CHAPTER 17

IP Version 6

IP version 6 (IPv6), the replacement protocol for IPv4, is well known for a couple of reasons. IPv6 provides the ultimate solution for the problem of running out of IPv4 addresses in the global Internet by using a 128-bit address—approximately 10^{38} total addresses, versus the mere (approximate) $4*10^{9}$ total addresses in IPv4. However, IPv6 has been the ultimate long-term solution for over ten years, in part because the interim solutions, including Network Address Translation/Port Address Translation (NAT/PAT), have thankfully delayed the day in which we truly run out of public unicast IP addresses.

This chapter focuses on IPv6 addressing and routing, in part because the primary motivation for the eventual migration to IPv6 is to relieve the address constraints of IPv4. This chapter also briefly introduces some of the other features of IPv6, as well as explains some of the reasons for the need for IPv6.

"Do I Know This Already?" Quiz

The "Do I Know This Already?" quiz allows you to assess whether you should read the entire chapter. If you miss no more than one of these nine self-assessment questions, you might want to move ahead to the section "Exam Preparation Tasks." Table 17-1 lists the major headings in this chapter and the "Do I Know This Already?" quiz questions covering the material in those headings so that you can assess your knowledge of these specific areas. The answers to the "Do I Know This Already?" quiz appear in Appendix A.

Table 17-1 *"Do I Know This Already?" Foundation Topics Section-to-Question Mapping*

Foundation Topics Section	Questions
Global Unicast Addressing, Routing, and Subnetting	1, 2
IPv6 Protocols and Addressing	3–5
Configuring IPv6 Routing and Routing Protocols	6–8
IPv6 Transition Options	9

1. Which of the following is the most likely organization from which an enterprise could obtain an administrative assignment of a block of IPv6 global unicast IP addresses?
   a. An ISP
   b. ICANN
   c. An RIR
   d. Global unicast addresses are not administratively assigned by an outside organization.
2. Which of the following is the shortest valid abbreviation for FE80:0000:0000:0100:0000:0000:0000:0123?
   a. FE80::100::123
   b. FE8::1::123
   c. FE80::100:0:0:0:123:4567
   d. FE80:0:0:100::123
3. Which of the following answers lists a multicast IPv6 address?
   a. 2000::1:1234:5678:9ABC
   b. FD80::1:1234:5678:9ABC
   c. FE80::1:1234:5678:9ABC
   d. FF80::1:1234:5678:9ABC
4. Which of the following answers list either a protocol or function that can be used by a host to dynamically learn its own IPv6 address?
   a. Stateful DHCP
   b. Stateless DHCP
   c. Stateless autoconfiguration
   d. Neighbor Discovery Protocol
5. Which of the following help allow an IPv6 host to learn the IP address of a default gateway on its subnet?
   a. Stateful DHCP
   b. Stateless RS
   c. Stateless autoconfiguration
   d. Neighbor Discovery Protocol

6. Which of the following are routing protocols that support IPv6?

   a. RIPng

   b. RIP-2

   c. OSPFv2

   d. OSPFv3

   e. OSPFv4

7. In the following configuration, this router's Fa0/0 interface has a MAC address of 4444.4444.4444. Which of the following IPv6 addresses will the interface use?

```
ipv6 unicast-routing
ipv6 router rip tag1
interface FastEthernet0/0
 ipv6 address 3456::1/64
```

   a. 3456::C444:44FF:FE44:4444

   b. 3456::4444:44FF:FE44:4444

   c. 3456::1

   d. FE80::1

   e. FE80::6444:44FF:FE44:4444

   f. FE80::4444:4444:4444

8. In the configuration text in the previous question, RIP was not working on interface Fa0/0. Which of the following configuration commands would enable RIP on Fa0/0?

   a. **network 3456::/64**

   b. **network 3456::/16**

   c. **network 3456::1/128**

   d. **ipv6 rip enable**

   e. **ipv6 rip tag1 enable**

9. Which of the following IPv4-to-IPv6 transition methods allows an IPv4-only host to communicate with an IPv6-only host?

   a. Dual-stack

   b. 6to4 tunneling

   c. ISATAP tunneling

   d. NAT-PT

Foundation Topics

The world has changed tremendously over the last 10–20 years as a result of the growth and maturation of the Internet and networking technologies in general. Twenty years ago, no global network existed to which the general populace could easily connect. Ten years ago, the public Internet had grown to the point where people in most parts of the world could connect to the Internet, but with most Internet users being the more computer-savvy people. Today, practically everyone seems to have access, through their PCs, handheld devices, phones, or even the refrigerator.

The eventual migration to IPv6 will likely be driven by the need for more addresses. Practically every mobile phone supports Internet traffic, requiring the use of an IP address. Most new cars have the ability to acquire and use an IP address, along with wireless communications, allowing the car dealer to contact the customer when the car's diagnostics detect a problem with the car. Some manufacturers have embraced the idea that all their appliances need to be IP enabled.

Besides the sheer growth in the need for IPv4 addresses, edicts from governmental agencies could drive demand for IPv6. As of this writing, the U.S. government had set a date in 2008 by which all government agencies should be running IPv6 in their core IP networks. Such initiatives can help drive adoption of IPv6.

While the two biggest reasons why networks might migrate to IPv6 are the need for more addresses and mandates from government organizations, at least IPv6 includes some attractive features and migration tools. Some of those advantages are as follows:

- **Address assignment features:** IPv6 address assignment allows easier renumbering, dynamic allocation, and recovery of addresses, with nice features for mobile devices to move around and keep their IP address (thereby avoiding having to close and reopen an application).
- **Aggregation:** IPv6's huge address space makes for much easier aggregation of blocks of addresses in the Internet.
- **No need for NAT/PAT:** Using publicly registered unique addresses on all devices removes the need for NAT/PAT, which also avoids some of the application layer and VPN-tunneling issues caused by NAT.
- **IPsec:** IPsec works with both IPv4 and IPv6, but it is required on IPv6 hosts, so you can rely on support for IPsec as needed for VPN tunneling.

- **Header improvements:** While it might seem like a small issue, the IPv6 header improves several things compared to IPv4. In particular, routers do not need to recalculate a header checksum for every packet, reducing per-packet overhead. Additionally, the header includes a flow label that allows easy identification of packets sent over the same single TCP or User Datagram Protocol (UDP) connection.
- **Transition tools:** As is covered in the last major section of this chapter, IPv6 has many tools to help with the transition from IPv4 to IPv6.

The worldwide migration from IPv4 to IPv6 will not be an event, or even a year on the calendar. Rather, it will be a long process, a process that has already begun. Network engineers have a growing need to learn more about IPv6. This chapter covers the basics of IPv6, ending with some discussions about the issues of living in a world in which both IPv4 and IPv6 will likely coexist for quite a long time.

NOTE *Information Week* (http://www.informationweek.com) published an interesting article about the need to migrate to IPv6, around the time this book was being completed. To see the article, search the website for the article "The Impending Internet Address Shortage."

Global Unicast Addressing, Routing, and Subnetting

One of the original design goals for the Internet was that all organizations would register and be assigned one or more public IP networks (Class A, B, or C). By registering to use a particular public network number, the company or organization using that network was assured by the numbering authorities that no other company or organization in the world would be using the addresses in that network. As a result, all hosts in the world would have globally unique IP addresses.

From the perspective of the Internet infrastructure, in particular the goal of keeping Internet routers' routing tables from getting too large, assigning an entire network to each organization helped to some degree. The Internet routers could ignore all subnets, instead having a route for each classful network. For example, if a company registered and was assigned Class B network 128.107.0.0/16, the Internet routers just needed one route for that entire network.

Over time, the Internet grew tremendously. It became clear by the early 1990s that something had to be done, or the growth of the Internet would grind to a halt when all the public IP networks were assigned, and no more existed. Additionally, the IP routing tables in Internet routers were becoming too large for the router technology of that day. So, the Internet community worked together to come up with both some short-term and long-term solutions to two problems: the shortage of public addresses and the size of the routing tables.

The short-term solutions included a much smarter public address assignment policy, where public addresses were not assigned as only Class A, B, and C networks, but as smaller subdivisions (prefixes), reducing waste. Additionally, the growth of the Internet routing tables was reduced by smarter assignment of the address ranges. For example, assigning the Class C networks that begin with 198 to only a particular Internet service provider (ISP) in a particular part of the world allowed other ISPs to use one route for 198.0.0.0/8—in other words, all addresses that begin with 198—rather than a route for each of the 65,536 different Class C networks that begin with 198. Finally, NAT/PAT achieved amazing results by allowing a typical home or small office to consume only one public IPv4 address, greatly reducing the need for public IPv4 addresses.

The ultimate solution to both problems is IPv6. The sheer number of IPv6 addresses takes care of the issue of running out of addresses. The address assignment policies already used with IPv4 have been refined and applied to IPv6, with good results for keeping the size of IPv6 routing tables smaller in Internet routers. The following sections provide a general discussion of both issues, in particular how global unicast addresses, along with good administrative choices for how to assign IPv6 address prefixes, aid in routing in the global Internet. These sections conclude with a discussion of subnetting in IPv6.

Global Route Aggregation for Efficient Routing

By the time IPv6 was being defined in the early 1990s, it was clear that thoughtful choices about how to assign the public IPv4 address space could help with the efficiency of Internet routers by keeping their routing tables much smaller. By following those same well-earned lessons, IPv6 public IP address assignment can make for even more efficient routing as the Internet migrates to IPv6.

The address assignment strategy for IPv6 is elegant, but simple, and can be roughly summarized as follows:

- Public IPv6 addresses are grouped (numerically) by major geographic region.
- Inside each region, the address space is further subdivided by ISP inside that region.
- Inside each ISP in a region, the address space is further subdivided for each customer.

The same organizations handle this address assignment for IPv6 as for IPv4. The Internet Corporation for Assigned Network Numbers (ICANN, http://www.icann.org) owns the process. ICANN assigns one or more IPv6 address ranges to each Regional Internet Registry (RIR), of which five exist at the time of publication, roughly covering North America, Central/South America, Europe, Asia/Pacific, and Africa. These RIRs then subdivide their assigned address space into smaller portions, assigning prefixes to different

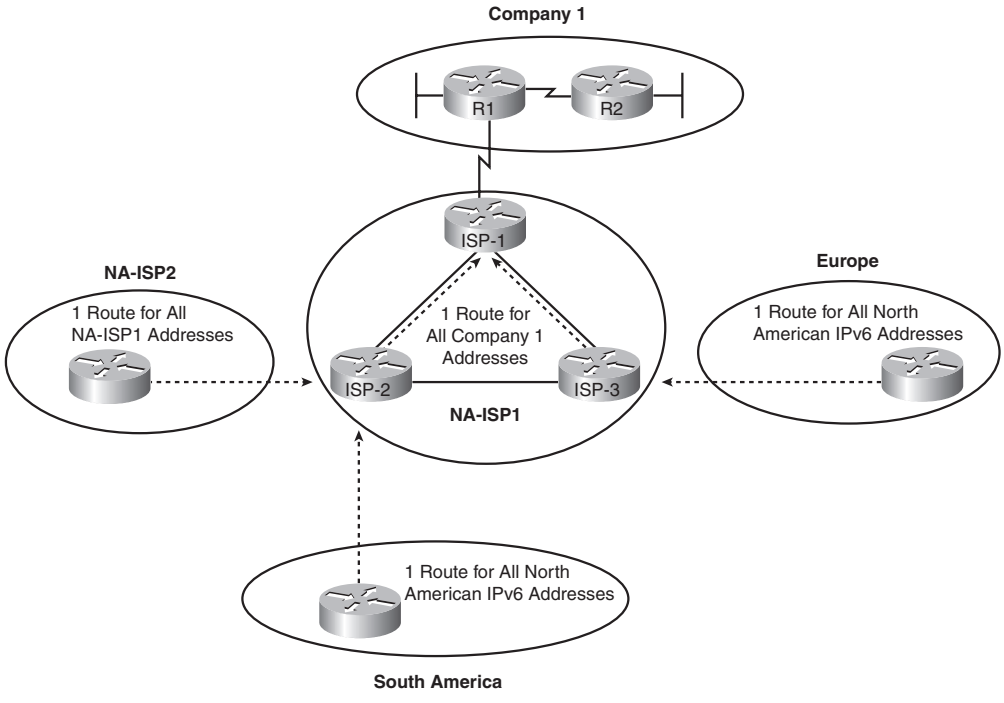
ISPs and other smaller registries, with the ISPs then assigning even smaller ranges of addresses to their customers.

> **NOTE** The Internet Assigned Numbers Authority (IANA) formerly owned the address assignment process, but it was transitioned to ICANN.

The IPv6 global address assignment plan results in more efficient routing, as shown in Figure 17-1. The figure shows a fictitious company (Company1) that has been assigned an IPv6 prefix by a fictitious ISP, NA-ISP1 (standing for North American ISP number 1). The figure lists the American Registry for Internet Numbers (ARIN), which is the RIR for North America.

Figure 17-1 *Conceptual View of IPv6 Global Routes*

Key Topic



As shown in the figure, the routers installed by ISPs in other major geographies of the world can have a single route that matches all IPv6 addresses in North America. While hundreds of ISPs might be operating in north America, and hundreds of thousands of enterprise customers of those ISPs, and tens of millions of individual customers of those ISPs, all the public IPv6 addresses can be from one (or a few) very large address blocks—requiring only one (or a few) routes on the Internet routers in other parts of the world. Similarly, routers

inside other ISPs in North America (for example, NA-ISP2, indicating North American ISP number 2 in the figure) can have one route that matches all address ranges assigned to NA-ISP2. And the routers inside NA-ISP1 just need to have one route that matches the entire address range assigned to Company1, rather than needing to know about all the subnets inside Company1.

Besides keeping the routers' routing table much smaller, this process also results in fewer changes to Internet routing tables. For example, if NA-ISP1 signed a service contract with another enterprise customer, NA-ISP1 could assign another prefix inside the range of addresses already assigned to NA-ISP1 by ARIN. The routers outside NA-ISP1's network—the majority of the Internet—do not need to know any new routes, because their existing routes already match the address range assigned to the new customer. The NA-ISP2 routers (another ISP) already have a route that matches the entire address range assigned to NA-ISP1, so they do not need any more routes. Likewise, the routers in ISPs in Europe and South America already have a route that works as well.

While the general concept might not be too difficult, a specific example can help. Before seeing a specific example, however, it helps to know a bit about how IPv6 addresses and prefixes are written.

Conventions for Representing IPv6 Addresses

IPv6 conventions use 32 hexadecimal numbers, organized into 8 quartets of 4 hex digits separated by a colon, to represent a 128-bit IPv6 address. For example:

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

Each hex digit represents 4 bits, so if you want to examine the address in binary, the conversion is relatively easy if you memorize the values shown in Table 17-2.

Table 17-2 *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

Writing or typing 32 hexadecimal digits, while more convenient than doing the same with 128 binary digits, can still be a pain. To make things a little easier, two conventions allow you to shorten what must be typed for an IPv6 address:

- Omit the leading 0s in any given quartet.
- Represent 1 or more consecutive quartets of all hex 0s with a double colon (::), but only for one such occurrence in a given address.

Key Topic

NOTE For IPv6, a quartet is one set of 4 hex digits in an IPv6 address. Eight quartets are in each IPv6 address.

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

FE00:**0000**:**0000**:**000**1:**0000**:**0000**:**0000**:**00**56

This address has two different locations in which one or more quartets have 4 hex 0s, so two main options exist for abbreviating this address, using the :: abbreviation in one or the other location. The following two options show the two briefest valid abbreviations:

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

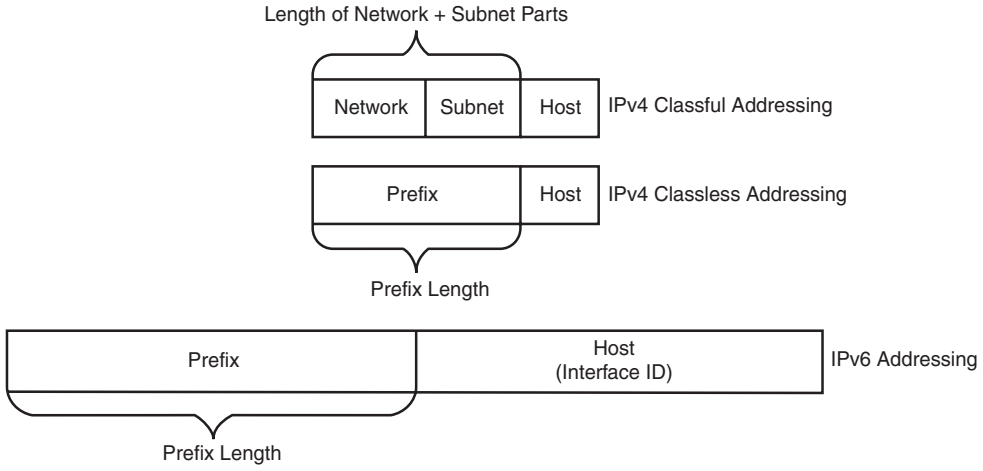
In particular, note that the :: abbreviation, meaning "one or more quartets of all 0s," cannot be used twice, because that would be ambiguous. So, the abbreviation FE00::1::56 would not be valid.

Conventions for Writing IPv6 Prefixes

IPv6 prefixes represent a range or block of consecutive IPv6 addresses. The number that represents the range of addresses, called a *prefix*, is usually seen in IP routing tables, just like you see IP subnet numbers in IPv4 routing tables.

Before examining IPv6 prefixes in more detail, it is helpful to review a few terms used with IPv4. IPv4 addresses can be analyzed and understood using either *classful addressing* rules or *classless addressing* rules. (This book and *CCENT/CCNA ICND1 Official Exam Certification Guide* both use classful terminology for the most part.) Classful addressing means that the analysis of an IP address or subnet includes the idea of a classful network number, with a separate network part of the address. The top part of Figure 17-2 reviews these concepts.

Figure 17-2 *IPv4 Classless and Classful Addressing, and IPv6 Addressing*



Thinking about IPv4 addressing as classful addresses helps to fully understand some issues in networking. With classful addressing, for example, the written value 128.107.3.0/24 means 16 network bits (because the address is in a Class B network) and 8 host bits (because the mask has 8 binary 0s), leaving 8 subnet bits. The same value, interpreted with classless rules, means prefix 128.107.3.0, prefix length 24. Same subnet/prefix, same meaning, same router operation, same configuration—it's just two different ways to think about the meaning of the numbers.

IPv6 uses a classless view of addressing, with no concept of classful addressing. Like IPv4, IPv6 prefixes list some value, a slash, and then a numeric prefix length. Like IPv4 prefixes, the last part of the number, beyond the length of the prefix, is represented by binary 0s. And finally, IPv6 prefix numbers can be abbreviated with the same rules as IPv4 addresses. 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 value represents the full 128-bit IP address; in fact, you have no opportunities to abbreviate this address. However, the /64 means that the prefix (subnet) in which this address resides is the subnet that includes all addresses that begin with the same first 64 bits as the address. Conceptually, it is the same logic as an IPv4 address. For example, address 128.107.3.1/24 is in the prefix (subnet) whose first 24 bits are the same values as address 128.107.3.1.

Like with IPv4, when writing or typing a prefix, the bits past the end of the prefix length are all binary 0s. In the IPv6 address shown previously, the prefix in which the address resides would be as follows:

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

When abbreviated, this would be:

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

Next, one last fact about the rules for writing prefixes before seeing some examples and moving on. If the prefix length is not a multiple of 16, the boundary between the prefix and the host part of the address is inside a quartet. In such cases, the prefix value should list all the values in the last octet in the prefix part of the value. For example, if the address just shown with a /64 prefix length instead had a /56 prefix length, the prefix would include all the first 3 quartets (a total of 48 bits), plus the first 8 bits of the fourth octet. The last 8 bits (last 2 hex digits) of the fourth octet should now be binary 0s. So, by convention, the rest of the fourth octet should be written, after being set to binary 0s, as follows:

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

The following list summarizes some key points about how to write IPv6 prefixes:

- The prefix has the same value as the IP addresses in the group for the first number of bits, as defined by the prefix length.
- Any bits after the prefix-length number of bits are binary 0s.
- The prefix can be abbreviated with the same rules as IPv6 addresses.
- If the prefix length is not on a quartet boundary, write down the value for the entire quartet.

Examples can certainly help a lot in this case. Table 17-3 shows several sample prefixes, their format, and a brief explanation.

Table 17-3 *Example IPv6 Prefixes and Their Meanings*

Prefix	Explanation	Incorrect Alternative
2000::/3	All addresses whose first 3 bits are equal to the first 3 bits of hex number 2000 (bits are 001)	2000/3 (omits ::) 2::/3 (omits the rest of the first quartet)
2340:1140::/26	All addresses whose first 26 bits match the listed hex number	2340:114::/26 (omits the last digit in the second quartet)
2340:1111::/32	All addresses whose first 32 bits match the listed hex number	2340:1111/32 (omits ::)

Almost as important to this convention is to note which options are not allowed. For example, 2::/3 is not allowed instead of 2000::/3, because it omits the rest of the octet, and

a device could not tell whether 2::/3 means "hex 0002" or "hex 2000." Only leading 0s in a quartet, and not trailing 0s, can be omitted when abbreviating an IPv6 address or prefix.

Now that you understand a few of the conventions about how to represent IPv6 addresses and prefixes, a specific example can show how ICANN's IPv6 global unicast IP address assignment strategy can allow the easy and efficient routing shown back in Figure 17-1.

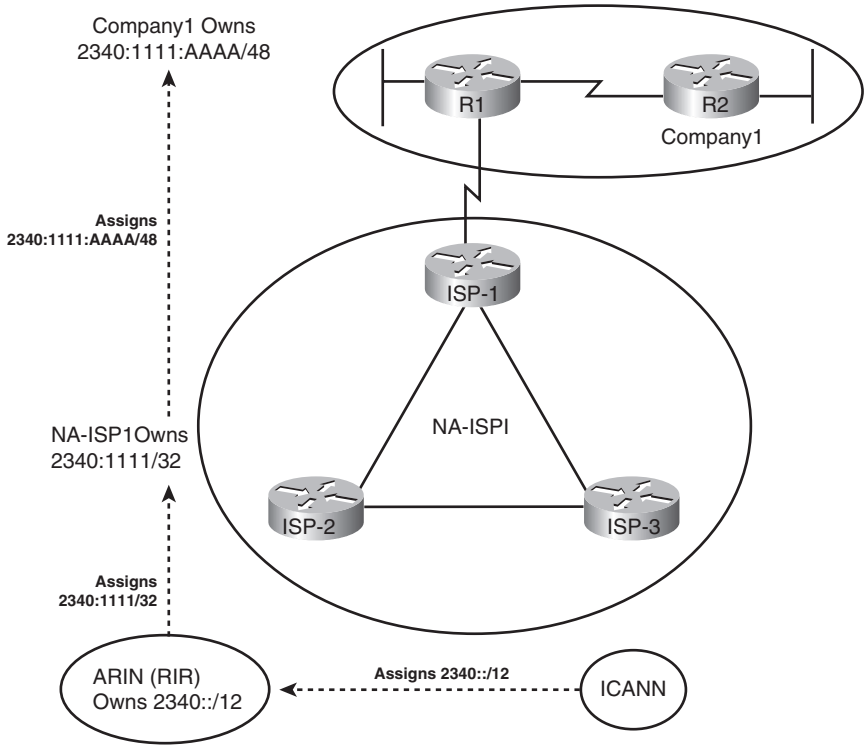
Global Unicast Prefix Assignment Example

IPv6 standards reserve the 2000::/3 prefix—which, when interpreted more fully, means all addresses that begin with binary 001 or either a hex 2 or 3—as global unicast addresses. Global unicast addresses are addresses that have been assigned as public and globally unique IPv6 addresses, allowing hosts using those addresses to communicate through the Internet without the need for NAT. In other words, these addresses fit the purest design for how to implement IPv6 for the global Internet.

Figure 17-3 shows an example set of prefixes that could result in a company (Company1) being assigned a prefix of 2340:1111:AAAA::/48.

Key Topic

Figure 17-3 *Example IPv6 Prefix Assignment in the Internet*

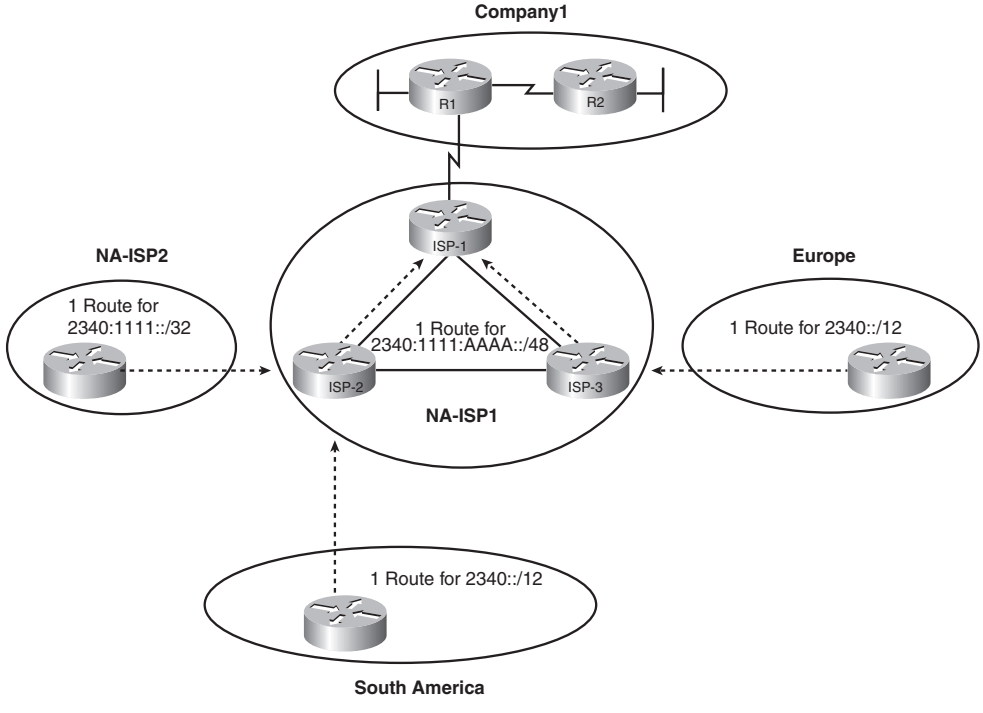


The process starts with ICANN, which owns the entire IPv6 address space, and assigns the rights to *registry prefix* 2340::/12 to one of the RIRs, ARIN in this case (North America). This means that ARIN has the rights to assign any IPv6 addresses that begin with the first 12 bits of hex 2340 (binary value 0010 0011 0100). For perspective, that's a large group of addresses—2^{116} to be exact.

Next, NA-ISP1 asks ARIN for a prefix assignment. After ARIN ensures that NA-ISP1 meets some requirements, ARIN might assign *ISP prefix* 2340:1111::/32 to NA-ISP1. This too is a large group—2^{96} addresses to be exact. For perspective, this one address block might well be enough public IPv6 addresses for even the largest ISP, without that ISP ever needing another IPv6 prefix.

Finally, Company1 asks its ISP, NA-ISP1, for the assignment of an IPv6 prefix. NA-ISP1 assigns Company1 the site prefix 2340:1111:AAAA::/48, which is again a large range of addresses—2^{80} in this case. In the next paragraph, the text shows what Company1 could do with that prefix, but first, examine Figure 17-4, which presents the same concepts as shown in Figure 17-1, but now with the prefixes shown.

Figure 17-4 *IPv6 Global Routing Concepts*



The figure shows the perspectives of routers outside North America, routers from another ISP in North America, and other routers in the same ISP. Routers outside North America can use a route for prefix 2340::/12, knowing that ICANN assigned this prefix to be used only by ARIN. This one route could match all IPv6 addresses assigned in North America. Routers in NA-ISP2, an example alternative ISP in North America, need one route for 2340:1111::/32, the prefix assigned to NA-ISP1. This one route could match all packets destined for all customers of NA-ISP1. Inside NA-ISP1, its routers need to know to which NA-ISP1 router to forward packets to for that particular customer (the router named ISP-1 in this case), so the routes inside NA-ISP1's routers lists a prefix of 2340:1111:AAAA/48.

Subnetting Global Unicast IPv6 Addresses Inside an Enterprise

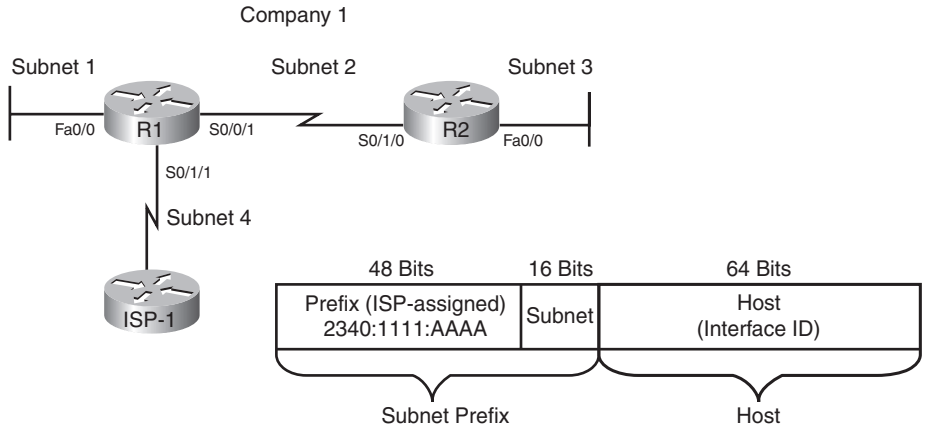
The original IPv4 Internet design called for each organization to be assigned a classful network number, with the enterprise subdividing the network into smaller address ranges by subnetting the classful network. This same concept of subnetting carries over from IPv4 to IPv6, with the enterprise subnetting the prefix assigned by its ISP into smaller prefixes. When thinking about the IPv6 subnetting concept, you could make the following general analogies with classful IPv4 subnetting to help understand the process:

- The prefix assigned to the enterprise by the ISP, which must be the same for all IPv6 addresses in one enterprise, is like the IPv4 network part of an address.
- The enterprise engineer extends the length of the prefix, borrowing host bits, to create a subnet part of the address.
- The last/third major part is the host part of the address, called the *interface ID* in IPv6, and is meant to uniquely identify a host inside a subnet.

For example, Figure 17-5 shows a more detailed view of the Company1 enterprise network shown in several of the earlier figures in this chapter. The design concepts behind how many subnets are needed with IPv6 are identical to those for IPv4: A subnet is needed for each VLAN and for each serial link, with the same options for subnets with Frame Relay. In this case, two LANs and two serial links exist, so Company1 needs four subnets.

The figure also shows how the enterprise engineer extended the length of the prefix as assigned by the ISP (/48) to /64, thereby creating a 16-bit subnet part of the address structure. The /48 prefix is generally called the *site prefix*, and the longer prefix used on each link is called a *subnet prefix*. To create this extra 16-bit subnet field, the engineer uses the same concept as with IPv4 when choosing a subnet mask by borrowing bits from the host field of an IPv4 address. In this case, think of the host field as having 80 bits (because the prefix assigned by the ISP is 48 bits long, leaving 80 bits), and the design in Figure 17-5 borrows 16 bits for the subnet field, leaving a measly 64 bits for the host field.

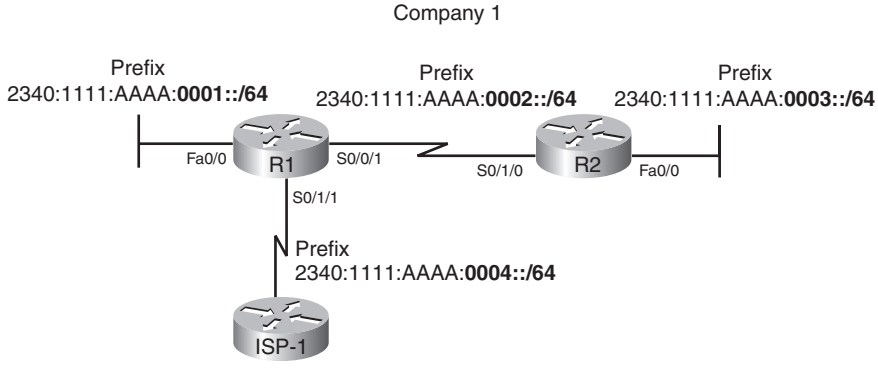
Figure 17-5 *Company1 Needs Four Subnets*



A bit of math about the design choices can help provide some perspective on the scale of IPv6. The 16-bit subnet field allows 2^{16}, or 65,536, subnets—overkill for all but the very largest organizations or companies. (There are also no worries about a zero or broadcast subnet in IPv6!) The host field is seemingly even more overkill: 2^{64} hosts per subnet, which is more than 1,000,000,000,000,000,000 addresses per subnet. However, a good reason exists for this large host or interface ID part of the address, because it allows one of the automatic IPv6 address assignment features to work well, as is covered in the section "IPv6 Host Address Assignment," later in this chapter.

Figure 17-6 takes the concept to the final conclusion, assigning the specific four subnets to be used inside Company1. Note that the figure shows the subnet fields and prefix lengths (64 in this case) in bold.

Figure 17-6 *Company1 with Four Subnets Assigned*



NOTE The subnet numbers in the figure could be abbreviated slightly, removing the three leading 0s from the last shown quartets.

Figure 17-6 just shows one option for subnetting the prefix assigned to Company1. However, any number of subnet bits could be chosen, as long as the host field retained enough bits to number all hosts in a subnet. For example, a /112 prefix length could be used, extending the /48 prefix by 64 bits (4 hex quartets). Then, for the design in Figure 17-6, you could choose the following four subnets:

- 2340:1111:AAAA::0001:0000/112
- 2340:1111:AAAA::0002:0000/112
- 2340:1111:AAAA::0003:0000/112
- 2340:1111:AAAA::0004:0000/112

By using global unicast IPv6 addresses, Internet routing can be very efficient and enterprises can have plenty of IP addresses and plenty of subnets, with no requirement for NAT functions to conserve the address space.

Prefix Terminology

Before wrapping up this topic, a few new terms need to be introduced. The process of global unicast IPv6 address assignment examines many different prefixes, with many different prefix lengths. The text scatters a couple of more specific terms, but for easier study, Table 17-4 summarizes the four key terms, with some reminders of what each means.

Table 17-4 *Example IPv6 Prefixes and Their Meanings*

Term	Assignment	Example from Chapter 17
Registry prefix	By ICANN to an RIR	2340::/12
ISP prefix	By an RIR to an ISP[1]	2340:1111/32
Site prefix	By an ISP to a customer (site)	2340:1111:AAAA/48
Subnet prefix	By an enterprise engineer for each individual link	2340:1111:AAAA:0001/64

[1]While an RIR can assign a prefix to an ISP, an RIR can also assign a prefix to other Internet registries, which can subdivide and assign additional prefixes, until eventually an ISP and then its customers are assigned some unique prefix.

The next sections of this chapter broaden the discussion of IPv6 to include additional types of IPv6 addresses, along with the protocols that control and manage several common functions for IPv6.

IPv6 Protocols and Addressing

IPv4 hosts need to know several basic facts before they can succeed in simple tasks like opening a web browser to view a web page. IPv4 hosts typically need to know the IP address of one or more Domain Name System (DNS) servers so that they can use DNS protocol messages to ask a DNS server to resolve that name into an IPv4 address. They need to know an IP address of a router to use as a default gateway (default router), with the host sending packets destined to a host in a different subnet to that default router. The host, of course, needs to know its unicast IPv4 IP address and mask—or, as stated with classless terminology, its IPv4 address and prefix length—from which the host can calculate the prefix (subnet) on that link.

IPv6 hosts need the same information—DNS IP addresses, default router IP address, and their own address/prefix length—for the same reasons. IPv6 hosts still use host names, and they need to have the host name resolved into an IPv6 address. IPv6 hosts still send packets directly to hosts on the same subnet, but they send packets to the default router for off-subnet destinations.

While IPv6 hosts need to know the same information, IPv6 changes the mechanisms for learning some of these facts compared to IPv4. The following sections examine the options and protocols through which a host can learn these key pieces of information. At the same time, these sections introduce several other types of IPv6 addresses that are used by the new IPv6 protocols. The end of these sections summarizes the details and terminology for the various types of IPv6 addresses.

DHCP for IPv6

IPv6 hosts can use Dynamic Host Configuration Protocol (DHCP) to learn and lease an IP address and corresponding prefix length (mask), the IP address of the default router, and the DNS IP address(es). The concept works basically like DHCP for IPv4: The host sends a (multicast) IPv6 packet searching for the DHCP server. When a server replies, the DHCP client sends a message asking for a lease of an IP address, and the server replies, listing an IPv6 address, prefix length, default router, and DNS IP addresses. The names and formats of the actual DHCP messages have changed quite a bit from IPv4 to IPv6, so DHCPv4 and DHCPv6 differ in detail, but the basic process remains the same. (DHCPv4 refers to the version of DHCP used for IPv4, and DHCPv6 refers to the version of DHCP used for IPv6.)

DHCPv4 servers retain information about each client, like the IP address leased to that client and the length of time for which the lease is valid. This type of information is called *state information*, because it tracks the state or status of each client. DHCPv6 servers happen to have two operational modes: stateful, in which the server tracks state information, and stateless, in which the server does not track state information. Stateful DHCPv6 servers fill the same role as the older DHCPv4 servers, while stateless DHCPv6

servers fill one role in an IPv6 alternative to stateful DHCP. (Stateless DHCP, and its purpose, is covered in the upcoming section "IPv6 Host Address Assignment.")

One difference between DHCPv4 and stateful DHCPv6 is that IPv4 hosts send IP broadcasts to find DHCP servers, while IPv6 hosts send IPv6 multicasts. IPv6 multicast addresses have a prefix of FF00::/8, meaning that the first 8 bits of an address are binary 11111111, or FF in hex. The multicast address FF02::1:2 (longhand FF02:0000:0000:0000:0000:0000:0001:0002) has been reserved in IPv6 to be used by hosts to send packets to an unknown DHCP server, with the routers working to forward these packets to the appropriate DHCP server.

IPv6 Host Address Assignment

When using IPv4 in enterprise networks, engineers typically configure static IPv4 addresses on each router interface with the **ip address** interface subcommand. At the same time, most end-user hosts use DHCP to dynamically learn their IP address and mask. For Internet access, the router can use DHCP to learn its own public IPv4 address from the ISP.

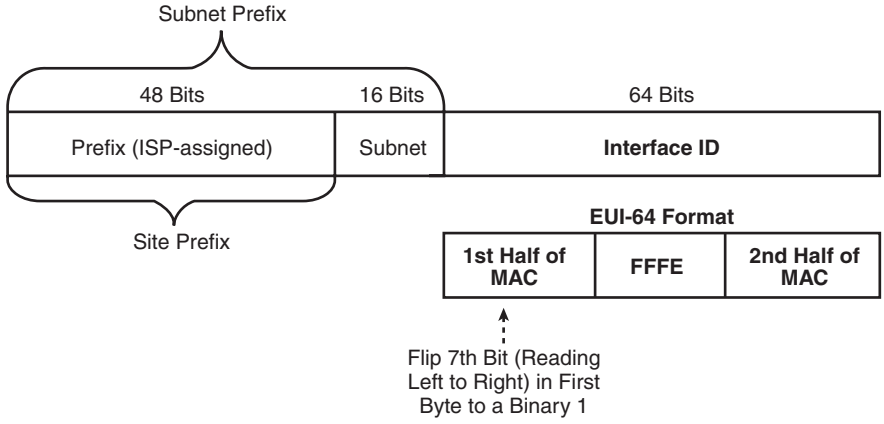
IPv6 follows the same general model, but with routers using one of two options for static IPv6 address assignment, and with end-user hosts using one of two options for dynamic IPv6 address assignment. The following sections examine all four options. But first, to appreciate the configuration options, you need a little more information about the low-order 64 bits of the IPv6 address format: the interface ID.

The IPv6 Interface ID and EUI-64 Format

Earlier in this chapter, Figure 17-5 shows the format of an IPv6 global unicast address, with the second half of the address called the host or interface ID. The value of the interface ID portion of a global unicast address can be set to any value, as long as no other host in the same subnet attempts to use the same value. (IPv6 includes a dynamic method for hosts to find out whether a duplicate address exists on the subnet before starting to use the address.) However, the size of the interface ID was purposefully chosen to allow easy autoconfiguration of IP addresses by plugging the MAC address of a network card into the interface ID field in an IPv6 address.

MAC addresses are 6 bytes (48 bits) in length, so for a host to automatically decide on a value to use in the 8-byte (64-bit) interface ID field, IPv6 cannot simply copy just the MAC address. To complete the 64-bit interface ID, IPv6 fills in 2 more bytes. Interestingly, to do so, IPv6 separates the MAC address into two 3-byte halves, and inserts hex FFFE in between the halves, to form the interface ID field, as well as setting 1 special bit to binary 1. This format, called the EUI-64 format, is shown in Figure 17-7.

Figure 17-7 *IPv6 Address Format with Interface ID and EUI-64*



Although it might seem a bit convoluted, it works. Also, with a little practice, you can look at an IPv6 address and quickly notice the FFFE late in the address, and then easily find the two halves of the corresponding interface's MAC address.

To be complete, the figure points out one other small detail regarding the EUI-64 interface ID value. Splitting the MAC address into two halves, and injecting FFFE, is easy. However, the EUI-64 format requires setting the seventh bit in the first byte of the value to binary 1. The underlying reason is that Ethernet MAC addresses are listed with the low-order bits of each byte on the left, and the high-order bits on the right. So, the eighth bit in a byte (reading from left to right) is the highest-order bit in the address, and the seventh bit (reading from left to right) is the second highest-order bit. This second highest-order bit in the first byte—the seventh bit reading from left to right—is called the universal/local (U/L) bit. Set to binary 0, it means that the MAC address is a burned-in MAC address. Set to 1, it means that the MAC address has been configured locally. EUI-64 says that the U/L bit should be set to 1, meaning local.

For example, the following two lines list a host's MAC address and corresponding EUI-64 format interface ID, assuming the use of an address configuration option that uses the EUI-64 format:

- 0034:5678:9ABC
- 0234:56FF:FE78:9ABC

NOTE To change the seventh bit (reading left-to-right) in the example, convert hex 00 to binary 00000000, change the seventh bit to 1 (00000010), and then convert back to hex, for hex 02 as the first two digits.

Static IPv6 Address Configuration

Two options for static IPv6 address configuration are covered in this book, and both are available on both routers and hosts: static configuration of the entire address, and static configuration of a /64 prefix with the host calculating its EUI-64 interface ID to complete the IP address. This section shows the concept using routers.

To configure an IPv6 address on an interface, the router needs an **ipv6 address** *address/prefix-length* [**eui-64**] interface subcommand on each interface. If the **eui-64** keyword is not included, the address must represent the entire 128-bit address. If the **eui-64** keyword is included, the address should represent the 64-bit prefix, with the router creating the interface ID using the EUI-64 format. The *prefix-length* parameter should be the length of the subnet prefix. For example, Example 17-1 lists the commands on Router R1 from Figure 17-6 earlier in this chapter, which is one of Company1's enterprise routers. It uses the site prefix length of /64. The example shows both versions of the command (with and without the **eui-64** keyword.)

Example 17-1 *Configuring Static IPv6 Addresses*

```
! The first interface is in subnet 1, and will use EUI-64 as the Interface ID
!
interface FastEthernet0/0
 ipv6 address 2340:1111:AAAA:1::/64 eui-64
! The next interface spells out the whole 128 bits, abbreviated. The longer
! version is 2340:1111:AAAA:0003:0000:0000:0001/64. It is in subnet 2.
!
interface Serial0/0/1
 ipv6 address 2340:1111:AAAA:2::1/64
! The third interface is in subnet 4, with EUI-64 format Interface ID again.
!
interface Serial0/1/1
 ipv6 address 2340:1111:AAAA:4::/64 eui-64
!
R1#show ipv6 interface fa0/0
FastEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::213:19FF:FE7B:5004
  Global unicast address(es):
    2340:1111:AAAA:1:213:19FF:FE7B:5004, subnet is 2340:1111:AAAA:1::/64 [EUI]
! Lines omitted for brevity
R1#show ipv6 interface S0/0/1
Serial0/0/1 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::213:19FF:FE7B:5004
  Global unicast address(es):
    2340:1111:AAAA:3::1, subnet is 2340:1111:AAAA:3::/64
! Lines omitted for brevity
R1#show ipv6 interface s0/1/1
Serial0/1/1 is up, line protocol is up
```

Example 17-1 *Configuring Static IPv6 Addresses (Continued)*

```
  IPv6 is enabled, link-local address is FE80::213:19FF:FE7B:5004
  Global unicast address(es):
    2340:1111:AAAA:4:213:19FF:FE7B:5004, subnet is 2340:1111:AAAA:4::/64 [EUI]
! Lines omitted for brevity
```

The end of the example lists the full global unicast IPv6 address as part of the **show ipv6 interface** command. When using the EUI-64 option, this command is particularly useful, because the configuration command does not list the entire IPv6 address. Note that if the EUI format is used, the **show ipv6 interface** command notes that fact (see interfaces Fa0/0 and S0/1/1, versus S0/0/1). Also, routers do not have MAC addresses associated with some interfaces, including serial interfaces, so to form the EUI-64–formatted interface ID on those interfaces, routers use the MAC address of a LAN interface. In this case, S0/1/1's interface ID is based on Fa0/0's MAC address.

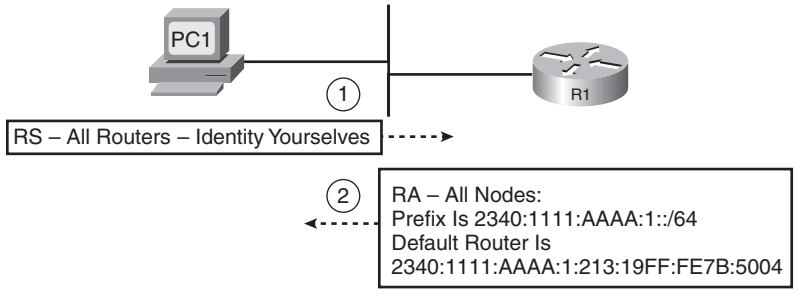
Stateless Autoconfiguration and Router Advertisements

IPv6 supports two methods of dynamic configuration of IPv6 addresses. One uses a stateful DHCPv6 server, which as mentioned earlier, works the same as DHCP in IPv4 in concept, although many details in the messages differ between DHCPv4 and DHCPv6. IPv6 also supplies an alternative called *stateless autoconfiguration* (not to be confused with stateless DHCP, which is covered in this section). With stateless autoconfiguration, a host dynamically learns the /64 prefix used on the subnet, and then calculates the rest of its address by using an EUI-64 interface ID based on its network interface card (NIC) MAC address.

The stateless autoconfiguration process uses one of many features of the IPv6 Neighbor Discovery Protocol (NDP) to discover the prefix used on the LAN. NDP performs many functions for IPv6, all related to something that occurs between two hosts in the same subnet. For example, one part of NDP replaces the IPv4 ARP protocol. IPv4 ARP allows devices on the same subnet—neighbors—to learn each other's MAC address. Because this and many other activities occur only inside the local subnet between neighbors on the same link, IPv6 collected these basic functions into one protocol suite, called NDP.

Stateless autoconfiguration uses two NDP messages, namely router solicitation (RS) and router advertisement (RA) messages, to discover the IPv6 prefix used on a LAN. The host sends the RS message as an IPv6 multicast message, asking all routers to respond to the questions "What IPv6 prefix(s) is used on this subnet?" and "What is the IPv6 address(s) of any default routers on this subnet?" Figure 17-8 shows the general idea, on subnet 1 from Figure 17-6, with PC1 sending an RS, and router R1 replying with the IPv6 prefix used on the LAN and R1's own IPv6 address as a potential default router.

Figure 17-8 *Example NDP RS/RA Process to Find the Default Routers*



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

IPv6 does not use broadcasts. In fact, there is no such thing as a subnet broadcast address, a network-wide broadcast address, or an equivalent of the all-hosts 255.255.255.255 broadcast IPv4 address. Instead, IPv6 uses multicast addresses. By using a different multicast IPv6 address for different functions, a computer that has no need to participate in a particular function can simply ignore those particular multicasts, reducing the impact to the host. For example, the RS message only needs to be received and processed by routers, so the RS message's destination IP address is FF02::2, which is the address reserved in IPv6 to be used only by IPv6 routers. RA messages are sent to a multicast address intended for use by all IPv6 hosts on the link (FF02::1), so not only will the host that sent the RS learn the information, but all other hosts on the link will also learn the details.

Table 17-5 summarizes some of the key details about the RS/RA messages.

Table 17-5 *Details of the RS/RA Process*

Message	RS	RA
Multicast destination	FF02::2	FF02::1
Meaning of multicast address	All routers on this link	All IPv6 nodes on this link

IPv6 Address Configuration Summary

This chapter covers four methods for assigning IPv6 addresses to hosts or router interfaces. Two variations use static configuration, while two dynamically learn the address. However, with both static and dynamic configuration, two alternatives exist—one that supplies the entire IPv6 address and one that allows the host to calculate the EUI-64 interface ID. Table 17-6 summarizes the configuration methods.

Table 17-6 *IPv6 Address Configuration Options*

Static or Dynamic	Option	Portion Configured or Learned
Static	Do not use EUI-64	Entire 128-bit address
Static	Use EUI-64	Just the /64 prefix
Dynamic	Stateful DHCPv6	Entire 128-bit address
Dynamic	Stateless autoconfiguration	Just the /64 prefix

Discovering the Default Router with NDP

In IPv4, hosts discover their default router (default gateway) either through static configuration on the host or, more typically, with DHCP. IPv6 can use both of these same options as well, plus the NDP RS/RA messages as explained in the previous section. The NDP router discovery process occurs by default on IPv6 hosts and routers, so while the stateful DHCPv6 server can supply the IP address(es) of the possible default routers, it is perfectly reasonable in IPv6 to simply not bother to configure these details in a stateful DHCP server, allowing the built-in NDP RS/RA messages to be used instead.

The default router discovery process is relatively simple. Routers automatically send RA messages on a periodic basis. These messages list not only the sending router's IPv6 address but also all the known routers on that subnet. A host can wait for the next periodic RA message or request that all local routers send an RA immediately by soliciting the routers using the RS message.

Learning the IP Address(es) of DNS Servers

Like IPv4 hosts, IPv6 hosts typically need to know the IP address of one or more DNS servers to resolve names into the corresponding IP address. Oftentimes, the host also needs to learn the DNS domain name to use. And like IPv4 hosts, IPv6 hosts can be told these IP addresses using (stateful) DHCP. When a host (or router for that matter) learns its IPv6 address using stateful DHCP, the host can also learn the DNS server IP addresses and the domain name, taking care of this particular detail.

Stateless DHCP, which is most useful in conjunction with stateless autoconfiguration, is an alternative method for finding the DNS server IP addresses and the domain name. A host that uses stateless autoconfiguration can learn its IPv6 address and prefix automatically, as well as learn its default router IP address, in both cases using NDP RS/RA messages. However, the stateless autoconfiguration process does not help a host learn the DNS IP addresses and domain name. So, stateless DHCP supplies that information using the same messages as stateful DHCP. However, to supply this information, the server does not need to track any state information about each client, so a stateless DHCP server can be used.

Table 17-7 summarizes some of the key features of stateful and stateless DHCPv6.

Key Topic

Table 17-7 *Comparison of Stateless and Stateful DHCPv6 Services*

Feature	Stateful DHCP	Stateless DHCP
Remembers IPv6 address (state information) of clients that make requests	Yes	No
Assigns IPv6 address to client	Yes	No
Supplies useful information, like DNS server IP addresses	Yes	Yes
Is most useful in conjunction with stateless autoconfiguration	No	Yes

IPv6 Addresses

This chapter has already introduced the concepts behind the general format of IPv6 addresses, the ideas behind global unicast IPv6 addresses, and some details about multicast IPv6 addresses. The following sections round out the coverage of addressing, specifically the three categories of IPv6 address:

Key Topic

- **Unicast:** IP addresses assigned to a single interface for the purpose of allowing that one host to send and receive data.
- **Multicast:** IP addresses that represent a dynamic group of hosts for the purpose of sending packets to all current members of the group. Some multicast addresses are used for special purposes, like with NDP messages, while most support end-user applications.
- **Anycast:** A design choice by which servers that support the same function can use the same unicast IP address, with packets sent by clients being forwarded to the nearest server, allowing load balancing across different servers.

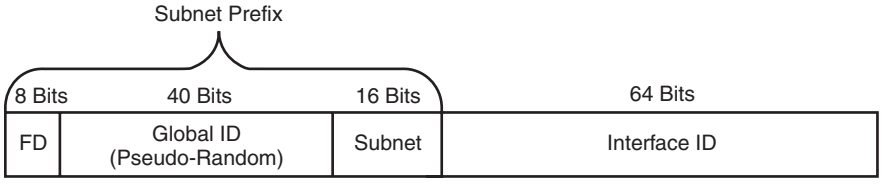
Unicast IPv6 Addresses

IPv6 supports three main classes of unicast addresses. One of these classes, global unicast IP addresses, closely matches the purpose of IPv4 public IP addresses. Global unicast addresses are assigned by ICANN and the RIRs for the purpose of allowing globally unique IPv6 addresses for all hosts. These addresses come from inside the 2000::/3 prefix, which includes all addresses that begin with 2 or 3 (hex).

The next class of IPv6 unicast addresses covered here, *unique local* unicast addresses, have the same function as IPv4 RFC 1918 private addresses. In IPv4, most every enterprise, and most every Internet-connected small or home office, uses IPv4 private networks. *Unique local* unicast addresses begin with hex FD (FD00::/8), with the format shown in Figure 17-9.

NOTE The original IPv6 RFCs defined a private address class called *site local*, meaning local within a site (organization). The original site local address class has been deprecated and replaced with unique local unicast addresses.

Figure 17-9 *Unique Local Address Format*



To use these addresses, an enterprise engineer would choose a 40-bit global ID in a pseudorandom manner, with the goal that hopefully the addresses will be unique in the universe. In reality, pseudorandom is probably a number made up by the engineer. The 16-bit subnet field and 64-bit interface ID work just like with global unicast addresses, numbering different subnets and hosts and allowing EUI-64 assignment of the interface ID. As usual, the engineer could avoid using EUI-64, using easier-to-remember values like 0000:0000:0000:0001 as the interface ID.

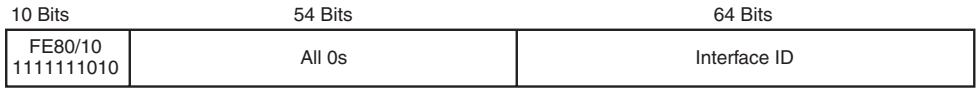
Link local addresses are the third class of unicast IPv6 addresses covered here. IPv4 has no concepts like the link local IP address. IPv6 uses these addresses when sending packets over the local subnet; routers never forward packets destined for link local addresses to other subnets.

Link local addresses can be useful for functions that do not need to leave the subnet, in particular because a host can automatically derive its own link local IP address without sending packets over the subnet. So, before sending the first packets, the host can calculate its own link local address so that the host has an IPv6 address to use when doing its first overhead messages. For example, before a host sends an NDP RS (router solicitation) message, the host will have already calculated its link local address. The host uses its link local address as the source IP address in the RS message.

Link local addresses come from the FE80::/10 range, meaning all addresses that begin with FE80, FE90, FEA0, and FEB0. No specific configuration is required, because a host forms these addresses by using the first 10 bits of hex FE80 (binary 1111111010), 54 more binary 0s, and the last 64 bits being the host's EUI-64 format interface ID. Figure 17-10 shows the format.

Key Topic

Figure 17-10 *Link Local Address Format*



Routers also use link local addresses on each interface enabled to support IPv6. Like hosts, routers automatically calculate their link local IP addresses. In fact, Example 17-1 earlier in this chapter listed the (R1) router's link local IP addresses in the output of the **show ipv6 interface** command output. Interestingly, routers normally use link local addresses as the next-hop IP address in IPv6 routes, rather than the neighboring router's global unicast or unique local unicast address.

Multicast and Other Special IPv6 Addresses

Multicast addresses can be used to communicate to dynamic groupings of hosts, with the sender sending a single packet and with the network replicating that packet as needed so that all hosts listening for packets sent to that multicast address receive a copy of the packet. IPv6 can limit the scope of where routers forward multicasts based on the value in the first quartet of the address. This book only examines multicasts that should stay on a local link; these addresses all begin with FF02::/16, so they are easily recognized.

For reference, Table 17-8 lists some of the more commonly seen IPv6 multicast addresses. Of particular interest are the addresses chosen for use by Routing Information Protocol (RIP), Open Shortest Path First (OSPF), and Enhanced IGRP (EIGRP), which somewhat mirror the multicast addresses each protocol uses for IPv4.

Table 17-8 *Common Link Local Multicast Addresses*

Purpose	IPv6 Address	IPv4 Equivalent
All IP nodes on the link	FF02::1	Subnet broadcast address
All routers on the link	FF02::2	N/A
OSPF messages	FF02::5, FF02::6	224.0.0.5, 224.0.0.6
RIP-2 messages	FF02::9	224.0.0.9
EIGRP messages	FF02::A	224.0.0.10
DHCP relay agents (routers that forward to the DHCP server)	FF02:1:2	N/A

Before completing the discussion of IPv6 addressing, you should know about a couple of special IPv6 addresses. First, IPv6 supports the concept of a loopback IP address, as follows:

::1 (127 binary 0s and a 1)

Just like the IPv4 127.0.0.1 loopback address, this address can be used to test a host's software. A packet sent by a host to this address goes down the protocol stack, and then right back up the stack, with no communication with the underlying network card. This allows testing of the software on a host, particularly when testing new applications.

The other special address is the :: address (all binary 0s). This address represents the unknown address, which hosts can use when sending packets in an effort to discover their IP addresses.

Summary of IP Protocols and Addressing

This chapter has covered a lot of concepts and details about IPv6 addresses, many of which require some work to remember or memorize. This short section pulls several concepts from throughout this major section on IPv6 protocols and addresses together before moving on to some details about routing protocols and router configuration.

When an IPv6 host first boots, it needs to do several tasks before it can send packets through a router to another host. When using one of the two methods of dynamically learning an IPv6 address that can be used to send packets past the local routers to the rest of a network, the first few initialization steps are the same, with some differences in the later steps. The following list summarizes the steps a host takes when first booting, at least for the functions covered in this chapter:

Step 1 The host calculates its IPv6 link local address (begins with FE80::/10).

Step 2 The host sends an NDP router solicitation (RS) message, with its link local address as the source address and the all-routers FF02::2 multicast destination address, to ask routers to supply a list of default routers and the prefix/length used on the LAN.

Step 3 The router(s) replies with an RA message, sourced from the router's link local address, sent to the all-IPv6-hosts-on-the-link multicast address (FF02::1), supplying the default router and prefix information.

Step 4 If the type of dynamic address assignment is stateless autoconfiguration, the following occur:

- **a.** The host builds the unicast IP address it can use to send packets through the router by using the prefix learned in the RA message and calculating an EUI-64 interface ID based on the NIC MAC address.
- **b.** The host uses DHCP messages to ask a stateless DHCP server for the DNS server IP addresses and domain name.

Step 4 If the type of dynamic address assignment is stateful DHCP, the host uses DHCP messages to ask a stateful DHCP server for a lease of an IP address/prefix length, as well as default router addresses, the DNS server IP addresses, and domain name.

NOTE Other tasks occur when a host initializes as well, but they are beyond the scope of this book.

IPv6 includes many different types of addresses, including unicast and multicast. By way of summary, Table 17-9 lists the types of IPv6 addresses mentioned by this chapter, with a few details, for easier reference when studying.

Key Topic

Table 17-9 *Common Link Local Multicast Addresses*

Type of Address	Purpose	Prefix	Easily Seen Hex Prefix(es)
Global unicast	Unicast packets sent through the public Internet	2000::/3	2 or 3
Unique local	Unicast packets inside one organization	FD00::/8	FD
Link Local	Packets sent in the local subnet	FE80::/10	FE8, FE9, FEA, FEB
Multicast (link local scope)	Multicasts that stay on the local subnet	FF02::/16	FF02

Configuring IPv6 Routing and Routing Protocols

To support IPv6, all the IPv4 routing protocols had to go through varying degrees of changes, with the most obvious being that each had to be changed to support longer addresses and prefixes. The following sections first examine a few details about routing protocols and then show how to configure IPv6 routing and routing protocols on Cisco routers.

IPv6 Routing Protocols

As with IPv4, most IPv6 routing protocols are interior gateway protocols (IGP), with Border Gateway Protocol (BGP) still being the only exterior gateway protocol (EGP) of note. All these current IGPs and BGP have been updated to support IPv6. Table 17-10 lists the routing protocols and their new RFCs (as appropriate).

Table 17-10 *Updates to Routing Protocols for IPv6*

Routing Protocol	Full Name	RFC
RIPng	RIP Next Generation	2080
OSPFv3	OSPF version 3	2740
MP-BGP4	Multiprotocol BGP-4	2545/4760
EIGRP for IPv6	EIGRP for IPv6	Proprietary

Each of these routing protocols has to make several changes to support IPv6. The actual messages used to send and receive routing information have changed, using IPv6 headers instead of IPv4 headers and using IPv6 addresses in those headers. For example, RIPng sends routing updates to the IPv6 destination address FF02::9, instead of the old RIP-2 IPv4 224.0.0.9 address. Also, the routing protocols typically advertise their link local IP address as the next hop in a route, as will be shown in the upcoming Example 17-2.

The routing protocols still retain many of the same internal features. For example, RIPng, being based on RIP-2, is still a distance vector protocol, with hop count as the metric and 15 hops as the longest valid route (16 is infinity). OSPFv3, created specifically to support IPv6, is still a link-state protocol, with cost as the metric but with many of the internals, including link-state advertisement (LSA) types, changed. As a result, OSPFv2, as covered in Chapter 9, "OSPF," is not compatible with OSPFv3. However, the core operational concepts remain the same.

IPv6 Configuration

Cisco router IOS enables the routing (forwarding) of IPv4 packets by default, with IPv4 being enabled on an interface when the interface has an IPv4 address configured. For IPv4 routing protocols, the routing protocol must be configured, with the **network** command indirectly enabling the routing protocol on an interface.

IPv6 configuration follows some of these same guidelines, with the largest difference being how to enable a routing protocol on an interface. Cisco router IOS does not enable IPv6 routing by default, so a global command is required to enable IPv6 routing. The unicast IP addresses need to be configured on the interfaces, similar to IPv4. The routing protocol needs to be globally configured, similar to IPv4. Finally, the routing protocol has to be configured on each interface as needed, but with IPv6, the process does not use the **network** router subcommand.

This section shows an example configuration, again showing Router R1 from the Company1 enterprise network shown in earlier figures in this chapter. The example uses

RIPng as the routing protocol. The following list outlines the four main steps to configure IPv6:

Step 1 Enable IPv6 routing with the **ipv6 unicast-routing** global command.

Step 2 Enable the chosen routing protocol. For example, for RIPng, use the **ipv6 router rip** *name* global configuration command.

Step 3 Configure an IPv6 unicast address on each interface using the **ipv6 address** *address/prefix-length* [**eui-64**] interface command.

Step 4 Enable the routing protocol on the interface, for example, with the **ipv6 rip** *name* **enable** interface subcommand (where the name matches the **ipv6 router rip** *name* global configuration command).

Example 17-2 shows the configuration, plus a few **show** commands. Note that the IP address configuration matches the earlier Example 17-1. Because Example 17-1 showed the address configuration, this example shows gray highlights on the new configuration commands only.

Example 17-2 *Configuring IPv6 Routing and Routing Protocols on R1*

```
R1#show running-config
! output is edited to remove lines not pertinent to this example
ipv6 unicast-routing
!
interface FastEthernet0/0
 ipv6 address 2340:1111:AAAA:1::/64 eui-64
 ipv6 rip atag enable
!
interface Serial0/0/1
 ipv6 address 2340:1111:AAAA:2::1/64
 ipv6 rip atag enable
!
interface Serial0/1/1
 ipv6 address 2340:1111:AAAA:4::/64 eui-64
 ipv6 rip atag enable
!
ipv6 router rip atag
!
R1#show ipv6 route
IPv6 Routing Table - 10 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
R   ::/0 [120/2]
```

Example 17-2 *Configuring IPv6 Routing and Routing Protocols on R1 (Continued)*

```
     via FE80::213:19FF:FE7B:2F58, Serial0/1/1
C    2340:1111:AAAA:1::/64 [0/0]
     via ::, FastEthernet0/0
L    2340:1111:AAAA:1:213:19FF:FE7B:5004/128 [0/0]
     via ::, FastEthernet0/0
C    2340:1111:AAAA:2::/64 [0/0]
     via ::, Serial0/0/1
L    2340:1111:AAAA:2::1/128 [0/0]
     via ::, Serial0/0/1
R    2340:1111:AAAA:3::/64 [120/2]
     via FE80::213:19FF:FE7B:5026, Serial0/0/1
C    2340:1111:AAAA:4::/64 [0/0]
     via ::, Serial0/1/1
L    2340:1111:AAAA:4:213:19FF:FE7B:5004/128 [0/0]
     via ::, Serial0/1/1
L    FE80::/10 [0/0]
     via ::, Null0
L    FF00::/8 [0/0]
     via ::, Null0
R1#show ipv6 interface brief
FastEthernet0/0            [up/up]
    FE80::213:19FF:FE7B:5004
    2340:1111:AAAA:1:213:19FF:FE7B:5004
FastEthernet0/1            [up/up]
    unassigned
Serial0/0/0                [administratively down/down]
    unassigned
Serial0/0/1                [up/up]
    FE80::213:19FF:FE7B:5004
    2340:1111:AAAA:2::1
Serial0/1/0                [administratively down/down]
    unassigned
Serial0/1/1                [up/up]
    FE80::213:19FF:FE7B:5004
    2340:1111:AAAA:4:213:19FF:FE7B:5004
```

The configuration itself does not require a lot of work beyond the IPv6 address configuration shown previously in Example 17-1. The **ipv6 router rip** *name* command requires a name (formally called a tag) that is just a text name for the routing process. Example 17-2 shows the configuration, using a RIP tag named "atag". This tag does not have to match between the various routers. Otherwise, the configuration itself is straightforward.

The **show ipv6 route** command lists all the IPv6 routes, listing some important differences as highlighted in the command output. First, note the first few lines of highlighted output

in that command, and the new routing code "L". For each interface with a unicast address, the router adds the usual connected route for the prefix connected to that interface. For example, the first highlighted line inside this command lists 2340:1111:AAAA:1::/64, which is the subnet connected to R1's Fa0/0 interface. The output also lists a host route—a /128 prefix length route—as a local route. Each of these local routes, as noted with the code "L," lists the specific address on each interface, respectively.

The next highlighted lines in that same **show ipv6 route** command list some interesting next-hop information in a RIP-learned route. The example highlights the route to subnet 3, listing outgoing interface S0/0/1, but the next-hop address is R2's link local IP address of FE80::213:19FF:FE7B:5026. IPv6 routing protocols typically advertise the link local addresses as next-hop addresses.

Finally, the last part of the example shows the output of the **show ipv6 interface brief** command, which lists the unicast IP addresses on each interface. The highlighted lines first show the link local address (each starts with FE8), and then the global unicast address, on R1's Fa0/0 interface. Each of the three interfaces used in this example has both the link local address, which is automatically generated, and the global unicast addresses configured, as shown in the first part of Example 17-2.

Configuring host names and DNS servers on routers for IPv4 can be a small convenience, but for IPv6, it might well be a necessity. Because of the length of IPv6 addresses, even a simple **ping** command requires a fair amount of typing and referring to other command output or documentation. So, just as with IPv4, you might want to configure static host names on routers, or refer to a DNS server, with the following two commands. Note that the commands and syntax are the same as the commands for IPv4, just with IPv6 addresses used as parameters.

- **ip host** *name ipv6-address* [*second-address* [*third-address* [*fourth-address*]]]
- **ip name-server** *server-address1* [*server-address2...server-address6*]

The first command configures a host name only known to the local routers, while the second refers to a DNS server. Note that the router attempts to act as a DNS client by default, based on the default **ip domain-lookup** global configuration command. However, if the **no ip domain-lookup** command has been configured, change the command back to **ip domain-lookup** to begin using DNS services.

While the configuration and **show** commands in Example 17-2 can be useful for learning the basics, much more is required before an internetwork can be ready for an IPv6 deployment. (*Deploying IPv6 Networks*, by Ciprian Popoviciu et al., published by Cisco Press, is a great resource if you want to read more.) The next section takes a brief look at

one of the larger deployment issues, namely, how to support users during a worldwide migration from IPv4 to IPv6, which might take decades.

IPv6 Transition Options

While IPv6 solves a lot of problems, an overnight migration from IPv4 to IPv6 is ridiculous. The number of devices on Earth that use IPv4 number is well into the billions, and in some cases, even if you wanted to migrate to IPv6, the devices or their software might not even have IPv6 support, or at least well-tested IPv6 support. The migration from IPv4 to IPv6 will at least take years, if not decades.

Thankfully, much time and effort have been spent thinking about the migration process and developing standards for how to approach the migration or transition issue. The following sections introduce the main options and explain the basics. In particular, these sections examine the idea of using dual stacks, tunneling, and translation between the two versions of IP. Note that no one solution is typically enough to solve all problems; in all likelihood, a combination of these tools will need to be used in most every network.

IPv4/IPv6 Dual Stacks

The term *dual stacks* means that the host or router uses both IPv4 and IPv6 at the same time. For hosts, this means that the host has both an IPv4 and IPv6 address associated with each NIC, that the host can send IPv4 packets to other IPv4 hosts, and that the host can send IPv6 packets to other IPv6 hosts. For routers, it means that in addition to the usual IPv4 IP addresses and routing protocols covered in many of the other chapters of this book, the routers would also have IPv6 addresses and routing protocols configured, as shown in this chapter. To support both IPv4 and IPv6 hosts, the router could then receive and forward both IPv4 packets and IPv6 packets.

The dual stack approach can be a reasonable plan of attack to migrate an enterprise to IPv6 for communications inside the enterprise. The routers could be easily migrated to use dual stacks, and most desktop operating systems (OS) support IPv6 today. In some cases, the upgrade my require new software or hardware, but this approach allows a slower migration, which is not necessarily a bad thing, because the support staff needs time to learn how IPv6 works.

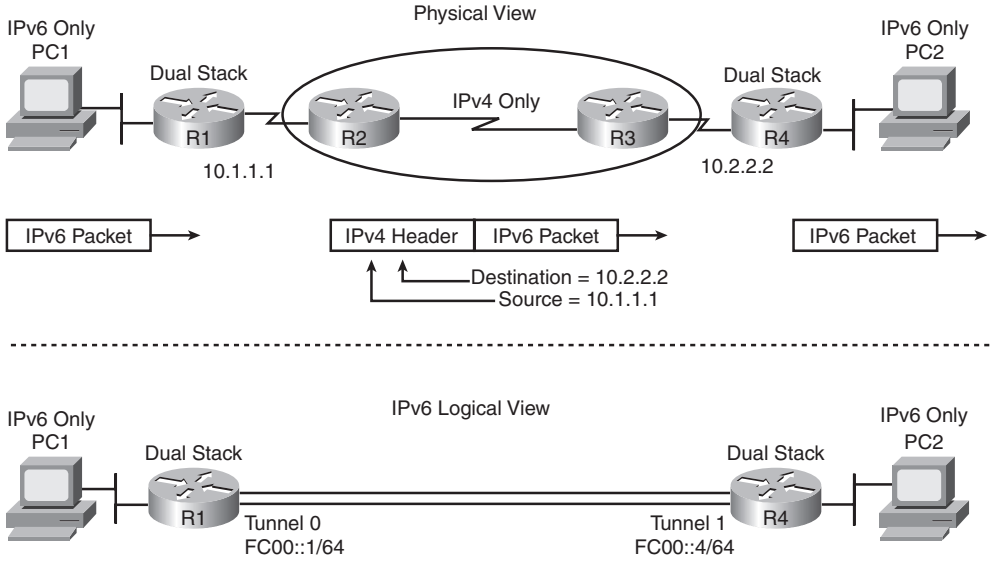
Tunneling

Another tool to support the IPv4-to-IPv6 transition is tunneling. Many types of tunneling exist, but in this case, the tunnel function typically takes an IPv6 packet sent by a host and encapsulates it inside an IPv4 packet. The IPv4 packet can then be forwarded over an existing IPv4 internetwork, with another device removing the IPv4 header, revealing the

original IPv6 packet. The concept is very much like a VPN tunnel, as explained in Chapter 15, "Virtual Private Networks."

Figure 17-11 shows a typical example with a type of tunnel generically called an IPv6-to-IPv4 tunnel, meaning IPv6 inside IPv4. The figure shows a sample enterprise internetwork in which hosts on some of the LANs have migrated to IPv6, but the core of the network still runs IPv4. This might be the case during an initial testing phase inside an enterprise, or it could be commonly done with an IPv4-based ISP that has customers wanting to migrate to IPv6.

Figure 17-11 *Example IPv6-to-IPv4 Tunnel, Physical and Logical View*



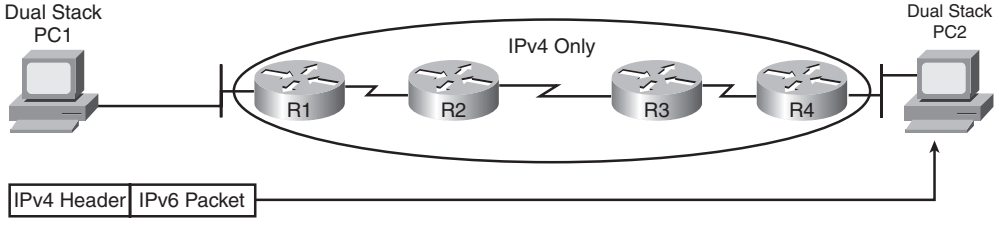
In the figure, the IPv6-based PC1 sends an IPv6 packet. Router R1 then encapsulates or tunnels the IPv6 packet into a new IPv4 header, with a destination IPv4 address of an address on Router R4. Routers R2 and R3 happily forward the packet, because it has a normal IPv4 header, while R4 de-encapsulates the original IPv6 packet, forwarding it to IPv6-based PC2. It's called a tunnel in part because the IPv6 packets inside the tunnel can't be seen while traversing the tunnel; the routers in the middle of the network, R2 and R3 in this case, perceive the packets as IPv4 packets.

Several types of IPv6-to-IPv4 tunnels exist. To perform the tunneling shown by the routers in Figure 17-11, the first three of the following types of tunnels could be used, with the fourth type (Teredo tunnels) being used by hosts:

- **Manually configured tunnels (MCT):** A simple configuration in which tunnel interfaces, a type of virtual router interface, are created, with the configuration referencing the IPv4 addresses used in the IPv4 header that encapsulates the IPv6 packet.
- **Dynamic 6to4 tunnels**: This term refers to a specific type of dynamically created tunnel, typically done on the IPv4 Internet, in which the IPv4 addresses of the tunnel endpoints can be dynamically found based on the destination IPv6 address.
- **Intra-site Automatic Tunnel Addressing Protocol (ISATAP):** Another dynamic tunneling method, typically used inside an enterprise. Unlike 6to4 tunnels, ISATAP tunnels do not work if IPv4 NAT is used between the tunnel endpoints.
- **Teredo tunneling:** This method allows dual-stack hosts to create a tunnel to another host, with the host itself both creating the IPv6 packet and encapsulating the packet inside an IPv4 header.

Figure 17-12 shows the basic idea behind the Teredo tunnel.

Figure 17-12 *Example Encapsulation for a Teredo Host-Host Tunnel*



Translating Between IPv4 and IPv6 with NAT-PT

Both classes of IPv6 transition features mentioned so far in this chapter, dual stack and tunnels, rely on the end hosts to at least support IPv6, if not both IPv4 and IPv6. However, in some cases, an IPv4-only host needs to communicate with an IPv6-only host. A third class of transition features needs to be used in this case: a tool that translates the headers of an IPv6 packet to look like an IPv4 packet, and vice versa.

In Cisco routers, Network Address Translation–Protocol Translation (NAT-PT), defined in RFC 2766, can be used to perform the translation. To do its work, a router configured with NAT-PT must know what IPv6 address to translate to which IPv4 address and vice versa, the same kind of information held in the traditional NAT translation table. And like traditional NAT, NAT-PT allows static definition, dynamic NAT, and dynamic PAT, which can be used to conserve IPv4 addresses.

Transition Summary

Table 17-11 summarizes the transition options for IPv6 for easier reference and study.

Table 17-11 *Summary of IPv6 Transition Options*

Name	Particular Type	Description
Dual stack	—	Supports both protocols, and sends IPv4 to IPv4 hosts and IPv6 to IPv6 hosts
Tunnel	MCT	Tunnel is manually configured; sends IPv6 through IPv4 network, typically between routers
Tunnel	6to4	Tunnel endpoints are dynamically discovered; sends IPv6 through IPv4 network, typically between routers
Tunnel	ISATAP	Tunnel endpoints are dynamically discovered; sends IPv6 through IPv4 network between routers; does not support IPv4 NAT
Tunnel	Teredo	Typically used by hosts; host creates IPv6 packet and encapsulates in IPv4
NAT-PT	—	Router translates between IPv4 and IPv6; allows IPv4 hosts to communicate with IPv6 hosts

Exam Preparation Tasks

Review All the Key Topics

Review the most important topics from this chapter, noted with the Key Topics icon in the outer margin of the page. Table 17-12 lists a reference of these key topics and the page numbers on which each is found.

Table 17-12 *Key Topics for Chapter 17*

Key Topic Element	Description	Page Number
Figure 17-1	Route aggregation concepts in the global IPv6 Internet	583
List	Rules for abbreviating IPv6 addresses	585
List	Rules for writing IPv6 prefixes	587
Figure 17-3	Example prefix assignment process	588
List	Major steps in subdividing a prefix into a subnet prefix in an enterprise	590
Figure 17-5	Example and structure of IPv6 subnets	591
Figure 17-7	Structure of IPv6 addresses and EUI-64 formatted interface ID	595
Table 17-6	List of four main options to IPv6 address configuration	599
Table 17-7	Comparisons of IPv6 stateful and stateless DHCP services	600
List	Different types and purposes of IPv6 addresses	600
Figure 17-10	Format and structure of link local addresses	602
List	Summary of the steps a host takes to learn its address, prefix length, DNS, and default router	603
Table 17-9	Summary of prefixes and purpose of most common types of IPv6 addresses	604
List	Configuration checklist for IPv6 configuration	606
Table 17-11	List of IPv6 transition options	612

Complete the Tables and Lists from Memory

Print a copy of Appendix J, "Memory Tables," (found on the CD) or at least the section for this chapter, and complete the tables and lists from memory. Appendix K, "Memory Tables Answer Key," also on the CD, includes completed tables and lists to check your work.

Definitions of Key Terms

Define the following key terms from this chapter, and check your answers in the glossary:

Dual stacks, global unicast address, ISP prefix, link local address, NAT-PT, Neighbor Discovery Protocol (NDP), Regional Internet Registry (RIR), registry prefix, site prefix, stateful DHCP, stateless autoconfiguration, stateless DHCP, subnet prefix, unique local address

Command Reference to Check Your Memory

While you should not necessarily memorize the information in the tables in this section, this section does include a reference for the configuration and EXEC commands covered in this chapter. Practically speaking, you should memorize the commands as a side effect of reading the chapter and doing all the activities in this exam preparation section. To check to see how well you have memorized the commands as a side effect of your other studies, cover the left side of the table with a piece of paper, read the descriptions on the right side, and see whether you remember the command.

Table 17-13 *Chapter 17 Configuration Command Reference*

Command	Description
ipv6 unicast-routing	Global command that enables IPv6 routing on the router
ipv6 router rip *tag*	Global command that enables RIPng
ipv6 rip *name* **enable**	Interface subcommand that enables RIPng on the interface
ipv6 address {*ipv6-address/prefix-length* | *prefix-name sub-bits/prefix-length*} **eui-64**	Interface subcommand that manually configures either the entire interface IP address, or a /64 prefix with the router building the EUI-64 format interface ID automatically
ipv6 host *name ipv6-address1* [*ipv6-address2...ipv6-address4*]	Global command to create a static host name definition
ip name-server *server-address1* [*server-address2...server-address6*]	Global command to point to one or more name servers, to resolve a name into either an IPv4 or IPv6 address
[**no**] **ip domain-lookup**	Global command that enables the router as a DNS client, or with the **no** option, disables the router as a DNS client

Table 17-14 *Chapter 17 EXEC Command Reference*

Command	Description
show ipv6 route	Lists IPv6 routes
show ipv6 route *ip-address*	Lists the route(s) this router would match for packets sent to the listed address
show ipv6 route [*prefix*/*prefix-length*]	Lists the route for the specifically listed prefix/length
show ipv6 interface [*type number*]	Lists IPv6 settings on an interface, including link local and other unicast IP addresses
show ipv6 interface brief	Lists interface status and IPv6 addresses for each interface