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

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

Part VII

IP Version 6

CHAPTER 25

Fundamentals of IP Version 6

This chapter covers the following exam topics:

1.0 Network Fundamentals

1.8 Configure and verify IPv6 addressing and prefix

1.9 Describe IPv6 address types

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

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

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

"Do I Know This Already?" Quiz

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

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

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

1. Which of the following was a short-term solution to the IPv4 address exhaustion problem?

a. IP version 6

b. IP version 5

- **c.** NAT/PAT
- **d.** ARP

2. A router receives an Ethernet frame that holds an IPv6 packet. The router then decides to route the packet out an Ethernet WAN link. Which statement is true about how a router forwards an IPv6 packet?

- **a.** The router discards the received frame's Ethernet data-link header and trailer.
- **b.** The router makes the forwarding decision based on the packet's source IPv6 address.
- **c.** The router keeps the incoming frame's Ethernet header, encapsulating the entire frame inside a new IPv6 packet before sending it over the outgoing Ethernet WAN link.
- **d.** The router uses the IPv4 routing table when choosing where to forward the packet.

3. Which of the following is the shortest valid abbreviation for FE80:0000:0000:0000:0100:0000:0000:0123?

- **a.** FE80::100::123
- **b.** FE8::1::123
- **c.** FE80::100:0:0:0:123:4567
- **d.** FE80::100:0:0:123

4. Which of the following is the shortest valid abbreviation for 2000:0300:0040:0005:6000:0700:0080:0009?

- **a.** 2:3:4:5:6:7:8:9
- **b.** 2000:300:40:5:6000:700:80:9
- **c.** 2000:300:4:5:6000:700:8:9
- **d.** 2000:3:4:5:6:7:8:9

5. Which of the following is the unabbreviated version of IPv6 address 2001:DB8::200:28?

- **a.** 2001:0DB8:0000:0000:0000:0000:0200:0028
- **b.** 2001:0DB8::0200:0028
- **c.** 2001:0DB8:0:0:0:0:0200:0028
- **d.** 2001:0DB8:0000:0000:0000:0000:200:0028

6. Which of the following is the correct abbreviated subnet prefix for address 2000:0000:0000:0005:6000:0700:0080:0009, assuming a mask of /64?

- **a.** 2000::5::/64
- **b.** 2000::5:0:0:0:0/64
- **c.** 2000:0:0:5::/64
- **d.** 2000:0:0:5:0:0:0:0/64

Foundation Topics

Introduction to IPv6

IP version 6 (IPv6) serves as the replacement protocol for IP version 4 (IPv4). To do so, the network can perform a slow migration that uses both, with IPv6 overcoming some of the issues that drove the need for a protocol to replace IPv4.

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

While this introductory chapter cannot get into every detail of why IPv4 needs to eventually be replaced by IPv6, the clearest and most obvious reason for migrating TCP/IP networks to use IPv6 is growth. IPv4 uses a 32-bit address, which totals to a few billion addresses. Interestingly, that seemingly large number of addresses is too small. IPv6 increases the address to 128 bits in length. For perspective, IPv6 supplies more than 10,000,000,000,000,000,000,000,000,000 times as many addresses as IPv4. IPv4 supplies just under 10 ^{10} addresses, while IPv6 supplies just under 10^{38} addresses.

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

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

The Historical Reasons for IPv6

In the last 50+ years, the Internet has gone from its infancy to being a huge influence in the world. It first grew through research at universities, from the ARPANET beginnings of the Internet in the late 1960s into the 1970s. The Internet kept growing fast in the 1980s, with the Internet's fast growth still primarily driven by research and the universities that joined in that research.

By the early 1990s, the Internet allowed commercial use, driving Internet growth even higher. Eventually, fixed Internet access from home became common, followed by the pervasive use of the Internet from mobile devices like smartphones. Now the Internet of Things (IoT) fuels Internet growth, adding all kinds of devices throughout industry that can communicate through an IP network. Figure 25-1 shows some of these major milestones with general dates.

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

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

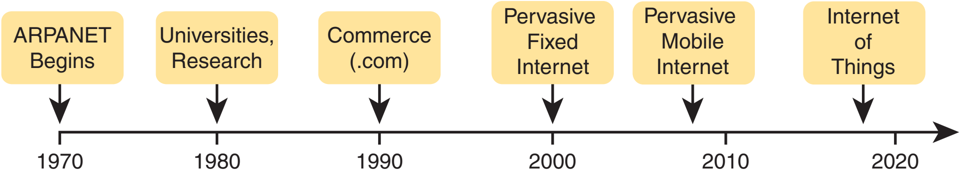


Figure 25-1 *Some Major Events in the Growth of the Internet*

The incredible growth of the Internet over a fairly long time created a big problem for public IPv4 addresses: the world was running out of addresses. As one example milestone, in 2011, the Internet Assigned Numbers Authority (IANA) allocated the final five /8 address blocks (the same size as a Class A network) to each of the five Regional Internet Registries (RIR). At that point, IANA had no more public IPv4 addresses to allocate to RIRs. The RIRs could no longer receive new allocations of public addresses from IANA but could continue to assign blocks from their remaining public address space.

In the 2010s, the five RIRs eventually assigned all of their public address space. For example, in late 2015, ARIN (North America) announced that it had exhausted its supply. All the RIRs have plans for how to deal with IPv4 address exhaustion, with all either being out of IPv4 address space or using a maintenance plan to reclaim unused IPv4 addresses for reassignment.

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

NOTE You can track ARIN's progress through this interesting transition in the history of the Internet at its IPv4 address depletion site: https://www.arin.net/resources/guide/ipv4/. You can also see a summary report at http://ipv4.potaroo.net.

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

The IETF came up with several short-term solutions to make IPv4 addresses last longer, and one long-term solution: IPv6. However, several other tools like Network Address Translation (NAT) and classless interdomain routing (CIDR) helped extend IPv4's life another couple of decades. IPv6 creates a more permanent and long-lasting solution, replacing IPv4, with a new IPv6 header and new IPv6 addresses. The address size supports a huge number of addresses, solving the address shortage problem for generations (we hope). Figure 25-2 shows some of the major IPv4 address exhaustion timing events.

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

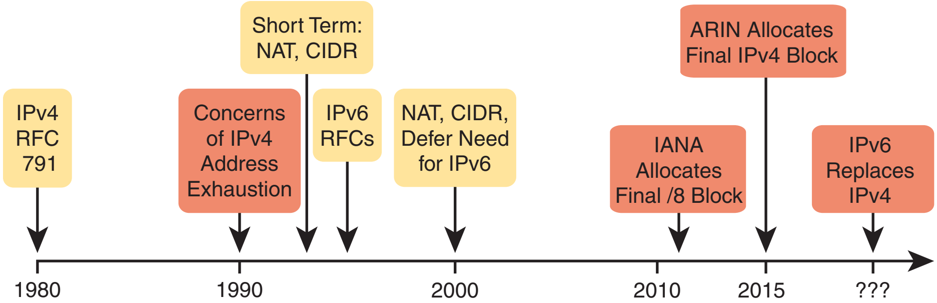

Short Term: NAT, CIDR
ARIN Allocates Final IPv4 Block
IPv4 RFC 791
Concerns of IPv4 Address Exhaustion
IPv6 RFCs
NAT, CIDR, Defer Need for IPv6
IANA Allocates Final /8 Block
IPv6 Replaces IPv4
1980
1990
2000
2010
2015
???

Figure 25-2 *Timeline for IPv4 Address Exhaustion and Short-/Long-Term Solutions*

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

The IPv6 Protocols

The primary purpose of the core IPv6 protocol mirrors the same purpose of the IPv4 protocol. That core IPv6 protocol, as defined in RFC 8200, defines a packet concept, addresses for those packets, and the role of hosts and routers. These rules allow the devices to forward packets sourced by hosts, through multiple routers, so that they arrive at the correct destination host. (IPv4 defines those same concepts for IPv4 back in RFC 791.)

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

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

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

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

NOTE A huge number of Internet RFCs define IPv6. A recent search for "IPv6" at https://www.rfc-editor.org showed over 550 such RFCs.

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

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

```
2345:1111:2222:3333:4444:5555:6666:AAAA
2000:1:2:3:4:5:6:A
FE80::1
```

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

Like IPv4, IPv6 defines a header, with places to hold both the source and destination address fields. Compared to IPv4, the IPv6 header does make some other changes besides simply making the address fields larger. However, even though the IPv6 header is larger than an IPv4 header, the IPv6 header is actually simpler (on purpose). Figure 25-3 shows the required 40-byte part of the IPv6 header.

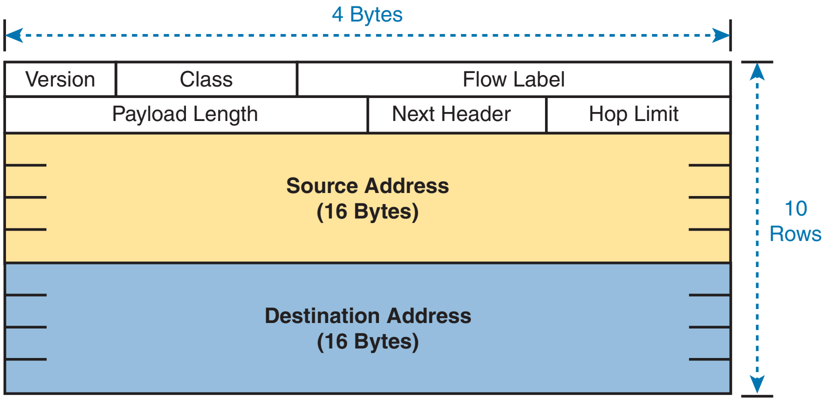


Figure 25-3 *IPv6 Header*

IPv6 Routing

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

Key Topic

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

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

The next few figures show the concepts with an example. First, Figure 25-4 shows a few settings on a host. The host (PC1) has an address of 2345::1. PC1 also knows its default gateway of 2345::2. (Both values are valid abbreviations for real IPv6 addresses.) To send an IPv6 packet to host PC2, on another IPv6 subnet, PC1 creates an IPv6 packet and sends it to R1, PC1's default gateway.

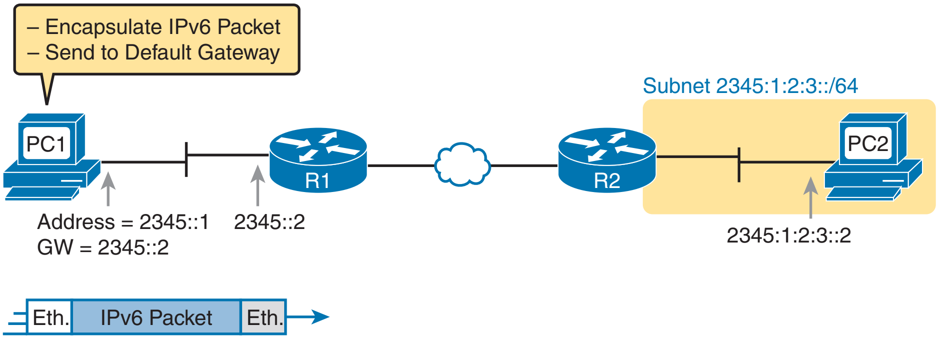


Figure 25-4 *IPv6 Host Building and Sending an IPv6 Packet*

The router (R1) has many small tasks to do when forwarding this IPv6 packet, but for now, focus on the work R1 does related to encapsulation. As seen in Step 1 of Figure 25-5, R1 receives the incoming data-link frame and extracts (de-encapsulates) the IPv6 packet from inside the frame, discarding the original data-link header and trailer. At Step 2, once R1 knows to forward the IPv6 packet to R2, R1 adds a correct outgoing data-link header and trailer to the IPv6 packet, encapsulating the IPv6 packet.

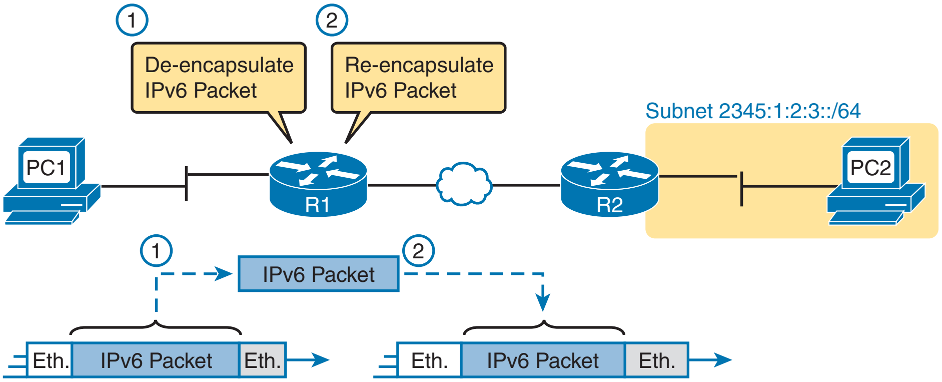


Figure 25-5 *IPv6 Router Performing Routine Encapsulation Tasks When Routing IPv6*

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

To route an IPv6 packet, a router must use its IPv6 routing table instead of the IPv4 routing table. The router must look at the packet's destination IPv6 address and compare that address to the router's current IPv6 routing table. The router uses the forwarding instructions in the matched IPv6 route to forward the IPv6 packet. Figure 25-6 shows the overall process.

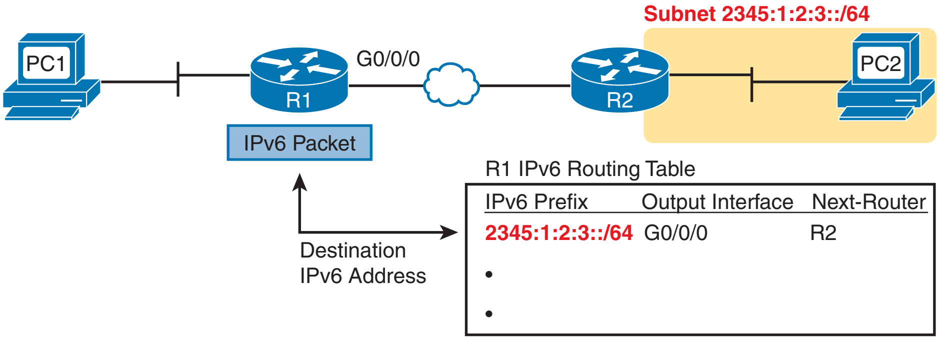


Figure 25-6 *Comparing an IPv6 Packet to R1's IPv6 Routing Table*

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

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

IPv6 Routing Protocols

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

None of the IPv4 routing protocols could be used to advertise IPv6 routes originally. They all required some kind of update to add messages, protocols, and rules to support IPv6. Over time, Routing Information Protocol (RIP), Open Shortest Path First (OSPF), Enhanced Interior Gateway Routing Protocol (EIGRP), and Border Gateway Protocol (BGP) were all updated to support IPv6. Table 25-2 lists the names of these routing protocols, with a few comments.

Table 25-2 IPv6 Routing Protocols

Routing Protocol	Defined By	Notes
RIPng (RIP next generation)	RFC	The "next generation" is a reference to a TV series, *Star Trek: The Next Generation*.
OSPFv3 (OSPF version 3)	RFC	The OSPF you have worked with for IPv4 is actually OSPF version 2, so the new version for IPv6 is OSPFv3.

Routing Protocol	Defined By	Notes
EIGRPv6 (EIGRP for IPv6)	Cisco	Cisco owns the rights to the EIGRP protocol, but Cisco also now publishes EIGRP as an informational RFC.
MP BGP-4 (Multiprotocol BGP version 4)	RFC	BGP version 4 was created to be highly extendable; IPv6 support was added to BGP version 4 through one such enhancement, MP BGP-4.

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

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

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

IPv6 Addressing Formats and Conventions

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

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

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

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

Representing Full (Unabbreviated) IPv6 Addresses

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

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

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

IPv6 addresses also have a binary format, but thankfully, most of the time you do not need to look at the binary version of the addresses. However, in those cases, converting from hex to binary is relatively easy. Just change each hex digit to the equivalent 4-bit value listed in Table 25-3.

Table 25-3 Hexadecimal/Binary Conversion Chart

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

Abbreviating and Expanding IPv6 Addresses

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

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

Abbreviating IPv6 Addresses

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

Key Topic

1. Inside each quartet of four hex digits, remove up to three leading 0s in the three positions on the left. (Note: At this step, a quartet of 0000 will leave a single 0.)
2. Replace the longest set of two or more consecutive 0000 quartets with a double colon. The abbreviation :: means "two or more quartets of all 0s." However, you can use this replacement only once inside a single address—for the longest set of consecutive 0000 quartets (or the first such set if a tie).

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

2100:**0000**:**0000**:**000**1:**0000**:**0000**:**0000**:**00**56

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

```
2100:0:0:1:0:0:0:56
```

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

```
2100:0:0:1::56
```

While 2100:0:0:1::56 is indeed the shortest abbreviation, this example happens to make it easier to see the two most common mistakes when abbreviating IPv6 addresses. First, never remove trailing 0s in a quartet (0s on the right side of the quartet). In this case, the first quartet of 2100 cannot be shortened at all because the two 0s trail. So, the following address, which begins now with only FE in the first quartet, is not a correct abbreviation of the original IPv6 address:

```
21:0:0:1::56
```

The second common mistake is to replace both sets of 0000 quartets with a double colon. For example, the following abbreviation would be incorrect for the original IPv6 address listed in this topic:

```
2100::1::56
```

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

Expanding Abbreviated IPv6 Addresses

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

Key Topic

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

The best way to get comfortable with these addresses and abbreviations is to do some yourself. Table 25-4 lists some practice problems, with the full 32-digit IPv6 address on the left and the best abbreviation on the right. The table gives you either the expanded or abbreviated address, and you need to supply the opposite value. The answers sit at the end of the chapter, in the section "Answers to Earlier Practice Problems."

Table 25-4 IPv6 Address Abbreviation and Expansion Practice

Full	Abbreviation
2340:0000:0010:0100:1000:ABCD:0101:1010	
	30A0:ABCD:EF12:3456:ABC:B0B0:9999:9009
2222:3333:4444:5555:0000:0000:6060:0707	
	3210::
210F:0000:0000:0000:CCCC:0000:0000:000D	

Full	Abbreviation
	34BA:B:B::20
FE80:0000:0000:0000:DEAD:BEFF:FEEF:CAFE	
	FE80::FACE:BAFF:FEBE:CAFE

Representing the Prefix Length of an Address

IPv6 uses a mask concept, called the **prefix length**, similar to IPv4 subnet masks. Similar to the IPv4 prefix-style mask, the IPv6 prefix length is written as a /, followed by a decimal number (there is no dotted-decimal equivalent mask for IPv6). The prefix length defines how many bits on the left side of an IPv6 address are the same value for all addresses within that subnet prefix. If you think of the ideas generically as prefix/length, all addresses in the subnet prefix begin with the same value in number of initial bits as defined by the length.

When writing an IPv6 address and prefix length in documentation, you can choose to leave a space before the /, or not, for readability. However, commands on Cisco devices typically do not allow spaces before or after the /.

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

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

Calculating the IPv6 Subnet Prefix (Subnet ID)

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

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

Finding the IPv6 Subnet Prefix

In IPv6, a subnet prefix represents a group of IPv6 addresses. For now, this section focuses on the math, and only the math, for finding the number that represents that subnet prefix. Chapter 26, "IPv6 Addressing and Subnetting," then starts putting more meaning behind the actual numbers.

Each IPv6 subnet prefix, or subnet if you prefer, has a number that represents the group. Many people just call it a subnet number or subnet ID, using the same terms as IPv4. As with IPv4, you can start with an IPv6 address and prefix length, and find the subnet prefix, with the same general rules that you use in IPv4. If the prefix length is /P, use these rules:

Key Topic

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

When using a prefix length that happens to be a multiple of 4, you do not have to think in terms of bits, but in terms of hex digits. A prefix length that is a multiple of 4 means that each hex digit is either copied or changed to hex 0. For example, a /44 prefix length implies 11 hex digits, a /48 prefix length implies 12 hex digits, and so on. Just for completeness, if the prefix length is indeed a multiple of 4, the process becomes

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

Figure 25-7 shows an example, with a prefix length of 64. In this case, Step 1 looks at the /64 prefix length and calculates that the subnet prefix has 16 hex digits. Step 2 copies the first 16 digits of the IPv6 address, while Step 3 records hex 0s for the rest of the digits.

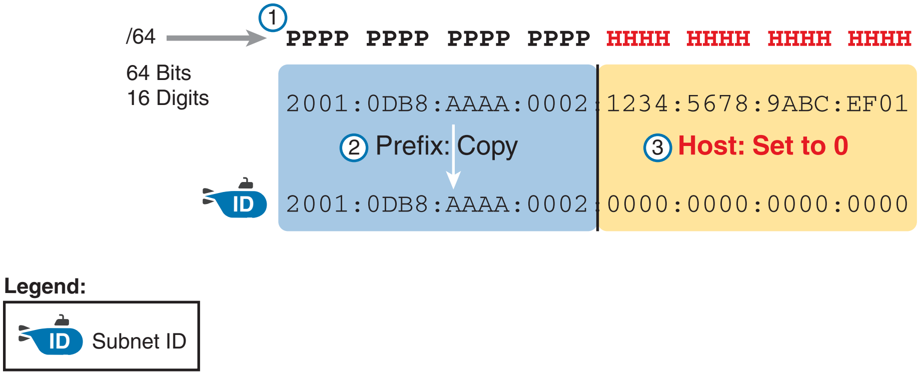

/64
1
PPPP PPPP PPPP PPPP HHHH HHHH HHHH HHHH
64 Bits
16 Digits
2001:0DB8:AAAA:0002:1234:5678:9ABC:EF01
2 Prefix: Copy
3 Host: Set to 0
ID
2001:0DB8:AAAA:0002:0000:0000:0000:0000
Legend:
ID Subnet ID

Figure 25-7 *Creating the IPv6 Subnet Prefix from an Address/Length*

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

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

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

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

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

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

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

To get better at the math, take some time to work through finding the subnet prefix for several practice problems, as listed in Table 25-5. The answers sit at the end of the chapter, in the section “Answers to Earlier Practice Problems.”

Table 25-5 Finding the IPv6 Subnet Prefix from an Address/Length Value

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

Working with More-Difficult IPv6 Prefix Lengths

In Chapter 26, you will read more about IPv6 subnetting—a much simpler topic to learn as compared with IPv4 subnetting. One reason why IPv6 subnetting happens to be simple is that several RFCs, including RFC 4291, "IPv6 Addressing Architecture," recommends that all deployed subnets use a /64 prefix length.

You will also on occasion need to work with prefix lengths shorter than /64 when working through your enterprise's subnetting plan. To get ready for that, this last topic of the chapter provides some practice with other shorter prefix lengths, using the easier cases of prefix lengths that are multiples of 4.

For example, consider the following IPv6 address and prefix length:

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

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

2000:1234:5678:9A**00:0000:0000:0000:0000**/56

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

2000:1234:5678:9A**00**::/56

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

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

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

Once again, some extra practice can help. Table 25-6 uses examples that have a prefix length that is a multiple of 4, but is not on a quartet boundary, just to get some extra practice. The answers sit at the end of the chapter, in the section "Answers to Earlier Practice Problems."

Table 25-6 Finding the IPv6 Subnet Prefix from an Address/Length Value

Address/Length	Subnet Prefix
34BA:B:B:0:5555:0:6060:707/48	
3124:1:20:DEAD:CAFE:FF:FE00:1/48	
2BCD::FACE:BEFF:FEBE:CAFE/48	
3FED:F:E0:D00:FACE:BAFF:FE00:0/48	
210F:A:B:C:CCCC:B0B0:9999:9009/40	
34BA:B:B:0:5555:0:6060:707/36	
3124::DEAD:CAFE:FF:FE00:1/60	
2BCD::FACE:1:BEFF:FEBE:CAFE/56	

Chapter Review

One key to doing well on the exams is to perform repetitive spaced review sessions. Review this chapter's material using either the tools in the book or interactive tools for the same material found on the book's companion website. Refer to the "Your Study Plan" element for more details. Table 25-7 outlines the key review elements and where you can find them. To better track your study progress, record when you completed these activities in the second column.

Table 25-7 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 table		Book, website

Review All the Key Topics

Key Topic

Table 25-8 Key Topics for Chapter 25

Key Topic Element	Description	Page Number
List	Similarities between IPv4 and IPv6	643
List	Rules for abbreviating IPv6 addresses	647
List	Rules for expanding an abbreviated IPv6 address	648
List	Process steps to find an IPv6 prefix, based on the IPv6 address and prefix length	649

Key Terms You Should Know

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

Additional Practice for This Chapter's Processes

For additional practice with IPv6 abbreviations, you may do the same set of practice problems based on Appendix H, "Practice for Chapter 25: Fundamentals of IP Version 6." 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 use these applications:

"Practice Exercise: Abbreviating and Expanding Addresses"

"Practice Exercise: Calculating the IPv6 Subnet Prefix"

"Practice Exercise: Calculating the IPv6 Subnet Prefix Round 2"

Answers to Earlier Practice Problems

This chapter includes practice problems spread around different locations in the chapter. The answers are located in Tables 25-9, 25-10, and 25-11.

Table 25-9 Answers to Questions in the Earlier Table 25-4

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

Table 25-10 Answers to Questions in the Earlier Table 25-5

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

Table 25-11 Answers to Questions in the Earlier Table 25-6

Address/Length	Subnet Prefix
34BA:B:B:0:5555:0:6060:707/48	34BA:B:B::/48
3124:1:20:DEAD:CAFE:FF:FE00:1/48	3124:1:20::/48
2BCD::FACE:BEFF:FEBE:CAFE/48	2BCD::/48
3FED:F:E0:D00:FACE:BAFF:FE00:0/48	3FED:F:E0::/48
210F:A:B:C:CCCC:B0B0:9999:9009/40	210F:A::/40
34BA:B:B:0:5555:0:6060:707/36	34BA:B::/36
3124::DEAD:CAFE:FF:FE00:1/60	3124:0:0:DEA0::/60
2BCD::FACE:1:BEFF:FEBE:CAFE/56	2BCD:0:0:FA00::/56

CHAPTER 26

IPv6 Addressing and Subnetting

This chapter covers the following exam topics:

1.0 Network Fundamentals

1.8 Configure and verify IPv6 addressing and prefix

1.9 Describe IPv6 address types

1.9.a Unicast (global, unique local, and link local)

The Internet Assigned Numbers Authority (IANA) assigns public IPv6 addresses using a process much like IPv4 with CIDR blocks. IANA first defines some IPv6 address space as unicast and other parts as multicast. The majority of the unicast address space serves as public addresses, with small parts of the address space reserved for particular purposes. IANA with the RIRs assign public address blocks as defined with a prefix length of any valid size, commonly in sizes that use /32 to /48 prefix lengths.

This chapter has two major sections. The first examines **global unicast addresses**, which serve as public IPv6 addresses. This section also discusses subnetting in IPv6, which happens to be much simpler than with IPv4. The second major section looks at **unique local addresses**, which serve as private IPv6 addresses.

"Do I Know This Already?" Quiz

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

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

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

1. Which of the following IPv6 addresses appears to be a unique local unicast address based on its first few hex digits?

a. 3123:1:3:5::1

b. FE80::1234:56FF:FE78:9ABC

c. FDAD::1

d. FF00::5

2. Which of the following IPv6 addresses appears to be a global unicast address, based on its first few hex digits?
   a. 3123:1:3:5::1
   b. FE80::1234:56FF:FE78:9ABC
   c. FDAD::1
   d. FF00::5
3. When subnetting an IPv6 address block, an engineer shows a drawing that breaks the address structure into three pieces. Comparing this concept to a three-part IPv4 address structure, which part of the IPv6 address structure is most like the IPv4 network part of the address?
   a. Subnet ID
   b. Interface ID
   c. Network ID
   d. Global routing prefix
   e. Subnet router anycast
4. When subnetting an IPv6 address block, an engineer shows a drawing that breaks the address structure into three pieces. Assuming that all subnets use the same /64 prefix length, which of the following answers lists the field's name on the far right side of the address?
   a. Subnet ID
   b. Interface ID
   c. Network ID
   d. Global routing prefix
   e. Subnet router anycast
5. For the IPv6 address FD00:1234:5678:9ABC:DEF1:2345:6789:ABCD, which part of the address is considered the global ID of the unique local address?
   a. None; this address has no global ID.
   b. 00:1234:5678:9ABC
   c. DEF1:2345:6789:ABCD
   d. 00:1234:5678
   e. FD00

Foundation Topics

Global Unicast Addressing Concepts

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

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

Public and Private IPv6 Addresses

The original plan for worldwide IPv4 addresses called for each organization connected to the Internet to be assigned a unique public IPv4 network. Each organization could then subnet that network and assign addresses from within that network so that every host in every organization used an IPv4 address unique in the universe. Unfortunately, because the IPv4 address space had too few addresses for that plan to work once every company, organization, and home wanted to connect to the Internet, those responsible for the IPv4 addressing updated their plan.

As part of that revised plan, in the 1990s, companies started using addresses from the private IPv4 address range, as defined in RFC 1918, along with Network Address Translation (NAT). Using NAT and private IPv4 addresses allowed one organization to share a few public globally unique IPv4 addresses for all host connections into the Internet.

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

The second IPv6 option uses *unique local* IPv6 addresses, which work like the IPv4 private addresses. Companies that do not plan to connect to the Internet and companies that plan to use IPv6 NAT can use unique local addresses. With IPv4, an organization simply picked numbers from the private networks in RFC 1918. With IPv6, you choose by referencing RFC 4193, which suggests a process to randomly choose a unique local prefix. And just as when using private networks with IPv4, when using IPv6 unique local addresses (ULAs), to connect to the Internet, the organization would need a small set of public IPv6 addresses and would need to use NAT.

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

Key Topic

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

Unique Local Addresses (ULAs): These addresses work somewhat like private IPv4 addresses, with the possibility that multiple organizations use the exact same addresses, and with no requirement for registering with any numbering authority.

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

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

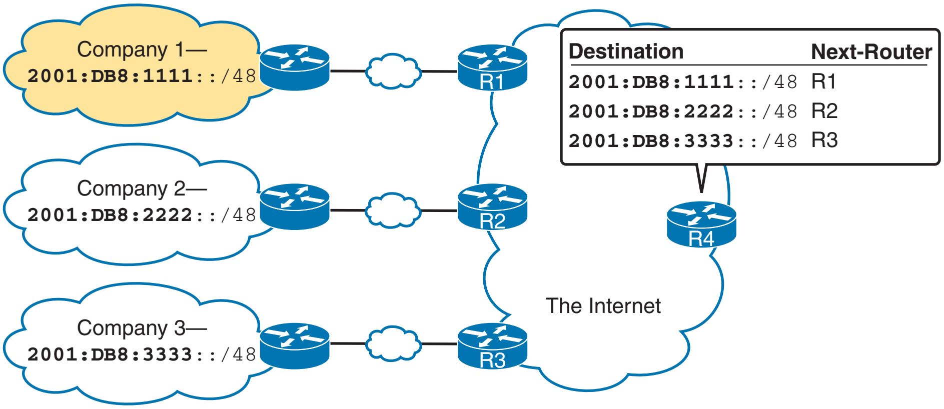
The rest of this first major section of the chapter examines global unicast addresses in more detail, while the second major section discusses unique local addresses.

The IPv6 Global Routing Prefix

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

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

The term *global routing prefix* might not make you think of a block of IPv6 addresses at first. The term refers to the idea that Internet routers can have one route that refers to all the addresses inside the address block, without a need to have routes for smaller parts of that block. For example, Figure 26-1 shows three companies with three different IPv6 global routing prefixes; the router on the right (R4) inside the Internet has one IPv6 route for each global routing prefix.



Destination	Next-Router
2001:DB8:1111::/48	R1
2001:DB8:2222::/48	R2
2001:DB8:3333::/48	R3

Figure 26-1 *Three Global Routing Prefixes, with One Route per Prefix*

The figure shows three global routing prefixes, each defining a range of GUAs reserved for use by the three companies in the figure. Consider prefix 2001:DB8:1111::/48. The prefix length, 48, happens to equal 12*4. So, the first phrase that follows gives a more literal and binary view of the meaning of "2001:DB8:1111::/48," while the second line shows the much easier hex view:

Addresses whose first 48 bits equal the first 48 bits of 2001:DB8:1111::

Addresses whose first 12 hex digits equal the first 12 hex digits of 2001:DB8:1111::

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

Both the IPv6 and IPv4 address assignment processes rely on the same organizations: IANA (along with ICANN), the Regional Internet Registries (RIR), and ISPs. For example, an imaginary company, Company1, received the assignment of a global routing prefix. The prefix means "All addresses whose first 12 hex digits are 2001:0DB8:1111," as represented by prefix 2001:0DB8:1111::/48. To receive that assignment, the process shown in Figure 26-2 happened.

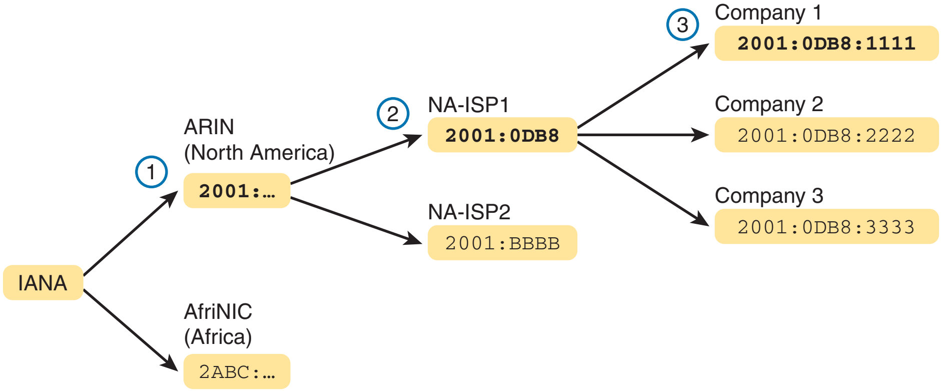


Figure 26-2 *Prefix Assignment with IANA, RIRs, and ISPs*

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

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

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

Address Ranges for Global Unicast Addresses

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

IANA allocates all IPv6 addresses that begin with hex 2 or 3 as global unicast addresses. The prefix 2000::/3 formally defines that range of numbers, meaning all addresses whose first three bits (per the /3 prefix length) match the first three bits of 2000::. To further explain:

- The first hex digit (2) is binary 0010.
- /3 means the number represents all addresses with the same first three bits as the listed prefix, or 001 binary in this case.
- The hex values whose first three bits are 001 are hex 2 and 3 but no others.
- Therefore, prefix 2000::/3 means all addresses that begin with hex 2 or 3.

IANA can expand the GUA address range beyond 2000::/3 over time if needed. RFC 4291, “IPv6 Addressing Architecture,” which lays the foundation of IPv6 addressing, reserves all addresses not otherwise reserved for the GUA address space. However, by current policy, IANA allocates global unicasts only from the 2000::/3 range.

NOTE For perspective, the 2000::/3 GUA address space, if allocated to organizations as only /48 prefixes, would provide more than 30 trillion /48 global routing prefixes. Each global routing prefix would allow for 65,536 subnets. IANA may never exhaust the 2000::/3 GUA address space.

Table 26-2 lists the address prefixes discussed in this book and their purpose.

Key Topic

Table 26-2 Some Types of IPv6 Addresses and Their First Hex Digit(s)

Address Type	First Hex Digits
Global unicast	2 or 3
Unique local	FD
Multicast	FF
Link local	FE80

IPv6 Subnetting Using Global Unicast Addresses

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

Subnetting IPv6 addresses generally works like IPv4, but with mostly simpler math (hoorah!). Many IPv6 RFCs dictate that all subnets deployed in a network (that is, by configuration on endpoints and routers) use a /64 prefix length. Using /64 as the prefix length for all subnets makes the IPv6 subnetting math just as easy as using a /24 mask for all IPv4 subnets. In addition, the dynamic IPv6 address assignment process works better with a /64 prefix length as well; so in practice, and in this book, expect IPv6 designs to use a /64 prefix length for subnets.

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

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

Deciding Where IPv6 Subnets Are Needed

First, IPv6 and IPv4 both use the same concepts about where a subnet is needed: one for each VLAN and one for each point-to-point WAN connection (serial and Ethernet). Figure 26-3 shows an example of the idea, using the small enterprise internetwork of Company 1. Company 1 has two LANs, with a point-to-point Ethernet WAN link connecting the sites. It also has an Ethernet WAN link connected to an ISP. Using the same logic you would use for IPv4, Company 1 needs four IPv6 subnets.

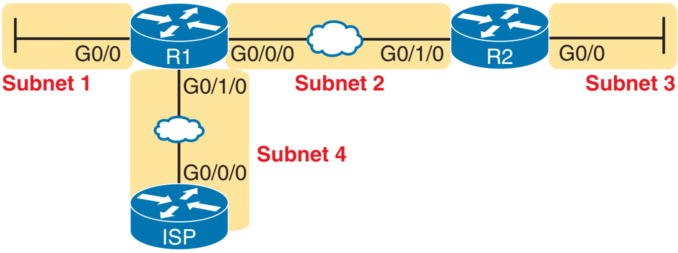


Figure 26-3 *Locations for IPv6 Subnets*

The Mechanics of Subnetting IPv6 Global Unicast Addresses

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

With IPv4, without subnetting, an address has two parts: a network part and a host part. Class A, B, and C rules define the length of the network part, with the host part making up the rest of the 32-bit IPv4 address, as shown in Figure 26-4.

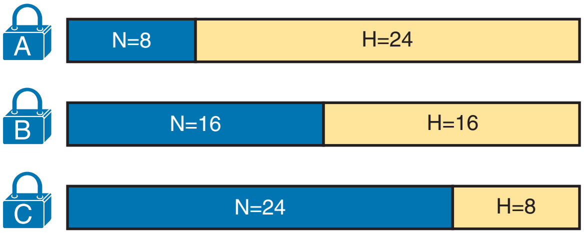


Figure 26-4 *Classful View of Unsubnetted IPv4 Networks*

To subnet an IPv4 Class A, B, or C network, the network engineer for the enterprise makes some choices. Conceptually, the engineer creates a three-part view of the addresses, adding a subnet field in the center while shortening the host field. (Many people call this "borrowing host bits.") The size of the network part stays locked per the Class A, B, and C rules, with the line between the subnet and host part being flexible, based on the choice of subnet mask. Figure 26-5 shows the field names and concepts idea for a subnetted Class B network.

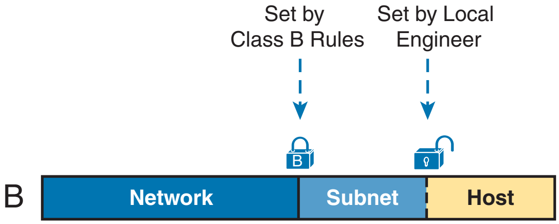


Figure 26-5 *Classful View of Subnetted IPv4 Networks with Field Names*

First, just think about the general idea with IPv6, comparing Figure 26-6 to Figure 26-5. 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 ID acts like the IPv4 subnet field. And the right side of the IPv6 address, formally called the **interface ID** (short for interface identifier), or simply *IID*, acts like the IPv4 host field.

Key Topic

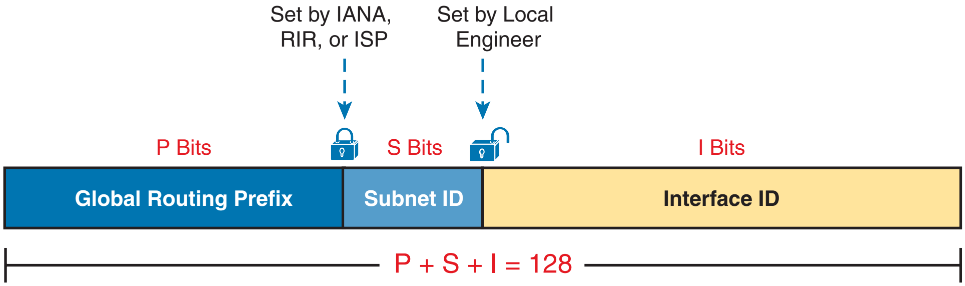


Figure 26-6 *Structure of Subnetted IPv6 Global Unicast Addresses*

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. 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 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 26-6 to the interface ID (IID) field. Several RFCs (including RFC 4291, "IPv6 Addressing Architecture") state that IIDs should be 64 bits long. A 64-bit IID works well with the IPv6 methods for dynamic IPv6 address assignment.

Finally, look to the subnet ID field in Figure 26-6. This field creates a place with which to number IPv6 subnets. The subnet ID field length depends on two facts: the global routing prefix's length and the interface ID's length. Assuming the typical 64-bit IID, the subnet ID field is typically 64–P bits, with P being the length of the global routing prefix.

As an example, consider the structure of a specific global unicast IPv6 address, 2001:0DB8:1111:0001:0000:0000:0000:0001, as seen in Figure 26-7. In this case:

Key Topic

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

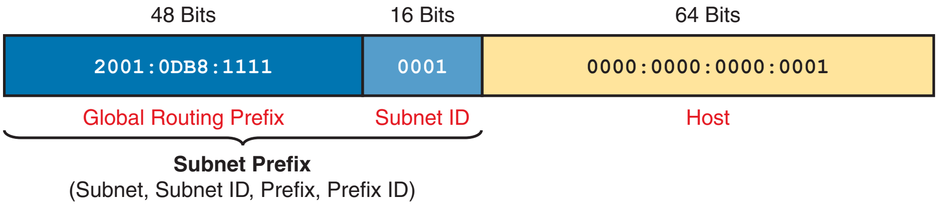


Figure 26-7 *Address Structure for Company 1 Example*

The example in Figure 26-7, along with a bit of math, shows one reason why so many companies embrace the recommendation of a 64-bit IID. With this structure, Company 1 can support 2^{16} possible subnets (65,536). Few companies need that many subnets. However, RIRs and ISPs can be generous in approving much larger blocks of public IPv6 addresses, so even if 65,536 subnets are enough, a company might apply for and receive a global routing prefix assignment with a shorter length, like /32, or /40. That would give them many more subnets (2^{32} and 2^{24}, respectively), opening up many possibilities for their subnet design and subnet numbering plans, beyond just having enough subnets.

Each subnet has far more than enough hosts per subnet with 64 bits in the interface ID. The motivation for the 64-bit IID comes from supporting various RFCs, particularly those related to dynamic IPv6 address assignment, including DHCP and SLAAC.

Listing the IPv6 Subnet Prefix (Subnet ID)

When working with IPv6 in Cisco routers, you often work with the numbers representing the subnet. For IPv6, you will use these terms:

Subnet prefix: The formal term for the number representing the subnet, including both the global routing prefix and subnet ID shown in Figure 26-7.

Subnet ID or **Prefix ID**: Informal terms used as synonyms for subnet prefix.

While the term *subnet prefix* comes from the RFCs, people tend to use the term *subnet ID*, given the history of that term in IPv4. However, note the potential miscommunication between the formal meaning of subnet ID (the middle portion of the address structure per Figure 26-7) or as a synonym for subnet prefix.

Chapter 25, “Fundamentals of IP Version 6,” already discussed finding the subnet prefix, given an IPv6 address and prefix length. The math works the same whether working with global unicast addresses or with the unique local addresses discussed later in the chapter. Just to review an example, the subnet prefix for the address in Figure 26-7 would be

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

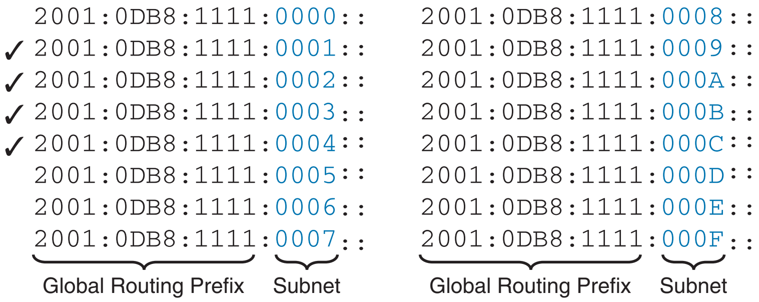
List All IPv6 Subnets

With IPv4, if you choose to use a single subnet mask for all subnets, you can find all the subnets of a Class A, B, or C network using that one subnet mask. With IPv6, the same ideas apply, and the /64 prefix length means you can find all subnets with no binary math.

To find all the subnet IDs, you simply need to find all the unique values that will fit inside the subnet ID part of the three-part IPv6 address, as shown in Figure 26-6 and Figure 26-7, basically following these rules:

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

As an example, take the IPv6 design shown in Figure 26-7, 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. The list of all possible IPv6 subnet prefixes provides all binary combinations in the subnet ID field, in this case, the fourth quartet. Also, represent the last four 0000 quartets with a :: symbol. Figure 26-8 shows the beginning of just such a list.



```
  2001:0DB8:1111:0000::    2001:0DB8:1111:0008::
✓ 2001:0DB8:1111:0001::    2001:0DB8:1111:0009::
✓ 2001:0DB8:1111:0002::    2001:0DB8:1111:000A::
✓ 2001:0DB8:1111:0003::    2001:0DB8:1111:000B::
✓ 2001:0DB8:1111:0004::    2001:0DB8:1111:000C::
  2001:0DB8:1111:0005::    2001:0DB8:1111:000D::
  2001:0DB8:1111:0006::    2001:0DB8:1111:000E::
  2001:0DB8:1111:0007::    2001:0DB8:1111:000F::
```

Figure 26-8 *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 Each 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. As with IPv4, each VLAN, each serial link, each Ethernet WAN link, and many other data-link instances need an IPv6 subnet.

Figure 26-9 shows an example using Company 1 again. The figure uses the four subnets from Figure 26-8 that have check marks beside them. The check marks are just a reminder not to use those four subnets in other locations.

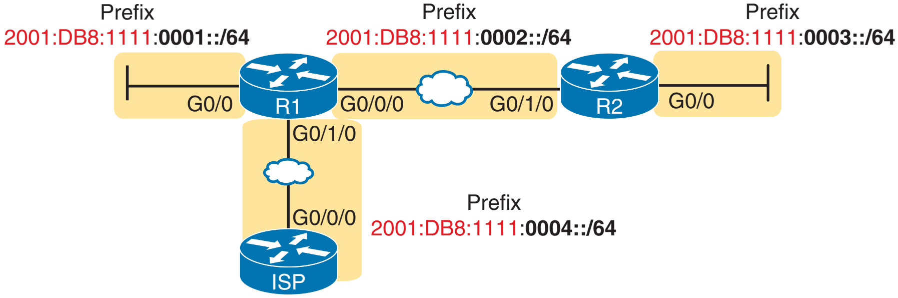


Figure 26-9 *Subnets in Company 1, with Global Routing Prefix of 2001:0DB8:1111::/48*

Assigning Addresses to Hosts in a Subnet

Now that the engineer has planned which IPv6 subnet to use 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). Chapter 28, "Implementing IPv6 Addressing on Hosts," discusses IPv6 address assignment in more depth.

For example, Figure 26-10 shows some static IP addresses chosen for the router interfaces based on the subnet choices shown in Figure 26-9. In each case, the router interfaces use an interface ID that is a relatively low number and easily remembered.

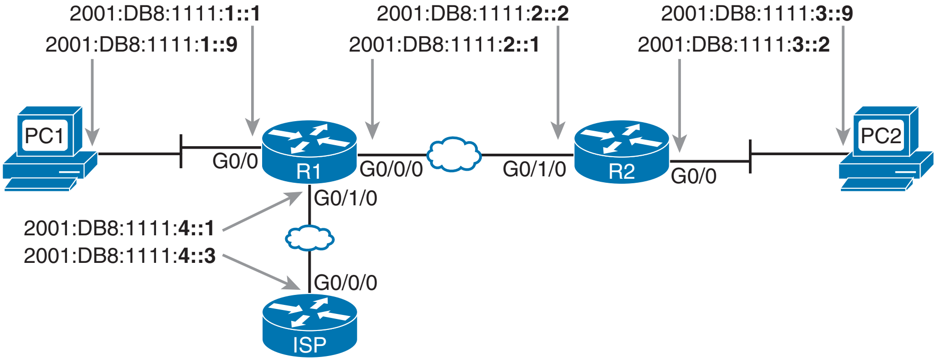


Figure 26-10 *Example Static IPv6 Addresses Based on the Subnet Design of Figure 26-9*

This chapter puts off the details of how to configure the IPv6 addresses until Chapter 27, "Implementing IPv6 Addressing on Routers."

Unique Local Unicast Addresses

Unique local addresses (ULAs) act as private IPv6 unicast addresses. These addresses have many similarities with global unicast addresses, particularly in how to subnet. The biggest difference lies in the literal number (ULAs begin with hex FD) and with the administrative process: the ULA 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 26-11 shows the format of these unique local unicast addresses.

Key Topic

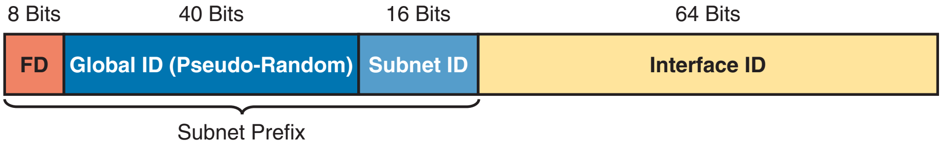


Figure 26-11 *IPv6 Unique Local Unicast Address Format*

NOTE Just to be entirely exact, IANA reserves prefix FC00::/7, and not FD00::/8, as the address range for ULAs. FC00::/7 includes all addresses that begin with hex FC and FD. However, the ULA RFC (4193) also requires setting the eighth bit to 1, which means that all ULAs begin with their first two hex digits as FD.

Subnetting with Unique Local IPv6 Addresses

Subnetting ULAs works like subnetting GUAs, assuming you begin with a 48-bit global routing prefix. The differences lie in that you must be assigned the GUA global routing prefix and prefix length. With ULAs, you create a prefix locally, with a prefix of /48.

To choose your 48-bit (12 hex digit) ULA prefix, you begin with hex FD and choose ten hex digits via a random number process. Per the ULA RFC, you should use a defined pseudo-random algorithm to determine your global ID. When practicing in lab, you can just make up a number. For example, imagine you chose a 10-hex-digit value of hex 00 0001 0001 and prepend a hex FD, making the entire prefix 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 the subnet ID field, as shown in Figure 26-11.

Figure 26-12 shows an example subnetting plan using unique local addresses. The example repeats the same topology shown earlier in Figure 26-9; 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.

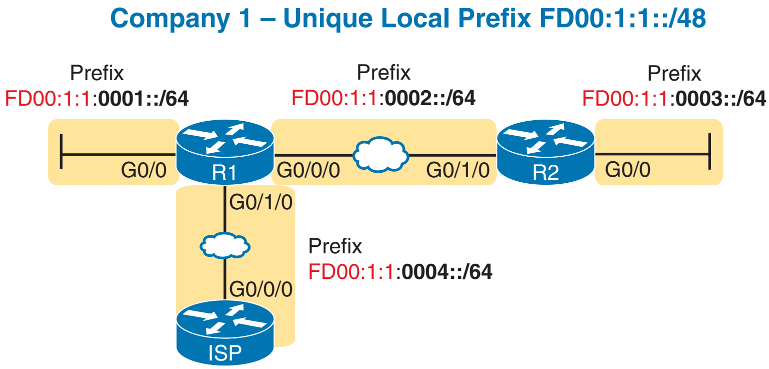


Figure 26-12 *Subnetting Using Unique Local Addresses*

The Need for Globally Unique Local Addresses

The example in Figure 26-12 shows an easy-to-remember prefix of FD00:1:1::/48. 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 ULAs 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 the 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. Instead, follow the ULA rules to make your ULAs unique in the universe—even without registering a prefix with an ISP or RIR.

RFC 4193 stresses the importance of choosing your global ID to make it statistically unlikely to be used by other companies. What is the result of unique global IDs at every company? It makes all these **ULA global IDs** unique across the globe. So, if you plan on using ULAs in a real network, use 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 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 ULAs, 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 significantly 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 26-3 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 26-3 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
Watch video		Website

Review All the Key Topics

Table 26-4 Key Topics for Chapter 26

Key Topic Element	Description	Page Number
List	Two types of IPv6 unicast addresses	656
Table 26-2	Values of the initial hex digits of IPv6 addresses, and the address type implied by each	659
Figure 26-6	Subnetting concepts for IPv6 global unicast addresses	661
List	Rules for how to find all IPv6 subnet IDs, given the global routing prefix, and prefix length used for all subnets	662
List	Rules for building unique local unicast addresses	665
Figure 26-11	Subnetting concepts for IPv6 unique local addresses	665

Key Terms You Should Know

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

CHAPTER 27

Implementing IPv6 Addressing on Routers

This chapter covers the following exam topics:

1.0 Network Fundamentals

1.8 Configure and verify IPv6 addressing and prefix

1.9 Describe IPv6 address types

1.9.a Unicast (global, unique local, and link local)

1.9.b Anycast

1.9.c Multicast

1.9.d 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, with the next chapter focusing on end-user devices. This 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 in comparison to IPv4, showing some other IPv6 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 27-1 "Do I Know This Already?" Foundation Topics Section-to-Question Mapping

Foundation Topics Section	Questions
Implementing Unicast IPv6 Addresses on Routers	1–4
Special Addresses Used by Routers	5–8

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 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 must 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. 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. On a router configured with an **ipv6 address** interface subcommand on its G0/0/0 interface, which of the following commands reveals the IPv6 prefix that the router computed based on the address/prefix-length? (Choose two answers.)
   a. **show ipv6 address**
   b. **show ipv6 route connected**
   c. **show ipv6 interface brief**
   d. **show ipv6 interface g0/0/0**
5. 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

6. 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

7. Host C must forward a packet to its default gateway, 2001:db8:1:1::1234:5678. Host C does not yet have a neighbor table entry for that address. To which solicited-node multicast address will host C send its NDP neighbor solicitation (NS) request?

a. FF02::1:ff34:5678

b. FF02::1:ff12:3456

c. FF02::1ff:1234:5678

d. FF02::1ff:34:5678

8. A router uses the **ipv6 enable** subcommand on interface G0/0/1, the **ipv6 address autoconfig** command on interface G0/0/2, and the **ipv6 address 2002:db8:1:1::1/64** command on interface G0/0/3. The commands succeed, dynamically or statically creating the expected addresses as typical for each command. Which answers are accurate about the link-local addresses (LLAs) on the various interfaces? (Choose two answers.)

a. G0/0/1 has no LLA.

b. G0/0/1 has an LLA.

c. G0/0/2 has an LLA and global unicast address with identical interface IDs.

d. G0/0/3 has a global unicast address but no LLA.

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 27-1. 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.

IPv6 might fully replace IPv4 one day; however, for a long migration period, most enterprises will run IPv4 and IPv6 on hosts and routers simultaneously, effectively like in the old days, with multiple protocol stacks. Over time, hosts will use IPv6 more and more until one day, it might be possible to disable IPv4 and run with only IPv6, as shown in Figure 27-2.

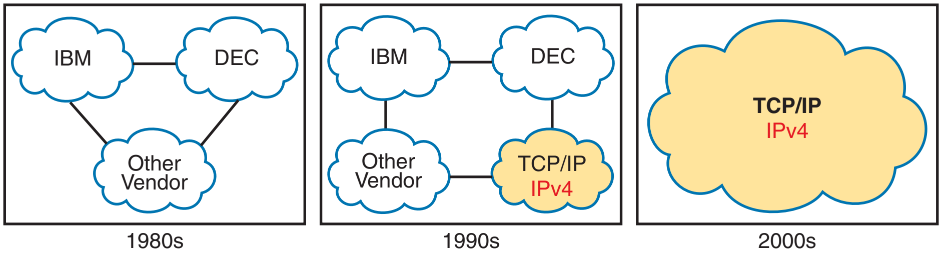


Figure 27-1 *Migration of Enterprise Networks to Use TCP/IP Stack Only, IPv4*

Key Topic

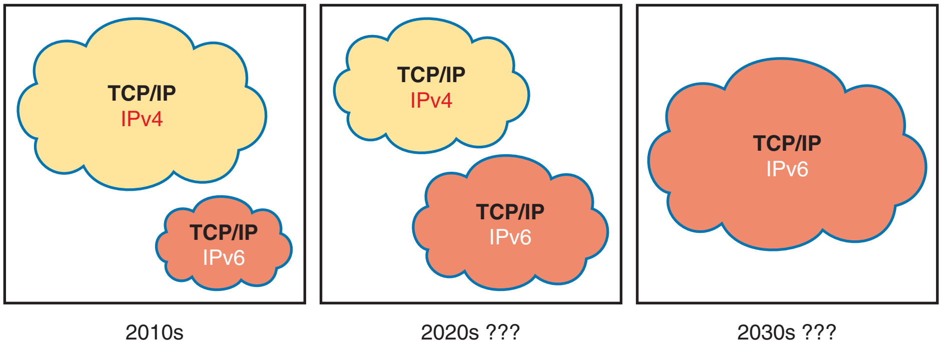


Figure 27-2 *Possible Path Through a Dual Stack (IPv4 and IPv6) over a Long Period*

Using **dual stacks**—that is, running both IPv4 and IPv6—makes great sense for today's enterprise networks. To do so, configure the routers to route IPv6 packets with IPv6 addresses on their interfaces, and advertise IPv6 routes with a routing protocol, while continuing to do the same with IPv4 addresses and routing protocols. On hosts and servers, implement IPv4 and IPv6 as well. The hosts have new methods to choose when to use IPv6 and when to use IPv4, giving preference to IPv6.

While using a dual-stack strategy works well, the configuration model for IPv6 works much as it does for IPv4. To that end, the first major section of this chapter shows how to configure and verify unicast IPv6 addresses on routers.

Static Unicast Address Configuration

Cisco routers give us two options for static configuration of IPv6 addresses. You configure the full 128-bit address and the standard /64 prefix length in one case. A second option allows you to configure a 64-bit prefix and let the router derive the second half of the address (the interface ID [IID]). The next few pages show both options.

Configuring the Full 128-Bit Address

To statically configure the full 128-bit unicast address—either global unicast address (GUA) or unique local address (ULA) —use the **ipv6 address** *address*/*prefix-length* interface subcommand. The command accepts abbreviated or expanded addresses, a slash, and then prefix length, with no spaces. Examples 27-1 and 27-2 show the configuration of the IPv6 GUAs on routers R1 and R2 from Figure 27-3, respectively.

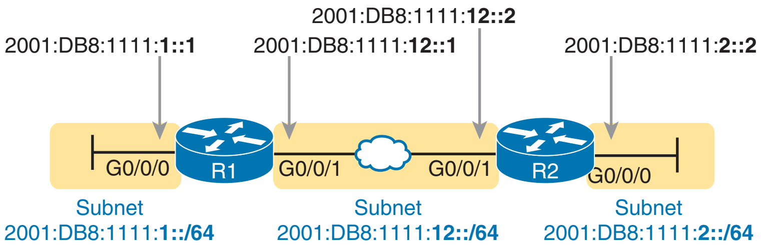


Figure 27-3 *Sample 128-bit IPv6 Addresses to Be Configured on Cisco Router Interfaces*

Example 27-1 *Configuring Static IPv6 Addresses on R1*

```
ipv6 unicast-routing
!
interface GigabitEthernet0/0/0
 ipv6 address 2001:DB8:1111:1::1/64
!
! Below, note the expanded address. IOS will abbreviate the address for you.
interface GigabitEthernet0/0/1
 ipv6 address 2001:0db8:1111:0012:0000:0000:0000:0001/64
```

Example 27-2 *Configuring Static IPv6 Addresses on R2*

```
ipv6 unicast-routing
!
interface GigabitEthernet0/0/0
 ipv6 address 2001:DB8:1111:2::2/64
!
interface GigabitEthernet0/0/1
 ipv6 address 2001:db8:1111:12::2/64
```

NOTE The configuration on R1 in Example 27-1 shows the commands as typed. One **ipv6 address** command uses an abbreviated address with uppercase hex digits, while the other uses an expanded address with lowercase hex digits. IOS accepts both commands but then changes them to their abbreviated form with uppercase hex digits.

Enabling IPv6 Routing

Interestingly, Cisco routers enable IPv4 routing by default but not IPv6 routing. The global command **ip routing** (the default setting) enables the routing of IPv4 packets. To route IPv6 packets, you must also configure the **ipv6 unicast-routing** global command.

If you forget to configure the **ipv6 unicast-routing** global command, the router will still accept your **ipv6 address** interface subcommands. In that case, the router acts as an IPv6

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

1 A **2** B **3** A, C **4** B, D **5** A **6** B **7** A **8** B, C host, so it can send IPv6 packets it generates. However, the router will not route IPv6 packets (that is, forward IPv6 packets received in an interface) until you configure the **ipv6 unicast-routing** global command.

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 27-3 lists a few samples from Router R1, with the explanations following.

Example 27-3 *Verifying Static IPv6 Addresses on Router R1*

```
! The first interface is in subnet 1
R1# show ipv6 interface GigabitEthernet 0/0/0
GigabitEthernet0/0/0 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:DB8:1111:1::1, subnet is 2001:DB8:1111:1::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF00:1
    FF02::1:FF11:1111
  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
GigabitEthernet0/0/0      [up/up]
    FE80::11FF:FE11:1111
    2001:DB8:1111:1::1
GigabitEthernet0/0/1     [up/up]
    FE80::32F7:DFF:FE29:8568
    2001:DB8:1111:12::1
GigabitEthernet0/0/2     [administratively down/down]
    unassigned
GigabitEthernet0/0/3     [administratively down/down]
    unassigned
```

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

To close the example, the **show ipv6 interface brief** command lists IPv6 addresses, not the prefix length or prefixes. Note that this command also lists all interfaces, whether configured with IPv6 addresses or not.

As with IPv4, the router adds IPv6 connected routes to the IPv6 routing table based on the IPv6 address configuration. Just as with IPv4, the router keeps these connected routes in the IPv6 routing table when the interface is working (up/up) and removes them when the interface is down. Example 27-4 shows the connected IPv6 on Router R1 from Figure 27-3.

Example 27-4 *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, R - RIP, H - NHRP, 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
       a - Application, m – OMP
C   2001:DB8:1111:1::/64 [0/0]
     via GigabitEthernet0/0/0, directly connected
C   2001:DB8:1111:12::/64 [0/0]
     via GigabitEthernet0/0/1, directly connected
```

Generating a Unique Interface ID Using Modified EUI-64

IOS supports two methods to configure a permanent, stable router IPv6 interface address. Examples 27-1 and 27-2 show the first method, in which you configure the entire 128-bit address. The second method uses the same **ipv6 address** command, but you configure only the 64-bit IPv6 prefix for the interface, letting the router automatically generate a unique IID.

To generate the second half of the address, this second method uses rules called *modified EUI-64* (extended unique identifier) or simply **EUI-64**. To use this method, add the **eui-64** keyword to the end of the command. The router then uses EUI-64 rules to create the IID part of the address, as follows:

Key Topic

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

Figure 27-4 shows a visual representation of the modified EUI-64 process.

Key Topic

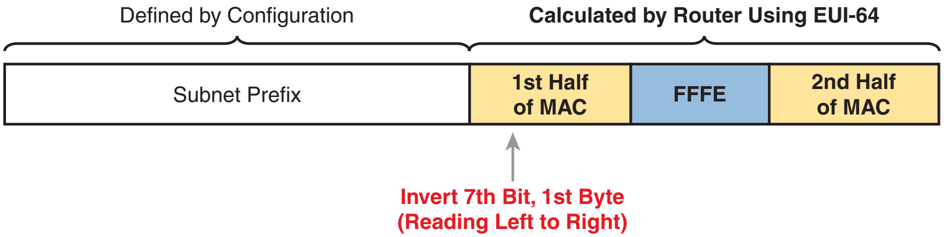

Defined by Configuration
Calculated by Router Using EUI-64
Subnet Prefix
1st Half of MAC
FFFE
2nd Half of MAC
Invert 7th Bit, 1st Byte (Reading Left to Right)

Figure 27-4 *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 IID 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 modified EUI-64 formatted IPv6 address on an interface.

Consider the two different examples in Figure 27-5, one on the left and another on the right. Both show all the work except the step to invert the 7th bit. Both start 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.

Key Topic

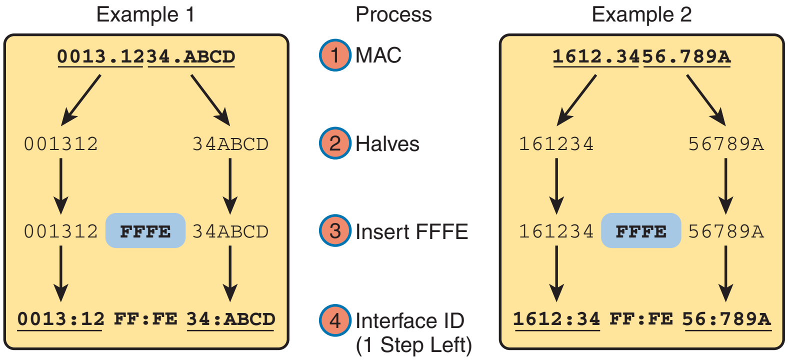


Figure 27-5 *Two Partial Examples of the EUI-64 Interface ID Process*

To complete the modified EUI-64 process, invert the 7th bit. Following the logic in Figure 27-6, convert the first byte (first two hex digits) from hex to binary. Then invert the seventh of the 8 bits: If it is a 0, make it a 1, or if it is a 1, make it a 0. Then convert the 8 bits back to two hex digits.

Key Topic

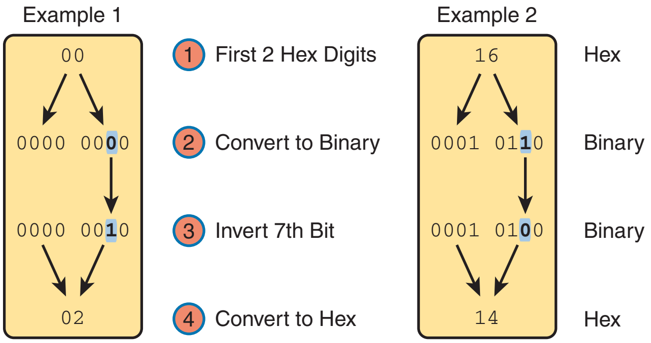


Figure 27-6 *Inverting the Seventh Bit of an EUI-64 Interface ID Field*

NOTE If you do not remember how to do hex-to-binary conversions, take a few moments to review the process. The conversion can be easy if you memorize the 16 hex values for digits 0 through F, with the corresponding binary values. 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 who prefer decimal shortcuts, you can do the bit-flip math without doing any hex-binary conversions with a little memorization. First, note that the process to invert the seventh bit, when working with a hexadecimal IPv6 address, flips the third of four bits one hex digit. With only 16 possible hex digits, you could memorize what each hex digit becomes if its third bit is inverted, and you can quickly memorize those values with Figure 27-7. To internalize the concepts, redraw the figure several times with these instructions. Just use any piece of scrap paper and use these steps:

Step 1. Write hex digits 0 through F, but arrange the digits as shown on the left of the figure, with spacing as shown. (You do not have to draw the arrow lines or the text "A Little Space"—those are instructions for you.)

Step 2. Draw lines to connect the nearby pairs of hex digits.

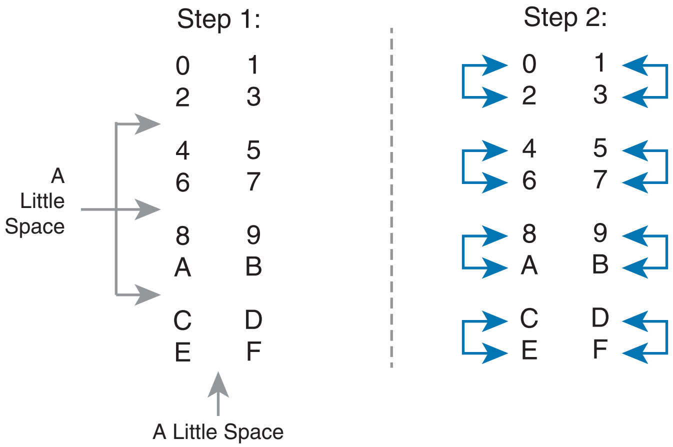


Figure 27-7 *A Mnemonic Device to Help Memorize Bit Inversion Shortcut*

When inverting the 7th bit, first remember that the 7th bit is in the 2nd hex digit. Then, refer to the right side of Figure 27-7 (or your version), and change the 2nd hex digit based on those pairs of digits. 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 arrange the hex digits in the pattern of Figure 27-7 on your notepad, you could avoid doing binary/hexadecimal conversion. Use whichever approach makes you more comfortable.

As usual, the best way to get comfortable forming these modified EUI-64 IIDs is to calculate some yourself. Table 27-2 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 modified EUI-64 rules. The answers are at the end of the chapter, in the section "Answers to Earlier Practice Problems."

Table 27-2 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	

The **ipv6 address** *prefix/prefix-length* **eui-64** interface subcommand has two key differences with the earlier examples of the **ipv6 address** command. First, the **eui-64** keyword tells the router to build the IID using modified EUI-64 rules. Second, the command should not list an address but instead a prefix, because the router will calculate the IID.

Example 27-5 shows an example that converts Router R1 from earlier Example 27-1 to use EUI-64 for the interface IIDs. It uses the same subnet prefixes as per the earlier example (based on Figure 27-3).

Example 27-5 *Configuring R1's IPv6 Interfaces Using EUI-64*

```
ipv6 unicast-routing
!
! The ipv6 address command below lists a prefix, not the full address
interface GigabitEthernet0/0/0
 mac-address 0200.1111.1111
 ipv6 address 2001:DB8:1111:1::/64 eui-64
!
interface GigabitEthernet0/0/1
 ipv6 address 2001:DB8:1111:12::/64 eui-64

R1# show ipv6 interface brief
GigabitEthernet0/0/0   [up/up]
    FE80::11FF:FE11:1111
    2001:DB8:1111:1:0:11FF:FE11:1111
GigabitEthernet0/0/1   [up/up]
    FE80::32F7:DFF:FE29:8568
    2001:DB8:1111:12:32F7:DFF:FE29:8568
GigabitEthernet0/0/2   [administratively down/down]
    unassigned
GigabitEthernet0/0/3   [administratively down/down]
    unassigned
```

For this example, interface G0/0/1 uses its universal MAC address (its burned-in address), but interface G0/0/0 uses a configured MAC address. Following that math:

`G0/0/0 – MAC 0200.1111.1111 – IID 0000.11FF.FE11.1111`

`G0/0/1 – MAC 30F7.0D29.8568 – IID 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 modified EUI-64, the **ipv6 address** command should list a prefix rather than the full 128-bit IPv6 address. However, suppose you mistakenly type the complete address and still use the **eui-64** keyword. In that case, IOS accepts the command; however, it converts the address to the matching prefix before putting it 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**.

IPv6 Address Attributes

IOS defines a set of attributes for some IPv6 addresses, displaying those with short all-caps codes in the output from commands like **show ipv6 interface** (but not listed in output from the **show ipv6 interface brief** command). The most common attributes include

EUI: The router generated the address using modified EUI-64.

TEN: Tentative. A temporary attribute that occurs while the router performs duplicate address detection (DAD).

CAL: Calendared. An address with a prescribed lifetime, including a valid and preferred lifetime. They are commonly listed when using DHCP and SLAAC.

PRE: Preferred. The preferred address to use on the interface that uses time-based (calendared) addresses. They are commonly listed when using DHCP and SLAAC.

Example 27-6 shows an example using R1's interface G0/0/0 in the previous example. The example shows two consecutive **show ipv6 interface g0/0/0** commands, the first issued immediately after the interface configuration in Example 27-5. At that time, the router had not completed DAD (which takes only a few seconds to complete), so the output lists the TEN attribute along with EUI. The subsequent repeat of the command no longer lists TEN, as DAD completed, but still lists EUI, confirming the router derived the address using EUI-64. (The section "Discovering Duplicate Addresses Using NDP NS and NA" in Chapter 28, "Implementing IPv6 Addressing on Hosts," explains DAD.)

Example 27-6 *An Example of IPv6 Address Attributes in* **show interfaces**

```
R1# show interfaces g0/0/0
GigabitEthernet0/0/0 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:DB8:1111:1:0:11FF:FE11:1111, subnet is 2001:DB8:1111:1::/64 [EUI/TEN]
! Lines omitted for brevity
! Next command gathered seconds later...
R1# show interfaces g0/0/0
GigabitEthernet0/0/0 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:DB8:1111:1:0:11FF:FE11:1111, subnet is 2001:DB8:1111:1::/64 [EUI]
! Lines omitted for brevity
```

Dynamic Unicast Address Configuration

In most cases, network engineers will statically configure the IPv6 addresses of router interfaces so that the addresses do not change until the engineer changes the router configuration. However, routers can dynamically learn IPv6 addresses. These can be useful for routers connecting to the Internet through some types of Internet access technologies, like cable modems or fiber Internet.

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 dynamic method to use to learn its IPv6 address. Example 27-7 shows the configuration, with one interface using stateful DHCP and one using SLAAC.

Example 27-7 *Router Configuration to Learn IPv6 Addresses with DHCP and SLAAC*

```
! This interface uses DHCP to learn its IPv6 address
interface GigabitEthernet0/0/0
 ipv6 address dhcp
!
! This interface uses SLAAC to learn its IPv6 address
interface GigabitEthernet0/0/1
 ipv6 address autoconfig
```

Endpoints often use one of the dynamic methods to learn their IPv6 address and related settings. Chapter 28, "Implementing IPv6 Addresses on Hosts," discusses the specifics of DHCP and SLAAC.

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 (LLA)** as a special kind of unicast IPv6 address. Every interface that supports IPv6 must have an LLA. A small number of important IPv6 protocols then use LLAs to send and receive packets to help with overhead IPv6 functions, like neighbor and address discovery—sometimes even before the device has a GUA or ULA to use. This next topic first looks at how IPv6 uses LLAs and then how routers create them.

Link-Local Address Concepts

LLAs provide every device that supports IPv6 the capability to send and receive unicast packets on the local link, before dynamically learning or being configured with their routable IPv6 GUA or ULA. While hosts could send and receive application data using LLAs over a single subnet, LLAs do not exist to support normal application data. Instead, LLAs exist to support some overhead IPv6 protocols—protocols that need some working address to use.

The IPv6 designers wanted better options than IPv4 to support protocols that need to send packets before hosts have their normal unicast address assignment. For instance, before an IPv4 host leases an address with DHCP, it needs to send DHCP messages inside IP packets. To do that, IPv4 defines special addresses 0.0.0.0 and 255.255.255.255. The IPv6 designers looked for a better way, and LLAs provide a solution in some cases.

Hosts self-generate an LLA for themselves and can then use their LLA to send and receive IPv6 packets on the local link. All interfaces that use IPv6 must have an LLA, with the option for the device to dynamically create it without outside help. Once created, the host ensures no other hosts use the same LLA by using duplicate address detection (DAD). If unique, the host keeps using the LLA, but if DAD finds another host using the same LLA, the new host picks a different LLA. Some IPv6 protocols rely on using a host's LLA as the source of IPv6 protocol messages, including Neighbor Discovery Protocol (NDP, which replaces IPv4 ARP) and Dynamic Host Configuration Protocol (DHCP).

LLAs also appear in a surprising place: as the next-hop IPv6 addresses in IPv6 routes, as shown in Figure 27-8. For routes learned by a routing protocol, the **show ipv6 route** command lists the LLA of the neighboring router, rather than the GUA or ULA. For the default router setting on a host, the host refers to the router's LLA.

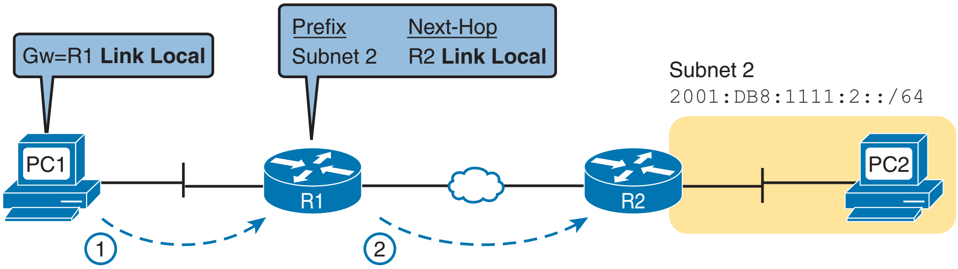


Figure 27-8 *IPv6 Using Link-Local Addresses as the Next-Hop Address*

The following list summarizes some key facts about link-local addresses:

Key Topic

Unicast (not multicast): LLAs 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 self-generated: Every IPv6 host interface (and router interface) can automatically create its own link-local address, 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

By default, routers self-assign an LLA to each IPv6-enabled interface. To do so, the router sets the first half of the address to FE80:0000:0000:0000, as shown on the left side of Figure 27-9. Then the router creates the IID using modified EUI-64 rules (see the earlier section "Generating a Unique Interface ID Using Modified EUI-64" for a review of the process). As a result, a router's complete LLA should be unique because the MAC address that feeds into the EUI-64 process should be unique.

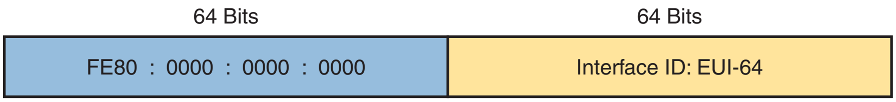


Figure 27-9 *Link-Local Address Format*

In practice, LLAs begin with FE80, but formally, RFC 4291 reserves prefix FE80::/10, which includes addresses that begin FE8, FE9, FEA, or FEB. However, that same RFC also dictates that the first 64 bits should be FE80:0000:0000:0000. So in practice, LLAs begin with FE80:0000:0000:0000.

When a router automatically generates an interface's LLA, it does not create a matching configuration command. To see the LLA, instead of the **show running-config** command, just use the usual commands that also list the unicast IPv6 address: **show ipv6 interface** and **show ipv6 interface brief**. Example 27-8 shows an example from Router R1 just after it was configured as shown in Example 27-5 (with the **eui-64** keyword on the **ipv6 address** commands).

Example 27-8 *Comparing Link-Local Addresses with EUI-Generated Unicast Addresses*

```
R1# show ipv6 interface brief
GigabitEthernet0/0/0   [up/up]
    FE80::11FF:FE11:1111
    2001:DB8:1111:1:0:11FF:FE11:1111
GigabitEthernet0/0/1   [up/up]
    FE80::32F7:DFF:FE29:8568
    2001:DB8:1111:12:32F7:DFF:FE29:8568
GigabitEthernet0/0/2   [administratively down/down]
    unassigned
GigabitEthernet0/0/3   [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 (G0/0/0 and G0/0/1), the output lists the GUAs, which happen to begin with 2001 in this case. At the same time, the output also lists the LLAs for each interface, beginning with FE80.

Next, focus on the two addresses listed under interface G0/0/0. If you look closely, you will see that both addresses have the same IID value. Because the GUA configuration used the **ipv6 address 2001:DB8:1111:1::/64 eui-64** command, the router used modified EUI-64 logic to form the IID portion of both the GUA and LLA. For both addresses, the router used the interface MAC address of 0200.1111.1111 and calculated the IID portion of both addresses as 0000:11FF:FE11:1111 (unabbreviated). After abbreviation, Router R1's LLA on interface G0/0/0 becomes FE80::11FF:FE11:1111.

NOTE While Cisco routers use the modified EUI-64 method to create the IID for LLAs, most end-user device OSs do not. Instead, to reduce security exposures, they randomly assign IPv6 address IIDs. Chapter 28, which focuses on host IPv6 address assignment, provides additional details and examples.

Cisco IOS also allows the direct configuration of the interface's LLA, with that configured LLA taking precedence, as follows:

- The **ipv6 address** *address* **link-local** interface subcommand configures the LLA; in this case, the router does not create an LLA using EUI-64 rules.
- If not configured, the IOS calculates the LLA using EUI-64 rules.

Routing IPv6 with Only Link-Local Addresses on an Interface

This chapter has shown four variations on the **ipv6 address** command to configure routable addresses (GUAs and ULAs) so far. To review:

ipv6 address *address/prefix-length*: Static configuration of the entire routable address and prefix length

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

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

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

To complete the list, now consider the **ipv6 enable** interface subcommand. It enables IPv6 processing on the interface and causes the router to self-assign an LLA, but it leaves the interface with no routable (GUA or ULA) address.

The purpose of the **ipv6 enable** command will not make sense until you realize that some links, particularly WAN links, do not need a GUA or ULA. Using the backdrop of Figure 27-10, 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. The hosts do not need to know the routers' WAN interface addresses. (That same logic applies with IPv4 as well.)

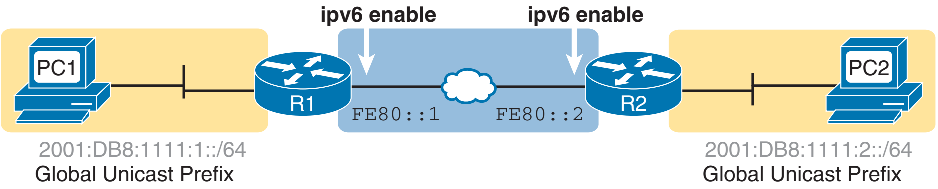


Figure 27-10 *Typical Use of the* **ipv6 enable** *Command*

Additionally, the routers do not need GUAs or ULAs on the WAN links for routing to work; they only need LLAs. For instance, IPv6 routing protocols use LLAs as the next-hop address when dynamically building IPv6 routes. Additionally, as discussed in Chapter 29, "Implementing IPv6 Routing," static routes can use LLAs for the next-hop address.

In short, IPv6 routing works on WAN links that use only LLAs, with no GUAs or ULAs. As a result, you would not even need to assign an IPv6 subnet prefix 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 used for 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. IANA and various RFCs use the term *well-known multicast address* to refer to these reserved predefined multicast addresses. IANA reserves address range FF00::/12 (all IPv6 addresses that begin FF0) for these addresses.

This next section focuses on the well-known permanent IPv6 multicast addresses. The first, **link-local multicast addresses**, have a scope that limits them to flow over a single link. The other type is a special overhead multicast address calculated for each host, called the **solicited-node multicast address**.

Well-Known Multicast Addresses

Stop for a moment and think about some of the control plane protocols discussed throughout this book so far. Some IPv4 control plane protocols use IPv4 broadcasts, meaning that the packet destination address was either 255.255.255.255 (the address for all hosts in the local subnet) or the subnet broadcast address (the address for all hosts in that specific subnet). The sender encapsulates the packet in an Ethernet broadcast frame, 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 needs to think about the message. LAN switches forward LAN broadcasts out every port in the same VLAN, so copies of each LAN broadcast arrive at each host in the subnet. As a result, each host has to process the frame, then the packet, read the type of message, and so on, before choosing to ignore the packet. For example, an IPv4 ARP Request—an IPv4 and LAN broadcast—requires a host to process the Ethernet and ARP details of the message before deciding whether to reply or not.

IPv6, instead of using Layer 3 and Layer 2 broadcasts, uses Layer 3 multicast addresses, which in turn causes Ethernet frames to use Ethernet multicast addresses. LAN switches can use a long-established multicast optimization feature that lets them forward multicast frames to only the hosts that care to receive a copy while not forwarding frames to disinterested hosts. As a result, the IPv6 protocols work more efficiently with fewer unnecessary interruptions to hosts.

For instance, OSPFv3 uses IPv6 multicast addresses FF02::5 and FF02::6. In a subnet, the OSPFv3 routers, when they initialize OSPF on an interface, send messages to register their interest in receiving packets sent to FF02::5 and FF02::6. However, hosts do not register an interest in receiving packets sent to those addresses. The LAN switches watch for those multicast registration messages and recall which ports connect to hosts interested in packets sent to each multicast address. When receiving frames sent to those multicast addresses, the switches forward them only to the OSPFv3 routers that preregistered to receive them, but no other hosts/routers.

All the IPv6 multicast addresses can benefit from the multicast optimizations. For reference, Table 27-3 lists the most common well-known IPv6 multicast addresses.

Table 27-3 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 27-9 repeats the output of the **show ipv6 interface** command to show the multicast addresses used by Router R1 on its G0/0/0 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). The highlighted section beginning with the heading "Joined group address(es)" lists the multicast addresses joined by the router on this interface.

Example 27-9 *Verifying Static IPv6 Addresses on Router R1*

```
R1# show ipv6 interface GigabitEthernet 0/0/0
GigabitEthernet0/0/0 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:DB8:1111:1::1, subnet is 2001:DB8:1111:1::/64
  Joined group address(es):
    FF02::1
    FF02::2
```

```
   FF02::5
   FF02::6
   FF02::1:FF00:1
   FF02::1:FF11:1111
! Lines omitted for brevity
```

Multicast Address Scopes

IPv6 RFC 4291 defines IPv6 unicast and multicast addressing details, 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, an LLA—a unicast IPv6 address—has 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 27-4.

Key Topic

Table 27-4 IPv6 Multicast Scope Terms

Scope Name	Fourth Hex Digit	Scope Defined by…	Meaning
Interface-Local	1	Derived by Device	The packet remains within the device. Useful for internally sending packets to services running on that same host.
Link-Local	2	Derived by Device	The host that creates the packet can send it onto the link, but no routers forward it.
Site-Local	5	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	8	Configuration on Routers	Intended to be broad, probably for an entire company or organization. Must be broader than Site-Local.
Global	E	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 27-4 shows that progression top to bottom, while Figure 27-11 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.

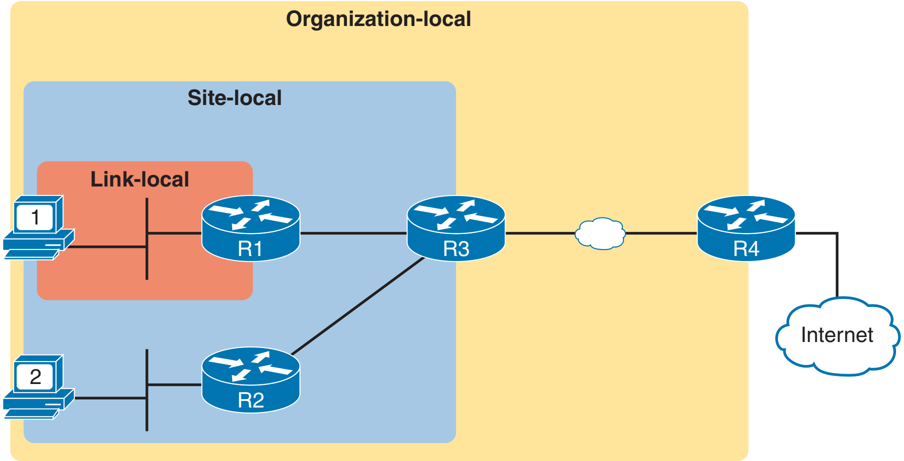


Figure 27-11 *IPv6 Multicast Scopes*

27

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

Key Topic

Link-local address: A unicast IPv6 address that begins FE80 with a link-local scope. Devices create their own LLAs. A more complete term for comparison would be *link-local unicast address*.

Link-local multicast address: An IPv6 address that begins with FF02. These addresses serve as reserved multicast addresses 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 28. NDP improves the MAC-discovery process compared to IPv4 ARP by sending IPv6 multicast packets to communicate with a second host before the first host knows the second host's MAC address. The process uses the solicited-node multicast address associated with the unicast IPv6 address.

Figure 27-12 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 27-12. 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

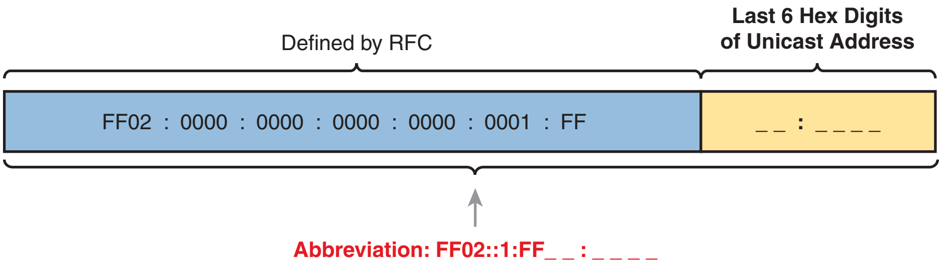

Defined by RFC
Last 6 Hex Digits of Unicast Address
FF02 : 0000 : 0000 : 0000 : 0000 : 0001 : FF
_ _ : _ _ _ _
Abbreviation: FF02::1:FF_ _ : _ _ _ _

Figure 27-12 *Solicited-Node Multicast Address Format*

Note that a host or router calculates and uses a matching solicited-node multicast address for every unicast address (GUA, ULA, and LLA) on an interface. The device then registers to receive incoming multicast messages sent to that address. Example 27-10 shows an example in which the router interface has a GUA of 2001:DB8:1111:1::1/64 and an LLA of FE80::11FF:FE11:1111. (You saw these addresses earlier in Examples 27-1 and 27-3.) As a result, the interface has two solicited-node multicast addresses, shown at the end of the output.

Example 27-10 *Verifying Static IPv6 Addresses on Router R1*

```
R1# show ipv6 interface GigabitEthernet 0/0/0
GigabitEthernet0/0/0 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:DB8:1111:1::1, subnet is 2001:DB8:1111:1::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::5
    FF02::1:FF00:1
    FF02::1:FF11:1111
 ! Lines omitted for brevity
```

Note that in this case, R1's expanded GUA ends with its last six hex digits as 00:0001, resulting in an unabbreviated solicited-node multicast address of FF02:0000:0000:0000:0000:0001:FF00:0001. This value begins with the 26-hex-digit prefix shown in Figure 27-12, followed by 00:0001. The solicited-node multicast address corresponding to link-local address FE80::11FF:FE11:1111 ends in 11:1111, as shown in the last line of the example.

The Unspecified and Loopback Addresses

All IPv6 hosts can use two additional special addresses:

Key Topic

- The unspecified IPv6 address, ::, that is, all 0s
- The loopback IPv6 address, ::1, that is, 127 binary 0s with a single 1

Seldom used, a host can use the unspecified address when it has not yet learned a unicast address to use. The address can be used only as a source address, and only used before the host has an LLA or other unicast address to use.

The IPv6 loopback address gives each IPv6 host a way to test its protocol stack. Just like the IPv4 127.0.0.1 loopback address, packets sent to ::1 do not leave the host but are 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 of the instances of the service can be used. Figure 27-13 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 of the routers that support the address.

Key Topic

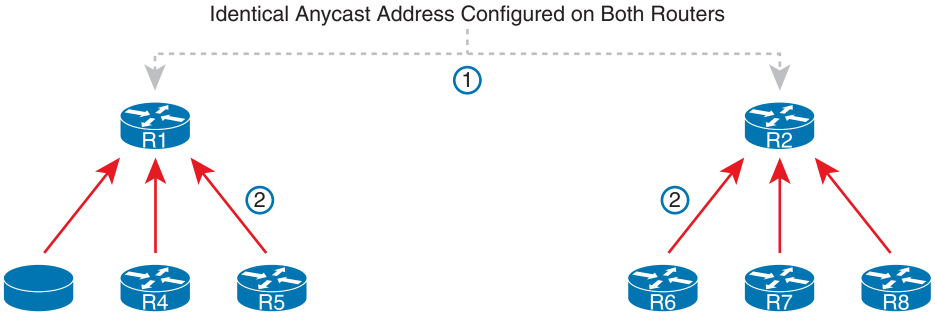


Figure 27-13 *IPv6 Anycast Addresses*

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 27-11 shows a sample configuration on a router. Note that the actual address (2001:1:1:2::99) looks like any other unicast address. 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 27-11 *Configuring and Verifying IPv6 Anycast Addresses*

```
R1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# interface gigabitEthernet 0/0/0
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 g0/0/0
GigabitEthernet0/0/0 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 g0/0/0
GigabitEthernet0/0/0 [up/up]
     FE80::11FF:FE11:1111
     2001:1:1:1::1
     2001:1:1:2::99
```

NOTE The term **subnet-router anycast address** represents a different concept than the anycast address just discussed. Instead, the *subnet-router anycast address* is one special anycast address in each subnet. It is reserved for routers 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; 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 27-5 summarizes the various commands and the automatically generated IPv6 addresses in one place for review and study.

Table 27-5 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** *address/prefix-length* **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 27-6 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 27-6 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 27-7 Key Topics for Chapter 27

Key Topic Element	Description	Page Number
Figure 27-2	Conceptual drawing about the need for dual stacks for the foreseeable future	671
List	Rules for creating an IPv6 address using EUI-64 rules	675
Figure 27-4	IPv6 EUI-64 Address Format and Rules	675
Figure 27-5	Conceptual drawing of how to create an IPv6 address using EUI-64 rules	675
Figure 27-6	Example of performing the bit inversion when using EUI-64	676
List	Functions IOS enables when an IPv6 address is configured on a working interface	680
List	Key facts about IPv6 link-local addresses	681
Table 27-4	Link-local scope terms and meanings	686
List	Comparisons of the use of the term *link-local*	687
Figure 27-12	Conceptual drawing of how to make a solicited-node multicast address	688
List	Other special IPv6 addresses	689
Figure 27-13	Concept of IPv6 anycast addresses	689
Table 27-5	IPv6 address summary with the commands that enable each address type	691

Key Terms You Should Know

all-nodes multicast address, all-routers multicast address, anycast address, dual stack, EUI-64, interface-local scope, IPv6 address scope, link-local address (LLA), link-local multicast address, link-local scope, organization-local scope, site-local scope, solicited-node 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 addresses using EUI-64 rules and finding the solicited-node multicast address based on a unicast address, use the exercises in Appendix I, "Practice for Chapter 27: Implementing IPv6 Addressing on Routers." You have two options to use:

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

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 27-8 and 27-9 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 27-8 Chapter 27 Configuration Command Reference

Command	Description
ipv6 unicast-routing	Global command that enables IPv6 routing on the router
ipv6 address *ipv6-address*/*prefix-length*	Interface subcommand that manually configures the entire interface IP address and prefix length
ipv6 address *ipv6-prefix*/*prefix-length* **eui-64**	Interface subcommand that manually configures an IPv6 prefix and prefix length, 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
ipv6 address autoconfig	Interface subcommand that tells the router to use SLAAC to find/build its interface IPv6 address

Table 27-9 Chapter 27 EXEC Command Reference

Command	Description
show ipv6 route [**connected**] [**local**]	Lists all IPv6 routes, or the connected routes only, or the local routes only
show ipv6 interface [*type number*]	Lists IPv6 settings, including link-local and other unicast IP addresses, on all interfaces (or for the listed interface)
show ipv6 interface brief [*type number*]	Lists IPv6 interface status and unicast IPv6 addresses for all interfaces (or for only the listed interface if included)

Answers to Earlier Practice Problems

Table 27-2, earlier in this chapter, listed several practice problems in which you needed to calculate the IPv6 address based on EUI-64 rules. Table 27-10 lists the answers to those problems.

Table 27-10 Answers to IPv6 EUI-64 Address Creation Practice

Prefix	MAC Address	Unabbreviated IPv6 Address
2001:DB8:1:1::/64	0013.ABAB.1001	2001:0DB8:0001:0001:0213:ABFF:FEAB:1001
2001:DB8:1:1::/64	AA13.ABAB.1001	2001:0DB8:0001:0001:A813:ABFF:FEAB:1001
2001:DB8:1:1::/64	000C.BEEF.CAFE	2001:0DB8:0001:0001:020C:BEFF:FEEF:CAFE
2001:DB8:1:1::/64	B80C.BEEF.CAFE	2001:0DB8:0001:0001:BA0C:BEFF:FEEF:CAFE
2001:DB8:FE:FE::/64	0C0C.ABAC.CABA	2001:0DB8:00FE:00FE:0E0C:ABFF:FEAC:CABA
2001:DB8:FE:FE::/64	0A0C.ABAC.CABA	2001:0DB8:00FE:00FE:080C:ABFF:FEAC:CABA

This page intentionally left blank

CHAPTER 28

Implementing IPv6 Addressing on Hosts

This chapter covers the following exam topics:

1.0 Network Fundamentals

1.8 Configure and verify IPv6 addressing and prefix

1.9 Describe IPv6 address types

1.9.a Unicast (global, unique local, and link local)

1.9.d Modified EUI 64

IPv6 hosts act like IPv4 hosts in many ways, using similar ideas, similar protocols, and even similar or identical commands for the same purpose. At the same time, IPv6 sometimes takes a different approach than IPv4, using a different solution with a new protocol or command. For example:

- Like IPv4, IPv6 hosts use a unicast address, prefix length (mask), default router, and DNS server.
- Like IPv4, IPv6 uses a protocol to dynamically learn the MAC addresses of other hosts in the same LAN-based subnet.
- Unlike IPv4, IPv6 hosts use the Neighbor Discovery Protocol (NDP) for many functions, including those done by IPv4's ARP.
- Like IPv4, IPv6 hosts can use DHCP to learn their four primary IPv6 settings.
- Unlike IPv4, IPv6 supports a dynamic address assignment process other than DHCP, called Stateless Address Autoconfiguration (SLAAC).

This chapter focuses on the four primary IPv6 settings on hosts: the address, prefix length, default router address, and DNS server address. However, to understand how hosts dynamically learn those addresses, this chapter begins its first major section devoted to NDP, which plays a crucial role in several IPv6 processes. The middle section of the chapter then focuses on how hosts dynamically learn their IPv6 settings with DHCP and SLAAC. The final major section of this chapter looks at the tools to verify a host's IPv6 settings, many of which use the same commands used for IPv4.

"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 28-1 "Do I Know This Already?" Foundation Topics Section-to-Question Mapping

Foundation Topics Section	Questions
The Neighbor Discovery Protocol	1–4
Dynamic Configuration of Host IPv6 Settings	5–7
Troubleshooting Host IPv6 Addressing	8

1. 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 message will PC1 use to begin the process of discovering PC2's MAC address?
   a. ARP Request
   b. NDP NS
   c. NDP RS
   d. SLAAC NS
2. 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. Router hostname
   c. IPv6 prefix(es) on the link
   d. IPv6 address of DHCP server
3. Three routers (R1, R2, and R3) connect to the same VLAN and IPv6 subnet. All three routers have responded to various NDP RS messages with NDP RA messages. Which of the answers best describes the kind of NDP information held in the output of the **show ipv6 neighbors** command on R1?
   a. IPv6 neighbors (both routers and hosts) plus their MAC addresses, without noting which are routers
   b. IPv6 neighbors (both routers and hosts) plus their MAC addresses, also noting which are routers
   c. IPv6 routers, with no information about nonrouters, with no MAC address info
   d. IPv6 routers, with no information about nonrouters, but with MAC address info
4. PC1 and Router R1 connect to the same VLAN and IPv6 subnet. The user of PC1 pings the IPv6 address of a host that sits at a remote site so that the packets flow through R1, PC1's default router. PC1 learned all its IPv6 settings dynamically. Which of the following answers lists a protocol or message that PC1 could have used to learn what IPv6 address to use as its default router?
   a. EUI-64
   b. NDP NS
   c. DAD
   d. NDP RS

5. Host PC1 dynamically learns its IPv6 settings using Stateless Address Autoconfiguration (SLAAC). Which one of PC1's settings is most likely to be learned from the stateless DHCPv6 server?
   - a. Host address
   - b. Prefix length
   - c. Default router address
   - d. DNS server address(es)
6. Host PC1 dynamically learns its IPv6 settings using Stateless Address Autoconfiguration (SLAAC). Think about the host's unicast address as two parts: the subnet prefix and the interface ID. Which answers list a way that SLAAC learns or builds the value of the interface ID portion of the host's address? (Choose two answers.)
   - a. Learned from a DHCPv6 server
   - b. Built by the host using EUI-64 rules
   - c. Learned from a router using NDP RS/RA messages
   - d. Built by the host using a random value
7. An IPv6 host is configured to use DHCP to lease its IPv6 address. The DHCPv6 server is not on the same link but is located at another site. Which answer describes a mechanism the client and routers use to make the DHCPv6 messages flow between the client and server?
   - a. The client sends the DHCPv6 Solicit message to multicast address FF02:1:2.
   - b. The client sends the DHCPv6 Solicit message to broadcast address FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF.
   - c. The client must learn the DHCPv6 server's unicast address from a local router using NDP messages.
   - d. The routers use IPv6 multicast routing to forward the Solicit message, unchanged, to the DHCPv6 server.
8. All routers in the network have global unicast addresses (GUAs) configured on their interfaces. The user of PC1, on a LAN, issues a **traceroute** command for a distant host's address. The command succeeds, listing four lines with IPv6 addresses. Which answer best describes the addresses in the **traceroute** output and its use of link-local addresses (LLAs) and GUAs?
   - a. All lines list LLAs with no GUAs.
   - b. The first line lists an address that matches PC1's default route.
   - c. All lines list GUAs with no LLAs.
   - d. The last line lists the GUA of the final router in the route.

Foundation Topics

The Neighbor Discovery Protocol

IPv4 and IPv6 define a wide range of control and management functions as part of the Internet Control Message Protocol (ICMP). To support similar features in IPv6, the Internet community created ICMPv6, which defines protocols appropriate for IPv6. (For easier comparison, ICMP for IPv4 is often called ICMPv4.)

The **Neighbor Discovery Protocol (NDP)**, a part of ICMPv6 defined in RFC 4861, provides several vital functions in every IPv6 network. Notably, NDP defines the IPv6 equivalent of the IPv4 ARP function. Some of its 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.

Prefix Discovery: Hosts learn the IPv6 subnet prefix and prefix length that the router(s) expect to exist on the link.

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

The next few pages explain the listed features.

Discovering Neighbor Link Addresses with NDP NS and NA

NDP replaces IPv4 ARP using the *Neighbor Solicitation (NS)* and *Neighbor Advertisement (NA)* messages. 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 following list summarizes the functions:

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.

Neighbor Advertisement (NA): This message lists the sender's IPv6 and MAC addresses. It can be sent in reply to an NS message; 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.

28

NOTE With NDP, the word *neighbor* refers to hosts on the same data link—for example, the same VLAN.

Figure 28-1 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.

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 target.

Key Topic

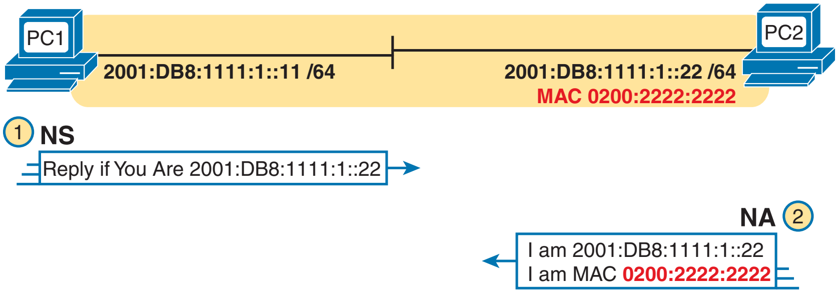


Figure 28-1 *Example NDP NS/NA Process to Find the Neighbor's Link Addresses*

As a brief aside, be aware that NDP NS messages use a destination IPv6 address of the target's solicited-node multicast address, in this case PC2's solicited-node multicast address FF02::1:FF00:22. PC1 would then encapsulate the IPv6 packet in a frame destined to a multicast Ethernet address. If the network engineers at this company also enabled the multicast optimization feature MLD Snooping, the switches would forward the multicast NS message only to hosts that had earlier registered to receive packets sent to that specific multicast address. The other hosts on the link will never see the NS message. If the LAN switches do not implement MLD, then the switches still flood the frame so that it reaches the intended destination.

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

Example 28-1 shows an example of the **IPv6 neighbor table** on Router R1 based on Figure 28-2. In this case, R1 has learned the MAC addresses for Routers R2 and R3, associated with their respective LLAs. R1 has also learned PC A's LLA and GUA and the matching MAC address. (To connect the output to the figure, pay close attention to the interface column on the far right of the output.)

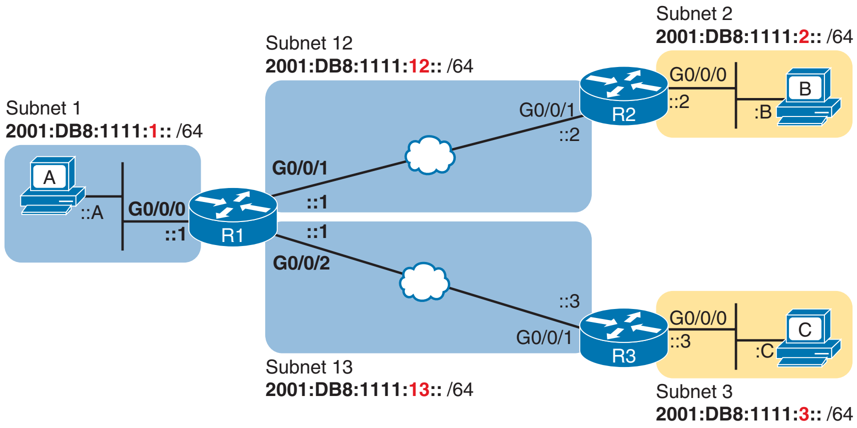


Figure 28-2 *Sample Network with IPv6 Addresses*

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

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

Example 28-1 *IPv6 Neighbor Table on Router R1*

```
R1# show ipv6 neighbors
IPv6 Address                              Age Link-layer Addr State Interface
2001:DB8:1111:1::a                          0 0200.aaaa.aaaa  REACH Gi0/0/0
2001:DB8:1111:1:9EA:93AC:F7CE:D39F          0 3c8c.f8eb.710d  REACH Gi0/0/0
2001:DB8:1111:1:706A:5FDF:AA40:E576        16 3c8c.f8eb.710d  STALE Gi0/0/0
2001:DB8:1111:1:70D0:AE1F:4786:907D         0 80fa.5b04.de8b  STALE Gi0/0/0
2001:DB8:1111:1:7C6B:7B02:DB5C:873F        16 3c8c.f8eb.710d  STALE Gi0/0/0
2001:DB8:1111:1:90A1:C742:1B11:6F10         0 00e0.4c68.1f26  STALE Gi0/0/0
2001:DB8:1111:1:BD2C:9AA4:83E2:6D8F        16 3c8c.f8eb.710d  STALE Gi0/0/0
FE80::AAAA:AAAA                             0 0200.aaaa.aaaa  REACH Gi0/0/0
FE80::184C:56F9:FD3B:D6E7                   0 00e0.4c68.1f26  REACH Gi0/0/0
FE80::552D:E285:4347:BDED                   0 80fa.5b04.de8b  DELAY Gi0/0/0
FE80::706A:5FDF:AA40:E576                   0 3c8c.f8eb.710d  REACH Gi0/0/0
FE80::FF:FE01:2                             0 2436.dadf.9281  REACH Gi0/0/1
FE80::72EA:1AFF:FE9A:D301                   0 70ea.1a9a.d301  REACH Gi0/0/2
```

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

Example 28-2 shows an excerpt from a Windows host neighbor table of a host in the same IPv6 subnet as PC A and Router R1 in Figure 28-2. The beginning output shows network shell command **netsh interface ipv6 show neighbors**, as issued in layers (which makes it much easier to issue additional **netsh** commands later). The highlighted entry lists R1's G0/0/0 link-local address (LLA) and MAC address. Also, the highlighted text at the far right of the line identifies this entry as representing a router. The output also lists several solicited-node multicast addresses.

Example 28-2 *Example Windows Neighbor Table with* **netsh interface ipv6 show neighbors** *Command*

```
C:\Users\Wendell> netsh
netsh> interface ipv6
netsh interface ipv6> show neighbors
Interface 7: En0

Internet Address                              Physical Address   Type
--------------------------------------------  -----------------  -----------
fe80::11ff:fe11:1111                          02-00-11-11-11-11  Reachable (Router)
ff02::1                                       33-33-00-00-00-01  Permanent
ff02::2                                       33-33-00-00-00-02  Permanent
ff02::1:ff11:1111                             33-33-ff-11-11-11  Permanent
! Lines omitted for brevity
! Next line shows a Powershell command

PS C:\Users\Wendell> get-NetNeighbor -AddressFamily IPv6
ifIndex IPAddress                             LinkLayerAddress      State       PolicyStore
------- ---------                             ----------------      -----       -----------
49      ff02::1:ff11:1111                     33-33-FF-11-11-11     Permanent   ActiveStore
49      fe80::11ff:fe11:1111                  02-00-11-11-11-11     Reachable   ActiveStore
49      ff02::2                               33-33-00-00-00-02     Permanent   ActiveStore
49      ff02::1                               33-33-00-00-00-01     Permanent   ActiveStore
! Lines omitted for brevity
```

The example ends with the PowerShell command equivalent to the **netsh interface ipv6 show neighbor** command: **get-NetNeighbor -AddressFamily IPv6**. The latter command lists the same info. Be aware that Microsoft favors using PowerShell commands over older commands like those from netshell.

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 IPv4 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 process using NDP. With IPv6, IPv6 hosts use NDP to dynamically discover all IPv6 routers on the link, learning what routers it can use as a default router. NDP defines two messages that allow any host to discover all routers in the subnet:

Key Topic

Router Solicitation (RS): Hosts send this message to the "all-IPv6-routers" local-scope multicast address of FF02::2 to ask all on-link routers to identify themselves.

Router Advertisement (RA): Routers send this message in response to an RS message, listing many facts, including the link-local IPv6 address of the router. The message flows to the unicast address of the host that sent the RS. Routers also send unsolicited RA messages, not in response to an RS, but periodically, announcing the same details to all hosts on the link. Routers send unsolicited RA messages to the all-IPv6-hosts local-scope multicast address of FF02::1.

For example, Figure 28-3 shows how host PC A can learn R1's LLA. The process is simple, with PC A first asking and R1 replying.

IPv6 does not use broadcasts, but it does use multicasts to improve efficiency compared to IPv4. 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 CPU cycles processing the RS message, and IPv6 hosts will ignore the message.

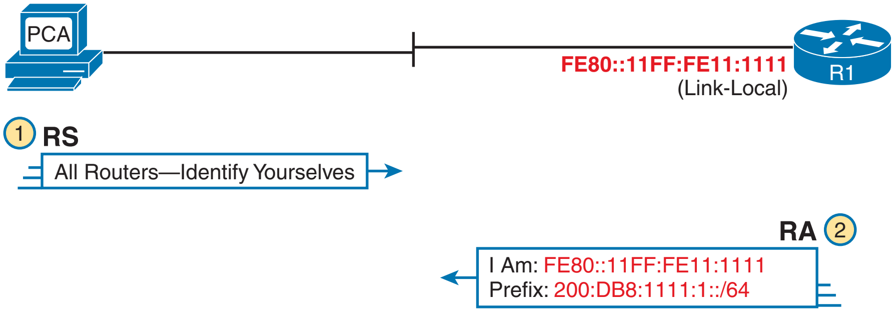

PCA
FE80::11FF:FE11:1111
(Link-Local)
R1
1 RS
All Routers—Identify Yourselves
RA 2
I Am: FE80::11FF:FE11:1111
Prefix: 200:DB8:1111:1::/64

Figure 28-3 *Example NDP RS/RA Process to Find the Default Routers*

Routers list any neighboring routers they learn about in NDP RA messages with the **show ipv6 routers** command. As an example, consider the topology in earlier Figure 28-2. No routers exist on the LAN connected to R1's G0/0/0 (on the left of the figure), but R1 should learn of both R2 and R3 on its WAN links. Example 28-3 shows the output, highlighting the LLA of the router that sent the NDP RA message, the local R1 interface on which it was received, and the **on-link prefix** advertised by the neighboring router.

Example 28-3 *Listing All Routers with the* **show ipv6 routers** *Command*

```
R1# show ipv6 routers
Router FE80::FF:FE01:2 on GigabitEthernet0/0/1, last update 2 min
  Hops 64, Lifetime 1800 sec, AddrFlag=0, OtherFlag=0, MTU=1500
  HomeAgentFlag=0, Preference=Medium
  Reachable time 0 (unspecified), Retransmit time 0 (unspecified)
  Prefix 2001:DB8:1111:12::/64 onlink autoconfig
    Valid lifetime 2592000, preferred lifetime 604800
Router FE80::72EA:1AFF:FE9A:D301 on GigabitEthernet0/0/2, last update 0 min
  Hops 64, Lifetime 1800 sec, AddrFlag=0, OtherFlag=0, MTU=1500
  HomeAgentFlag=0, Preference=Medium
  Reachable time 0 (unspecified), Retransmit time 0 (unspecified)
  Prefix 2001:DB8:1111:13::/64 onlink autoconfig
    Valid lifetime 2592000, preferred lifetime 604800
```

NOTE To view the routers learned by a host, use these commands: (Windows) **netsh interface ipv6 show neighbors**; (Linux) **ip -6 neighbor**; (macOS) **ndp -rn**.

Discovering Prefixes with NDP RS and RA

Beyond identifying routers on a link, the NDP RA message also lists the IPv6 prefix and prefix length used on the link. As a result, hosts dynamically learn the subnet(s) that should exist on-link. For example, Figure 28-3 shows an RS/RA exchange example. The RA in the lower right of the figure not only shows the router identifying itself using its link-local address (LLA), but the message also lists one on-link prefix of 2001:db8:1111:1::/64.

IPv6 hosts use a routing table much like IPv4 hosts but build their routes differently. IPv4 hosts build a route for each on-link subnet, with that subnet calculated from the host's IPv4 address and mask. IPv6 hosts build a route for each on-link subnet, but hosts do no calculations. Instead, they rely on the prefixes advertised in the RA messages sent by routers. Routers can advertise multiple on-link prefixes, listing the subnet prefix and prefix length, with the hosts then considering destinations in those subnets as on-link. Also, hosts build a default route, using the information listed in the NDP RA message. To summarize, IPv6 hosts create two important routes based on the RA message, as follows:

Key Topic

- Create a route for each on-link prefix/length learned from a router in an NDP RA message. These on-link routes allow the host to forward packets directly to on-link destinations without using a router.
- Create a default route with a next-hop router address of the router LLA identified in the NDP RA message. The default route allows the host to forward packets destined off-link to the router that resides on-link.

Example 28-4 shows excerpts from two hosts on the same subnet as PC A in Figure 28-3—with the initial output from macOS and the latter output from a Windows PC. Both commands display the host's routing table entries for the default route and one on-link prefix. The example highlights each command's default route, the next-hop address ("gateway"), and on-link prefix 2001:db8:1111:1::/64.

Example 28-4 *Example macOS Host Routing Table with* **netstat -rn** *Command*

```
Mac% netstat -rnf inet6
                              fe80::11ff:fe11:1111%en8        UGcg          en8
                              link#20                         UC            en8
fe80::11ff:fe11:1111%en8      2:0:11:11:11:11                 UHLWIir       en8
! Lines omitted for brevity
```

```
Windows PC> netsh
netsh> interface ipv6
netsh interface ipv6> show route

Publish  Type      Met  Prefix                    Idx  Gateway/Interface Name
-------  --------  ---  ------------------------  ---  ------------------------
No       Manual    256  ::/0                      49   fe80::11ff:fe11:1111
No       Manual    256  2001:db8:1111:1::/64      49   Ethernet 6
! Lines omitted for brevity
```

You see some differences when looking closely at the output from macOS versus Windows. However, both hosts use information from RA messages from R1. Notably, macOS lists the default route with the word *default*, while Windows uses the numeric equivalent of ::/0, a number meant to represent all IPv6 addresses. Both list the router's LLA rather than its GUA as the next-hop address.

Discovering Duplicate Addresses Using NDP NS and NA

IPv6 hosts use the Duplicate Address Detection (DAD) process before they begin using a unicast address to ensure that no other node on that link is already using it. Hosts perform DAD when first using the address, and any time a host interface initializes. Hosts also use DAD, whether using static address configuration or any of the dynamic address configuration options. When performing DAD, if another host already uses that address, the first host simply does not use the address until the problem is resolved.

DAD refers to the function, but the function uses NDP NS and NA messages. 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 28-4 shows an example of DAD. PC1 initializes and does a DAD check, but PC2 already uses the same 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 for the address PC2 currently uses, so PC2 sends back an NA.
4. PC1, on receiving the NA message for its IPv6 address, realizes a duplicate address exists.

Key Topic

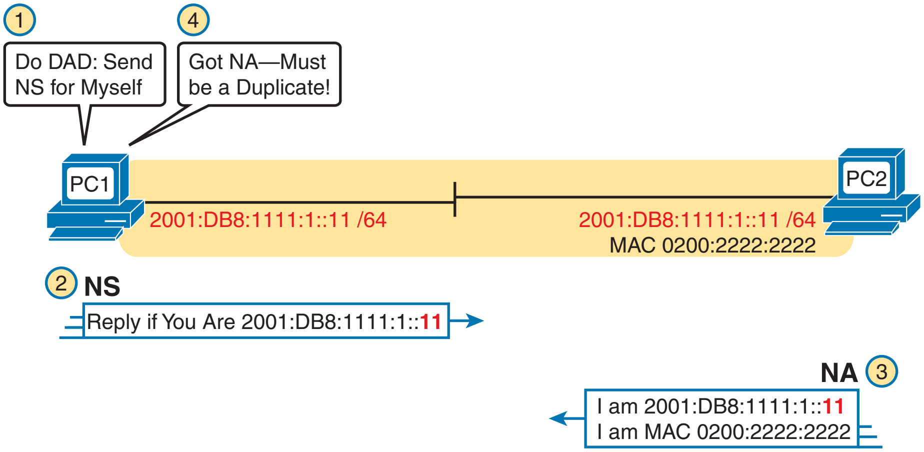


Figure 28-4 *Example Duplicate Address Detection (DAD) with NDP NS/NA*

NDP Summary

This chapter explains some of the many important functions performed by NDP. Use Table 28-2 as a study reference for the four NDP features discussed here.

Key Topic

Table 28-2 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	A simple confirmation of whether a unicast address is already in use

Dynamic Configuration of Host IPv6 Settings

Dynamic Host Configuration Protocol (DHCP) worked well for the dynamic assignment of IPv4 addresses. When the creators of IPv6 protocols looked for a solution for dynamic host address assignment, creating new DHCP protocols for IPv6 made perfect sense. Today, the DHCP Version 6 (DHCPv6) RFC 8415 defines one dynamic IPv6 address assignment method.

However, the creators of IPv6 also wanted another method to assign IPv6 addresses. DHCPv4 uses a server, requiring preconfiguration of the address pools used for each subnet. That model works well in some cases, but using the DHCPv4 model also requires the server to know all the address leases, keeping that information (called state information) about each host (client) and its address.

The creators of IPv6 made two methods for dynamic address assignment:

- **DHCPv6 (Stateful DHCPv6):** This method uses the same stateful model as DHCPv4 using a DHCP server.
- **Stateless Address Autoconfiguration (SLAAC):** The client self-assigns its IPv6 address. This method requires no preconfiguration of address pools and no need for servers to keep state information about the client.

This next major section of the chapter first looks at stateful DHCPv6, followed by SLAAC.

Using Stateful DHCP

DHCPv6 gives an IPv6 host a way to learn host IPv6 configuration settings using the same general concepts as DHCP for IPv4. The host exchanges messages with a DHCP server. The server supplies the host with configuration information, including a lease of an IPv6 address and DNS server address information.

More specifically, stateful DHCPv6 works like the more familiar DHCP for IPv4 in many other general ways, as follows:

Key Topic

- DHCP clients on a LAN send messages that flow only on the local LAN, hoping to find a DHCP server.
- If the DHCP server sits on the same LAN as the client, the client and server can exchange DHCP messages directly, without needing help from a router.
- If the DHCP server sits on another link as compared to the client, the client and server rely on a router to forward the DHCP messages.
- The router that forwards messages from one link to a server in a remote subnet must be configured as a DHCP relay agent, with knowledge of the DHCP server's IPv6 address.
- Servers have configuration that lists pools of addresses for each subnet from which the server allocates addresses.
- Servers offer a lease of an IP address to a client, from the pool of addresses for the client's subnet; the lease lasts a set time (usually days or weeks).
- The server tracks state information, specifically a client identifier (often based on the MAC address), along with the address currently leased to that client.

DHCPv6 defines two branches: **stateful DHCPv6** and **stateless DHCPv6**. Stateful DHCPv6 works more like the DHCPv4 model, especially related to that last item in the list. A stateful DHCPv6 server tracks information about which client has a lease for what IPv6 address; the fact that the server knows information about a specific client is called state information, making the DHCP server a stateful DHCP server.

Stateless DHCP servers do not lease an address to the client, so a stateless DHCP server does not track any per-client information. The upcoming section, "Using Stateless Address Autoconfiguration," discusses how stateless DHCPv6 servers have an important role when a company decides to use SLAAC.

Differences Between Stateful DHCPv6 and DHCPv4

While stateful DHCPv6 has many similarities to DHCPv4, many particulars differ as well. Figure 28-5 shows the differences: Stateful DHCPv6 supplies the address and the DNS server list. However, the host already relies on NDP RA messages to learn the default router address and the prefix length to use, with enough information to build default and on-link routes. So, with stateful DHCPv6, the server does not supply a default router address or prefix length.

DHCPv6 also updates the protocol messages to use IPv6 packets instead of IPv4 packets, with new messages and fields. For example, Figure 28-6 shows the names of the DHCPv6 messages, which replace the DHCPv4 Discover, Offer, Request, and Acknowledgment (DORA) messages. Instead, DHCPv6 uses the Solicit, Advertise, Request, and Reply (SARR) messages.

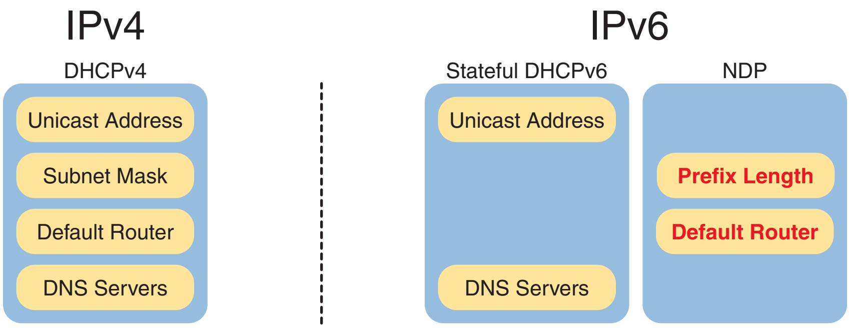


Figure 28-5 *Sources of Specific IPv6 Settings When Using Stateful DHCP*

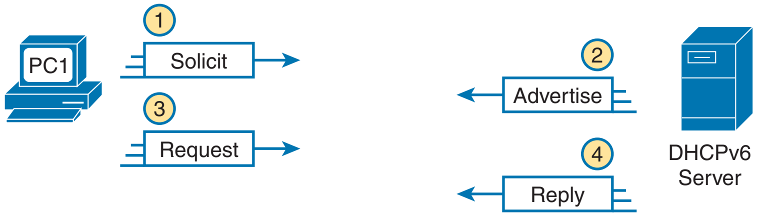


Figure 28-6 *Four Stateful DHCPv6 Messages Between Client and Server*

The four DHCPv6 messages work in two matched pairs with the same general flow as the similar DHCPv4 messages. The Solicit and Advertise messages complete the process of the client searching for the IPv6 address of a DHCPv6 server (the Solicit message) and the server advertising an address (and other configuration settings) for the client to possibly use (the Advertise message). The Request and Reply messages let the client ask to lease the address, with the server confirming the lease in the Reply message. (Note that stateful DHCPv6 supports a rapid commit option that completes the lease using only the Solicit and Reply messages.)

DHCPv6 Relay Agents

For enterprises that choose to use stateful DHCPv6, often the DHCP server sits at a central site, far away from many of the clients that use the DHCPv6 server. In those cases, the local router at each site must act as a DHCP relay agent.

The concepts of DHCPv6 relay work like DHCPv4 relay, as discussed in the section "Configuring DHCP Relay" in Chapter 19, "IP Addressing on Hosts." The client sends a message that normally has a link-local scope so that routers do not forward the packet. By enabling DHCPv6 relay, the router then changes the source and destination IP address, forwarding the packet to the DHCP server. When the server sends a reply, it flows to an address on the router (the relay agent). The router again changes the addresses in the packet for correct delivery to the client.

The differences for IPv6 become more apparent when you look at some of the IPv6 addresses used in DHCPv6 messages, like the Solicit message used to lead off a DHCPv6 flow. As shown in Figure 28-7, the client uses the following addresses in the solicit message:

Source of link-local: The client uses its link-local address as the packet's source address.

Destination address of "all-DHCP-agents" FF02::1:2: The client sends the Solicit message to the link-local scope multicast address FF02::1:2. Only DHCP servers and routers acting as DHCP relay agents listen for these packets.

With a link-local scope multicast destination address, the Solicit message sent by a host would flow only on the local LAN. Figure 28-7 shows how R1, acting as a DHCPv6 relay agent, assists DHCPv6 clients like host A to deliver DHCPv6 packets to the DHCPv6 server.

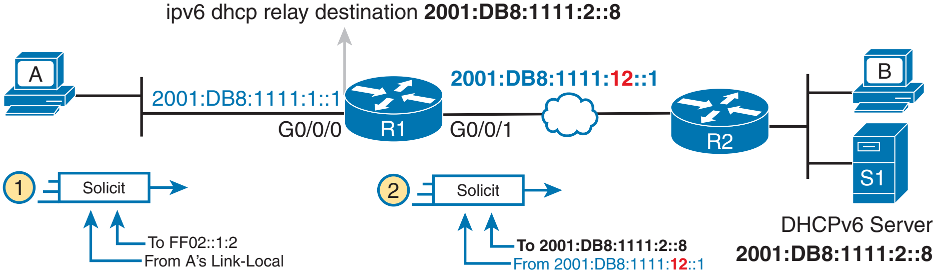


Figure 28-7 *DHCPv6 Relay Agent and DHCP IPv6 Addresses*

In the figure, Step 1 shows the DHCPv6 Solicit message, which would otherwise stay on the link due to the link-local scope of destination multicast address FF02::1:2. Step 2 then shows the action of the DHCP relay agent on Router R1. Router R1 changes the destination IPv6 address to the configured DHCP server's address (2001:DB8:1111:2::8). The DHCP relay agent also sets the source IPv6 address to the address of its outgoing interface (G0/0/1) as the source IPv6 address, which is slightly different from the DHCPv4 relay agent. R1 then forwards the Solicit message to the server.

Continuing the story beyond the figure, the server sends a reply, specifically a DHCPv6 Advertise message. That message reverses the IPv6 addresses used compared to the Solicit message, so the Advertise message uses a destination address of 2001:DB8:1111:12::1. The relay agent in Router R1 reacts by converting the destination address to host A's LLA and forwarding the packet out the interface toward the client.

Example 28-5 shows the DHCPv6 relay agent configuration for R1 in Figure 28-6. The top of the example shows the **ipv6 dhcp relay** interface subcommand, with reference to the IPv6 address of the DHCPv6 server. The bottom of the figure shows the output of the **show ipv6 interface** command, which confirms that R1 is now listening for multicasts sent to the all-DHCP-agents multicast address FF02::1:2.

Example 28-5 *Configuring Router R1 to Support Remote DHCPv6 Server*

```
interface GigabitEthernet0/0/0
 ipv6 dhcp relay destination 2001:DB8:1111:2::8

R1# show ipv6 interface g0/0/0
GigabitEthernet0/0/0 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:DB8:1111:1::1, subnet is 2001:DB8:1111:1::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::5
    FF02::6
    FF02::1:2
    FF02::1:FF00:1
    FF02::1:FF11:1111
! Lines omitted for brevity
```

As an aside, note that of the multicast addresses listed under the heading "Joined group address(es)," the first five are well-known multicast addresses (FF02::1, FF02::2, FF02::5, FF02::6, and FF02::1:2), with two solicited-node multicast addresses that begin with FF02::1:FF.

Using Stateless Address Autoconfiguration

Most companies extensively use DHCPv4, and stateful DHCPv6 makes sense for those same reasons; however, using a stateful DHCP process does have some negatives. Someone has to configure, administer, and manage the DHCP server(s). The configuration includes ranges of IP addresses for every subnet. Then, when a host (client) leases the address, the server notes which client is using which address. All these functions work well, and knowing the information in the DHCP server can be pretty valuable; however, the reliance on a stateful DHCP server requires some thought and attention from the IT staff.

IPv6's **Stateless Address Autoconfiguration (SLAAC)** provides an alternative method for dynamic IPv6 address assignment—without needing a stateful server. In other words, SLAAC does not require a server to assign or lease the IPv6 address, does not require the IT staff to preconfigure a pool of addresses per subnet, and does not require the server to track state information about which device uses which IPv6 address.

The term *SLAAC* refers to both a specific part of how a host learns one IPv6 setting—its IPv6 address—plus the overall process of learning all four key host IPv6 settings (address, prefix length, default router, and the list of DNS server addresses). This next topic begins by looking at the tasks done by SLAAC related to the IPv6 address. Then the text looks at the overall SLAAC process to find all four host settings—a process that also uses NDP and stateless DHCP.

Building an IPv6 Address Using SLAAC

When using SLAAC, a host does not lease its IPv6 address. Instead, the host learns part of the address from the nearby router—the prefix—and then makes up the rest of its IPv6 address. Specifically, a host using SLAAC to choose its IPv6 address uses the following steps:

1. Learn the IPv6 prefix used on the link from any router, using NDP RS/RA messages.

2. Choose its IPv6 address by making up the interface ID (IID) value to follow the just-learned IPv6 prefix.
3. Before using the address, use DAD to ensure that no other host is already using the same address.

Figure 28-8 depicts the first two steps while noting the two most common ways a host completes the address. Hosts can use modified EUI-64 rules, as discussed in the section, “Generating a Unique Interface ID Using Modified EUI-64,” in Chapter 27, “Implementing IPv6 Addressing on Routers,” or a random number.

Key Topic

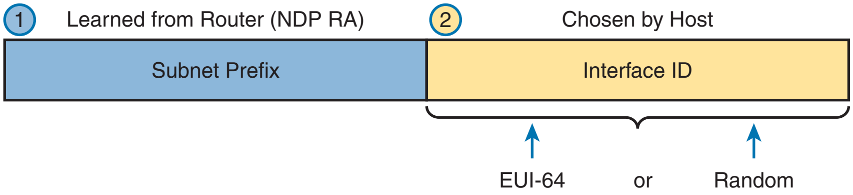


Figure 28-8 *Host IPv6 Address Formation Using SLAAC*

Combining SLAAC with Stateless DHCP

When using SLAAC, a host uses three tools to find its four IPv6 settings, as noted in Figure 28-9. SLAAC itself focuses on the IPv6 address only. The host then uses NDP messages to learn the prefix length and the IPv6 addresses of the default routers on the link. Finally, the host uses stateless DHCP to learn the IPv6 addresses of any DNS servers.

Key Topic

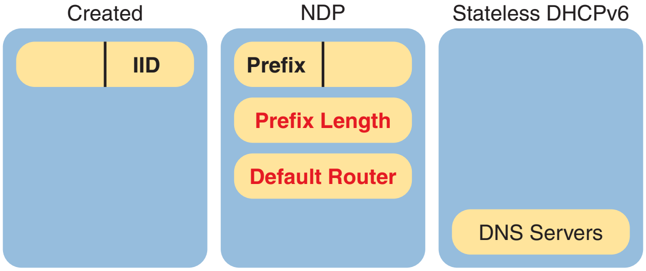


Figure 28-9 *Sources of Specific IPv6 Settings When Using SLAAC*

When SLAAC uses DHCP to supply the list of DNS servers, the server implements stateless DHCP. With stateless DHCPv6, the DHCPv6 server

- Needs simple configuration only, specifically the short list of DNS server addresses
- Needs no per-subnet configuration: no subnet list, no per-subnet address pools, no list of excluded addresses per subnet, and no per-subnet prefix lengths
- Has no need to track state information about DHCP leases—that is, which devices lease which IPv6 address—because the server does not lease addresses to any clients

Table 28-3 summarizes the key comparison points between stateful and stateless DHCP.

Table 28-3 Comparison of Stateless and Stateful DHCPv6 Services

Feature	Stateful DHCP	Stateless DHCP
Remembers IPv6 address (state information) of clients	Yes	No
Leases IPv6 address to client	Yes	No
Supplies list of DNS server addresses	Yes	Yes
Commonly used with SLAAC	No	Yes

Combining SLAAC with RA-Based DNS Server Configuration

SLAAC originally relied on a stateless DHCP server to supply the DNS server list. IPv6 now supports another option to deliver the DNS server list, called RA-based DNS configuration, which removes the need for a stateless DHCP server.

With RA-based DNS Server (RDNSS) configuration, you configure each router interface with the list of DNS servers. Then, when sending each NDP RA message, the router supplies the DNS list in its NDP RA (Router Advertisement) messages. As a result, RDNSS configuration provides a means for automatic assignment of all client IPv6 settings using router configuration only, with no DHCP server at all. However, note that it also requires configuration of the DNS server list on all routers that support IPv6 so that a centralized stateless DHCPv6 server may be more practical to manage.

Permanent and Temporary SLAAC Addresses

In practice, hosts use SLAAC to generate multiple unicast addresses, specifically a **permanent IPv6 address** plus a series of **temporary IPv6 addresses** over time. The permanent address remains unchanged over time. If that host runs as a server, the host will use the permanent address for all incoming connections. You would also typically add a DNS entry for that address in the DNS server so clients could easily connect.

Hosts use temporary addresses for client applications. For instance, when you open a web browser tab and connect to a website, your host connects using the temporary address rather than the permanent address.

Using temporary addresses makes hosts more secure, as detailed in a series of related RFCs that revolve around RFC 8981. Attackers may gather information and packets sent by a host through various attacks. If the captured packets over time have the same address, the attacker can more easily correlate the information to find a way to gain access to the host or deny services to it. By frequently changing the address, you can prevent the attacker's data analysis from giving them an edge in attacking your host.

When created, a host assigns all SLAAC-created addresses a **preferred lifetime** and **valid lifetime** setting. Each temporary address moves through stages called preferred, valid, and deprecated. During the preferred phase, the host uses the address for new connections. When the preferred lifetime expires, the address becomes valid. The host can continue using a valid address for existing connections but not for new ones. After the valid lifetime expires, the host considers the address deprecated and does not use the address again. Figure 28-10 shows the concepts.

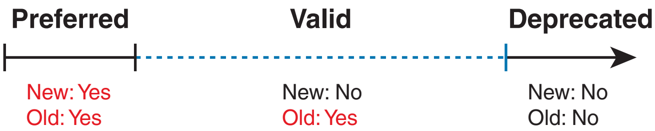


Figure 28-10 *SLAAC Preferred and Valid Lifetimes for One Temporary Address*

With temporary addresses, the preferred and valid lifetimes have short settings. A typical setting might be 24 hours for the preferred lifetime and one week for the valid lifetime. In that case, every 24 hours, the host would need a new temporary address because the last temporary address moves from a preferred to a valid state. Basically, at any one time, the host needs one temporary address in its preferred lifetime state. Figure 28-11 shows the idea, with the timelines for three consecutive temporary addresses appearing 24 hours after the previous one. At any point, the host has one temporary address with some preferred lifetime remaining, so the host can use the new preferred address for new application connections.

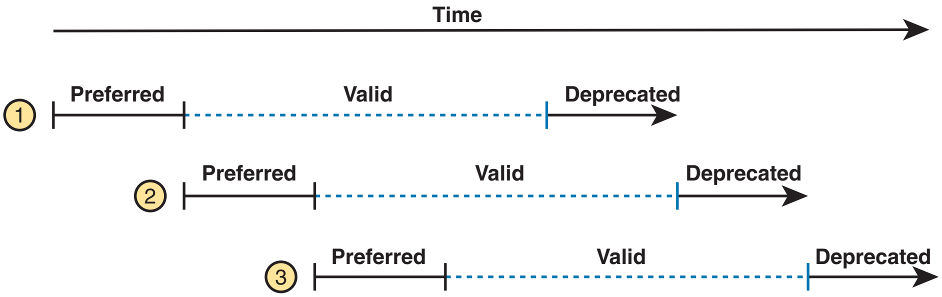


Figure 28-11 *Creating New Temporary Addresses to Ensure One Has Preferred Life Remaining*

You can see the permanent and temporary addresses in host commands like these:

macOS: **ifconfig en0 inet 6** or **ifconfig -aL inet6**

Linux: **ip -6 address**

Windows: **ipconfig /all** or **netsh interface ipv6 show address** (Windows netshell)

Windows: **Get-NetIPConfiguration -Detailed** (Windows PowerShell)

Example 28-6 shows output from the **netsh interface ipv6 show address** command on Windows. The example shows an excerpt of the output for the one working wireless LAN interface. The output shows its permanent (public) address used for incoming connections, with an infinite valid and preferred lifetime. It also shows the current temporary address, with a preferred lifetime of just under 24 hours and a valid lifetime of just under one week. Also, examine the IIDs of the addresses for ff:fe—again, their absence signals that this host did not use EUI-64 but instead generated a random IID.

Example 28-6 *Windows SLAAC Addresses with the* **netsh interface ipv6 show address** *Command*

```
C:\Users\Wendell> netsh
netsh> interface ipv6
netsh interface ipv6> show address

Interface 7: Wi-Fi

Addr Type  DAD State   Valid Life Pref. Life Address
---------  ----------- ---------- ---------- ------------------------
Temporary  Preferred  6d23h57m58s   23h49m3s 2001:db8:1111:1:c1cd:7a44:45a5:58f1
Public     Preferred     infinite   infinite 2001:db8:1111:1:f1f5:5cbb:f395:6c51
Other      Preferred     infinite   infinite fe80::f1f5:5cbb:f395:6c51%7
```

Note that upcoming Example 28-7 shows a sample of similar commands for macOS.

Troubleshooting Host IPv6 Addressing

This chapter's third and final major section examines a few commands to verify and troubleshoot IPv6 addressing configuration on hosts. Specifically, this section examines the host's IPv6 settings and then looks at the usual commands to test whether a host can send packets: **ping** and **traceroute**.

Note that this section lists some commands on different host OSs; however, be aware that this and other chapters do not attempt to show each variation of every networking command on every OS. Instead, the host command examples reinforce the concepts seen earlier in the chapter.

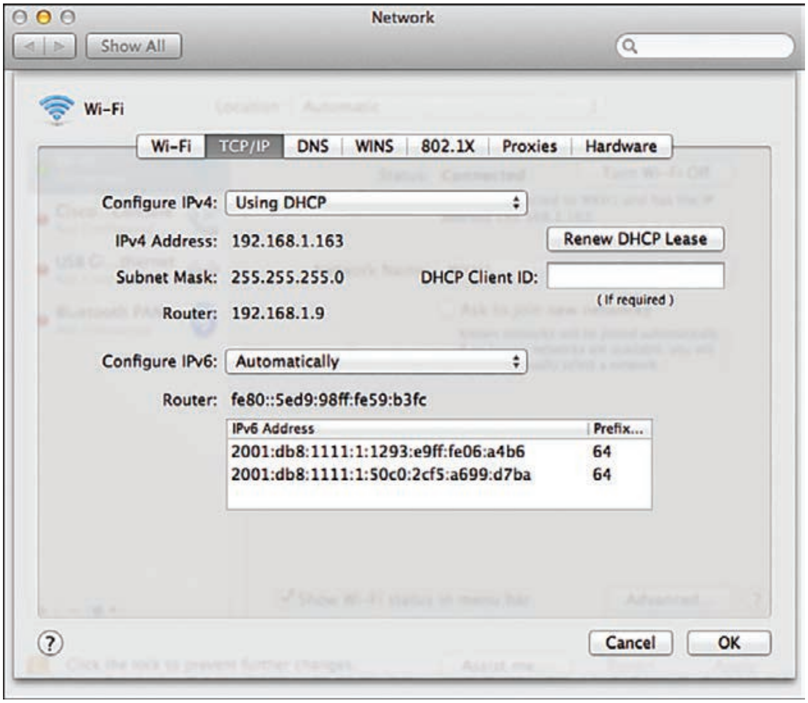
Verifying IPv6 Connectivity from Hosts

Most end-user OSs support a convenient way to look at IPv6 settings from the graphical user interface. In some cases, all four of the key IPv6 host settings can be on the same window, whereas in other cases, seeing all the settings might require navigation to multiple windows or tabs in the same window. The following few pages first focus on how to find the IPv6 addresses used by hosts, followed by how to test connectivity using the **ping** and **traceroute** commands.

Host Commands to Find IPv6 Interface Addresses

For example, Figure 28-12 shows a macOS window that lists three IPv6 host settings. The one missing setting, the DNS server setting, is in another tab (as seen near the top of the image).

Take a moment to look at the details in Figure 28-12's image. The image shows the IPv4 settings at the top. The lower half of the window shows the IPv6 settings as having been learned "Automatically," which means that the host will use either stateful DHCP or SLAAC. In this case, the host used SLAAC to give itself two IPv6 addresses inside the same 2001:DB8:1111:1::/64 subnet. However, the graphical interface does not identify the permanent and temporary addresses or lifetimes.



Configure IPv4: Using DHCP

IPv4 Address: 192.168.1.163

Subnet Mask: 255.255.255.0

Router: 192.168.1.9

DHCP Client ID:

Configure IPv6: Automatically

Router: fe80::5ed9:98ff:fe59:b3fc

IPv6 Address	Prefix...
2001:db8:1111:1:1293:e9ff:fe06:a4b6	64
2001:db8:1111:1:50c0:2cf5:a699:d7ba	64

Figure 28-12 *Three IPv6 Settings for Dynamic Address Assignment on macOS*

Hosts support a range of commands to find more detail about the addresses, including lifetimes. Example 28-6 earlier showed the **netsh interface ipv6 show address** command for Windows, for example, with Linux using the **show -6 address** command. Example 28-7 shows these exact details on macOS with two commands. The first command, **ifconfig en8**, lists details about Ethernet, IPv4, and IPv6, on one specific interface (internally numbered as Ethernet number 8, or en8). The command also identifies the preferred address (using the keyword *secured*) and temporary address, as highlighted in the upper part of the example.

Example 28-7 *Sample* **ifconfig** *Commands from a Mac*

```
Mac% ifconfig en8
en8: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
       options=6467<RXCSUM,TXCSUM,VLAN_MTU,TSO4,TSO6,CHANNEL_IO,PARTIAL_
CSUM,ZEROINVERT_CSUM>
        ether 00:e0:4c:68:1f:26
        inet6 fe80::184c:56f9:fd3b:d6e7%en8 prefixlen 64 secured scopeid 0x14
        inet6 2001:db8:1111:1:106a:dd3e:8e22:a6fb prefixlen 64 autoconf secured
        inet6 2001:db8:1111:1:ec69:15f9:b4fc:fe2c prefixlen 64 autoconf temporary
        inet 192.168.1.120 netmask 0xffffffff broadcast 192.168.1.120
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect (1000baseT <full-duplex>)
        status: active
```

```
Mac% ifconfig -aL inet6
! Only interface en8 shown for brevity
en8: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
     options=6467<RXCSUM,TXCSUM,VLAN_MTU,TSO4,TSO6,CHANNEL_IO,PARTIAL_
CSUM,ZEROINVERT_CSUM>
     inet6 fe80::8ad:f140:a952:9a46%en8 prefixlen 64 secured scopeid 0xc
     inet6 2001:db8:1111:1:106a:dd3e:8e22:a6fb prefixlen 64 autoconf secured
pltime 604654 vltime 2591854
     inet6 2001:db8:1111:1:a968:a6d9:7fbf:38a6 prefixlen 64 autoconf temporary
pltime 85782 vltime 604654
     nd6 options=201<PERFORMNUD,DAD>
```

The lower part of the output shows how to find the preferred (pltime) and valid (vltime) lifetimes for each address using the **ifconfig -aL inet6** command. The output also lists the word *autoconf*, which implies the host used SLAAC. The output also identifies one permanent address (secured) and the other as the temporary address. Finally, note that the SLAAC-derived addresses appear to use random interface IDs, because the string ff:fe does not exist in the middle of the interface ID.

Testing IPv6 Connectivity with ping and traceroute

The **ping** and **traceroute** commands make for great connectivity testing tools for IPv4 as well as for IPv6. Some OSs (notably Microsoft Windows variants and Cisco routers and switches) let you use the same **ping** and **traceroute** commands used with IPv4. Some other OSs require a different command, like the **ping6** and **traceroute6** commands used with macOS and Linux. (The upcoming examples show both variations.)

As for the output of the **ping** and **traceroute** commands, most people who understand the commands as used with IPv4 need no coaching to use the commands with IPv6. The output is mostly unchanged compared to the IPv4 equivalents, other than the obvious differences with listing IPv6 addresses. For comparison, the upcoming examples use the internetwork displayed in Figure 28-13.

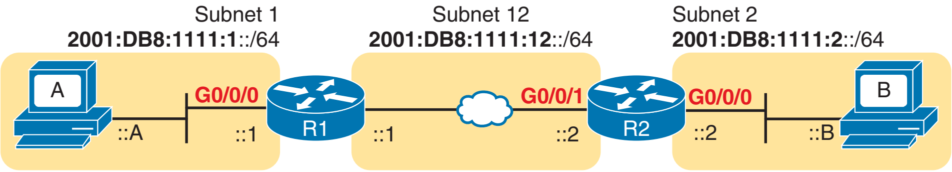


Figure 28-13 *IPv6 Internetwork for* **ping** *and* **traceroute** *Examples*

Example 28-8 shows three **ping6** commands, taken from PC1, a Linux host. (Linux uses **ping6** and **traceroute6** commands for IPv6.) The first two commands show IPv6 pings, the first to R1's LAN IPv6 address, followed by PC1 pinging PC B's IPv6 address. The final command shows an IPv4 ping for comparison.

Key Topic

Example 28-8 *The* **ping6** *Command from PC1, for R1 and PC2*

```
! An IPv6 ping, PC A pings R1's address in the same subnet
Linux_A:$ ping6 2001:db8:1111:1::1
PING 2001:db8:1111:1::1 (2001:db8:1111:1::1) 56 data bytes
64 bytes from 2001:db8:1111:1::1: icmp_seq=1 ttl=64 time=1.26 ms
64 bytes from 2001:db8:1111:1::1: icmp_seq=2 ttl=64 time=1.15 ms
^C
--- 2001:db8:1111:1::1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001 ms
rtt min/avg/max/mdev = 1.156/1.210/1.263/0.062 ms

! An IPv6 ping next, ping of PC B from PC A
Linux_A:$ ping6 2001:db8:1111:2::b
PING 2001:db8:1111:2::b (2001:db8:1111:2::b) 56 data bytes
64 bytes from 2001:db8:1111:2::b: icmp_seq=1 ttl=64 time=2.33 ms
64 bytes from 2001:db8:1111:2::b: icmp_seq=2 ttl=64 time=2.59 ms
64 bytes from 2001:db8:1111:2::b: icmp_seq=3 ttl=64 time=2.03 ms
^C
--- 2001:db8:1111:2::b ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003 ms
rtt min/avg/max/mdev = 2.039/2.321/2.591/0.225 ms

! An IPv4 ping next, for comparison - ping of PC B from PC A
Linux_A:$ ping 10.1.2.22
PING 10.1.3.22 (10.1.2.22) 56 data bytes
64 bytes from 10.1.2.22: icmp_seq=1 ttl=64 time=2.45 ms
64 bytes from 10.1.2.22: icmp_seq=2 ttl=64 time=2.55 ms
64 bytes from 10.1.2.22: icmp_seq=3 ttl=64 time=2.14 ms
^C
--- 10.1.3.22 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2014 ms
rtt min/avg/max/mdev = 2.04/2.318/2.604/0.224 ms
```

Example 28-9 shows a **traceroute6** command on PC A, finding the route to PC B. The output mirrors the style of output for most IPv4 **traceroute** commands, other than the obvious difference of listing IPv6 addresses. Note that the output lists R1's G0/0/0 IPv6 address, R2's G0/0/1 IPv6 address, and then PC B's address to end the output.

Example 28-9 *The* **traceroute6** *Command from PC1, for PC2*

```
Linux_A:$ traceroute6 2001:db8:1111:2::b
traceroute to 2001:db8:1111:2::b (2001:db8:1111:2::b) from 2001:db8:1111:1::a,
   30 hops max, 24 byte packets
1  2001:db8:1111:1::1 (2001:db8:1111:1::1)  0.794 ms  0.648 ms  0.604 ms
2  2001:db8:1111:12::2 (2001:db8:1111:12::2)  1.606 ms  1.49 ms  1.497 ms
3  2001:db8:1111:2::b (2001:db8:1111:2::b)  2.038 ms  1.911 ms  1.899 ms
```

NOTE The **traceroute/traceroute6** commands learn the addresses using the ICMPv6 time exceeded message. That mechanism results in the command output listing the GUA of the various routers and destination host.

Verifying Host Connectivity from Nearby Routers

For router verification commands for IPv6, some IPv6 features use the same command as with IPv4, but some substitute "ipv6" for "ip." And in some cases, particularly with functions that do not exist in IPv4 or have changed quite a bit, routers support brand-new commands. This section looks at a couple of router commands useful to verify IPv6 host connectivity, some old and some new for IPv6.

First, for connectivity testing, Cisco routers and switches support the **ping** and **traceroute** commands with the same basic features for IPv6 as with IPv4. The commands accept either an IPv4 or an IPv6 address as input for the standard version of the commands. For the extended versions of these commands, the first prompt question asks for the protocol. Just type **ipv6** instead of using the default of **ip**, and answer the rest of the questions.

Of course, an example helps, particularly for the extended commands. Example 28-10 begins with an extended IPv6 **ping** from R1 to PC B from Figure 28-13, using R1's G0/0/0 interface as the source of the packets. The second command shows a standard IPv6 **traceroute** from R1 to PC B.

Example 28-10 *Extended* **ping** *and Standard* **traceroute** *for IPv6 from Router R1*

```
R1# ping
Protocol [ip]: ipv6
Target IPv6 address: 2001:db8:1111:2::b
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands? [no]: yes
Source address or interface: GigabitEthernet0/0/0
UDP protocol? [no]:
Verbose? [no]:
Precedence [0]:
DSCP [0]:
Include hop by hop option? [no]:
Include destination option? [no]:
Sweep range of sizes? [no]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:1111:2::b, timeout is 2 seconds:
Packet sent with a source address of 2001:DB8:1111:1::1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/1/4 ms
```

```
R1# traceroute 2001:db8:1111:2::b
Type escape sequence to abort.
Tracing the route to 2001:DB8:1111:2::b

  1 2001:DB8:1111:12::2 4 msec 0 msec 0 msec
  2 2001:DB8:1111:2::b 0 msec 4 msec 0 msec
```

Another way to verify host settings from a router is to look at the router's NDP neighbor table. All IPv6 hosts, routers included, keep an IPv6 neighbor table: a list of all neighboring IPv6 addresses and matching MAC addresses.

One way to verify whether a neighboring host is responsive is to determine whether it will send back an NDP NA when the router sends it an NDP NS (to discover the host's MAC address). To do so, the router could clear its neighbor table (**clear ipv6 neighbor**) and ping a host on some connected interface. The router will first need to send an NDP NS, and the host must send an NDP NA back. If the router shows that host's MAC address in the neighbor table, the host must have just replied with an NDP NA. Example 28-11 shows a sample of an IPv6 neighbor table from Router R2 in Figure 28-13, using the **show ipv6 neighbors** command.

Example 28-11 *The* **show ipv6 neighbors** *Command on Router R2*

```
R2# show ipv6 neighbors
IPv6 Address                              Age Link-layer Addr State Interface
2001:DB8:1111:2::B                          0 0200.bbbb.bbbb  STALE Gi0/0/0
FE80::BBFF:FEBB:BBBB                        0 0200.bbbb.bbbb  STALE Gi0/0/0
FE80::FF:FE01:1                             0 2436.dadf.5681  REACH Gi0/0/1
```

Finally, routers can also list information about the available routers on a LAN subnet, which impacts the connectivity available to hosts. As a reminder, routers send NDP RA messages to announce their willingness to act as an IPv6 router on a particular LAN subnet. Cisco routers watch for RA messages from other routers, typically receiving unsolicited RA messages that routers send to the FF02::1 all IPv6 hosts multicast address. The **show ipv6 routers** command lists any other routers but not the local router. Refer to earlier Example 28-3 for a sample of the **show ipv6 routers** command output.

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 28-4 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 28-4 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 table		Book, website
Review command tables		Book

Review All the Key Topics

Key Topic

Table 28-5 Key Topics for Chapter 28

Key Topic Element	Description	Page Number
List	Four functions for which NDP plays a major role	699
List	Descriptions of the NDP NS and NA messages	699
Figure 28-1	Example use of NDP RS and RA	700
List	Descriptions of the NDP RS and RA messages	702
Figure 28-3	Example use of NDP NS and NA	703
List	Two important routes created by IPv6 hosts based on the RA message	704
Figure 28-4	Example use of NDP for Duplicate Address Detection (DAD)	705
Table 28-2	Summary of NDP functions discussed in this chapter	706
List	Similarities between DHCP for IPv4 and stateful DHCP for IPv6	707
Figure 28-5	Sources of host IPv6 configuration when using stateful DHCPv6	708
Figure 28-8	SLAAC address creation concepts	711
Figure 28-9	Sources of host IPv6 configuration when using SLAAC	711
Example 28-8	Examples of the **ping6** command	717
Example 28-10	Using extended **ping** and standard **traceroute** for IPv6	718

Key Terms You Should Know

Duplicate Address Detection (DAD), IPv6 neighbor table, Neighbor Advertisement (NA), Neighbor Discovery Protocol (NDP), Neighbor Solicitation (NS), on-link prefix, permanent IPv6 address, preferred lifetime, prefix discovery, Router Advertisement (RA), Router Solicitation (RS), stateful DHCPv6, Stateless Address Autoconfiguration (SLAAC), stateless DHCPv6, temporary IPv6 address, valid lifetime

Command References

Tables 28-6, 28-7, and 28-8 list configuration and verification commands used in this chapter, respectively. 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 28-6 Chapter 28 Configuration Command Reference

Command	Description
ipv6 dhcp relay destination *server-address*	Interface subcommand that enables the IPv6 DHCP relay agent

Table 28-7 Chapter 28 EXEC Command Reference

Command	Description
ping {*host-name* | *ipv6-address*}	Tests IPv6 routes by sending an ICMP packet to the destination host
traceroute {*host-name* | *ipv6-address*}	Tests IPv6 routes by discovering the IP addresses of the routes between a router and the listed destination
show ipv6 neighbors	Lists the router's IPv6 neighbor table
show ipv6 routers	Lists any neighboring routers that advertised themselves through an NDP RA message

Table 28-8 Chapter 28 Host Command Reference

Command (Windows/macOS/Linux)	Description
ipconfig / ifconfig / ifconfig	(Windows/macOS/Linux) Lists interface settings, including IPv4 and IPv6 addresses
ping / ping6 / ping6	(Windows/macOS/Linux) Tests IP routes by sending an ICMPv6 packet to the destination host
tracert / traceroute6 / traceroute6	(Windows/macOS/Linux) Tests IP routes by discovering the IPv6 addresses of the routes between a router and the listed destination
netsh interface ipv6 show neighbors / get-Neighbor -AddressFamily IPv6	(Windows only) Lists IPv6 neighbors with network shell and PowerShell
ndp -an / ip -6 neighbor show	(macOS/Linux) Lists IPv6 neighbors
netsh interface ipv6 show route / netstat -rnf inet6 / ip -6 route	(Windows/macOS/Linux) Lists a host's IPv6 routing table
netsh interface ipv6 show address / ifconfig -aL inet6 / ip -6 address	(Windows/macOS/Linux) Lists a host's interface IPv6 addresses

CHAPTER 29

Implementing IPv6 Routing

This chapter covers the following exam topics:

- 3.0 IP Connectivity
  - 3.1 Interpret the components of routing table
    - 3.1.a Routing protocol code
    - 3.1.b Prefix
    - 3.1.c Network mask
    - 3.1.d Next hop
    - 3.1.e Administrative distance
  - 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 four major topics. The first section examines IPv6 connected and **local routes**—the routes a router adds to its routing table in reaction to IPv6 address configuration. The second major section discusses static network routes, that is, routes configured by the **ipv6 route** command with a destination of an IPv6 prefix (subnet). The third major section examines other IPv6 static route topics, including default routes, host routes, and floating static routes, with some notes about troubleshooting all static IPv6 routes. The final short section discusses a few tools and tips to troubleshoot IPv6 static routes.

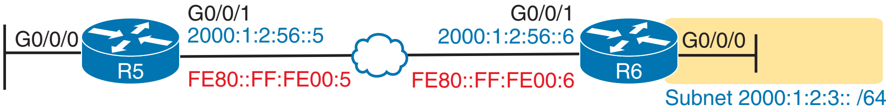
"Do I Know This Already?" Quiz

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 29-1 "Do I Know This Already?" Foundation Topics Section-to-Question Mapping

Foundation Topics Section	Questions
Connected and Local IPv6 Routes	1, 2
Static IPv6 Network Routes	3, 4
Static Default, Host, and Floating Static IPv6 Routes	5, 6
Troubleshooting Static IPv6 Routes	7

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



1. Router R6 in the figure has been configured with the **ipv6 address 2000:1:2:3::1/64** command on its G0/0/0 interface. The router also creates a link-local address of FE80::FF:FE00:1. All router interfaces have an up/up status. Which of the following routes will R6 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 G0/0/0 interface and **ipv6 address 3222:2:2:2::1/64** on its G0/0/1 interface. Both interfaces have an up/up state. 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 in the figure. Which of the following **ipv6 route** commands would result in a working IPv6 route for this subnet prefix? (Choose two answers.)
   - **a.** **ipv6 route 2000:1:2:3::/64 G0/0/1 2000:1:2:56::6**
   - **b.** **ipv6 route 2000:1:2:3::/64 2000:1:2:56::6**
   - **c.** **ipv6 route 2000:1:2:3::/64 G0/0/1** 2001:1:2:56::5
   - **d.** **ipv6 route 2000:1:2:3::/64 G0/0/1**
4. An engineer needs to add a static IPv6 route for prefix 2000:1:2:3::/64 to Router R5 in the figure. 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. When displaying an IPv6 static default route, how does the **show ipv6 route** command represent the concept of the default destination?
   - **a.** With the phrase "Gateway of Last Resort" just above the list of routes
   - **b.** With a prefix value ::/0 in the line for the route
   - **c.** With a prefix value ::/128 in the line for the route

   d. With a prefix value 2000::/3 in the line for the route
   e. With the keyword "default" in the line for the route

6. Router R1 has a useful IPv6 configuration so that Router R1 learns a route for subnet 2001:db8:1:2::/64 with OSPF. The route lists an outgoing interface that is an Ethernet WAN link. The engineer then installs cellular interfaces on both routers for WAN backup. As part of the configuration, the engineer configures the **ipv6 route 2001:db8:1:2::/64 cellular0/2/0 200** command. What does the router do in response?
   a. It rejects the **ipv6 route** command because of the existing OSPF route for the same prefix.
   b. It accepts the **ipv6 route** command into the configuration but does not add a route to the routing table.
   c. It accepts the **ipv6 route** command into the configuration and adds the route to the routing table, but it leaves the OSPF-learned route in the routing table.
   d. It accepts the **ipv6 route** command into the configuration, adds the route to the routing table, and removes the OSPF-learned route in the routing table.
7. 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 lists no routes 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 uses a next-hop global unicast address, which is not allowed, preventing the router from adding a route.
   b. The command must include an outgoing interface parameter, so IOS rejected the **ipv6 route** command.
   c. The router has no routes that match the next-hop address 2001:DB8:9:9::9.
   d. A route for 2001:DB8:8:8::/64 with administrative distance 110 already exists.

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 residing in the CCNP Enterprise exams.

Also, as an early reminder of a few essential acronyms from earlier chapters, remember that GUA refers to global unicast addresses, the most common routable unicast address configured on Enterprise router interfaces. Unique local addresses (ULA) serve as private addresses. Finally, routes frequently use a neighboring router's link-local address (LLA), a unicast address that exists on every router and host interface, useful for sending packets over the local link.

Rules for Connected and Local Routes

Routers add and remove IPv6 connected and **IPv6 local routes** based on the interface configuration and state. First, the router looks for 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 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.

For example, consider a router configured with a working interface 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 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 local route for that address, with a /128 prefix length. (With IPv4, local 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:

29

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 local 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 29-1 details 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.

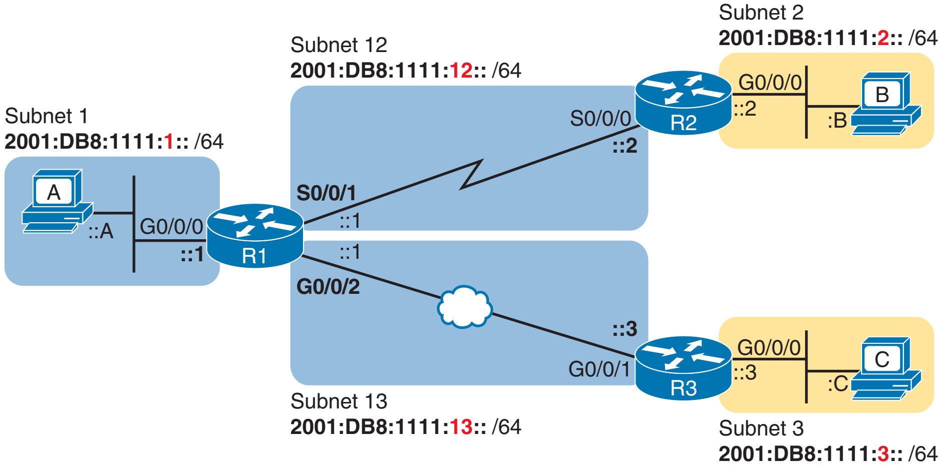


Figure 29-1 *Sample Network Used to Show Connected and Local Routes*

To clarify the notes in Figure 29-1, 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 G0/0/0 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 29-1, as shown in Example 29-1. The excerpt from the **show running-config** command on R1 shows three working interfaces. Also note that no static route or routing protocol configuration exists.

Example 29-1 *IPv6 Addressing Configuration on Router R1*

```
ipv6 unicast-routing
!
! Unused interfaces omitted
!
interface GigabitEthernet0/0/0
 ipv6 address 2001:DB8:1111:1::1/64
!
interface Serial0/0/1
 ipv6 address 2001:db8:1111:12::1/64
!
interface GigabitEthernet0/0/2
 ipv6 address 2001:db8:1111:13::1/64
```

Based on Figure 29-1 and Example 29-1, R1 should have three connected IPv6 routes, as highlighted in Example 29-2.

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

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

Example 29-2 *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 GigabitEthernet0/0/0, directly connected
L   2001:DB8:1111:1::1/128 [0/0]
     via GigabitEthernet0/0/0, receive
C   2001:DB8:1111:12::/64 [0/0]
     via Serial0/0/1, directly connected
L   2001:DB8:1111:12::1/128 [0/0]
     via Serial0/0/1, receive
C   2001:DB8:1111:13::/64 [0/0]
     via GigabitEthernet0/0/2, directly connected
L   2001:DB8:1111:13::1/128 [0/0]
     via GigabitEthernet0/0/2, 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 details the connected route for subnet 2001:DB8:1111:1::/64. The first pair of highlighted lines states: The route is a "directly connected" route; the interface ID is GigabitEthernet0/0/0; 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 29-3 shows only the local routes, as listed by the **show ipv6 route local** command, with highlights of one particular local route for discussion.

Example 29-3 *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 GigabitEthernet0/0/0, receive
L   2001:DB8:1111:4::1/128 [0/0]
     via Serial0/0/1, receive
L   2001:DB8:1111:5::1/128 [0/0]
     via GigabitEthernet0/0/2, 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 29-1, and note R1's IPv6 address on its G0/0/0 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 Network Routes

IPv6 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 network route is the most common type of static route among the types mentioned in the CCNA exam topics (network, host, default, and floating static). A network route, by definition, defines a route to the addresses in a subnet based on the listed prefix and prefix length. The **ipv6 route** global command also lists the forwarding instructions of how this router should forward packets toward that destination prefix by listing the outgoing interface or the address of the next-hop router.

Figure 29-2 shows the general concepts behind a single **ipv6 route** command for a network route. The figure shows the ideas behind a static route on Router R1 for the subnet on the right (subnet 2, or 2001:DB8:1111:2::/64). The command begins with **ipv6 route 2001:DB8:1111:2::/64**, defining the destination subnet. The rest of the command defines the forwarding instructions—either the outgoing interface (S0/0/1), the next-hop IPv6 address, or both.

Now that you understand the big ideas with IPv6 static network 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 GUA, and then with a next-hop LLA.

Key Topic

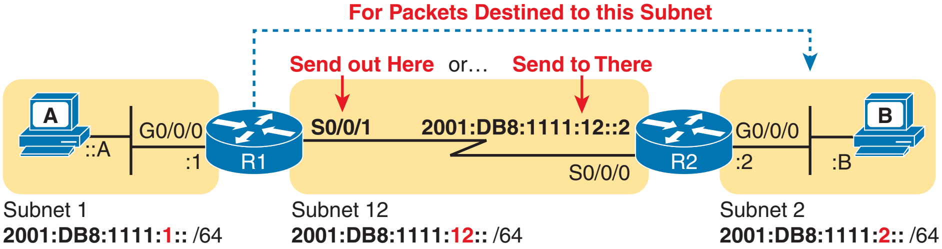


Figure 29-2 *Logic Behind IPv6 Static Route Commands (IPv6 Route)*

Static Network Routes Using an 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 29-2, R1's **ipv6 route** command would use interface S0/0/1, as shown in Example 29-4.

Example 29-4 *Static IPv6 Routes on Router R1*

```
! Static route on router R1
R1(config)# ipv6 route 2001:db8:1111:2::/64 S0/0/1
```

If you were attempting to support packet flow between all hosts, like PCs A and B in the figure, you need static routes on each router for each remote subnet. 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 destined for host B's subnet out R1's S0/0/1 interface to R2 next. However, Router R2 does not yet have a route that matches addresses in host A's subnet (2001:DB8:1111:1::/64), so a complete static routing solution requires more routes.

Example 29-5 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 29-5 *Static IPv6 Routes on Router R2*

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

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 29-6 shows that output, with the legend omitted.

Example 29-6 *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 Serial0/0/1, 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 29-4), as does the outgoing interface (S0/0/1).

When working with IPv6 routes, you will often wonder which route the router will match for a given destination address. You should be ready to think through that question while examining a full IPv6 routing table, but if answering a lab question on the exam, you can use a command that tells you the specific route, as seen in Example 29-7. For example, if host A sent an IPv6 packet to host B (2001:DB8:1111:2::B), 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::B** command. This command asks the router to list the route that the router would use when forwarding packets to that particular address.

Example 29-7 *Displaying the Route R1 Uses to Forward to Host B*

```
R1# show ipv6 route 2001:db8:1111:2::b
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 Serial0/0/1
      Last updated 00:01:29 ago
```

NOTE Only serial interfaces, and not Ethernet interfaces, support using an **ipv6 route** command with only an outgoing interface as the forwarding instructions. With Ethernet interfaces, the router will accept the command, but the router cannot forward packets using the route. See the later section, "Troubleshooting Static IPv6 Routes," for more details on this case and other challenges with static IPv6 routes.

Static Network Routes Using Next-Hop IPv6 Address

IPv6 static route commands can use only the next-hop router address or a combination of the next-hop router address plus the outgoing interface. Note that these options work when the route forwards packets out any kind of interface, including Ethernet and serial links, but the upcoming examples show an Ethernet WAN link.

Interestingly, IPv6 supports using the neighbor's GUA, ULA, or LLA as the next-hop IPv6 address. The following list provides the working combinations, with Figure 29-3 depicting the logic from R1's perspective for a route with R3 as the next-hop router:

- Next-hop GUA or ULA only
- Next-hop GUA or ULA and outgoing interface
- Next-hop LLA and outgoing interface

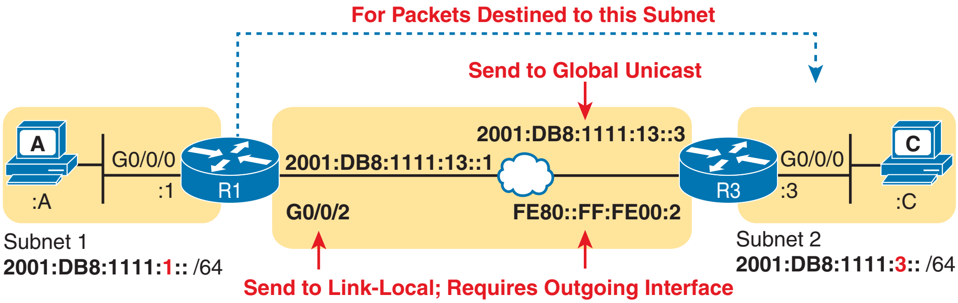


Figure 29-3 *Using GUA or LLA as the Next-Hop Address for Static IPv6 Routes*

The next few pages walk you through examples, first with a GUA and then with an LLA as a next-hop router.

Example Static Network Route with a Next-Hop GUA

In Example 29-8, R1 adds a static route that refers to neighbor R3's GUA per Figure 29-3. The command lists subnet 3 (2001:DB8:1111:3::/64), which exists on the far side of Router R3. The route then lists R3's GUA (ending in 13::3), the GUA on the shared subnet between R1 and R3, as the next-hop address.

Example 29-8 *Static IPv6 Routes Using Global Unicast Addresses*

```
!
R1(config)# ipv6 route 2001:db8:1111:3::/64 2001:DB8:1111:13::3
```

The first two commands in Example 29-9 list detail about the static route configured in Example 29-8. The first command lists all static routes on Router R1, currently just the one static route per Example 29-8. The second command, **show ipv6 route 2001:DB8:1111:3::33**, asks R1 to identify the route it would use to forward packets to the address listed in the command (Host C in the figure uses address **2001:DB8:1111:3::33**). The output details the just-configured static route, proving that R1 uses this new static route when forwarding packets to that host.

Example 29-9 *Verification of Static Routes to a Next-Hop Global Unicast Address*

```
R1# show ipv6 route static
! Legend omitted for brevity
S   2001:DB8:1111:3::/64 [1/0]
     via 2001:DB8:1111:13::3
```

```
R1# show ipv6 route 2001:db8:1111:3::33
Routing entry for 2001:DB8:1111:3::/64
  Known via "static", distance 1, metric 0
  Route count is 1/1, share count 0
  Routing paths:
    2001:DB8:1111:13::3
      Route metric is 0, traffic share count is 1
      Last updated 00:07:43 ago
```

Interestingly, the **ipv6 route 2001:db8:1111:3::/64 2001:db8:1111:13::3** command in Example 29-8 works, but you might wonder how the router knows what outgoing interface to use because the command does not list it. The router does need to identify the outgoing interface, so it uses an iterative process by finding the route to reach the next-hop GUA. The first iteration happens when the router matches the packet's destination address to the static route. The second iteration occurs when the router finds the route to use when forwarding packets to the next-hop address in the first route. That second iteration matches the route for the subnet to which both the local router and the next-hop router connect—in this case, subnet prefix 2001.db8:1111:13::/64 connected to R1's G0/0/2.

Example 29-10 shows some notes about the iterative lookup to find the outgoing interface. The first command lists the connected route that Router R1 uses that matches next-hop GUA 2002:db8:1111:13::3, namely the route for prefix 2001:db8:1111:13::/64. But the **show ipv6 static detail** command shows the results of the router's iterative route lookup, listing interface G0/0/2 as the outgoing interface. It also lists a code of *, informing us that the router added the route to the Routing Information Base (RIB), referring to the IPv6 routing table.

Example 29-10 *Understanding the Iterative Lookup to Find the Outgoing Interface*

```
R1# show ipv6 route connected
! Lines omitted to reveal route to the subnet between R1 and R3
C   2001:DB8:1111:13::/64 [0/0]
     via GigabitEthernet0/0/2, directly connected

R1# show ipv6 static detail
IPv6 Static routes Table - default
Codes: * - installed in RIB, u/m - Unicast/Multicast only
Codes for []: P - permanent I - Inactive permanent
       U - Per-user Static route
       N - ND Static route
       M - MIP Static route
       P - DHCP-PD Static route
       R - RHI Static route
       V - VxLan Static route
*   2001:DB8:1111:3::/64 via 2001:DB8:1111:13::3, distance 1
     Resolves to 1 paths (max depth 1)
     via GigabitEthernet0/0/2
```

You can also configure a static IPv6 route with the next-hop GUA plus the outgoing interface (sometimes called a fully specified route). Doing so means that the router does not need to perform the iterative route lookup. Example 29-11 shows just such a case, replacing the static route in Example 29-8 with one that also lists both the next-hop GUA plus G0/0/2 as the outgoing interface. The end of the example confirms that the route listed in the IPv6 routing table lists the next-hop GUA and outgoing interface.

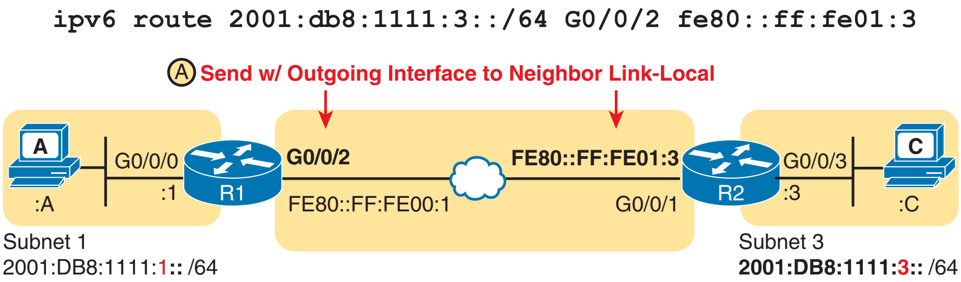
Example 29-11 *Alternate IPv6 Static Route with Outgoing Interface and GUA*

```
R1# show running-config | include ipv6 route
ipv6 route 2001:DB8:1111:3::/64 GigabitEthernet0/0/2 2001:DB8:1111:13::3

R1# show ipv6 route | section 2001:DB8:1111:3::/64
S   2001:DB8:1111:3::/64 [1/0]
     via 2001:DB8:1111:13::3, GigabitEthernet0/0/2

R1# show ipv6 static detail
! Legend omitted
*   2001:DB8:1111:3::/64 via 2001:DB8:1111:13::3, GigabitEthernet0/0/2, distance 1
```

Example Static Network Route with a Next-Hop LLA

Static routes that refer to a neighbor's LLA work a little like the preceding two styles of static routes. First, the **ipv6 route** command refers to a next-hop address, namely an LLA; however, the command must also refer to the router's local outgoing interface. Why both? The **ipv6 route** command cannot simply refer to an LLA next-hop address by itself because the iterative route lookup used with GUAs will not work with LLAs. There are no routes for LLAs, so there is no way for a router to use only the LLA as a static route's forwarding instructions. In short, if using the LLA, you also need to configure the outgoing interface.

Figure 29-4 depicts the logic for using both the LLA and outgoing interface to replace the example static route configured in Example 29-8 and explained with Examples 29-9 and 29-10. In this case, the R1 configuration refers to R1's outgoing interface and to Router R2's LLA. The figure also shows the global configuration command required on Router R1.



```
ipv6 route 2001:db8:1111:3::/64 G0/0/2 fe80::ff:fe01:3
```

Figure 29-4 *Example Using LLAs and Outgoing Interface for IPv6 Static Routes*

Example 29-12 shows the configuration of the LLA-based static route on Router R1, with some purposeful mistakes along the way. The first attempt with the **ipv6 route** command omits the outgoing interface, so IOS rejects the command. The second attempt shows the command after using the up-arrow to retrieve the command and then adding the outgoing interface (G0/0/2) to the end of the command, but that command uses incorrect syntax. The interface ID should *precede* the address. The final **ipv6 route** command shows the correct syntax with IOS accepting the command.

Example 29-12 *Static IPv6 Network Route Using LLA*

```
R1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# ipv6 route 2001:db8:1111:3::/64 fe80::ff:fe01:3
% Interface has to be specified for a link-local nexthop
R1(config)# ipv6 route 2001:db8:1111:3::/64 fe80::ff:fe00:2 g0/0/2
                                                            ^

% Invalid input detected at '^' marker.

R1(config)# ipv6 route 2001:db8:1111:3::/64 g0/0/2 fe80::ff:fe01:3
R1(config)#^Z
```

Example 29-13 verifies the configuration in Example 29-12 by repeating the **show ipv6 route static** and **show ipv6 route 2001:DB8:1111:3::33** commands used in Example 29-9. Note that the output from both commands differs slightly versus the earlier example. Because the new configuration commands list both the next-hop address and outgoing interface, the **show** commands in Example 29-13 also list both the next-hop (LLA) and the outgoing interface. If you refer back to Example 29-9, you will see only a next-hop address listed.

Example 29-13 *Verification of Static Routes to a Next-Hop Link-Local Address*

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

S   2001:DB8:1111:3::/64 [1/0]
      via FE80::FF:FE00:2, GigabitEthernet0/0/2

R1# show ipv6 route 2001:db8:1111:3::33
Routing entry for 2001:DB8:1111:3::/64
  Known via "static", distance 1, metric 0
  Route count is 1/1, share count 0
  Routing paths:
    FE80::FF:FE00:2, GigabitEthernet0/0/2
      Route metric is 0, traffic share count is 1
      Last updated 00:01:01 ago
```

In summary, IPv6 static network routes list a prefix/prefix-length along with forwarding instructions. The forwarding instructions allow for unicast address types—GUA, ULA, or even LLAs. However, there are some restrictions about which options you can use with forwarding instructions. Table 29-2 summarizes the options.

Key Topic

Table 29-2 Summary Table for IPv6 Network Route Configuration

Forwarding Instructions	Serial	Ethernet
Interface only	Yes	No*
GUA/ULA only	Yes	Yes
Interface plus GUA/ULA	Yes	Yes
Interface plus LLA	Yes	Yes
LLA only	No	No

* The **ipv6 route** command can be configured with an outgoing Ethernet interface but no next-hop address, but the route does not forward packets.

Static Default, Host, and Floating Static IPv6 Routes

If you have been reading this book sequentially, the following topics should be relatively easy. Engineers use IPv6 default, host, and floating static routes for the same purposes as their IPv4 cousins. The only difference comes in the combinations of how to configure the next-hop addresses and outgoing interfaces for IPv6, a topic you just finished learning about in this chapter. Hopefully, most of the following few pages will be both a review of IPv4 default, host, and floating static concepts, along with helpful examples for IPv6 routes.

Static IPv6 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 straightforward router forwarding logic when no routes match a packet's destination is

- 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 beneficial 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. When using a routing protocol in a large network, the branch router could learn thousands of routes, all of which forward packets 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 29-5 shows just such an example, with two sample branch routers on the right and a core site router on the left, with the default route shown as the dashed line.

To configure a static default route, use a specific value to note the route as a default route: ::/0. An entire prefix of a double colon (::) with no other digits represents an all 0s prefix, and the /0 prefix length signifies that zero of the initial bits must match to be part of the same group. So, ::/0 serves as the abbreviation of "all IPv6 addresses." This idea mirrors the IPv4 convention to refer to the default route as 0.0.0.0/0 in **show** command output or 0.0.0.0 0.0.0.0 in the **ip route** command.

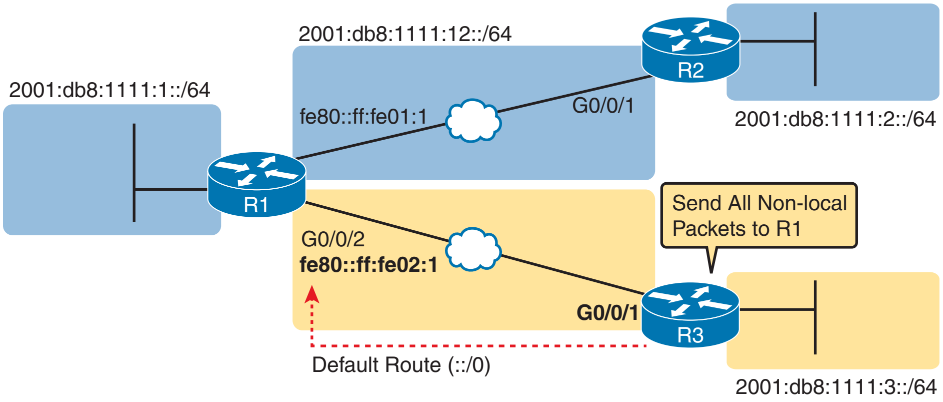


Figure 29-5 *Using Static Default Routes at Branch Routers*

To configure a static default route, use ::/0 as the destination, and otherwise, just configure the **ipv6 route** command as normal, with the same conventions about forwarding instructions as discussed earlier in this chapter. Example 29-14 shows one such sample static default route on Router R3 from Figure 29-5. The configuration uses next-hop LLA and, therefore, also requires the outgoing interface.

Example 29-14 *Static Default Route for Branch Router R3*

```
! Forward out R3's G0/0/1 local interface to R1's G0/0/2 LLA FE80::FF:FE02:1.
R3(config)# ipv6 route ::/0 g0/0/1 fe80::ff:fe02:1
```

Cisco IOS provides a simpler representation of IPv6 default routes compared to IPv4 default routes. For instance, the **show ipv6 route** command output does not list a "Gateway of Last Resort" as with IPv4. Instead, it lists the default route destination as ::/0, just as in the **ipv6 route** configuration command. Example 29-15 shows Router R3's routing table, with only the default route, plus the expected connected and local routes for its two working interfaces. The **show ipv6 route ::/0** command output, at the bottom of the example, shows more detail about the default route, including confirmation of the configured prefix of ::/0 and the next-hop LLA and outgoing interface.

Example 29-15 *Router R3's Static Default Route (Using Outgoing Interface)*

```
R3# show ipv6 route
IPv6 Routing Table - default - 6 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, R - RIP, H - NHRP, 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
       a - Application, m - OMP
S   ::/0 [1/0]
     via FE80::FF:FE02:1, GigabitEthernet0/0/1
```

```
C   2001:DB8:1111:3::/64 [0/0]
     via GigabitEthernet0/0/0, directly connected
L   2001:DB8:1111:3::3/128 [0/0]
     via GigabitEthernet0/0/0, receive
C   2001:DB8:1111:13::/64 [0/0]
     via GigabitEthernet0/0/1, directly connected
L   2001:DB8:1111:13::3/128 [0/0]
     via GigabitEthernet0/0/1, receive
L   FF00::/8 [0/0]
     via Null0, receive

R3# show ipv6 route ::/0
Routing entry for ::/0
  Known via "static", distance 1, metric 0
  Route count is 1/1, share count 0
  Routing paths:
    FE80::FF:FE02:1, GigabitEthernet0/0/1
      Route metric is 0, traffic share count is 1
      Last updated 00:04:04 ago
```

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.

As for the syntax, an **IPv6 host route** follows the same rules as an IPv6 network route regarding the forwarding instructions. The only difference comes in the configuration of a full 128-bit address along with a prefix length of /128 so that the route matches a single IPv6 address.

You might never need to use a host route in a real network, but the motivation usually comes from some business reason to forward packets to one host over a different route that the routing protocol would choose. Figure 29-6 shows just such an example. The route for all hosts in the subnet flows over the lower part of the topology; however, you might have business reasons to forward traffic sent to host D over the upper path through Router R2, requiring a host route.

Example 29-16 shows the configuration based on Figure 29-6, with the host route for host D using R2 as the next-hop router, and a static network route for host D's subnet using Router R3 as the next-hop router.

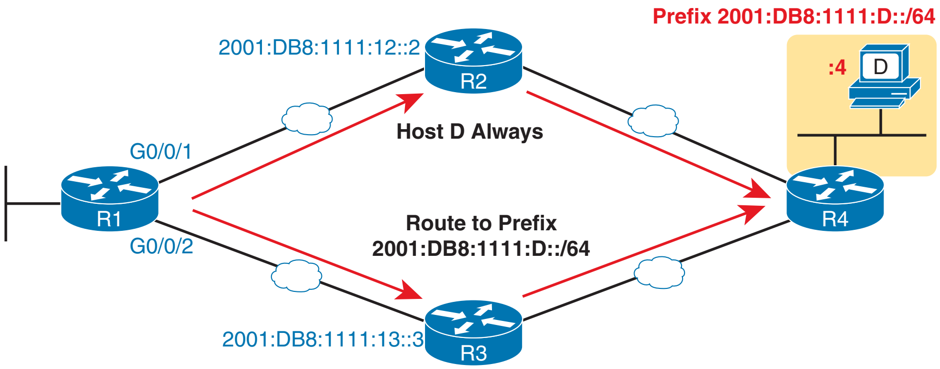


Figure 29-6 *An Example Topology for an IPv6 Host Route*

Example 29-16 *Static Host IPv6 Routes on R1, for Host D*

```
! The first command shows the host route, and the second shows the network route.
! Both use next-hop GUA and outgoing interface for clarity.
R1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# ipv6 route 2001:db8:1111:d::4/128 g0/0/1 2001:db8:1111:12::2
R1(config)# ipv6 route 2001:db8:1111:d::/64 g0/0/2 2001:db8:1111:13::3
R1(config)#^Z
R1#
```

Adding IPv6 host routes typically causes overlapping routes, as seen in this case. When choosing to place routes into the IPv6 routing table, a router always attempts to place the best route for each prefix (subnet) into the IP routing table. That rule applies to each subnet prefix, defined by each combination of prefix and prefix length. In this case, the two static routes have different prefix lengths, so it places both routes into the routing table, as seen in the output from the first command in Example 29-17.

Example 29-17 *IPv6 Host Route for Host D with Overlapping Network Route*

```
R1# show ipv6 route static
! Legend omitted
S   2001:DB8:1111:D::/64 [1/0]
     via 2001:DB8:1111:13::3, GigabitEthernet0/0/2
S   2001:DB8:1111:D::4/128 [1/0]
     via 2001:DB8:1111:12::2, GigabitEthernet0/0/1

R1# show ipv6 route 2001:db8:1111:d::4
Routing entry for 2001:DB8:1111:D::4/128
  Known via "static", distance 1, metric 0
  Route count is 1/1, share count 0
  Routing paths:
    2001:DB8:1111:12::2, GigabitEthernet0/0/1
      Route metric is 0, traffic share count is 1
      Last updated 00:02:28 ago

R1# show ipv6 route 2001:db8:1111:d::3
Routing entry for 2001:DB8:1111:D::/64
  Known via "static", distance 1, metric 0
  Route count is 1/1, share count 0
  Routing paths:
    2001:DB8:1111:13::3, GigabitEthernet0/0/2
      Route metric is 0, traffic share count is 1
      Last updated 00:00:43 ago
```

As with IPv4 routes, IPv6 matches a packet to the IPv6 routing table using longest-match logic. Packets sent to host D's address match both routes configured in Example 29-16, but the router chooses the longer prefix (more binary 1s in the prefix length) /128 host route. Stated differently, when matching more than one route, the router uses the route that defines the most specific set of addresses.

The final two **show** commands in Example 29-17 demonstrate Router R1's matching logic. The **show ipv6 route 2001:db8:1111:d::4** command asks the router what route it would use to match packets sent to host D's address, and the output describes the /128 host route, with R2's GUA as the next-hop address. Conversely, the final command lists a different IPv6 address in the same subnet as host D, so R1 will not match the host route. Instead, it matches the network route, listing outgoing interface G0/0/2 with R3's GUA as next-hop address.

Floating Static IPv6 Routes

Floating static routes are the last of the four specific static route types mentioned in the CCNA exam topics. The general concept works the same for both IPv4 and IPv6 floating static routes, so this section will provide a brief introduction. For more detail on the concepts, please refer to the section "Floating Static Routes" in Chapter 17, "Configuring IPv4 Addresses and Static Routes."

Floating static routes let a router direct traffic over a switched WAN connection, for instance, over a cellular interface—but only when the normally working WAN link fails. For instance, in Figure 29-7, the Ethernet WAN link typically works for months. OSPF learns IPv6 routes using that link. During those same times, the cellular interfaces remain unused, with no cellular call/link between the routers, and no IPv6 routes that use the cellular interfaces.

Floating static routes provide a method to direct packets over the cellular link, which triggers a phone call to create a link between the two routers—but only when the routes that use the primary link fail. To do so, you configure a static route that uses the cellular interface, and the router stores that route in its configuration. However, the router does not add the route to the routing table because of the existing OSPF-learned route for the exact same prefix/length. Once the primary routes fail, the routers add the floating static routes to their routing tables, directing traffic over the cellular link. In this scenario, the cellular link and the floating static route get used only during outages to the primary link.

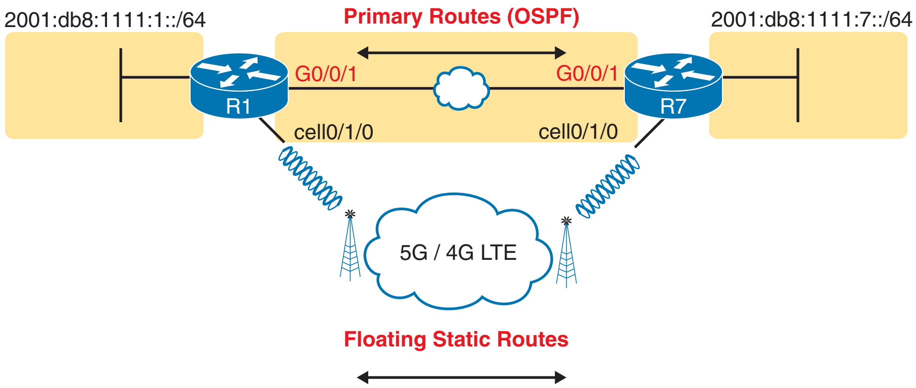


Figure 29-7 *Using a Floating Static Route to Key Subnet 2001:DB8:1111:7::/64*

Routers rely on each route's **IPv6 administrative distance** (AD) parameter to judge between different sources of routing information (for example, OSPF, EIGRP, or static routes) for routes for the same prefix/length. For instance, in Figure 29-7, an OSPF-learned route on R1, for prefix 2001:db8:1111:7::/64, would have a default AD of 110. Static routes default to an AD of 1, so Router R1 would normally prefer the static route configured with the **ipv6 route 2001:db8:1111:7::/64 cell0/1/0** command versus the OSPF-learned AD 110 route to that same subnet, and replace the OSPF-learned route with the static route.

Floating static routes remain out of the routing table by using a higher (worse) AD setting than the normally used routes. In the recent example, by setting the static route's AD to something higher than OSPF's 110—for instance, with the **ipv6 route 2001:db8:1111:7::/64 cell0/1/0 130** command—Router R1 will instead prefer the AD 110 OSPF-learned route instead of the AD 130 static route.

You can find the AD of a route in the routing table with the usual **show** commands. For instance, the **show ipv6 route** command lists each route's administrative distance as the first of the two numbers inside the brackets. The second number in brackets is the metric. Also, commands like **show ipv6 route** *prefix/length* give detail about one specific route; this command lists the AD simply as "distance."

Table 29-3 lists the various default IPv6 AD values. These defaults mirror the same default AD settings for IPv4 as seen in Table 24-4 in this book. Compared to Table 24-4, this table adds the entry for NDP (which does not apply to IPv4) and removes the mention of the IPv4 DHCP-based default route (AD 254) because IPv6 does not advertise default routes with DHCP.

Key Topic

Table 29-3 Default IPv6 Administrative Distances

Route Type	Administrative Distance
Connected	0
Static	1
NDP default route	2
BGP (external routes [eBGP])	20
EIGRP (internal routes)	90
IGRP	100

Route Type	Administrative Distance
OSPF	110
IS-IS	115
RIP	120
EIGRP (external routes)	170
BGP (internal routes [iBGP])	200
Unusable	255

Troubleshooting Static IPv6 Routes

IPv6 static routes have the same potential issues and mistakes as do static IPv4 routes, as discussed in Chapter 17. 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 router adds the route to the routing table but the route does not work, and cases in which the router does not add the route to 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, next-hop address, or 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, a static route that references a next-hop address should list the address of some other router on a subnet that connects the two routers. Imagine Router R1 uses an address 2001:1:1:1::1 on that subnet, with Router R2 using 2001:1:1:1::2. R1's **ipv6 route** command ought to refer to Router R2' 2001:1:1:1::2 address, but nothing prevents the mistake of configuring Router R1's 2001:1:1:1::1 address or any other address in that subnet other than 2001:1:1:1::/64. Router R1 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. Mentally review this list of questions that confirm correct configuration and reveal common mistakes, ensuring an answer of yes to each relevant question.

Key Topic

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

Step 2. **Next-hop LLA:** If configuring a next-hop LLA:

a. Does the command also include an outgoing interface?

- **b** Is the outgoing interface the correct interface (on the router on which the command is configured) based on the topology?
- **c.** Is the configured LLA the LLA of the neighboring router on the same link as the listed outgoing interface?

Step 3. **Next-hop GUA/ULA only:** If configuring a next-hop GUA or ULA but no outgoing interface:

- **a.** Does the configured GUA/ULA refer to the neighbor's address, specifically on the neighbor's interface that connects to the local router?
- **b.** Does the local router have a working route that matches the next-hop GUA/ULA? (Usually a connected route.)

Step 4. **Outgoing Interface only:** If configuring an outgoing interface but no next-hop address:

- **a.** Is the outgoing interface a serial interface—the only interface type that works in the **ipv6 route** command without also configuring a next-hop address?
- **b.** Is the outgoing interface the correct interface (on the router on which the command is configured) based on the topology?

The preceding 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 29-8 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 (note that this network uses different IPv6 addresses than the earlier figures).

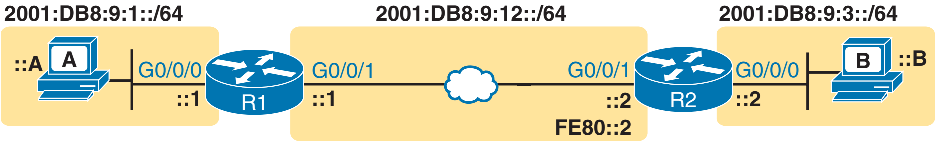


Figure 29-8 *Sample Topology for Incorrect IPv6 Route Examples*

Example 29-18 shows several **ipv6 route** commands. Of the commands shown, IOS rejects the second command in the example (step 2A), but it accepts and adds routes to the routing table for the others. However, the routes in the accepted commands all have some issue that prevents the route from working as designed. Look for the short comment at the end of each configuration command to see why each is incorrect, and the list following the example for more detail.

Example 29-18 **ipv6 route** *Commands with Correct Syntax but Incorrect Ideas*

```
ipv6 route 2001:DB8:9:33::/64 2001:DB8:9:12::2 ! Step 1: Wrong prefix
ipv6 route 2001:DB8:9:3::/64 FE80::2           ! Step 2A: Missing outgoing interface
ipv6 route 2001:DB8:9:3::/64 G0/0/0 FE80::2    ! Step 2B: Wrong interface on R1
ipv6 route 2001:DB8:9:3::/64 G0/0/1 FE80::1    ! Step 2C: Wrong neighbor link local
ipv6 route 2001:DB8:9:3::/64 2001:DB8:9:12::3  ! Step 3A: Wrong neighbor address
ipv6 route 2001:DB8:9:3::/64 G0/0/1            ! Step 4A: Also needs next-hop
```

Step 1. The prefix (2001:DB8:9:33::) has a typo in the fourth quartet (33 instead of 3). IOS accepts the command and now has a route for a nonexistent subnet.

Step 2A. The command uses the correct LLA as the next-hop address but fails to include any outgoing interface. IOS rejects the command.

Step 2B. The command uses the correct R2 LLA on the R1-R2 link (FE80::2 per the figure) but uses the incorrect R1 local interface of G0/0/0. (See the next example for more detail.)

Step 2C. The figure shows R2's G0/0/1 as using link-local address FE80::2, but the command uses FE80::1, R1's LLA.

Step 3A. The figure shows the subnet in the center as 2001:DB8:9:12::/64, with R1 using the ::1 address and R2 using ::2. This command uses a next-hop address ending in ::3 in that subnet that neither router uses.

Step 4A. The command lists only an outgoing interface of G0/0/1 (the correct interface), but no next-hop address. Only serial interfaces can be used with this syntax and have the route function correctly. Because it uses an Ethernet outgoing interface, the router needs the next-hop address in the command.

The key takeaway for this section is to know that a route in the IPv6 routing table might be incorrect due to poor choices for the parameters. The fact that a route is in the IPv6 routing table, particularly a static route, does not mean that it is correct.

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 has a second step in which it considers whether to add that route 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 GUA or ULA (that is, not an LLA), 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 than the competing route.

Note that the first two bullet items can change even without any configuration. For instance, as interfaces fail and recover, the interface state will change and the connected IPv6 routes will be removed and reappear in the IPv6 routing table. Depending on the current status, the static routes that rely on those interfaces and routes will be added and removed from the IPv6 routing table.

Chapter Review

One key to doing well on the exams is to perform repetitive spaced review sessions. Review this chapter's material using either the tools in the book or interactive tools for the same material found on the book's companion website. Refer to the "Your Study Plan" element for more details. Table 29-4 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 29-4 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
Watch video		Website

Review All the Key Topics

Key Topic

Table 29-5 Key Topics for Chapter 29

Key Topic Element	Description	Page Number
List	Methods by which a router can build IPv6 routes	724
List	Rules for IPv6 connected and local routes	725
Figure 29-2	IPv6 static route concepts	729
Table 29-2	Summary of IPv6 network route configuration	735
Table 29-3	IPv6 default administrative distance values	740
Checklist	List of reasons for IPv6 static route problems that result in a route in the routing table but the route does not work correctly	741
Checklist	List of reasons for IPv6 static route problems that result in the route not appearing in the routing table	743

Key Terms You Should Know

IPv6 administrative distance, IPv6 host route, IPv6 local route, local route

Command References

Tables 29-6 and 29-7 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 29-6 Chapter 29 Configuration Command Reference

Command	Description
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-lla*	Global command to define an IPv6 static route, with a next-hop link-local address and the required local router outgoing interface
ipv6 route *prefix/length [outgoing-interface] next-hop-gua*	Global command to define an IPv6 static route, with the next-hop global unicast address and the optional local router outgoing interface
ipv6 route ::/0 *{[next-hop-address] [outgoing-interface]}*	Global command to define a default IPv6 static route

Table 29-7 Chapter 29 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 router's route to forward packets to the IPv6 address listed in the command. If no route matches, it displays "No route found."
show ipv6 route *prefix/length*	If a route that matches the listed prefix/length exists, the router displays detailed information about the route.
show ipv6 route | section *prefix/length*	Displays an excerpt of the output from the **show ipv6 route** command of only the lines related to routes for the listed prefix/length.
show ipv6 static detail	Displays a static route legend with multiple output lines about each static route, noting whether it is installed in the routing table (RIB) and the administrative distance.

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		
Use Per-Chapter Interactive Review		

Repeat All DIKTA Questions

For this task, use the PTP 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 or with Cisco Packet Tracer in about 15 minutes. To find them, open https://www.certskills.com and look under the Labs menu item.

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 Video

The companion website includes a variety of common mistake and Q&A videos organized by part and chapter. Use these videos to challenge your thinking, dig deeper, review topics, and better prepare for the exam. Make sure to bookmark a link to the companion website and use the videos for review whenever you have a few extra minutes.