and an OSPFv3-learned route over the top path would be given an administrative distance of 110. R1 would use the lower path to reach subnet 2001:DB8:1111:7::/64 in this case, which is not the intended design. Instead, the engineer prefers to use the OSPF-learned routes over the much-faster primary link and use the static route over the backup link only as needed when the primary link fails.

To instead prefer the OSPF routes, the configuration would need to change the administrative distance settings and use what many networkers call a floating static route. Like an IPv4 floating static route, an IPv6 *floating static* route floats or moves into and out of the IPv6 routing table depending on whether the better (lower) administrative distance route learned by the routing protocol happens to exist currently. Basically, the router ignores the static route during times when the better routing protocol route is known.

To implement an IPv6 floating static route, just override the default administrative distance on the static route, making the value larger than the default administrative distance of the routing protocol. For example, the **ipv6 route 2001:db8:1111:7::/64 2001:db8:1111:9::3 130** command on R1 would do exactly that, setting the static route's administrative distance to 130. As long as the primary link (G0/0) stays up, and OSPFv3 on R1 learns a route for **2001:db8:1111:7**::/64 with OSPF's default administrative distance of 110, R1 ignores the static route whose administrative distance is explicitly configured as 130.

Finally, note that both the **show ipv6 route** and **show ipv6 route 2001:db8:1111:7::/64** commands list the administrative distance. Example 25-16 shows a sample matching this most recent example. Note that in this case, the static route is in use in the IPv6 routing table.

Example 25-16 *Displaying the Administrative Distance of the Static Route*

```
R1# show ipv6 route static
! Legend omitted for brevity
S   2001:db8:1111:7::/64 [130/0]
      via 2001:db8:1111:9::3

R1# show ipv6 route 2001:db8:1111:7::/64
Routing entry for 2001:db8:1111:7::/64
  Known via "static", distance 130, metric 0
  Route count is 1/1, share count 0
  Routing paths:
    2001:db8:1111:9::3
      Last updated 00:00:58 ago
```

Table 25-2 lists some of the default administrative distance values used with IPv6.

Table 25-2 IOS Defaults for Administrative Distance

Route Source	Administrative Distance
Connected routes	0
Static routes	1
NDP	2
EIGRP	90
OSPF	110
RIP	120
Unknown or unbelievable	255

Troubleshooting Static IPv6 Routes

IPv6 static routes have the same potential issues and mistakes as do static IPv4 routes, as discussed in Chapter 16. However, IPv6 static routes do have a few small differences. This last part of the static route content in the chapter looks at troubleshooting IPv6 static routes, reviewing many of the same troubleshooting rules applied to IPv4 static routes, while focusing on the details specific to IPv6.

This topic breaks static route troubleshooting into two perspectives: cases in which the route is in the routing table but is incorrect, and cases in which the route is not in the routing table.

Troubleshooting Incorrect Static Routes That Appear in the IPv6 Routing Table

A static route is only as good as the input typed into the **ipv6 route** command. IOS checks the syntax of the command, of course. However, IOS cannot tell if you choose the incorrect outgoing interface, incorrect next-hop address, or incorrect prefix/prefix-length in a static route. If the parameters pass the syntax checks, IOS places the **ipv6 route** command into the

running-config file. Then, if no other problem exists (as discussed at the next heading), IOS puts the route into the IP routing table—even though the route may not work because of the poorly chosen parameters.

For instance, an exam question might show a figure with Router R1 having an address of 2001:1:1:1::1 and neighboring Router R2 with an address of 2001:1:1:1::2. If R1 lists a static route with the command **ipv6 route 3333::/64 2001:1:1:1::1**, the command would be accepted by IOS with correct syntax, but it would not be effective as a route. Note that the command lists R1's address as the next-hop address, and R1 cannot use its own IPv6 address as a next-hop address. IOS does not prevent the configuration of the command, however; it allows the command and adds the route to the IPv6 routing table, but the route cannot possibly forward packets correctly.

When you see an exam question that has static routes, and you see them in the output of **show ipv6 route**, remember that the routes may have incorrect parameters. Check for these types of mistakes:

Key Topic

Step 1. Prefix/Length: Does the **ipv6 route** command reference the correct subnet ID (prefix) and mask (prefix length)?

Step 2. If using a next-hop IPv6 address that is a link-local address:

- **A.** Is the link-local address an address on the correct neighboring router? (It should be an address on another router on a shared link.)
- **B.** Does the **ipv6 route** command also refer to the correct outgoing interface on the local router?

Step 3. If using a next-hop IPv6 address that is a global unicast or unique local address, is the address the correct unicast address of the neighboring router?

Step 4. If referencing an outgoing interface, does the **ipv6 route** command reference the interface on the local router (that is, the same router where the static route is configured)?

This troubleshooting checklist works through the various cases in which IOS would accept the configuration of the static IPv6 route, but the route would not work because of the incorrect parameters in context. It helps to see a few examples. Figure 25-7 shows a sample network to use for the examples; all the examples focus on routes added to Router R1, for the subnet on the far right.

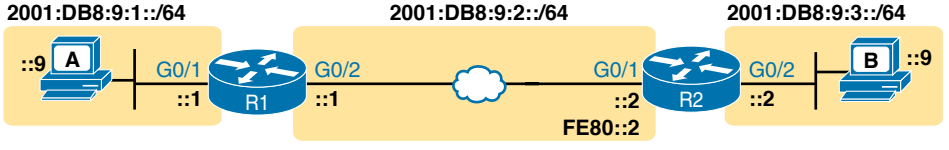


Figure 25-7 *Sample Topology for Incorrect IPv6 Route Examples*

Example 25-17 shows five **ipv6 route** commands. All have correct syntax, but all have one incorrect value; that is, the route will not work because of the types of problems in the

troubleshooting checklist. Look for the short comment at the end of each configuration command to see why each is incorrect.

Example 25-17 **ipv6 route** *Commands with Correct Syntax but Incorrect Ideas*

```
ipv6 route 2001:DB8:9:33::/64 2001:DB8:9:2::2    ! Step 1: Wrong prefix
ipv6 route 2001:DB8:9:3::/64 G0/2 FE80::AAA9     ! Step 2A: Wrong neighbor link local
ipv6 route 2001:DB8:9:3::/64 FE80::2             ! Step 2B: Missing outgoing interface
ipv6 route 2001:DB8:9:3::/64 2001:DB8:9:2::1     ! Step 3: Wrong neighbor address
ipv6 route 2001:DB8:9:3::/64 G0/1 FE80::2        ! Step 4: Wrong interface on R1
```

All these incorrect examples have correct syntax and would be added to R1's IPv6 routing table if configured on R1. However, all have flaws. Working through the examples in order:

Step 1. The prefix (2001:DB8:9:33::) has a typo in the fourth quartet (33 instead of 3).

Step 2A. The figure shows R2's G0/1 with link-local address FE80::2, but the command uses FE80::AAA9.

Step 2B. The command uses the correct link-local address on R2's address on the common link (FE80::2 per the figure), but it omits the outgoing interface of R1's G0/2 interface. (See the next example for more detail.)

Step 3. The figure shows the subnet in the center as 2001:DB8:9:2::/64, with R1 using the ::1 address and R2 using ::2. For the fourth command, R1's command should use R2's address 2001:DB8:9:2::2, but it uses R1's own 2001:DB8:9:2::1 address instead.

Step 4. As a command on R1, the outgoing interface references R1's own interfaces. R1's G0/1 is the interface on the left, whereas R1 should use its G0/2 interface on the right when forwarding packets to subnet 2001:DB8:9:3::/64.

The key takeaway for this section is to know that a route in the IPv6 routing table may be incorrect due to poor choices for the parameters. The parameters should always include the neighboring router's IPv6 addresses, but the local router's interface type/number, and in all cases, the correct prefix/length. The fact that a route is in the IPv6 routing table, particularly a static route, does not mean it is a correct route.

Note that of the five example commands in Example 25-17, IOS would accept all of them except the third one. IOS can notice the case of omitting the outgoing interface if the next-hop address is a link-local address. Example 25-18 shows a sample of the error message from IOS.

Example 25-18 *IOS Rejects the* **ipv6 route** *Command with Link-Local and No Outgoing Interface*

```
R1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# ipv6 route 2001:DB8:9:3::/64 FE80::2
% Interface has to be specified for a link-local nexthop
```

```
R1(config)# ^Z
R1#
R1# show running-config | include ipv6 route
R1#
```

The Static Route Does Not Appear in the IPv6 Routing Table

The preceding few pages focused on IPv6 static routes that show up in the IPv6 routing table but unfortunately have incorrect parameters. The next page looks at IPv6 routes that have correct parameters, but IOS does not place them into the IPv6 routing table.

When you add an **ipv6 route** command to the configuration, and the syntax is correct, IOS considers that route to be added to the IPv6 routing table. IOS makes the following checks before adding the route; note that IOS uses this same kind of logic for IPv4 static routes:

Key Topic

- For **ipv6 route** commands that list an outgoing interface, that interface must be in an up/up state.
- For **ipv6 route** commands that list a global unicast or unique local next-hop IP address (that is, not a link-local address), the local router must have a route to reach that next-hop address.
- If another IPv6 route exists for that exact same prefix/prefix-length, the static route must have a better (lower) administrative distance.

The Neighbor Discovery Protocol

Similar to ICMP for IPv4, IPv6 defines the ICMP protocol for IPv6 (ICMPv6). However, ICMPv6 reaches further than ICMPv4, pulling in functions done by other miscellaneous protocols in IPv4. For instance, with IPv4, ARP works as a separate protocol; with IPv6, the Neighbor Discovery Protocol (NDP), a part of ICMPv6, performs the same functions.

As it turns out, routers play a key role in several NDP protocol functions, so this final major section of the chapter explains a few of the functions of the NDP protocol (RFC 4861). Some of those NDP functions are

Key Topic

Neighbor MAC Discovery: An IPv6 LAN-based host will need to learn the MAC address of other hosts in the same subnet. NDP replaces IPv4's ARP, providing messages that replace the ARP Request and Reply messages.

Router Discovery: Hosts learn the IPv6 addresses of the available IPv6 routers in the same subnet.

SLAAC: When using Stateless Address Auto Configuration (SLAAC), the host uses NDP messages to learn the subnet (prefix) used on the link plus the prefix length.

DAD: Before using an IPv6 address, hosts use NDP to perform a Duplicate Address Detection (DAD) process, to ensure no other host uses the same IPv6 address before attempting to use it.

Discovering Neighbor Link Addresses with NDP NS and NA

NDP replaces IPv4 ARP using a pair of matched solicitation and advertisement messages: the *Neighbor Solicitation* (NS) and *Neighbor Advertisement* (NA) messages. Basically, the NS

acts like an IPv4 ARP request, asking the host with a particular unicast IPv6 address to send back a reply. The NA message acts like an IPv4 ARP Reply, listing that host's MAC address.

The process of sending the NS and NA messages follows the same general process with IPv4 ARP: the NS message asks for information, and the NA supplies the information, as summarized in this list:

Key Topic

Neighbor Solicitation (NS): This message asks the host with a particular IPv6 address (the target address) to reply with an NA message that lists its MAC address. The NS message is sent to the solicited-node multicast address associated with the target address, so the message is processed only by hosts whose last six hex digits match the address that is being queried.

Neighbor Advertisement (NA): This message lists the sender's IPv6 and MAC addresses. It can be sent in reply to an NS message, and if so, the packet is sent to the IPv6 unicast address of the host that sent the original NS message. A host can also send an unsolicited NA, announcing its IPv6 and MAC addresses, in which case the message is sent to the all-IPv6-hosts local-scope multicast address FF02::1.

NOTE With NDP, the word *neighbor* refers to the fact that the devices will be on the same data link—for example, the same VLAN.

Figure 25-8 shows an example of how a host (PC1) uses an NS message to learn the MAC address used by another host. The NS message lists a target IPv6 unicast address, with the implied question: "What is your link address?" The NA message, in this example sent back to the original host that asked the question, lists that link address.

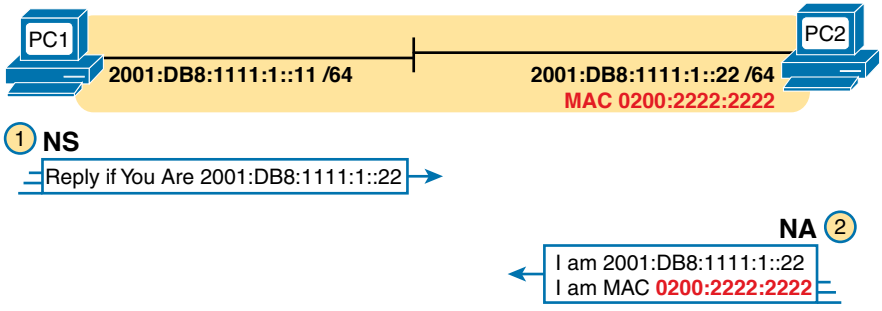


Figure 25-8 *Example NDP NS/NA Process to Find the Neighbor's Link Addresses*

At Step 1 of this particular example, PC1 sends the solicitation to find PC2's MAC address. PC1 first looks in its NDP neighbor table, the equivalent of the IPv4 ARP cache, and does not find the MAC address for IPv6 address 2001:DB8:1111:1::22. So, at Step 1, PC1 sends the NDP NS message to the matching solicited-node multicast address for 2001:DB8:1111:1::22 or FF02::1:FF00:22. Only IPv6 hosts whose address ends with 00:0022 will listen for this solicited-node multicast address. As a result, only a small subset of hosts on this link will process the received NDP NS message.

At Step 2, PC2 reacts to the received NS message. PC2 sends back an NA message in reply, listing PC2's MAC address. PC1 records PC2's MAC address in PC1's NDP neighbor table.

Example 25-19 shows an example of the IPv6 neighbor table on Router R3, as seen originally back in Figure 25-1. In this case, R3 has learned the MAC addresses of Router R1's WAN interface (G0/1/0)—both its global unicast address as well as the link-local address on that same interface.

Example 25-19 *IPv6 Neighbor Table on Router R3*

```
R3# show ipv6 neighbors
IPv6 Address                              Age Link-layer Addr State Interface
2001:DB8:1111:5::1                          0 0201.a010.0001  REACH Gi0/0/0
FE80::1:A0FF:FE10:1                         0 0201.a010.0001  REACH Gi0/0/0
```

NOTE To view a host's NDP neighbor table, use these commands: (Windows) **netsh interface ipv6 show neighbors**; (Linux) **ip -6 neighbor show**; (Mac OS) **ndp -an**.

Discovering Routers with NDP RS and RA

IPv4 hosts use the concept of an IPv4 default gateway or default router. When the host needs to send a packet to some IPv4 subnet other than the local subnet, the host sends the IPv4 packet to the default router, expecting the router to be able to route the packet to the destination. Note that hosts either statically set the IP address of their default gateway or learn it from a server called a Dynamic Host Configuration Protocol (DHCP) server.

IPv6 uses the same concept of a default gateway, but it improves the method for hosts to learn the identity of possible default gateways using NDP. NDP defines two messages that allow any host to discover all routers in the subnet:

Key Topic

Router Solicitation (RS): This message is sent to the "all-IPv6-routers" local-scope multicast address of FF02::2 so that the message asks all routers, on the local link only, to identify themselves.

Router Advertisement (RA): This message, sent by the router, lists many facts, including the link-local IPv6 address of the router. When sent in response to an RS message, it flows back to either the unicast address of the host that sent the RS or to the all-IPv6-hosts address FF02::1. Routers also send RA messages without being asked, sent to the all-IPv6-hosts local-scope multicast address of FF02::1.

For example, Figure 25-9 shows how host PC1 can learn R1's link-local address. The process is indeed simple, with PC1 first asking and R1 replying.

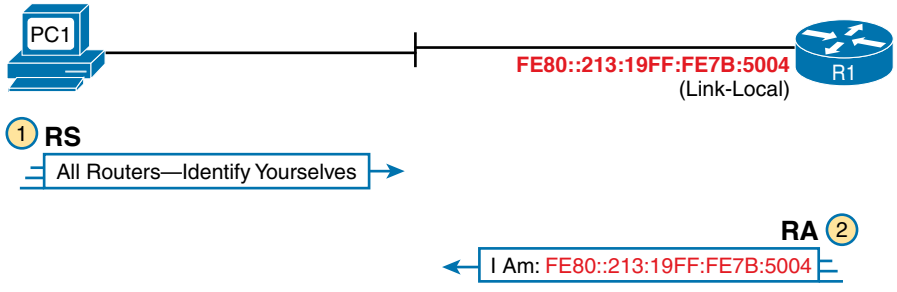


Figure 25-9 *Example NDP RS/RA Process to Find the Default Routers*

NOTE IPv6 allows multiple prefixes and multiple default routers to be listed in the RA message; Figure 25-9 just shows one of each for simplicity's sake.

IPv6 does not use broadcasts, but it does use multicasts. In this case, the RS message flows to the all-routers multicast address (FF02::2) so that all routers will receive the message. It has the same good effect as a broadcast with IPv4, without the negatives of a broadcast. In this case, only IPv6 routers will spend any CPU cycles processing the RS message, and IPv6 hosts will ignore the message. The RA message can flow either to the unicast IPv6 address of PC1 or to the all-nodes FF02::1 address.

Note that while Figure 25-9 shows how a host can ask to learn about any routers, routers also periodically send unsolicited RA messages, even without an incoming RS. When routers send these periodic RA messages, they basically advertise details about IPv6 on the link. In this case, the RA messages flow to the FF02::1 all-nodes IPv6 multicast address.

Using SLAAC with NDP RS and RA

Both IPv4 and IPv6 support the idea of dynamic address assignment for hosts via the Dynamic Host Configuration Protocol (DHCP). To find an address to use with DHCP, the DHCP client sends messages to a DHCP server, and the server assigns a currently unused address in the correct subnet for the endpoint host to use. The process relies on DHCP client functions in each device and a DHCP server configured and working in the network.

IPv6 supports an alternative method for IPv6 hosts to dynamically choose an unused IPv6 address to use—a process that does not require a server like a DHCP server. The process goes by the name *Stateless Address Autoconfiguration* (SLAAC). SLAAC uses a simple three-step process that begins by learning the prefix/length as shown in the figure. The steps are as follows:

1. Learn the IPv6 prefix used on the link, from any router, using NDP RS/RA messages.
2. Build an address from the prefix plus an interface ID, chosen either by using EUI-64 rules or as a random value.
3. Before using the address, first use DAD to make sure that no other host is already using the same address.

Figure 25-10 shows the structure of an IPv6 address created with SLACC using Steps 1 and 2 in the process, with the next topic detailing the third step (DAD).

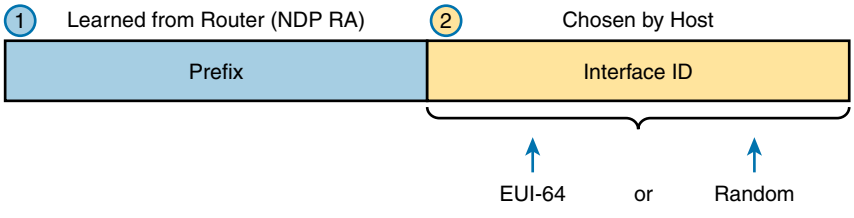


Figure 25-10 *Host IPv6 Address Formation Using SLAAC*

Discovering Duplicate Addresses Using NDP NS and NA

IPv6 uses the Duplicate Address Detection (DAD) process before using a unicast address to make sure that no other node on that link is already using the address. Hosts use DAD not only at the end of the SLAAC process, but also any time that a host interface initializes, no matter whether using SLAAC, DHCP, or static address configuration. When performing DAD, if another host already uses that address, the first host simply does not use the address until the problem is resolved.

The term *DAD* refers to the function, but the function uses NDP NS and NA messages. Basically, a host sends an NS message for its own IPv6 address. No other host should be using that address, so no other host should send an NDP NA in reply. However, if another host already uses that address, that host will reply with an NA, identifying a duplicate use of the address.

Figure 25-11 shows an example. PC1 initializes and does a DAD check, but PC2 happens to already be working and already be using the address. The figure shows the following steps:

1. PC1, before using address 2001:DB8:1111:1::11, must use DAD.
2. PC1 sends an NS message, listing the address PC1 now wants to use (2001:DB8:1111:1::11) as the target.
3. PC2 receives the NS, sees what PC2 already uses as its own address, and sends back an NA.
4. PC1, on receiving the NA message for its own IPv6 address, realizes a duplicate address exists.

Key Topic

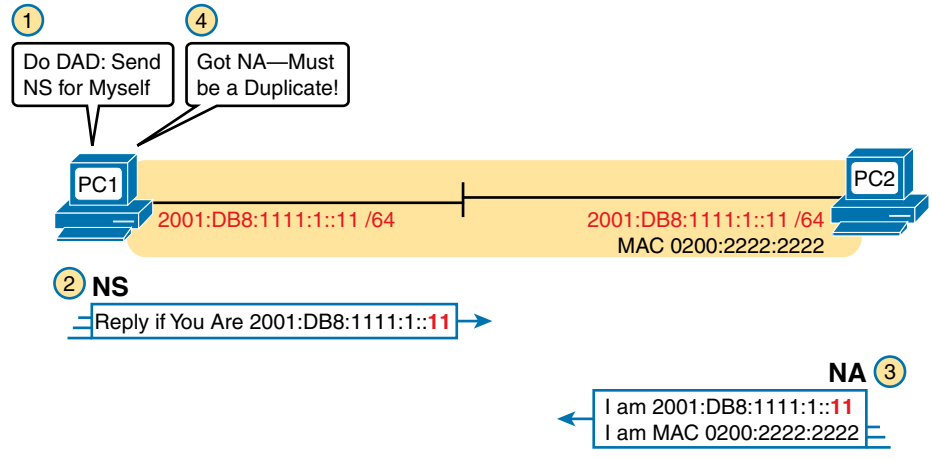


Figure 25-11 *Example Duplicate Address Detection (DAD) with NDP NS/NA*

Hosts do the DAD check for each of their unicast addresses, link-local addresses included, both when the address is first used and each time the host's interface comes up.

NDP Summary

This chapter explains some of the more important functions performed by NDP. NDP does more than what is listed in this chapter, and the protocol allows for addition of other functions, so NDP might continue to grow over time. For now, use Table 25-3 as a study reference for the four NDP features discussed here.

Key Topic

Table 25-3 NDP Function Summary

Function	Protocol Messages	Who Discovers Info	Who Supplies Info	Info Supplied
Router discovery	RS and RA	Any IPv6 host	Any IPv6 router	Link-local IPv6 address of router
Prefix/length discovery	RS and RA	Any IPv6 host	Any IPv6 router	Prefix(es) and associated prefix lengths used on local link
Neighbor discovery	NS and NA	Any IPv6 host	Any IPv6 host	Link-layer address (for example, MAC address) used by a neighbor
Duplicate Address Detection	NS and NA	Any IPv6 host	Any IPv6 host	Simple confirmation whether a unicast address is already in use

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 25-4 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 25-4 Chapter Review Tracking

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

Review All the Key Topics

Key Topic

Table 25-5 Key Topics for Chapter 25

Key Topic Element	Description	Page Number
List	Methods by which a router can build IPv6 routes	583
List	Rules for IPv6 connected and local routes	583
Figure 25-2	IPv6 static route concepts	587
Checklist	Items to check on **ipv6 route** command that cause problems with IPv6 static routes	596
Checklist	Items to check other than the **ipv6 route** command that cause problems with IPv6 static routes	598
List	Four functions that use NDP messages	598
List	NDP NS and NA messages and meanings	599
List	NDP RS and RA messages and meanings	600
Figure 25-11	Example DAD check	602
Table 25-3	NDP Function summary table	603

Key Terms You Should Know

IPv6 host route, local route, IPv6 local route, IPv6 administrative distance, IPv6 multicast scope

Command References

Tables 25-6 and 25-7 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 25-6 Chapter 25 Configuration Command Reference

Command	Description
ipv6 route *prefix/length next-hop-address*	Global command to define an IPv6 static route to a next-hop router IPv6 address.
ipv6 route *prefix/length outgoing-interface*	Global command to define an IPv6 static route, with packets forwarded out the local router interface listed in the command.
ipv6 route *prefix/length outgoing-interface next-hop-address*	Global command to define an IPv6 static route, with both the next-hop address and local router outgoing interface listed.

Command	Description
ipv6 route ::/0 *{[next-hop-address] [outgoing-interface]}*	Global command to define a default IPv6 static route.
ipv6 address autoconfig [default]	Interface subcommand that tells the router to use SLAAC to find/build its own interface IPv6 address, and with the **default** parameter, to add a default route with a next hop of the router that responds with the RA message.

Table 25-7 Chapter 25 EXEC Command Reference

Command	Description
show ipv6 route [**connected** | **local** | **static**]	Lists routes in the routing table.
show ipv6 route *address*	Displays detailed information about the route this router uses to forward packets to the IPv6 address listed in the command.
show ipv6 neighbors	Lists the contents of the IPv6 neighbor table, which lists the MAC address associated with IPv6 addresses on common subnets.

Part VII Review

Keep track of your part review progress with the checklist in Table P7-1. Details on each task follow the table.

Table P7-1 Part VII 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, use the PCPT software to answer the "Do I Know This Already?" questions again for the chapters in this part of the book.

Answer Part Review Questions

For this task, use PTP to answer the Part Review questions for this part of the book.

Review Key Topics

Review all key topics in all chapters in this part, either by browsing the chapters or 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 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: Configure IPv6 addresses on interfaces, and before using any show commands, predict the connected and local routes that should be added to the IPv6 routing table, and predict the link-local (unicast) address and various multicast addresses you expect to see in the output of the **show ipv6 interfaces** command.

Watch Videos

Chapter 24 mentions that the companion website's section for Chapter 24 review includes a video about the EUI-64 address generation process, so consider using the video as a review.

This book began with an overview of the fundamentals of LANs, WANs, and IP routing. It then described Ethernet LANs (wired LANs) in some depth over the course of seven chapters. The book then meandered through many chapters exploring the many concepts of IPv4 and IPv6 addressing, routing, and how to implement those features in Cisco devices.

This final part of Volume 1 turns our attention back to the LAN, not to wired Ethernet LANs, but to IEEE 802.11 wireless LANs—in other words, Wi-Fi. The four chapters in this part of the book lay down the foundations of how wireless LANs work and then show how to implement wireless LANs using Cisco devices.

Building wireless LANs requires some thought because the endpoints that use the LAN do not sit in one place and connect via a known cable and known switch port. To explain those details, Chapter 26 begins with the basics of how a wireless client can connect to the wireless network through a wireless access point (AP). After you learn the foundations in Chapter 26, Chapter 27 takes an architectural view of wireless LANs to discuss how you might build a wireless LAN for an enterprise, which requires much different thinking than, for instance, building a wireless LAN for your home.

Chapter 28 completes the three concepts-focused wireless LAN chapters by working through the alphabet soup that is wireless LAN security. The fact that wireless LAN clients come and go means that the LAN may be under constant attack as an easy place for an attacker to gain access to the network, so wireless LANs must use effective security. Finally, Chapter 29 closes by showing how to configure an enterprise wireless LAN using Cisco APs and the Cisco Wireless LAN Controller (WLC) from the WLC's graphical interface.

Part VIII

Wireless LANs

CHAPTER 26

Fundamentals of Wireless Networks

This chapter covers the following exam topics:

1.0 Network Fundamentals

1.1 Explain the role and function of network components

1.1.d Access Points

1.11 Describe wireless principles

1.11.a Nonoverlapping Wi-Fi channels

1.11.b SSID

1.11.c RF

Wireless communication usually involves a data exchange between two devices. A wireless LAN goes even further; many devices can participate in sharing the medium for data exchanges. Wireless LANs must transmit a signal over radio frequencies (RF) to move data from one device to another. Transmitters and receivers can be fixed in consistent locations, or they can be mobile and free to move around. This chapter explains the topologies that can be used to control access to the wireless medium and provide data exchange between devices.

"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 26-1 "Do I Know This Already?" Foundation Topics Section-to-Question Mapping

Foundation Topics Section	Questions
Comparing Wired and Wireless Networks	1
Wireless LAN Topologies	2–4
Other Wireless Topologies	5–6
Wireless Bands and Channels	7–8

1. Wired Ethernet and Wi-Fi are based on which two IEEE standards, respectively?
   a. 802.1, 802.3
   b. 802.3, 802.1
   c. 802.3, 802.11
   d. 802.11, 802.3
2. Devices using a wireless LAN must operate in which one of the following modes?
   a. Round-robin access
   b. Half duplex
   c. Full duplex
   d. None of these answers
3. An access point is set up to offer wireless coverage in an office. Which one of the following is the correct 802.11 term for the resulting standalone network?
   a. BSA
   b. BSD
   c. BSS
   d. IBSS
4. Which one of the following is used to uniquely identify an AP and the basic service set it maintains with its associated wireless clients?
   a. SSID
   b. BSSID
   c. Ethernet MAC address
   d. Radio MAC address
5. Which one of the following can be used to provide wireless connectivity to a nonwireless device?
   a. Wireless repeater
   b. Workgroup bridge
   c. Transparent bridge
   d. Adaptive bridge
6. Which one of the following is not needed in a Cisco outdoor mesh network?
   a. A BSS function
   b. Ethernet cabling to each AP
   c. A workgroup bridge
   d. A backhaul network

7. Which of the following are frequency bands commonly used for Wi-Fi?
   a. 2.5 KHz
   b. 2.5 MHz
   c. 5 MHz
   d. 2.5 GHz
   e. 5 GHz
8. Which of the following are considered to be nonoverlapping channels?
   a. Channels 1, 2, and 3 in the 2.4-GHz band
   b. Channels 1, 5, and 10 in the 2.4-GHz band
   c. Channels 1, 6, and 11 in the 2.4-GHz band
   d. Channels 40, 44, and 48 in the 5-GHz band

Foundation Topics

Comparing Wired and Wireless Networks

In a wired network, any two devices that need to communicate with each other must be connected by a wire. (That was obvious!) The "wire" might contain strands of metal or fiber-optic material that run continuously from one end to the other. Data that passes over the wire is bounded by the physical properties of the wire. In fact, the IEEE 802.3 set of standards defines strict guidelines for the Ethernet wire itself, in addition to how devices may connect, send, and receive data over the wire.

Wired connections have been engineered with tight constraints and have few variables that might prevent successful communication. Even the type and size of the wire strands, the number of twists the strands must make around each other over a distance, and the maximum length of the wire must adhere to the standard.

Therefore, a wired network is essentially a bounded medium; data must travel over whatever path the wire or cable takes between two devices. If the cable goes around a corner or lies in a coil, the electrical signals used to carry the data must also go around a corner or around a coil. Because only two devices may connect to a wire, only those two devices may send or transmit data. Even better: the two devices may transmit data to each other simultaneously because they each have a private, direct path to each other.

Wired networks also have some shortcomings. When a device is connected by a wire, it cannot move around very easily or very far. Before a device can connect to a wired network, it must have a connector that is compatible with the one on the end of the wire. As devices get smaller and more mobile, it just is not practical to connect them to a wire.

As its name implies, a wireless network removes the need to be tethered to a wire or cable. Convenience and mobility become paramount, enabling users to move around at will while staying connected to the network. A user can (and often does) bring along many different wireless devices that can all connect to the network easily and seamlessly.

Wireless data must travel through free space, without the constraints and protection of a wire. In the free space environment, many variables can affect the data and its delivery. To minimize the variables, wireless engineering efforts must focus on two things:

- Wireless devices must adhere to a common standard (IEEE 802.11).
- Wireless coverage must exist in the area where devices are expected to use it.

As you study for the CCNA 200-301 exam, keep in mind that the exam is geared more toward a functional view of wireless technology. More detailed topics like RF characteristics, antenna performance, and so on are reserved for the Implementing Cisco Enterprise Network Core Technologies ENCOR 300-401 exam.

Wireless LAN Topologies

Wireless communication takes place over free space through the use of radio frequency (RF) signals. The theory behind RF signals can be complex, and is described further in the "RF Overview" section in this chapter. For now, just assume that one device, the transmitter, sends RF signals to another device, the receiver. As Figure 26-1 shows, the transmitter can contact the receiver at any and all times, as long as both devices are tuned to the same frequency (or channel) and use the same scheme to carry the data between them. That all sounds simple, except that it is not really practical.

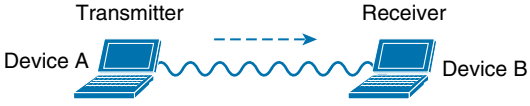


Figure 26-1 *Unidirectional Communication*

To fully leverage wireless communication, data should travel in *both* directions, as shown in Figure 26-2. Sometimes Device A needs to send data to Device B, while Device B would like to take a turn to send at other times.

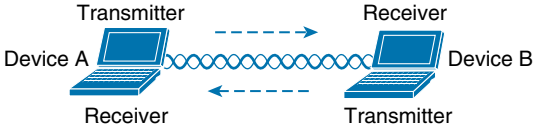


Figure 26-2 *Bidirectional Communication*

Because the two devices are using the same channel, two phrases in the preceding sentence become vitally important: *take a turn* and *send at other times*. With wireless communication, if multiple signals are received at the same time, they can interfere with each other. The likelihood of interference increases as the number of wireless devices grows. For example, Figure 26-3 shows four devices tuned to the same channel and what might happen if some or all of them transmit at the same time.

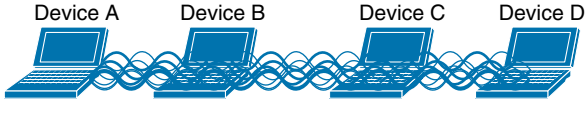


Figure 26-3 *Interference from Simultaneous Transmissions*

All this talk about waiting turns and avoiding interference might remind you of a traditional (nonswitched) Ethernet LAN, where multiple hosts can connect to a shared media and share a common bandwidth. To use the media effectively, all the hosts must operate in half-duplex mode so that they try to avoid colliding with other transmissions already in progress. The side effect is that no host can transmit and receive at the same time on a shared medium.

A wireless LAN is similar. Because multiple hosts can share the same channel, they also share the "airtime" or access to that channel at any given time. Therefore, to keep everything clean, only one device should transmit at any given time. To contend for use of the channel, devices based on the 802.11 standard have to determine whether the channel is clear and available before transmitting anything.

NOTE IEEE 802.11 WLANs are always half duplex because transmissions between stations use the same frequency or channel. Only one station can transmit at any time; otherwise, collisions occur. To achieve full-duplex mode, one station's transmission would have to occur on one frequency while it receives over a different frequency—much like full-duplex Ethernet links work. Although this is certainly possible and practical, the 802.11 standard does not permit full-duplex operation. Some amendments to the standard do provide a means for multiple devices to transmit on the same channel at the same time, but this is beyond the scope of this book.

At the most basic level, there is no inherent organization to a wireless medium or any inherent control over the number of devices that can transmit and receive frames. Any device that has a wireless network adapter can power up at any time and try to communicate. At a minimum, a wireless network should have a way to make sure that every device using a channel can support a common set of parameters. Beyond that, there should be a way to control which devices (and users) are allowed to use the wireless medium and the methods that are used to secure the wireless transmissions.

Basic Service Set

The solution is to make every wireless service area a closed group of mobile devices that forms around a fixed device; before a device can participate, it must advertise its capabilities and then be granted permission to join. The 802.11 standard calls this a *basic service set* (BSS). At the heart of every BSS is a wireless *access point* (AP), as shown in Figure 26-4. The AP operates in *infrastructure mode*, which means it offers the services that are necessary to form the infrastructure of a wireless network. The AP also establishes its BSS over a single wireless channel. The AP and the members of the BSS must all use the same channel to communicate properly.

Because the operation of a BSS hinges on the AP, the BSS is bounded by the area where the AP's signal is usable. This is known as the *basic service area* (BSA) or *cell*. In Figure 26-4, the cell is shown as a simple shaded circular area that centers around the AP itself. Cells can

Answers to the "Do I Know This Already?" quiz:

1 C **2** B **3** C **4** B **5** B **6** B **7** D, E **8** C, D

have other shapes too, depending on the antenna that is connected to the AP and on the physical surroundings that might affect the AP's signals.

The AP serves as a single point of contact for every device that wants to use the BSS. It advertises the existence of the BSS so that devices can find it and try to join. To do that, the AP uses a unique BSS identifier (BSSID) that is based on the AP's own radio MAC address.

NOTE Recall that wired Ethernet devices each have a unique MAC address to send frames from a source to a destination over a Layer 2 network. Wireless devices must also have unique MAC addresses to send wireless frames at Layer 2 over the air.

Key Topic

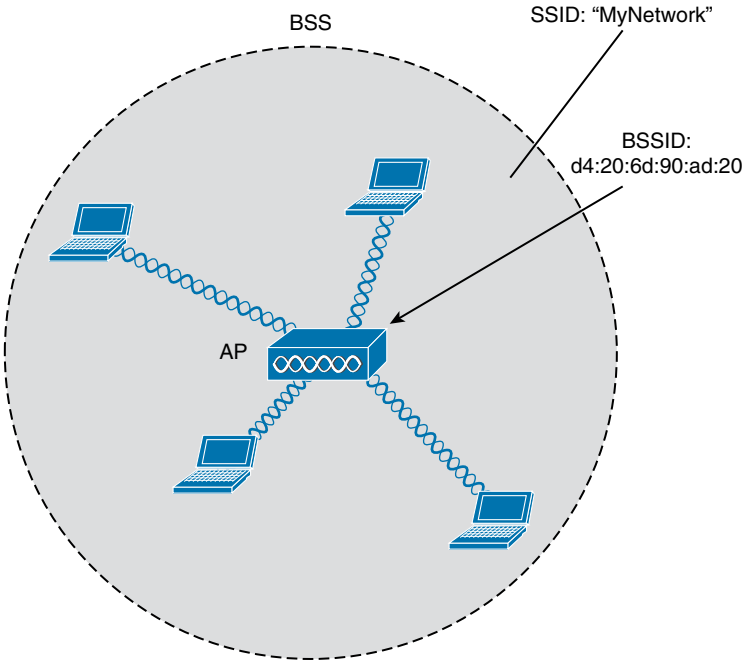


Figure 26-4 *802.11 Basic Service Set*

In addition, the AP advertises the wireless network with a Service Set Identifier (SSID), which is a text string containing a logical name. Think of the BSSID as a machine-readable name tag that uniquely identifies the BSS ambassador (the AP), and the SSID as a nonunique, human-readable name tag that identifies the wireless service.

Membership with the BSS is called an *association*. A wireless device must send an association request to the AP and the AP must either grant or deny the request. Once associated, a device becomes a client, or an 802.11 *station* (STA), of the BSS. What then? As long as a wireless client remains associated with a BSS, most communications to and from the client must pass *through* the AP, as indicated in Figure 26-5. By using the BSSID as a source or destination address, data frames can be relayed to or from the AP.

You might be wondering why all client traffic has to traverse the AP at all. Why can two clients not simply transmit data frames directly to each other and bypass the middleman? If clients

are allowed to communicate directly, then the whole idea of organizing and managing a BSS is moot. By sending data through the AP first, the BSS remains stable and under control.

NOTE Even though data frames are meant to pass through an AP, keep in mind that other devices in the same general area that are listening on the same channel can overhear the transmissions. After all, wireless frames are not contained within a wire that connects a device to an AP. Instead, the frames are freely available over the air to anyone that is within range to receive them. If the frames are unencrypted, then anyone may inspect their contents. Only the BSSID value contained within the frames indicates that the intended sender or recipient is the AP.

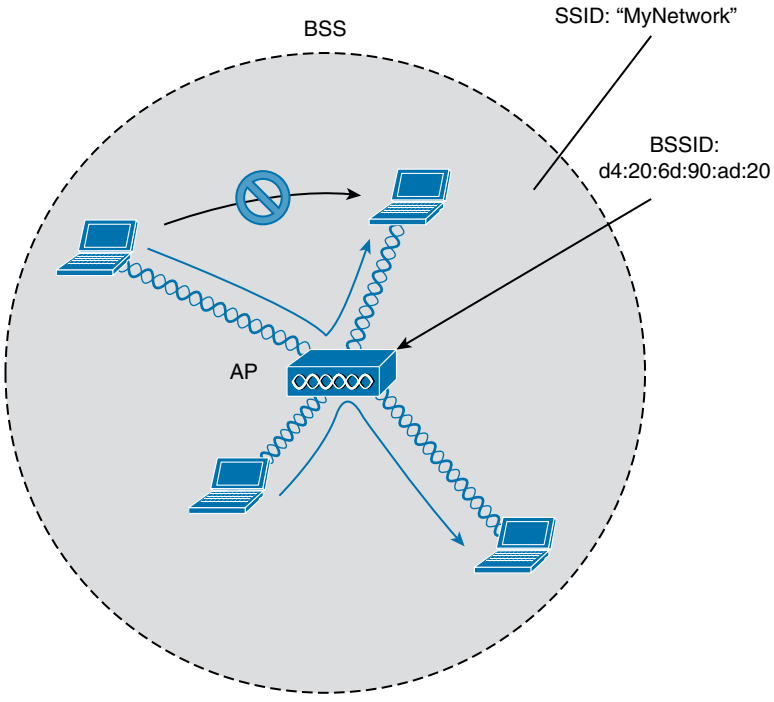


Figure 26-5 *Traffic Flows Within a BSS*

Distribution System

Notice that a BSS involves a single AP and no explicit connection into a regular Ethernet network. In that setting, the AP and its associated clients make up a standalone network. But the AP's role at the center of the BSS does not just stop with managing the BSS; sooner or later, wireless clients will need to communicate with other devices that are not members of the BSS. Fortunately, an AP can also uplink into an Ethernet network because it has both wireless and wired capabilities. The 802.11 standard refers to the upstream wired Ethernet as the *distribution system* (DS) for the wireless BSS, as shown in Figure 26-6.

You can think of an AP as a translational bridge, where frames from two dissimilar media (wireless and wired) are translated and then bridged at Layer 2. In simple terms, the AP is in charge of mapping a virtual local-area network (VLAN) to an SSID. In Figure 26-6, the AP

maps VLAN 10 to the wireless LAN using SSID "MyNetwork." Clients associated with the "MyNetwork" SSID will appear to be connected to VLAN 10.

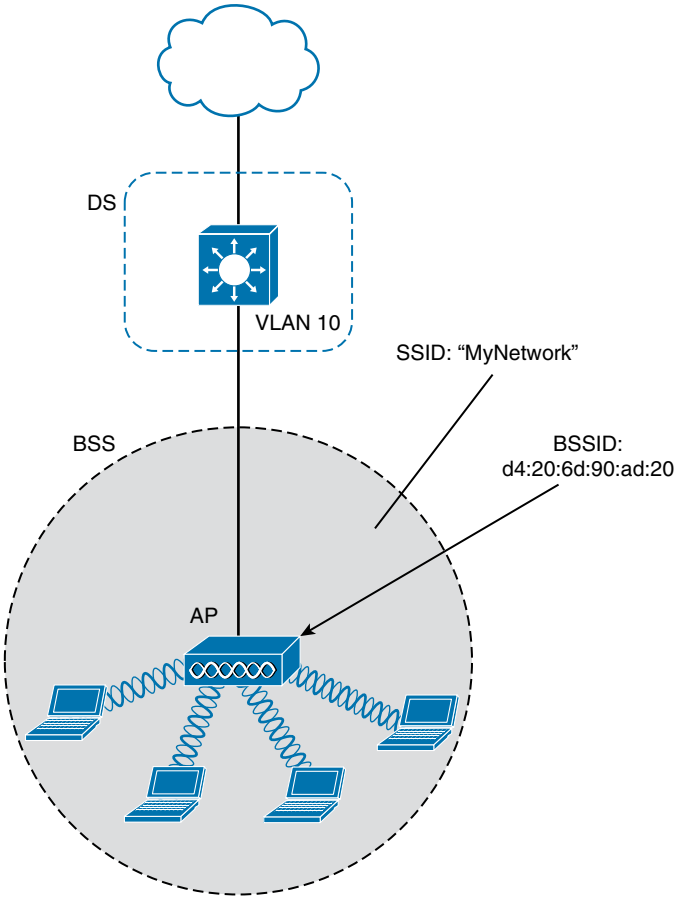


Figure 26-6 *Distribution System Supporting a BSS*

This concept can be extended so that multiple VLANs are mapped to multiple SSIDs. To do this, the AP must be connected to the switch by a trunk link that carries the VLANs. In Figure 26-7, VLANs 10, 20, and 30 are trunked to the AP over the DS. The AP uses the 802.1Q tag to map the VLAN numbers to the appropriate SSIDs. For example, VLAN 10 is mapped to SSID "MyNetwork," VLAN 20 is mapped to SSID "YourNetwork," and VLAN 30 to SSID "Guest."

In effect, when an AP uses multiple SSIDs, it is trunking VLANs over the air, and over the same channel, to wireless clients. The clients must use the appropriate SSID that has been mapped to the respective VLAN when the AP was configured. The AP then appears as multiple logical APs—one per BSS—with a unique BSSID for each. With Cisco APs, this is usually accomplished by incrementing the last digit of the radio's MAC address for each SSID.

Even though an AP can advertise and support multiple logical wireless networks, each of the SSIDs covers the same geographic area. The reason is that the AP uses the same transmitter, receiver, antennas, and channel for every SSID that it supports. Beware of one misconception though: multiple SSIDs can give an illusion of scale. Even though wireless clients can be

distributed across many SSIDs, all of those clients must share the same AP's hardware and must contend for airtime on the same channel.

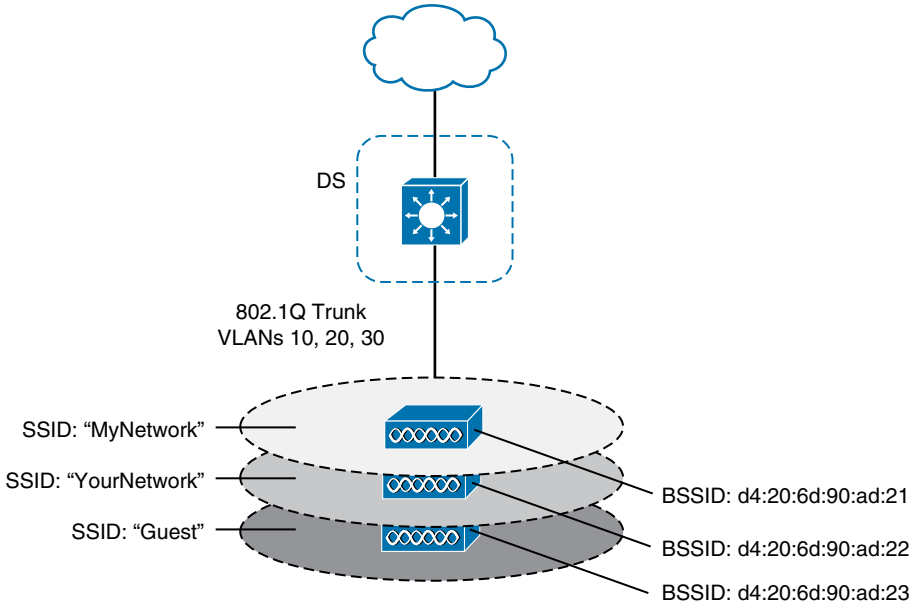


Figure 26-7 *Supporting Multiple SSIDs on One AP*

Extended Service Set

Normally, one AP cannot cover the entire area where clients might be located. For example, you might need wireless coverage throughout an entire floor of a business, hotel, hospital, or other large building. To cover more area than a single AP's cell can cover, you simply need to add more APs and spread them out geographically.

When APs are placed at different geographic locations, they can all be interconnected by a switched infrastructure. The 802.11 standard calls this an extended service set (ESS), as shown in Figure 26-8.

The idea is to make multiple APs cooperate so that the wireless service is consistent and seamless from the client's perspective. Ideally, any SSIDs that are defined on one AP should be defined on all the APs in an ESS; otherwise, it would be very cumbersome and inconvenient for a client to be reconfigured each time it moves into a different AP's cell.

Notice that each cell in Figure 26-8 has a unique BSSID, but both cells share one common SSID. Regardless of a client's location within the ESS, the SSID will remain the same but the client can always distinguish one AP from another.

In an ESS, a wireless client can associate with one AP while it is physically located near that AP. If the client later moves to a different location, it can associate with a different nearby AP automatically. Passing from one AP to another is called *roaming*. Keep in mind that each AP offers its own BSS on its own channel, to prevent interference between the APs. As a client device roams from one AP to another, it must scan the available channels to find a new AP (and BSS) to roam toward. In effect, the client is roaming from BSS to BSS, and from channel to channel.

Key Topic

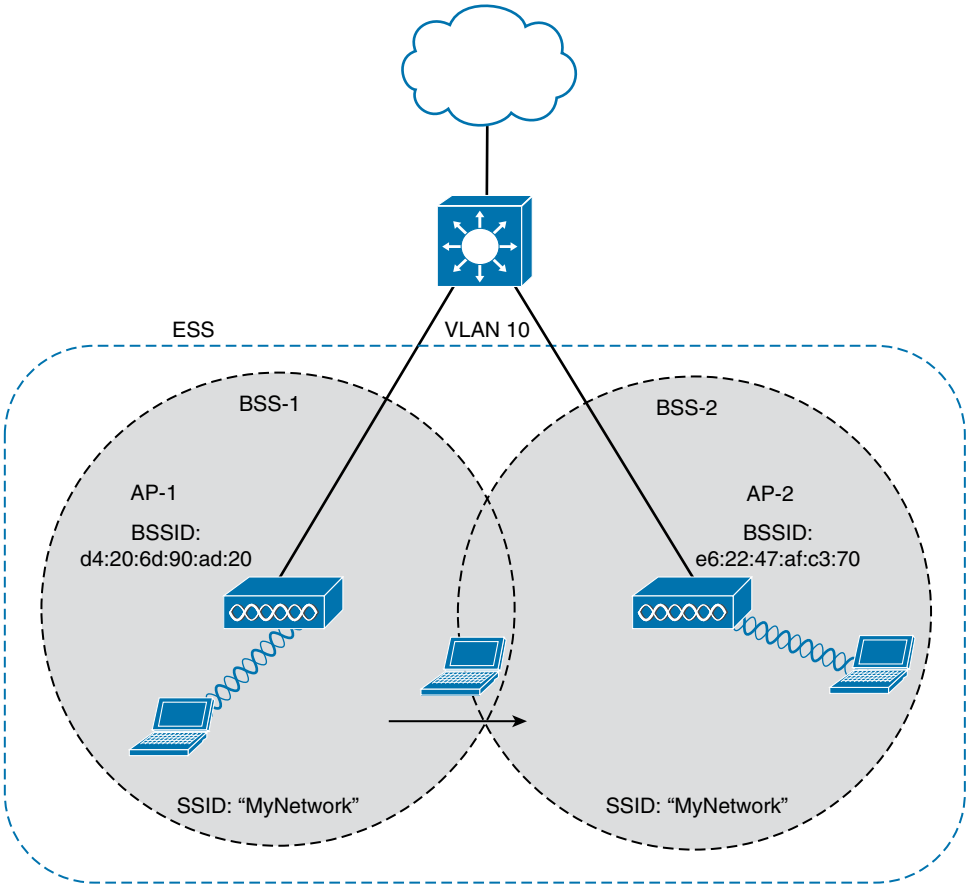


Figure 26-8 *Scaling Wireless Coverage with an 802.11 Extended Service Set*

Independent Basic Service Set

Usually a wireless network leverages APs for organization, control, and scalability. Sometimes that is not possible or convenient in an impromptu situation. For example, two people who want to exchange electronic documents at a meeting might not be able to find a BSS available or might want to avoid having to authenticate to a production network. In addition, many personal printers have the capability to print documents wirelessly, without relying on a regular BSS or AP.

The 802.11 standard allows two or more wireless clients to communicate directly with each other, with no other means of network connectivity. This is known as an *ad hoc* wireless network, or an *independent basic service set* (IBSS), as shown in Figure 26-9. For this to work, one of the devices must take the lead and begin advertising a network name and the necessary radio parameters, much like an AP would do. Any other device can then join as needed. IBSSs are meant to be organized in an impromptu, distributed fashion; therefore, they do not scale well beyond eight to ten devices.

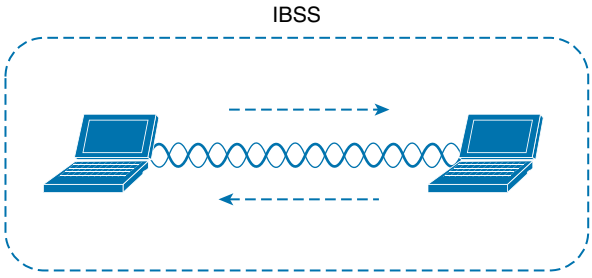


Figure 26-9 *802.11 Independent Basic Service Set*

Other Wireless Topologies

Wireless APs can be configured to operate in noninfrastructure modes when a normal BSS cannot provide the functionality that is needed. The following sections cover the most common modes.

Repeater

Normally, each AP in a wireless network has a wired connection back to the DS or switched infrastructure. To extend wireless coverage beyond a normal AP's cell footprint, additional APs and their wired connections can be added. In some scenarios, it is not possible to run a wired connection to a new AP because the cable distance is too great to support Ethernet communication.

In that case, you can add an additional AP that is configured for *repeater mode*. A wireless repeater takes the signal it receives and repeats or retransmits it in a new cell area around the repeater. The idea is to move the repeater out away from the AP so that it is still within range of both the AP and the distant client, as shown in Figure 26-10.

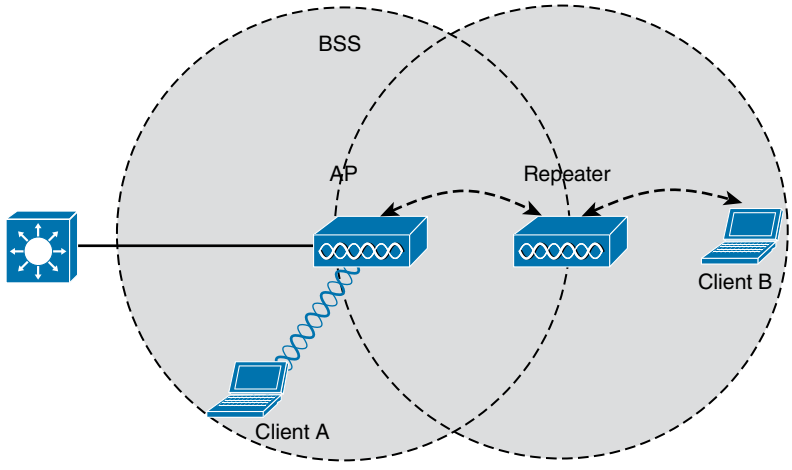


Figure 26-10 *Extending the Range of an AP with a Wireless Repeater*

If the repeater has a single transmitter and receiver, it must operate on the same channel that the AP is using. That can create the possibility that the AP's signal will be received and retransmitted by the repeater, only to be received again by the AP—halving the effective

throughput because the channel will be kept busy twice as long as before. As a remedy, some repeaters can use two transmitters and receivers to keep the original and repeated signals isolated on different channels. One transmitter and receiver pair is dedicated to signals in the AP's cell, while the other pair is dedicated to signals in the repeater's own cell.

Workgroup Bridge

Suppose you have a device that supports a wired Ethernet link but is not capable of having a wireless connection. For example, some mobile medical devices might be designed with only a wired connection. While it is possible to plug the device into an Ethernet connection when needed, a wireless connection would be much more practical. You can use a workgroup bridge (WGB) to connect the device's wired network adapter to a wireless network.

Rather than providing a BSS for wireless service, a WGB becomes a wireless client of a BSS. In effect, the WGB acts as an external wireless network adapter for a device that has none. In Figure 26-11, an AP provides a BSS; Client A is a regular wireless client, while Client B is associated with the AP through a WGB.

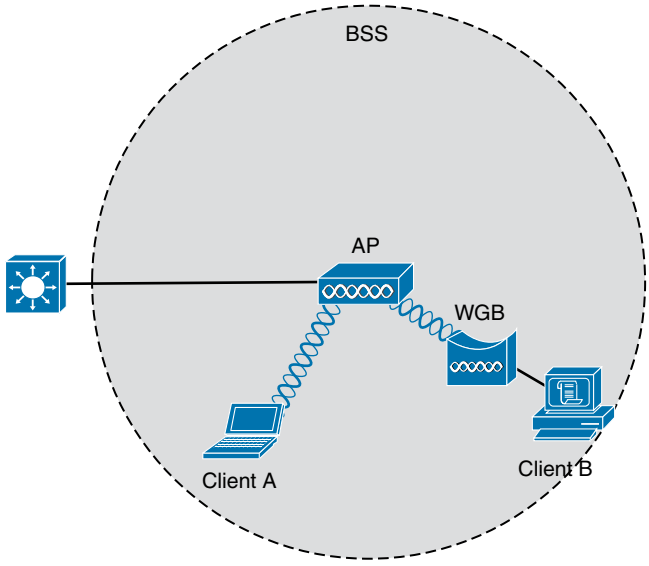


Figure 26-11 *Nonwireless Device Connecting Through a Workgroup Bridge*

You might encounter two types of workgroup bridges:

- **Universal workgroup bridge (uWGB):** A single wired device can be bridged to a wireless network.
- **Workgroup bridge (WGB):** A Cisco-proprietary implementation that allows multiple wired devices to be bridged to a wireless network.

Outdoor Bridge

An AP can be configured to act as a bridge to form a single wireless link from one LAN to another over a long distance. Outdoor bridged links are commonly used for connectivity between buildings or between cities.

If the LANs at two locations need to be bridged, a point-to-point bridged link can be used. One AP configured in bridge mode is needed on each end of the wireless link. Special purpose antennas are normally used with the bridges to focus their signals in one direction—toward the antenna of the AP at the far end of the link. This maximizes the link distance, as shown in Figure 26-12.

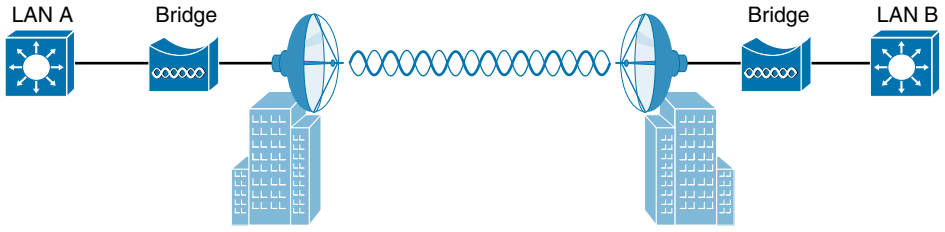


Figure 26-12 *Point-to-Point Outdoor Bridge*

Sometimes the LANs at multiple sites need to be bridged together. A point-to-multipoint bridged link allows a central site to be bridged to several other sites. The central site bridge is connected to an omnidirectional antenna, such that its signal is transmitted equally in all directions so that it can reach the other sites simultaneously. The bridges at each of the other sites can be connected to a directional antenna aimed at the central site. Figure 26-13 shows the point-to-multipoint scenario.

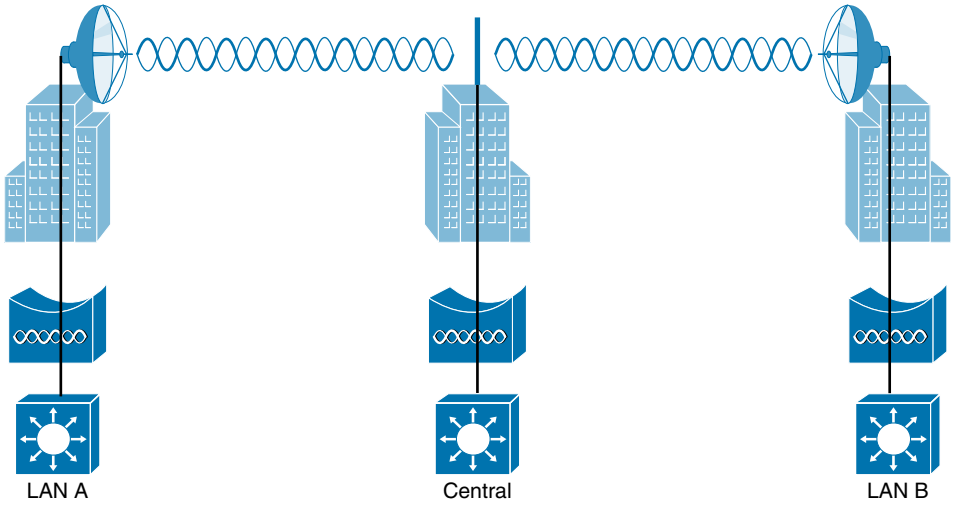


Figure 26-13 *Point-to-Multipoint Outdoor Bridge*

Mesh Network

To provide wireless coverage over a very large area, it is not always practical to run Ethernet cabling to every AP that would be needed. Instead, you could use multiple APs configured in mesh mode. In a mesh topology, wireless traffic is bridged from AP to AP, in a daisy-chain fashion, using another wireless channel.

Mesh APs can leverage dual radios—one using a channel in one range of frequencies and one a different range. Each mesh AP usually maintains a BSS on one channel, with which

wireless clients can associate. Client traffic is then usually bridged from AP to AP over other channels as a backhaul network. At the edge of the mesh network, the backhaul traffic is bridged to the wired LAN infrastructure. Figure 26-14 shows a typical mesh network. With Cisco APs, you can build a mesh network indoors or outdoors. The mesh network runs its own dynamic routing protocol to work out the best path for backhaul traffic to take across the mesh APs.

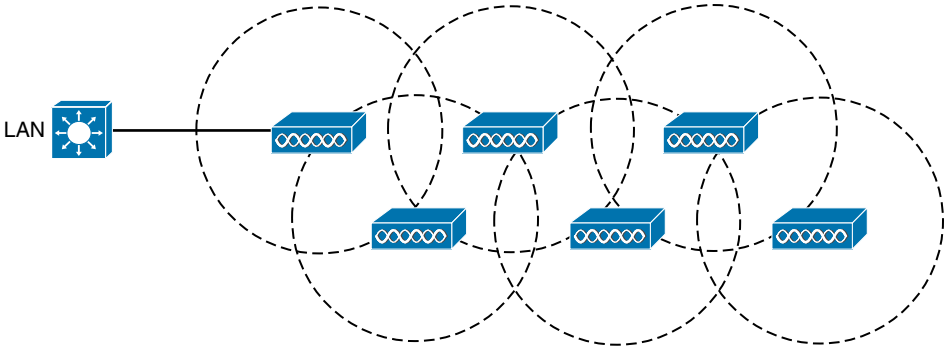


Figure 26-14 *Typical Wireless Mesh Network*

RF Overview

To send data across a wired link, an electrical signal is applied at one end and carried to the other end. The wire itself is continuous and conductive, so the signal can propagate rather easily. A wireless link has no physical strands of anything to carry the signal along.

How, then, can an electrical signal be sent across the air, or free space? Consider a simple analogy of two people standing far apart. One person wants to signal something to the other. They are connected by a long and somewhat loose rope; the rope represents free space. The sender at one end decides to lift his end of the rope high and hold it there so that the other end of the rope will also rise and notify the partner. After all, if the rope were a wire, he knows that he could apply a steady voltage at one end of the wire and it would appear at the other end. Figure 26-15 shows the end result; the rope falls back down after a tiny distance, and the receiver never notices a change.

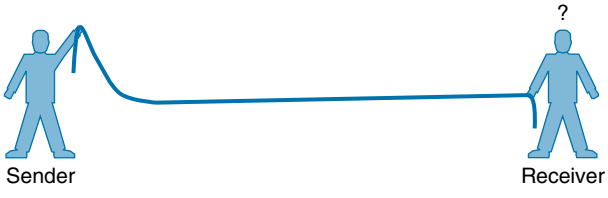


Figure 26-15 *Failed Attempt to Pass a Message Down a Rope*

The sender tries a different strategy. He cannot push the rope, but when he begins to wave it up and down in a steady, regular motion, a curious thing happens. A continuous wave pattern appears along the entire length of the rope, as shown in Figure 26-16. In fact, the waves (each representing one up and down cycle of the sender's arm) actually travel from the sender to the receiver.

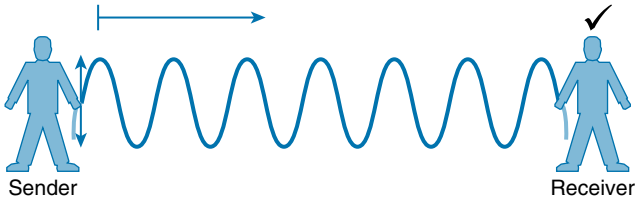


Figure 26-16 *Sending a Continuous Wave Down a Rope*

In free space, a similar principle occurs. The sender (a transmitter) can send an alternating current into a section of wire (an antenna), which sets up moving electric and magnetic fields that propagate out and away as traveling waves. The electric and magnetic fields travel along together and are always at right angles to each other, as shown in Figure 26-17. The signal must keep changing, or alternating, by cycling up and down, to keep the electric and magnetic fields cycling and pushing ever outward.

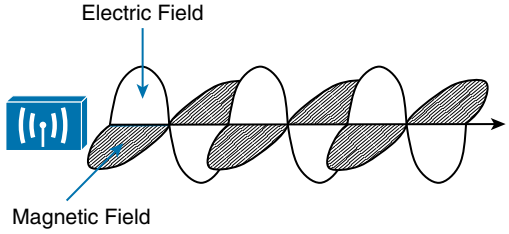


Figure 26-17 *Traveling Electric and Magnetic Waves*

Electromagnetic waves do not travel in a straight line. Instead, they travel by expanding in *all* directions away from the antenna. To get a visual image, think of dropping a pebble into a pond when the surface is still. Where it drops in, the pebble sets the water's surface into a cyclic motion. The waves that result begin small and expand outward, only to be replaced by new waves. In free space, the electromagnetic waves expand outward in all three dimensions.

Figure 26-18 shows a simple idealistic antenna that is a single point at the end of a wire. The waves produced expand outward in a spherical shape. The waves will eventually reach the receiver, in addition to many other locations in other directions.

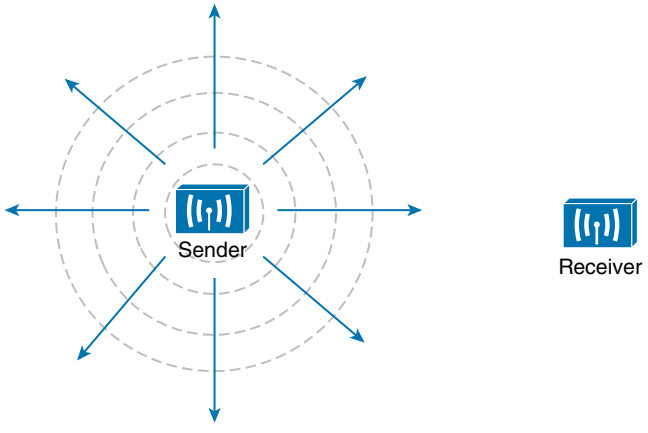


Figure 26-18 *Wave Propagation with an Idealistic Antenna*

At the receiving end of a wireless link, the process is reversed. As the electromagnetic waves reach the receiver's antenna, they induce an electrical signal. If everything works right, the received signal will be a reasonable copy of the original transmitted signal.

The electromagnetic waves involved in a wireless link can be measured and described in several ways. One fundamental property is the *frequency* of the wave, or the number of times the signal makes one complete up and down *cycle* in 1 second. Figure 26-19 shows how a cycle of a wave can be identified. A cycle can begin as the signal rises from the center line, falls through the center line, and rises again to meet the center line. A cycle can also be measured from the center of one peak to the center of the next peak. No matter where you start measuring a cycle, the signal must make a complete sequence back to its starting position where it is ready to repeat the same cyclic pattern.

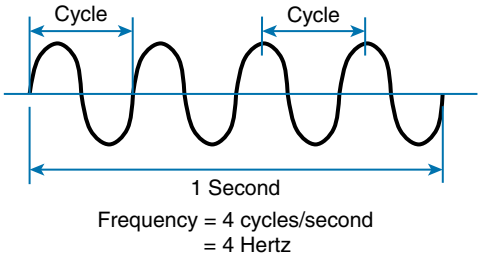


Figure 26-19 *Cycles Within a Wave*

In Figure 26-19, suppose that 1 second has elapsed, as shown. During that 1 second, the signal progressed through four complete cycles. Therefore, its frequency is 4 cycles/second or 4 hertz. A *hertz* (Hz) is the most commonly used frequency unit and is nothing other than one cycle per second.

Frequency can vary over a very wide range. As frequency increases by orders of magnitude, the numbers can become quite large. To keep things simple, the frequency unit name can be modified to denote an increasing number of zeros, as listed in Table 26-2.

Table 26-2 Frequency Unit Names

Unit	Abbreviation	Meaning
Hertz	Hz	Cycles per second
Kilohertz	kHz	1000 Hz
Megahertz	MHz	1,000,000 Hz
Gigahertz	GHz	1,000,000,000 Hz

Figure 26-20 shows a simple representation of the continuous frequency spectrum ranging from 0 Hz to 10^{22} (or 1 followed by 22 zeros) Hz. At the low end of the spectrum are frequencies that are too low to be heard by the human ear, followed by audible sounds. The highest range of frequencies contains light, followed by X, gamma, and cosmic rays.

Frequency (Hz)	Frequency Notation
10^{21}	
10^{20}	
10^{19}	
10^{18}	
10^{17}	
10^{16}	
10^{15}	
10^{14}	
10^{13}	
10^{12}	
10^{11}	100 GHz
10^{10}	10 GHz
10^{9}	1 GHz
10^{8}	100 MHz
10^{7}	10 MHz
10^{6}	1 MHz
10^{5}	100 kHz
10^{4}	10 kHz
10^{3}	1 kHz
10^{2}	100 Hz
10^{1}	10 Hz
0	0 Hz

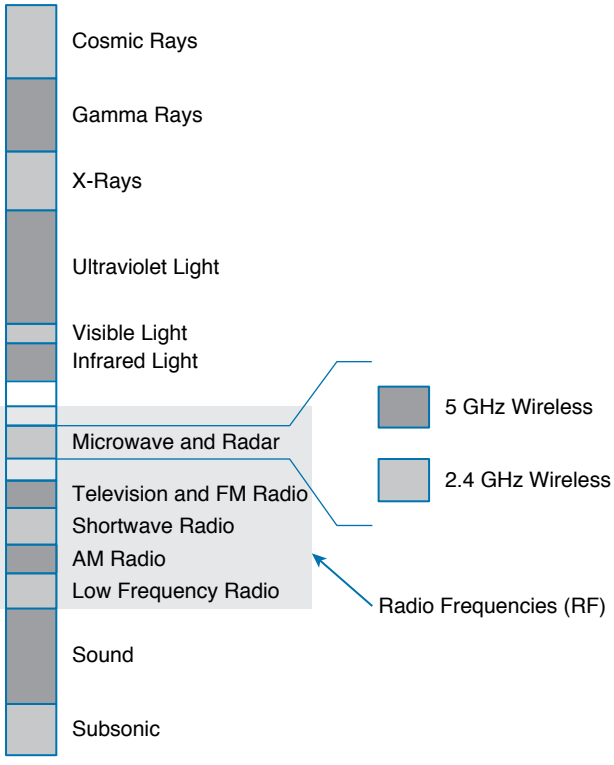


Figure 26-20 *Continuous Frequency Spectrum*

The frequency range from around 3 kHz to 300 GHz is commonly called *radio frequency* (RF). It includes many different types of radio communication, including low-frequency radio, AM radio, shortwave radio, television, FM radio, microwave, and radar. The microwave category also contains the two main frequency ranges that are used for wireless LAN communication: 2.4 and 5 GHz.

Wireless Bands and Channels

Because a range of frequencies might be used for the same purpose, it is customary to refer to the range as a *band* of frequencies. For example, the range from 530 kHz to around 1710 kHz is used by AM radio stations; therefore, it is commonly called the AM band or the AM broadcast band.

One of the two main frequency ranges used for wireless LAN communication lies between 2.400 and 2.4835 GHz. This is usually called the *2.4-GHz band*, even though it does not encompass the entire range between 2.4 and 2.5 GHz. It is much more convenient to refer to the band name instead of the specific range of frequencies included.

The other wireless LAN range is usually called the *5-GHz band* because it lies between 5.150 and 5.825 GHz. The 5-GHz band actually contains the following four separate and distinct bands:

5.150 to 5.250 GHz

5.250 to 5.350 GHz

5.470 to 5.725 GHz

5.725 to 5.825 GHz

TIP You might have noticed that most of the 5-GHz bands are contiguous except for a gap between 5.350 and 5.470. At the time of this writing, this gap exists and cannot be used for wireless LANs. However, some governmental agencies have moved to reclaim the frequencies and repurpose them for wireless LANs. Efforts are also underway to add 5.825 through 5.925 GHz.

It is interesting that the 5-GHz band can contain several smaller bands. Remember that the term *band* is simply a relative term that is used for convenience. Do not worry about memorizing the band names or exact frequency ranges; just be aware of the two main bands at 2.4 and 5 GHz.

A frequency band contains a continuous range of frequencies. If two devices require a single frequency for a wireless link between them, which frequency can they use? Beyond that, how many unique frequencies can be used within a band? To keep everything orderly and compatible, bands are usually divided into a number of distinct *channels*. Each channel is known by a channel number and is assigned to a specific frequency. As long as the channels are defined by a national or international standards body, they can be used consistently in all locations. Figures 26-21 and 26-22 show the channel layout for the 2.4 and 5 GHz bands, respectively.

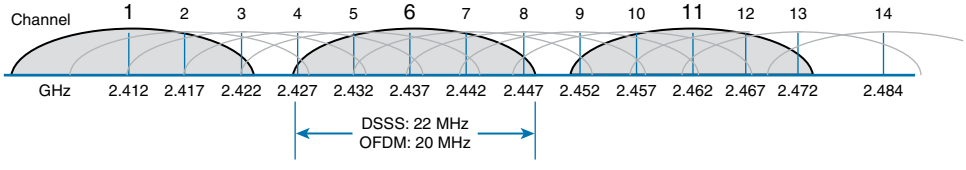


Figure 26-21 *Channel Layout in the 2.4-GHz Band*

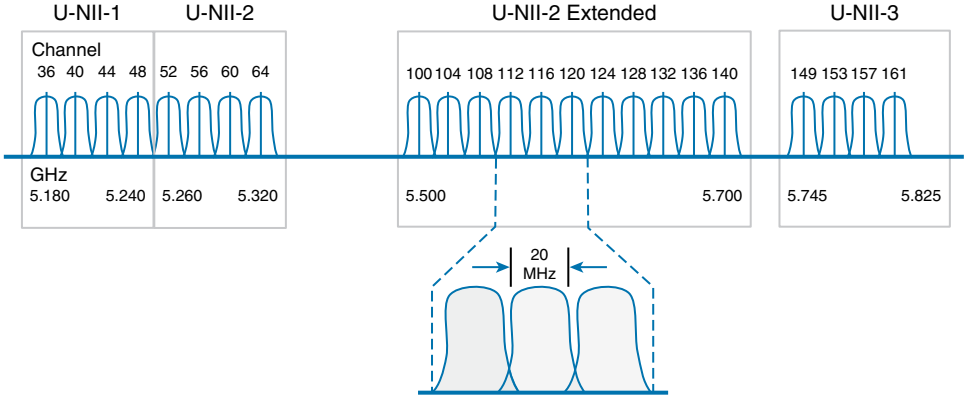


Figure 26-22 *Channel Layout in the 5-GHz Band*

You might assume that an AP can use any channel number without affecting any APs that use other channel numbers. In the 5-GHz band, this is the case because each channel is allocated a frequency range that does not encroach on or overlap the frequencies allocated for any other channel. In other words, the 5-GHz band consists of *nonoverlapping channels*.

Key Topic

The same is *not* true of the 2.4-GHz band. Each of its channels is much too wide to avoid overlapping the next lower or upper channel number. In fact, each channel covers the frequency range that is allocated to more than four consecutive channels! Notice the width of the channel spacing in Figure 26-21 as compared to the width of one of the shaded signals centered on channels 1, 6, and 11. The only way to avoid any overlap between adjacent channels is to configure APs to use only channels 1, 6, and 11. Even though there are 14 channels available to use, you should always strive for nonoverlapping channels in your network.

APs and Wireless Standards

It might be obvious that wireless devices and APs should all be capable of operating on the same band. For example, a 5-GHz wireless phone can communicate only with an AP that offers Wi-Fi service on 5-GHz channels. In addition, the devices and APs must also share a compatibility with the parts of the 802.11 standard they support.

As the IEEE 802.11 Wi-Fi standard evolves and develops, new amendments with new functionality get proposed. These amendments are known by "802.11" followed by a one- or two-letter suffix until they are accepted and rolled up into the next generation of the complete 802.11 standard. Even then, it is common to see the amendment suffixes still used to distinguish specific functions.

You should be aware of several amendments that define important characteristics such as data rates, methods used to transmit and receive data, and so on. For the CCNA 200-301 exam, you should know which band each of the amendments listed in Table 26-3 uses. The ENCOR 300-401 exam goes further into the data rates and modulation and coding schemes used by each.

Key Topic

Table 26-3 Basic Characteristics of Some IEEE 802.11 Amendments

Amendment	2.4 GHz	5 GHz	Max Data Rate	Notes
802.11-1997	Yes	No	2 Mbps	The original 802.11 standard ratified in 1997
802.11b	Yes	No	11 Mbps	Introduced in 1999
802.11g	Yes	No	54 Mbps	Introduced in 2003
802.11a	No	Yes	54 Mbps	Introduced in 1999
802.11n	Yes	Yes	600 Mbps	HT (high throughput), introduced in 2009
802.11ac	No	Yes	6.93 Gbps	VHT (very high throughput), introduced in 2013
802.11ax	Yes	Yes	4x 802.11ac	High Efficiency Wireless, Wi-Fi6; expected late 2019; will operate on other bands too, as they become available

The 802.11 amendments are not mutually exclusive. Wireless client devices and APs can be compatible with one or more amendments; however, a client and an AP can communicate only if they both support and agree to use the same amendment. When you look at the specifications for a wireless device, you may find supported amendments listed in a single string, separated by slashes. For example, a device that supports 802.11b/g will support both 802.11b and 802.11g. One that supports b/g/a/n/ac will support 802.11b, 802.11g, 802.11n, and 802.11ac. You should become familiar with Table 26-3 so that you can know which bands a device can use based on its 802.11 amendment support.

If a device can operate on both bands, how does it decide which band to use? APs can usually operate on both bands simultaneously to support any clients that might be present on each band. However, wireless clients typically associate with an AP on one band at a time, while scanning for potential APs on both bands. The band used to connect to an AP is chosen according to the operating system, wireless adapter driver, and other internal configuration. A wireless client can initiate an association with an AP on one band and then switch to the other band if the signal conditions are better there.

NOTE Cisco APs have dual radios (sets of transmitters and receivers) to support BSSs on one 2.4-GHz channel and other BSSs on one 5-GHz channel simultaneously. Some models also have two 5-GHz radios that can be configured to operate BSSs on two different channels at the same time, providing wireless coverage to higher densities of users that are located in the same vicinity.

You can configure a Cisco AP to operate on a specific channel number. As the number of APs grows, manual channel assignment can become a difficult task. Fortunately, Cisco wireless architectures can automatically and dynamically assign each AP to an appropriate channel. The architecture is covered in Chapter 27, "Analyzing Cisco Wireless Architectures," while dynamic channel assignment is covered on the ENCOR 300-401 exam.

In open space, RF signals propagate or reach further on the 2.4-GHz band than on the 5-GHz band. They also tend to penetrate indoor walls and objects easier at 2.4 GHz than 5 GHz. However, the 2.4-GHz band is commonly more crowded with wireless devices. Remember that only three nonoverlapping channels are available, so the chances of other neighboring APs using the same channels is greater. In contrast, the 5-GHz band has many more channels available to use, making channels less crowded and experiencing less interference.

Chapter Review

Review this chapter's material using either the tools in the book or the interactive tools for the same material found on the book's companion website. Table 26-4 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 26-4 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 memory tables		Website

Review All the Key Topics

Key Topic

Table 26-5 Key Topics for Chapter 26

Key Topic Element	Description	Page Number
Figure 26-4	Basic service set	615
Figure 26-7	Multiple SSIDs	618
Figure 26-8	Extended service set	619
Paragraph	Nonoverlapping channels and bands	628
Table 26-3	Basic Characteristics of Some 802.11 Amendments	628

Key Terms You Should Know

access point (AP), ad hoc network, Band, basic service set (BSS), Basic Service Set Identifier (BSSID), channel, cell, distribution system (DS), extended service set (ESS), independent basic service set (IBSS), infrastructure mode, mesh network, nonoverlapping channels, point-to-point bridge, repeater, roaming, Service Set Identifier (SSID), station (STA), workgroup bridge (WGB)

This page intentionally left blank

CHAPTER 27

Analyzing Cisco Wireless Architectures

This chapter covers the following exam topics:

2.0 Network Access

2.6 Compare Cisco Wireless Architectures and AP modes

In Chapter 26, "Fundamentals of Wireless Networks," you learned about how a single access point (AP) can provide a basic service set (BSS) for a cell area and how multiple APs can be connected to form an extended service set (ESS) for a larger network. In this chapter, you learn more about different approaches or architectures that allow APs to be networked together for an enterprise. You also learn how some architectures are more scalable than others and how to manage each type of wireless network architecture.

As you work through this chapter, think about how each architecture can be applied to specific environments—how easy it would be to manage, deploy, and troubleshoot the network, how the APs can be controlled, and how data would move through the network.

"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 27-1 "Do I Know This Already?" Section-to-Question Mapping

Foundation Topics Section	Questions
Autonomous AP Architectures	1
Cloud-based AP Architecture	2
Split-MAC Architectures	3–5
Comparing Wireless LAN Controller Deployments	6
Cisco AP Modes	7–8

1. Which one of the following terms best describes a Cisco wireless access point that operates in a standalone, independent manner?
   - **a.** Autonomous AP
   - **b.** Independent AP
   - **c.** Lightweight AP
   - **d.** Embedded AP

2. The Cisco Meraki cloud-based APs are most accurately described by which one of the following statements?
   a. Autonomous APs joined to a WLC
   b. Autonomous APs centrally managed
   c. Lightweight APs joined to a WLC
   d. Lightweight APs centrally managed
3. A lightweight access point is said to participate in which one of the following architectures?
   a. Light-MAC
   b. Tunnel-MAC
   c. Split-MAC
   d. Big-MAC
4. How does a lightweight access point communicate with a wireless LAN controller?
   a. Through an IPsec tunnel
   b. Through a CAPWAP tunnel
   c. Through a GRE tunnel
   d. Directly over Layer 2
5. Which one of the following is not needed for a lightweight AP in default local mode to be able to support three SSIDs that are bound to three VLANs?
   a. A trunk link carrying three VLANs
   b. An access link bound to a single VLAN
   c. A WLC connected to three VLANs
   d. A CAPWAP tunnel to a WLC
6. Which one of the following WLC deployment models would be best for a large enterprise with around 3000 lightweight APs?
   a. Cisco Mobility Express
   b. Embedded
   c. Unified
   d. Cloud-based
7. If a lightweight AP provides at least one BSS for wireless clients, which one of the following modes does it use?
   a. Local
   b. Normal
   c. Monitor
   d. Client

8. Regarding lightweight AP modes, which one of the following is true?

   a. An AP can operate in multiple modes at the same time.
   b. An AP only has one possible mode of operation.
   c. The Run mode is the default mode.
   d. The SE-Connect mode is used for spectrum analysis.

Foundation Topics

Autonomous AP Architecture

An access point's primary function is to bridge wireless data from the air to a normal wired network. An AP can accept "connections" from a number of wireless clients so that they become members of the LAN, as if the same clients were using wired connections.

APs act as the central point of access (hence the AP name), controlling client access to the wireless LAN. An *autonomous AP* is self-contained; it is equipped with both wired and wireless hardware so that the wireless client associations can be terminated onto a wired connection locally at the AP. The APs and their data connections must be distributed across the coverage area and across the network.

Autonomous APs offer one or more fully functional, standalone basic service sets (BSSs). They are also a natural extension of a switched network, connecting wireless service set identifiers (SSIDs) to wired virtual LANs (VLANs) at the access layer. Figure 27-1 shows the basic architecture; even though only four APs are shown across the bottom, a typical enterprise network could consist of hundreds or thousands of APs.

Key Topic

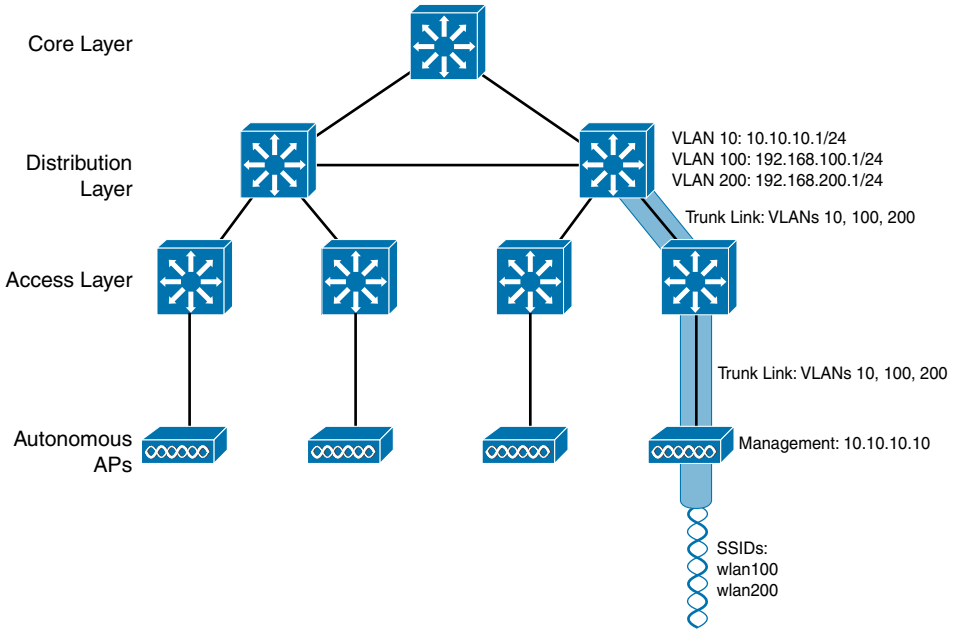


Figure 27-1 *Wireless Network Architecture with Autonomous APs*

What exactly does an autonomous AP need to become a part of the network? The wireless network in Figure 27-1 consists of two SSIDs: wlan100 and wlan200. These correspond to wired VLANs 100 and 200, respectively. As shown by the shaded links, the VLANs must be trunked from the distribution layer switch (where routing commonly takes place) to the access layer, where they are extended further over a trunk link to the AP.

An autonomous AP offers a short and simple path for data to travel between the wireless and wired networks. Data has to travel only through the AP to reach the network on the other side. Two wireless users that are associated to the same autonomous AP can reach each other through the AP without having to pass up into the wired network. As you work through the wireless architectures discussed in the rest of the chapter, notice the data path that is required for each.

An autonomous AP must also be configured with a management IP address (10.10.10.10 in Figure 27-1) so that you can remotely manage it. After all, you will want to configure SSIDs, VLANs, and many RF parameters like the channel and transmit power to be used. The management address is not normally part of any of the data VLANs, so a dedicated management VLAN (i.e., VLAN 10) must be added to the trunk links to reach the AP. Each AP must be configured and maintained individually unless you leverage a management platform such as Cisco Prime Infrastructure or Cisco DNA Center.

Because the data and management VLANs may need to reach every autonomous AP, the network configuration and efficiency can become cumbersome as the network scales. For example, you will likely want to offer the same SSID on many APs so that wireless clients can associate with that SSID in most any location or while roaming between any two APs. You might also want to extend the corresponding VLAN (and IP subnet) to each and every AP so that clients do not have to request a new IP address for each new association.

Because SSIDs and their VLANs must be extended at Layer 2, you should consider how they are extended throughout the switched network. The shaded links in Figure 27-2 show an example of a single VLAN's extent in the data plane. Working top to bottom, follow VLAN 100 as it reaches through the network. VLAN 100 is routed within the distribution layer and must be carried over trunk links to the access layer switches and then to each autonomous AP. In effect, VLAN 100 must extend end to end across the whole infrastructure—something that is usually considered to be a bad practice.

That might sound straightforward until you have to add a new VLAN and configure every switch and AP in your network to carry and support it. Even worse, suppose your network has redundant links between each layer of switches. The Spanning Tree Protocol (STP) running on each switch becomes a vital ingredient to prevent bridging loops from forming and corrupting the network. For these reasons, client roaming across autonomous APs is typically limited to the Layer 2 domain, or the extent of a single VLAN. As the wireless network expands, the infrastructure becomes more difficult to configure correctly and becomes less efficient.

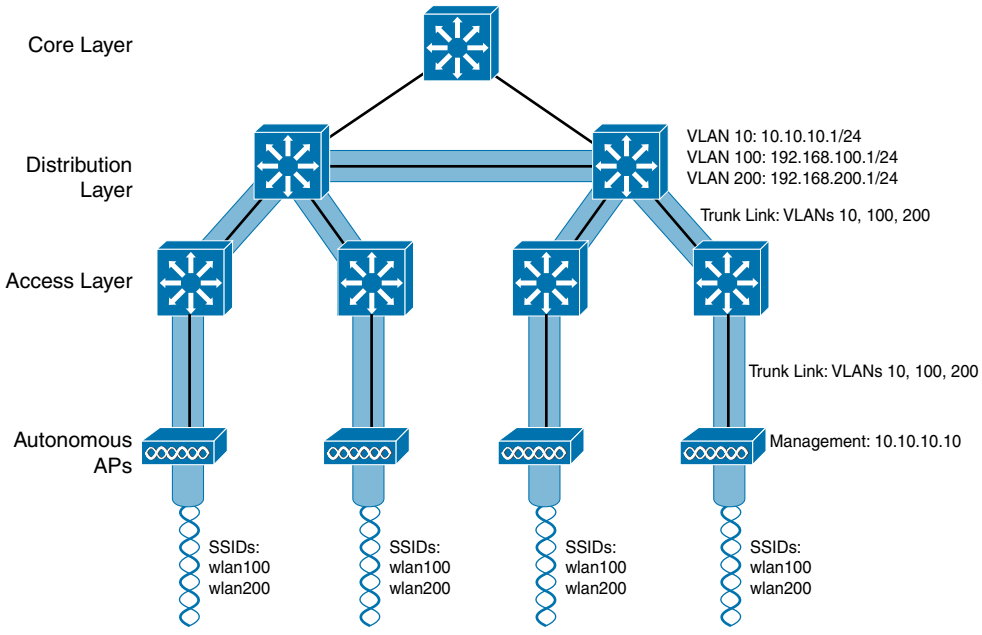


Figure 27-2 *Extent of a Data VLAN in a Network of Autonomous APs*

Cloud-based AP Architecture

Recall that an autonomous AP needs quite a bit of configuration and management. To help manage more and more autonomous APs as the wireless network grows, you could place an AP management platform such as Cisco Prime Infrastructure in a central location within the enterprise. The management platform would need to be purchased, configured, and maintained too.

A simpler approach is a *cloud-based AP* architecture, where the AP management function is pushed out of the enterprise and into the Internet cloud. Cisco Meraki is cloud-based and offers centralized management of wireless, switched, and security networks built from Meraki products. For example, through the cloud networking service, you can configure and manage APs, monitor wireless performance and activity, generate reports, and so on.

Cisco Meraki APs can be deployed automatically, once you register with the Meraki cloud. Each AP will contact the cloud when it powers up and will self-configure. From that point on, you can manage the AP through the Meraki cloud dashboard.

Figure 27-3 illustrates the basic cloud-based architecture. Notice that the network is arranged identically to that of the autonomous AP network. The reason is that the APs in a cloud-based network are all autonomous, too. The most visible difference is that all of the APs are managed, controlled, and monitored centrally from the cloud.

Answers to the "Do I Know This Already?" quiz:

1 A **2** B **3** C **4** B **5** A **6** C **7** A **8** D

Key Topic

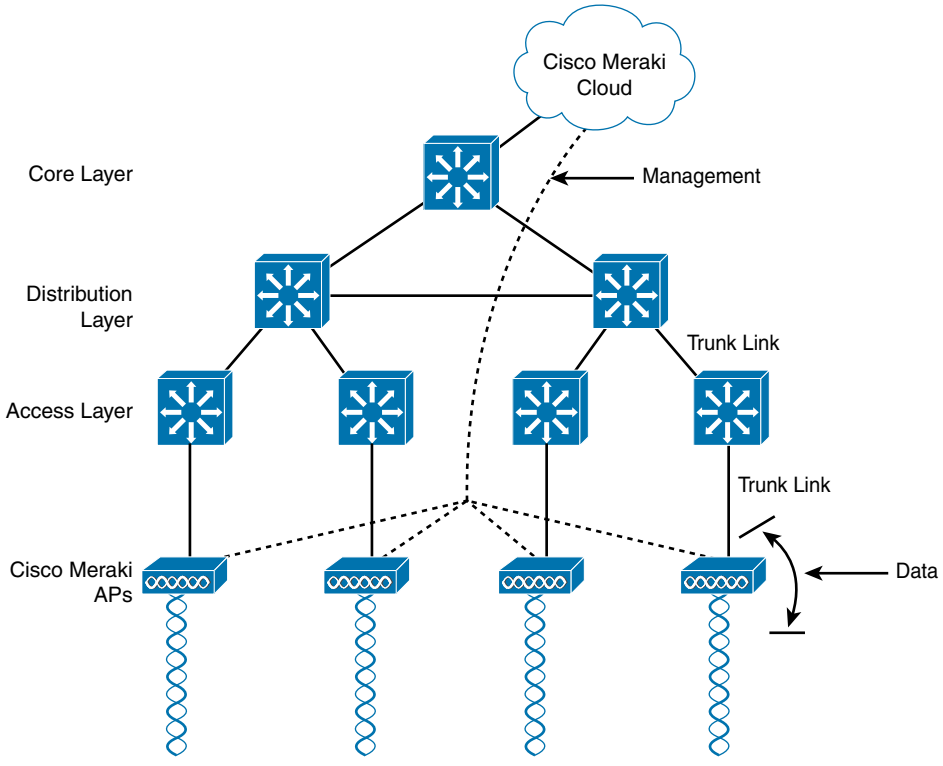


Figure 27-3 *Cisco Meraki Cloud-Based Wireless Network Architecture*

From the cloud, you can push out code upgrades and configuration changes to the APs in the enterprise. The Cisco Meraki cloud also adds the intelligence needed to automatically instruct each AP on which channel and transmit power level to use. It can also collect information from all of the APs about things such as RF interference, rogue or unexpected wireless devices that were overheard, and wireless usage statistics.

Finally, there are a couple of things you should observe about the cloud-based architecture. The data path from the wireless network to the wired network is very short; the autonomous AP links the two networks. Data to and from wireless clients does not have to travel up into the cloud and back; the cloud is used to bring management functions into the data plane.

Also, notice that the network in Figure 27-3 consists of two distinct paths—one for data traffic and another for management traffic, corresponding to the following two functions:

- **A control plane:** Traffic used to control, configure, manage, and monitor the AP itself
- **A data plane:** End-user traffic passing through the AP

This division will become important in the following sections as other types of architecture are discussed.

Split-MAC Architectures

Because autonomous APs are…well, autonomous, managing their RF operation can be quite difficult. As a network administrator, you are in charge of selecting and configuring the channel used by each AP and detecting and dealing with any rogue APs that might be interfering. You must also manage things such as the transmit power level to make sure that the wireless coverage is sufficient, it does not overlap too much, and there aren't any coverage holes—even when an AP's radio fails.

Managing wireless network security can also be difficult. Each autonomous AP handles its own security policies, with no central point of entry between the wireless and wired networks. That means there is no convenient place to monitor traffic for things such as intrusion detection and prevention, quality of service, bandwidth policing, and so on.

To overcome the limitations of distributed autonomous APs, many of the functions found within autonomous APs have to be shifted toward some central location. In Figure 27-4, most of the activities performed by an autonomous AP on the left are broken up into two groups—management functions on the top and real-time processes on the bottom.

Key Topic

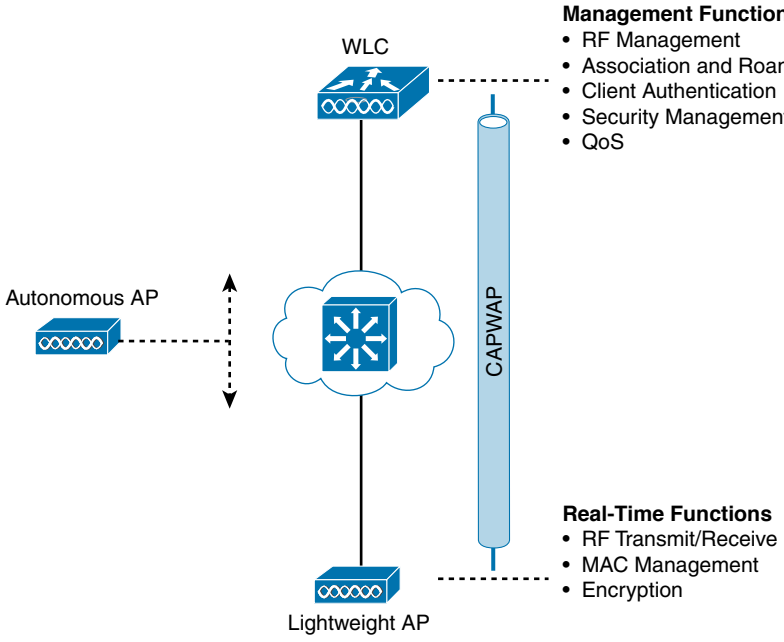


Figure 27-4 *Autonomous Versus Lightweight Access Point*

The real-time processes involve sending and receiving 802.11 frames, beacons, and probe messages. 802.11 data encryption is also handled in real time, on a per-packet basis. The AP must interact with wireless clients on some low level, known as the *Media Access Control* (MAC) layer. These functions must stay with the AP hardware, closest to the clients.

The management functions are not integral to handling frames over the RF channels, but are things that should be centrally administered. Therefore, those functions can be moved to a centrally located platform away from the AP.

When the functions of an autonomous AP are divided, the AP hardware is known as a *lightweight access point*, and performs only the real-time 802.11 operation. The lightweight AP gets its name because the code image and the local intelligence are stripped down, or lightweight, compared to the traditional autonomous AP.

The management functions are usually performed on a *wireless LAN controller* (WLC), which controls many lightweight APs. This is shown in the bottom right portion of Figure 27-4. Notice that the AP is left with duties in Layers 1 and 2, where frames are moved into and out of the RF domain. The AP becomes totally dependent on the WLC for every other WLAN function, such as authenticating users, managing security policies, and even selecting RF channels and output power.

NOTE Remember that a lightweight AP cannot normally operate on its own; it is very dependent on a WLC somewhere in the network. The only exception is the FlexConnect architecture, which is discussed later in this chapter.

The lightweight AP-WLC division of labor is known as a *split-MAC architecture*, where the normal MAC operations are pulled apart into two distinct locations. This occurs for every AP in the network; each one must boot and bind itself to a WLC to support wireless clients. The WLC becomes the central hub that supports a number of APs scattered about in the network.

How does a lightweight AP bind with a WLC to form a complete working access point? The two devices must use a tunneling protocol between them, to carry 802.11-related messages and also client data. Remember that the AP and WLC can be located on the same VLAN or IP subnet, but they do not have to be. Instead, they can be located on two entirely different IP subnets in two entirely different locations.

The Control and Provisioning of Wireless Access Points (*CAPWAP*) tunneling protocol makes this all possible by encapsulating the data between the LAP and WLC within new IP packets. The tunneled data can then be switched or routed across the campus network. As Figure 27-5 shows, the CAPWAP relationship actually consists of two separate tunnels, as follows:

- **CAPWAP control messages:** Carries exchanges that are used to configure the AP and manage its operation. The control messages are authenticated and encrypted, so the AP is securely controlled by only the appropriate WLC, then transported over the control tunnel.
- **CAPWAP data:** Used for packets traveling to and from wireless clients that are associated with the AP. Data packets are transported over the data tunnel but are not encrypted by default. When data encryption is enabled for an AP, packets are protected with Datagram Transport Layer Security (DTLS).

NOTE CAPWAP is defined in RFCs 5415, 5416, 5417, and 5418. CAPWAP is based on the Lightweight Access Point Protocol (LWAPP), which was a legacy Cisco proprietary solution.

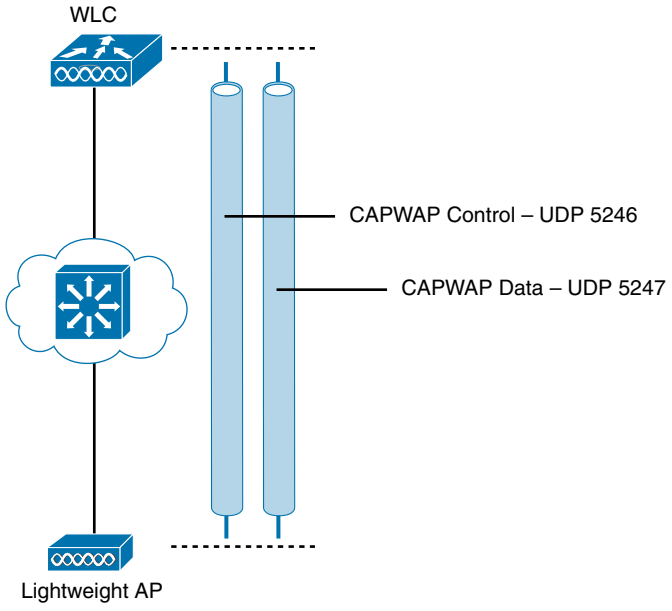


Figure 27-5 *Linking a Lightweight AP and WLC with CAPWAP*

Every AP and WLC must also authenticate each other with digital certificates. An X.509 certificate is preinstalled in each device when it is purchased. By using certificates behind the scenes, every device is properly authenticated before becoming part of the wireless network. This process helps assure that no one can add an unauthorized AP to your network.

The CAPWAP tunneling allows the AP and WLC to be separated geographically and logically. It also breaks the dependence on Layer 2 connectivity between them. For example, Figure 27-6 uses shaded areas to show the extent of VLAN 100. Notice how VLAN 100 exists at the WLC and in the air as SSID 100, near the wireless clients—but not in between the AP and the WLC. Instead, traffic to and from clients associated with SSID 100 is transported across the network infrastructure encapsulated inside the CAPWAP data tunnel. The tunnel exists between the IP address of the WLC and the IP address of the AP, which allows all of the tunneled packets to be routed at Layer 3.

Also, notice how the AP is known by only a single IP address: 10.10.10.10. Because the AP sits on the access layer where its CAPWAP tunnels terminate, it can use one IP address for both management and tunneling. No trunk link is needed because all of the VLANs it supports are encapsulated and tunneled as Layer 3 IP packets, rather than individual Layer 2 VLANs.

As the wireless network grows, the WLC simply builds more CAPWAP tunnels to reach more APs. Figure 27-7 depicts a network with four APs. Each AP has a control and a data tunnel back to the centralized WLC. SSID 100 can exist on every AP, and VLAN 100 can reach every AP through the network of tunnels.

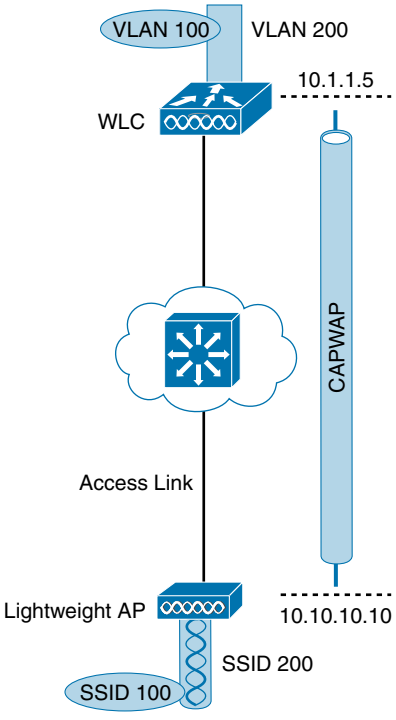


Figure 27-6 *Extent of VLAN 100 in a Cisco Wireless Network*

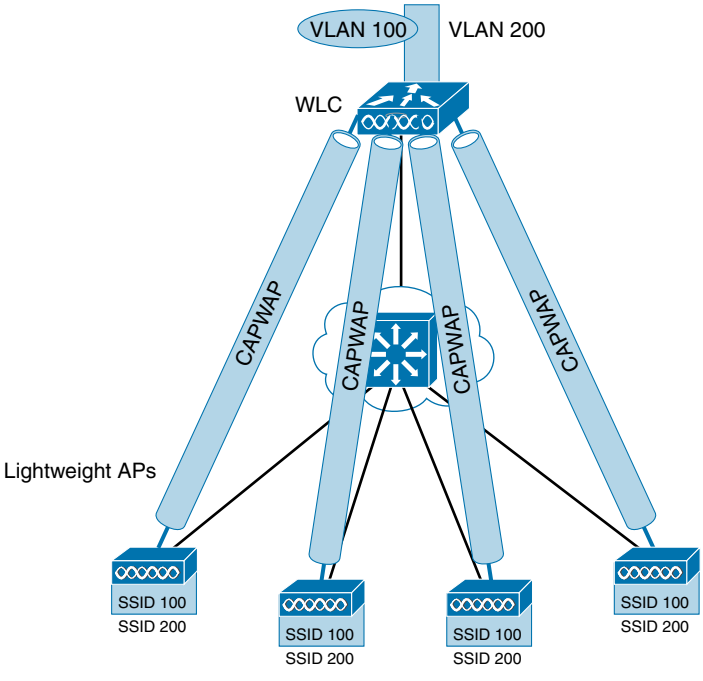


Figure 27-7 *Using CAPWAP Tunnels to Connect APs to One Central WLC*

Once CAPWAP tunnels are built from a WLC to one or more lightweight APs, the WLC can begin offering a variety of additional functions. Think of all the puzzles and shortcomings that were discussed for the traditional autonomous WLAN architecture as you read over the following list of WLC activities:

- **Dynamic channel assignment:** The WLC can automatically choose and configure the RF channel used by each AP, based on other active access points in the area.
- **Transmit power optimization:** The WLC can automatically set the transmit power of each AP based on the coverage area needed.
- **Self-healing wireless coverage:** If an AP radio dies, the coverage hole can be "healed" by turning up the transmit power of surrounding APs automatically.
- **Flexible client roaming:** Clients can roam between APs with very fast roaming times.
- **Dynamic client load balancing:** If two or more APs are positioned to cover the same geographic area, the WLC can associate clients with the least used AP. This distributes the client load across the APs.
- **RF monitoring:** The WLC manages each AP so that it scans channels to monitor the RF usage. By listening to a channel, the WLC can remotely gather information about RF interference, noise, signals from neighboring APs, and signals from rogue APs or ad hoc clients.
- **Security management:** The WLC can authenticate clients from a central service and can require wireless clients to obtain an IP address from a trusted DHCP server before allowing them to associate and access the WLAN.
- **Wireless intrusion protection system:** Leveraging its central location, the WLC can monitor client data to detect and prevent malicious activity.

Comparing Wireless LAN Controller Deployments

Suppose you want to deploy a WLC to support multiple lightweight APs in your network. Where should you put the WLC? The split-MAC concept can be applied to several different network architectures. Each architecture places the WLC in a different location within the network—a choice that also affects how many WLCs might be needed to support the number of APs required.

One approach is to locate the WLC in a central location so that you can maximize the number of APs joined to it. This is usually called a *unified* or *centralized WLC deployment*, which tends to follow the concept that most of the resources users need to reach are located in a central location such as a data center or the Internet. Traffic to and from wireless users would travel over CAPWAP tunnels that reach into the center of the network, near the core, as shown in Figure 27-8. A centralized WLC also provides a convenient place to enforce security policies that affect all wireless users.

Key Topic

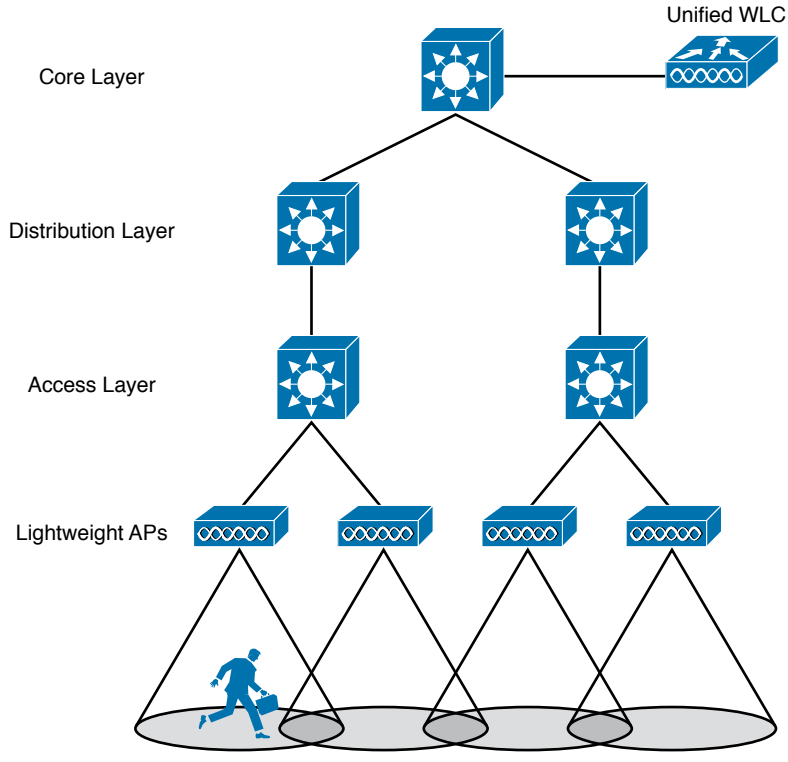


Figure 27-8 *WLC Location in a Unified Deployment*

Figure 27-8 shows four APs joined to a single WLC. Your network might have more APs—many, many more. A large enterprise network might have thousands of APs connected to its access layer. Scalability then becomes an important factor in the centralized design. Typical unified WLCs can support a maximum of 6000 APs. If you have more APs than the maximum, you will need to add more WLCs to the design, each located centrally.

A WLC can also be located in a central position in the network, inside a data center in a private cloud, as shown in Figure 27-9. This is known as a *cloud-based WLC deployment*, where the WLC exists as a virtual machine rather than a physical device. If the cloud computing platform already exists, then deploying a cloud-based WLC becomes straightforward. Such a controller can typically support up to 3000 APs. If your wireless network scales beyond that, then additional WLCs can be added as more virtual machines.

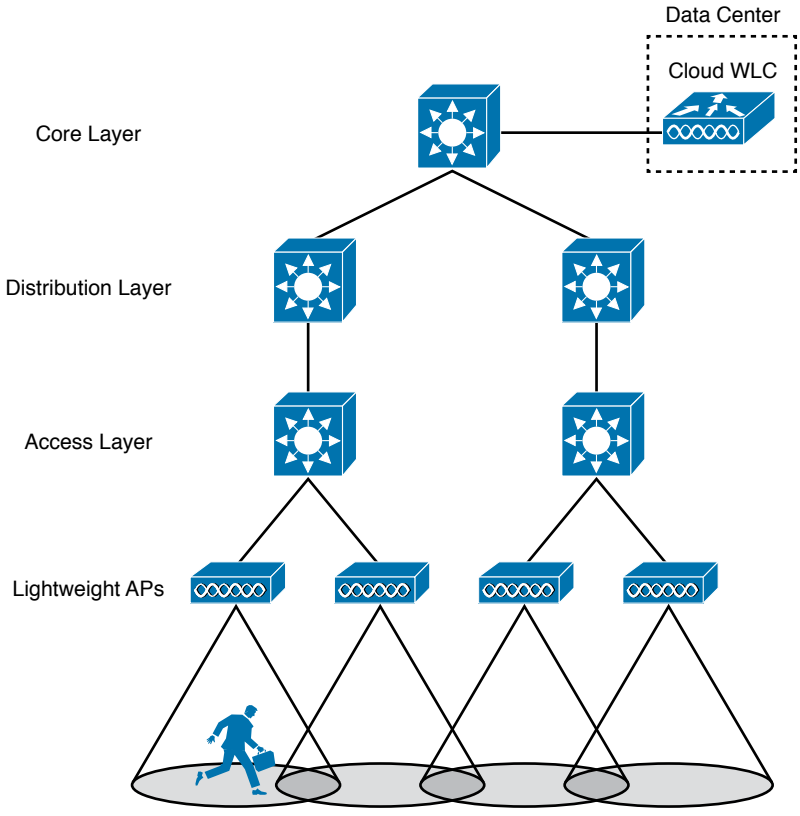


Figure 27-9 *WLC Location in a Cloud-based Deployment*

For small campuses or distributed branch locations, where the number of APs is relatively small in each, the WLC can be co-located with a stack of switches, as shown in Figure 27-10. This is known as an *embedded WLC deployment* because the controller is embedded within the switching hardware. Typical Cisco embedded WLCs can support up to 200 APs. The APs do not necessarily have to be connected to the switches that host the WLC; APs connected to other switches in other locations can join the embedded WLC too. As the number of APs grows, additional WLCs can be added by embedding them in other switch stacks at the site.

Key Topic

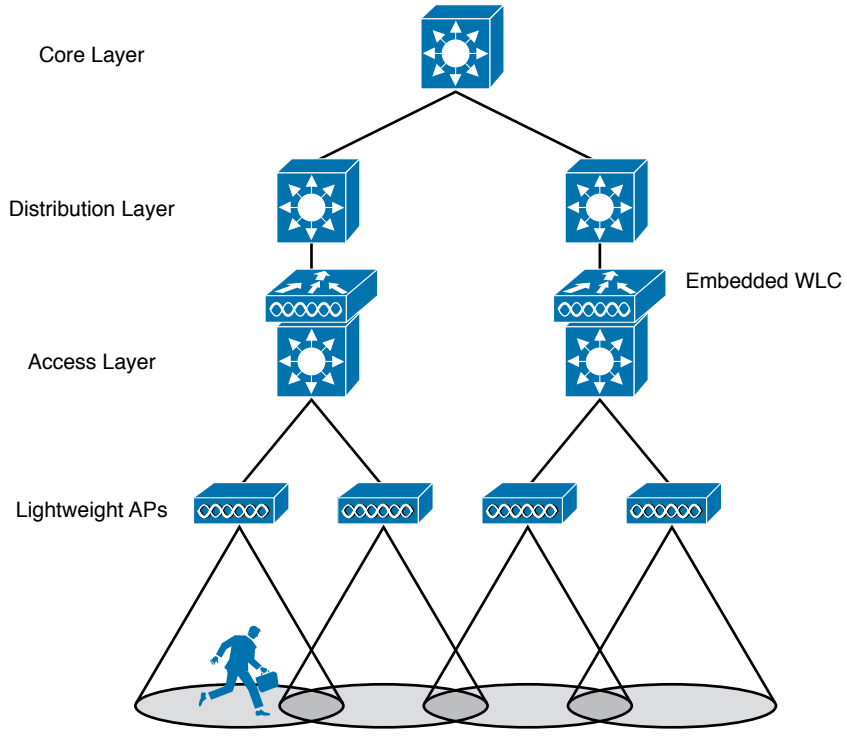


Figure 27-10 *WLC Location in an Embedded Deployment*

Finally, in small-scale environments, such as small, midsize, or multisite branch locations, you might not want to invest in dedicated WLCs at all. In this case, the WLC function can be co-located with an AP that is installed at the branch site. This is known as a Cisco *Mobility Express WLC deployment*, as shown in Figure 27-11. The AP that hosts the WLC forms a CAPWAP tunnel with the WLC, along with any other APs at the same location. A Mobility Express WLC can support up to 100 APs.

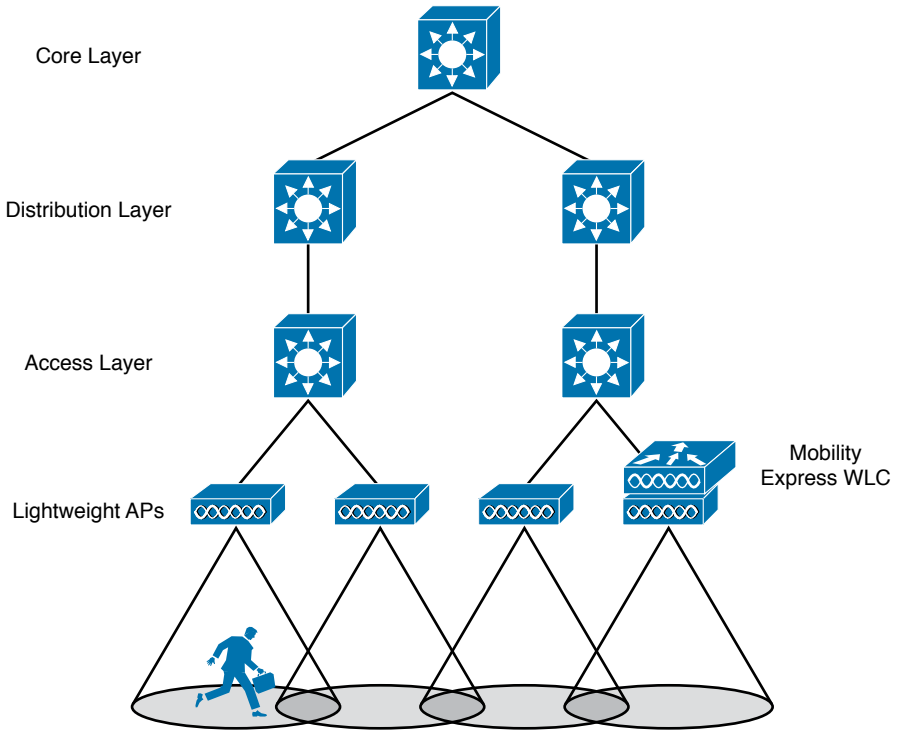


Figure 27-11 *WLC Location in a Mobility Express Deployment*

See Table 27-2 for a summary of WLC deployment models, WLC locations, and a typical maximum number of APs and clients that each one supports.

Table 27-2 Summary of WLC Deployment Models

Deployment Model	WLC Location (DC, Access, Central, AP)	APs Supported	Clients Supported	Typical Use
Unified	Central	6000	64,000	Large enterprise
Cloud	DC	3000	32,000	Private cloud
Embedded	Access	200	4000	Small campus
Mobility Express	Other	100	2000	Branch location
Autonomous	N/A	N/A	N/A	N/A