CHAPTER 2

IPv6 Overview

This chapter assumes the reader is very familiar with IPv4 addressing and operation. IPv6 is an extension of IP addressing with several advanced features:

- Larger address space.
- Simpler header for increased router efficiency.
- No more broadcasts.
- Stateless autoconfiguration.
- Built-in support for Mobile IP.
- Built-in support for IPsec security.
- Rich transition features.
- Easy IP address renumbering.
- Support for multiple addresses per interface.
- Routers create link-local addresses for use by Interior Gateway Protocols (IGPs).
- As with IPv4, the addresses can be obtained from an Internet service provider (ISP) or can be provider independent.

The primary adoption of IPv6 is driven by the need for more addresses. Given the growth in Internet use and the emergence of large groups of Internet users worldwide, this is a significant requirement. Another reason to use IPv6 is growth in the size of the current Internet routing table. IPv4 addresses are not summarized enough to keep the size down, increasing the load on Internet routers. Additionally, although the use of Network Address Translation (NAT) has postponed the need for IPv6, it breaks TCP/IP's end-to-end networking model.

IPv6 is not enabled by default on Cisco routers. To enable IPv6 routing, the command is **ipv6 unicast-routing** at the global configuration mode.

IPv6 Addressing

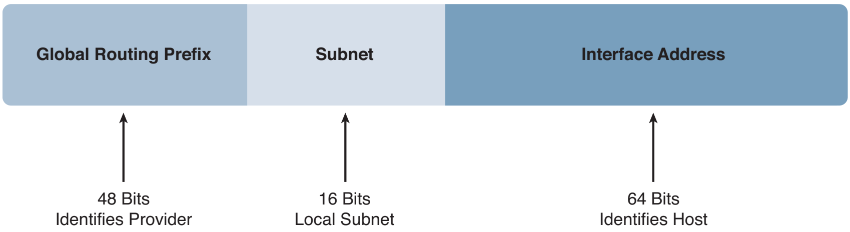
IPv4 addresses are 32 bits long and written in dotted decimal, whereas IPv6 addresses are 128 bits and written in hexadecimal. IPv6 addresses are written in groups of four numbers, separated by colons. Each number represents 4 bits and thus has a value of 0–F. Addresses are typically divided into a 64-bit network portion and a 64-bit host portion. An IPv6 address might look something like 2001:db80:aabb:1111:2222:3333:4444:ffff/64. There is no equivalent of the IPv4 dotted decimal subnet mask—the prefix length (network portion of the address) is written in CIDR notation. Having host addresses that end in 0 or F is legal.

The first 48 bits of the network portion are considered *global address space*. These bits consist of the following elements:

- The first three bits (/3) of a unicast address are always 001.
- The next 13 bits (/16) identify the top-level aggregator (TLA), the upstream ISP or address authority.
- The subsequent 24 bits (/40) identify the next-level aggregator, a regional ISP or address authority.

The next 16 bits are available for creating subnetworks within the enterprise. The host interface portion of the address is the last 64 bits. Because IPv6 does not use broadcasts and separates the interface address portion from the network portion, interface addresses of all zeroes and all ones are legal. Figure 2-1 shows the address components.

Figure 2-1 IPv6 Address Structure



Simplifying an IPv6 Address

There are two ways to shorten the representation of an IPv6 address. Take the example address 2001:0000:0001:0002:0000:0000:0000:ABCD.

- Leading zeros can be omitted, which reduces the preceding address to 2001:0:1:2:0:0:0:ABCD.
- Sequential zeros can be shown as double colons. *This is allowed only once per address*. This simplifies the aforementioned address even further, to 2001:0:1:2::ABCD.

Be sure that you can distinguish between correct and incorrect IPv6 addresses. For instance, the address 2001::1:2::ABCD is incorrect because it uses double colons twice.

Special Addresses

IPv6 does not support broadcasts but replaces them with multicasts. IPv6 also uses *anycast*, which involves using the same address on multiple devices. Anycast is used to implement redundancy and has been backported to IPv4.

Each IPv6 system must recognize the following addresses:

- Its unicast addresses.
- Link local address (begins with FE80/10).
- Loopback (::1/128).
- All-nodes multicast (FF00::1).
- Site-local multicast (FF02::2).
- Solicited-nodes multicast (FF02::1:FF00/104).
- Default route (::/0).
- Routers must support subnet-router anycast (all zeros EUI-64).
- Routers must support local all-routers multicast (FF01::2), link-local (FF02::2), and site-local (FF05:2).
- Routers must support routing protocol multicast groups.

Additionally, some systems also use the following addresses:

- IPv4-compatible address (::/96 | 32-bit, IPv4 address)
- Second unicast address shared with another system (anycast)
- Additional multicast groups

IPv6 Host Addressing

IPv6 uses one special type of address called a *link-local address*. In IPv6, this is composed of the network prefix FE80:: and the host or interface MAC address. This address is valid only on the local network segment. To make sure that the address is not already being used, a host sends a neighbor solicitation message to that address to make sure no other device answers. Even if you assign a routable, global address to the interface, it still has its link-local address. IGP routing protocols use the link-local address to form neighbor relationships. The link-local address is also the next-hop address that is installed in the routing table by IGPs. You can see an example of a link-local address later in Example 2-2.

An IPv6 host can obtain a routable IP address by manual assignment; physically designating the network part of the address only using Stateless Address Autoconfiguration (SLAAC) or by using DHCPv6. DHCPv6 is covered in Chapter 9, "Infrastructure Services."

To ping any IPv6 address, including link-local addresses, use the command **ping** [**ipv6**] *destination-address* [**source** *exit-interface*]. Note that the source interface is required when pinging a link-local address.

Neighbor Discovery Protocol

Neighbor Discovery Protocol uses ICMPv6 and assists in IPv6 addressing in several ways:

- **Duplicate address discovery (DAD):** The host uses neighbor solicitation (NS) to send a message to its own address. No response means that the address is unique. This is used when a host creates its link-local address.

- **Neighbor discovery:** Similarly to ARP, the host discovers the link-local address of neighbors using an NS message. This is ICMP type 135. Neighbors respond with an ICMP type 136 message.

- **Router discovery:** IPv6 routers periodically send router advertisements (RAs) listing the network prefix. When a host comes online, it immediately sends a router solicitation (RS) message asking for prefix information rather than waiting for the RA. This is sent to the all-routers multicast address.

Manual IP Address Assignment

To manually assign an IPv6 address to a router interface, use the command **ipv6** *address* ipv6-address/prefix-length. Example 2-1 shows two router interfaces configured with IPv6 addresses. In the first address, note that leading zeroes are omitted in two of the quartets. In the second address, note the use of the double colons.

Example 2-1 IPv6 Address Configuration

```
Router(config)# ipv6 unicast-routing
!
Router(config)# interface gigabitethernet0/0
Router(config-if)# ipv6 address 2001:db8:aabb:1:2222:3333:4444:f
  fff/64
!
Router(config)# interface gigabitethernet0/1
Router(config-if)# ipv6 address 2001:db8:aabb:2::1 /64
```

Manual Network Assignment

Although statically assigning IP addresses to router interfaces is a best practice, routers can create their own IPv6 address when they know the network prefix. Assuming the end system has a 48-bit MAC address, the router or host flips the global/local bit (the seventh bit) and inserts 0xFFEE into the middle of the MAC address. The resulting 64-bit number is called the *EUI-64 address*. The prefix and EUI-64 address are concatenated to form the host IPv6 address. The command is **ipv6 address** *ipv6-prefix*::/*prefix-length* **eui-64**.

Example 2-2 shows this command and the resulting link-local and global unicast address. Note the interface MAC address and how it relates to the IPv6 addresses.

Example 2-2 Configuring an EUI-64 Address

```
Router(config)# interface ethernet0/0
Router(config-if)# ipv6 address 2001:db8:1234:aabb::/64 eui-64
!
Router# show int e0/0
Ethernet0/0 is up, line protocol is up
 Hardware is AMDP2, address is 001d.a188.33c1 (bia 001d.a188.33c1)
<output omitted>
!
Router# show ipv6 int e0/0
Ethernet0/0 is up, line protocol is up
 IPv6 is enabled, link-local address is FE80::21D:A1FF:FE88:33C1
 No Virtual link-local address(es):
```

Example 2-2 Continued

```
 Global unicast address(es):
  2001:DB8:1234:AABB:21D:A1FF:FE88:33C1,subnet is
 2001:DB8:1234:AABB::/64[EUI]
 Joined group address(es):
  FF02::1
  FF02::2
  FF02::1:FF00:1A00
<output omitted>
```

Stateless Address Autoconfiguration

One big benefit of IPv6 is the capability of a host to automatically acquire an IP address without DHCP, called *Stateless Address Autoconfiguration* or *SLAAC*. To enable stateless autoconfiguration, use the interface command **ipv6 address autoconfig**. Acquiring an address involves the following steps:

Step 1. The host creates a link-local address.

Step 2. It sends an NS message to its link-local address out the interface.

Step 3. If there is no reply, DAD declares the address unique.

Step 4. If the host doesn't receive an RA, it sends an RS.

Step 5. A router on the subnet sends an RA, listing its interface prefix.

Step 6. The host uses that prefix and the interface MAC address to create its IPv6 address.

Use the command **show ipv6 interface** to verify your configuration. Example 2-3 shows this command and the resulting IPv6 address.

Example 2-3 IPv6 Autoconfiguration

```
Router(config)# int e0/0
Router(config-if)# ipv6 address autoconfig
!
Router# show ipv6 int e0/0
Ethernet0/0 is up, line protocol is up
 IPv6 is enabled, link-local address is FE80::21D:A1FF:FE6C:D238
 No Virtual link-local address(es):
 Global unicast address(es):
  2001:DB8::21D:A1FF:FE6C:D238, subnet is 2001:DB8::/64 [EUI/CAL/
 PRE]
   valid lifetime 2591828 preferred lifetime 604628
<output omitted>
```

Securing NDP

NDP is an essential part of IPv6, enabling SLAAC and allowing MAC addresses to be learned and hosts to be redirected to better routes. But because it starts operating when IPv6 unicast routing is enabled, it can expose the router to problems such as man-in-the-middle attacks and blackholing of traffic.

If there is no need for router advertisement messages on a link, disable them with the command **ipv6 nd suppress-ra**.

Another option is to use Secure Neighbor Discovery (SeND). SeND is an extension to NDP that provides some additional security, such as

- Proof of address ownership to prevent address hijacking
- Message integrity and replay protection
- Router authorization for specific prefixes only

SeND requires a pair of public and private keys for all IPv6 nodes. Its deployment can become complex and its use can be CPU intensive.

DHCPv6

DHCP can also be used to assign IP addresses on an IPv6 network. DHCP for IPv6 is covered in Part I in the "DHCP" section of Chapter 9.

Renumbering

IPv6 supports easy network renumbering. Note in Example 2-3 that lifetimes are listed for the subnet address. When it is time to change the subnet, simply configure the router to advertise the old prefix with a short lifetime and the new prefix with a longer one. You can even configure the router to expire a prefix at a certain date and time. The router sends out an RA with both prefixes and their lifetimes. Hosts then update their addresses. Anyone who has had to renumber a large range of IPv4 addresses can testify to how useful this feature is!

IPv6 Routing

Routing with IPv6 will seem familiar to you. The same IGPs—RIP, EIGRP, and OSPF—are used as in IPv4; they have been adapted to carry IPv6 routes. BGP extensions enable it to perform IPv6 routing. The same rules for metric and administrative distance apply. The commands are similar, too. The main difference in commands is that you must specify that the command pertains to IPv6 because IPv4 is the default. One big configuration difference is that IGPs no longer use the **network** command to initiate routing; it is enabled at each interface instead. BGP still uses the network command to designate which networks to advertise.

The specific commands and examples are given in the chapters for each routing protocol. The following is some general information that applies to all protocols.

Static Routing

Static routing with IPv6 works exactly like it does with version 4. Aside from understanding the address format, there are no differences. The syntax for the IPv6 static route command is

```
Router(config)# ipv6 route ipv6-prefix/prefix-length {ipv6-
  address | interface-type
interface-number [ipv6-address]} [administrative-distance]
[administrative-multicast-distance | unicast | multicast] [tag
  tag]
```

The following configuration shows the command in context as it might be applied. The first line shows a static route that lists a next-hop address. The second line shows a directly connected static default route that lists an outbound interface. The third line shows a fully specified static route, which lists both the next-hop address and the outbound interface:

```
Router(config)# ipv6 route 2001:db8:1:2::/64 2001:db8:1:1::1
!
Router(config)# ipv6 route ::/0 serial1/0/0
!
Router(config)# ipv6 route 2001:db8:1:2::/64 serial1/0/1
  2001:db8:1:1::1
```

Verify your configuration with the **show ipv6 route** command.

IPv6 Route Summarization

You must summarize IPv6 routes for the same reasons as you summarize IPv4 routes. There might even be a greater need because you can have so many more IPv6 routes! Originally, the idea was to use the 16-bit local subnet portion of the address for local routes, but you can use part of the 64 bits allocated for host addresses for subnetting, too. As in IPv4, this just gives you fewer possible hosts. After you understand the concept, you can apply it to any portion of the IPv6 address you need.

Recall that each number in an IPv6 address is 4 bits, which gives you 16 possible values: 0–F. Each of the quartets separated by colons is thus 16 bits. For instance, consider the following set of networks. How would you summarize them?

- 2001:db8:1:0::/64
- 2001:db8:1:1::/64
- 2001:db8:1:2::/64
- 2001:db8:1:3::/64

The fourth quartet is the one we are interested in. The binary equivalents for the fourth quartet of the preceding networks are

- 0000 0000 0000 0000
- 0000 0000 0000 0001
- 0000 0000 0000 0010
- 0000 0000 0000 0011

You can see that the last two bits are the ones you can summarize. It doesn't matter whether their value is 0 or 1. You need only to focus on the value of the first 14 bits in this quartet. The bit mask for the summary breaks down this way:

```
  2001:     db8:     1:          0::
16 bits + 16 bits + 16 bits + 14 bits = 62 bits
```

Thus, the summarized address is 2001:db8:1:0::/62.

Integrating IPv4 and IPv6

Several strategies exist for migrating from IPv4 to IPv6. Each strategy must be considered when organizations decide to make the move to IPv6 because each has positive points to aiding a smooth migration. It should also be said that there does not have to be a global decision on strategy—your organization might choose to run dual-stack in the United States, convert completely to IPv6 in Japan, and use tunneling in Europe. The transition mechanisms include

- **Dual stack:** Running IPv6 and IPv4 concurrently on the same interface.

- **Tunneling:** Routers that straddle the IPv4 and IPv6 worlds encapsulate IPv6 traffic inside IPv4 packets.
- **Translation:** Using an extension of NAT, NAT64, to translate between IPv4 and IPv6 addresses. This is covered in Chapter 9.

Tunneling IPv6 over IPv4

A tunnel serves as a virtual point-to-point link between IPv6 domains. It doesn't matter what the underlying IPv4 structure is if there is IP reachability between the tunnel endpoints.

There are several ways to tunnel IPv6 over IPv4:

- Manual tunnels
- GRE tunnels
- 6to4 tunnels
- Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)

Manual Tunnels

When you manually create the tunnel, the source and destination IP addresses are IPv4 addresses because IPv4 is the transport protocol. Use loopback addresses for increased stability. IPv6 addresses go on the tunnel interfaces because IPv6 is the passenger protocol. Because IPv6 considers the tunnel a point-to-point link, the address of each end of the tunnel is in the same subnet. Include the command **tunnel mode ipv6ip** in tunnel configuration mode to enable IPv6 over IP encapsulation.

To verify your configuration use the commands **debug tunnel** or **show interface tunnel** *int-number*.

GRE Tunnels

GRE is the default tunnel mode for Cisco routers. It provides more flexibility because it is protocol-agnostic. It can carry multiple protocols, including IPv6 and routing protocols.

Configuring an IPv4 GRE tunnel to carry IPv6 traffic is the same as configuring a manual tunnel except you do not have to specify the tunnel mode because GRE is the default. You can allow a routing protocol on the tunnel interface, too. The process is the same as enabling it on a physical interface.

To configure a completely IPv6 GRE tunnel, use IPv6 interface addresses as the tunnel source and destination. Give the tunnel endpoints IPv6 addresses, too. You need a command to identify that the transport protocol is IPv6. That command, given in tunnel configuration mode, is **tunnel mode gre ipv6**.

6to4 Tunnels

This technique dynamically creates tunnels that IPv6 considers point-to-multipoint interfaces. Use the reserved prefix 2002::/16 in your IPv6 domain, and then add the IPv4 address of the dual-stack router on the other side of the IPv4 domain as the next 32 bits of the network address. This means you must translate that IP address into hexadecimal.

When IPv6 traffic arrives at an edge dual-stack router with a destination IPv6 prefix of 2002::/16, the router examines the first 48 bits, derives the embedded IPv4 address from them, and uses it to determine the packet destination. The router then encapsulates the IPv6 packet in an IPv4 packet with the extracted IPv4 address as the packet destination.

Configure a tunnel as before, using IPv4 addresses as the source, but do not manually specify a destination. Give the tunnel an IPv6 address as previously described, with the tunnel destination embedded in its prefix. The tunnel mode command is **tunnel mode ipv6ip 6to4**.

Each router needs a route to its peer on the other side of the IPv4 network. The only current options for this are static routes and BGP.

ISATAP Tunnels

ISATAP tunnels are similar to the other two tunnel techniques in that an IPv4 address is encoded into the IPv6 address. It is meant to be used within a site, between hosts and routers, although it can be used between sites.

The tunnel source address is an IPv4 address. Do not specify a tunnel destination. The IPv6 address of the tunnel itself combines the network prefix, 0000:5EFE, and the 32-bit IPv4 tunnel source address. The IPv4 address is encoded into the least significant 32 bits of the address. Use any network prefix. The tunnel interface link-local address still starts with FE80 and then uses 0000:5EFE plus the encoded IPv4 address.

For instance, the link-local address of a tunnel that uses 10.8.8.8 as its source is

```
FE80::5EFE:A08:808
```

The unicast IPv6 address of that same tunnel interface, assuming that prefix 2001:db8:1:3/64 was assigned to the interface, is

```
2001:db8:1:3:0:5EFE:A08:808
```

ISATAP tunnels do not support multicast. A route is needed to the tunnel destination if it is in a different subnet; this can be either a static route or a BGP route.

IPv6 Link Types

IPv6 recognizes three types of links:

- Point-to-point
- Point-to-multipoint
- Multiaccess

Point-to-Point Links

Recall that an IPv6 interface uses its MAC address to create its link-local address. A serial link has no MAC address associated with it, so it uses one from an Ethernet interface. You can manually configure the link-local address to make it more recognizable. Be sure to begin the IPv6 address with the link-local prefix FE80.

Point-to-point links do not necessarily need global unicast addresses. The routers can communicate with only link-local addresses, but you could not reach those interfaces from off the network because the link-local is not a routable address.

Point-to-Multipoint Links

For point-to-multipoint links, such as Frame Relay, you must map the destination IPv6 address to the correct DLCI, just as with IPv4. The difference is that with IPv6 you must also map the link-local address to the DLCI because it is used as the next hop for routing. So for each DLCI, you must have at least two mappings: the remote router's IPv6 global unicast address and the remote router's link-local address. The map command is

```
frame relay map ipv6 destination-address out dlci dlci-number
 broadcast
```

In a hub-and-spoke topology, the hub must be configured for IPv6 unicast routing for the spokes to communicate with each other.

Multiaccess Links

Devices on multiaccess links, such as Ethernet, build a table mapping destination Layer 3 addresses to Layer 2 addresses, whether you use IPv4 or IPv6. IPv4 uses a separate protocol, ARP, to accomplish this. In IPv6, the process is built in to the IPv6 protocol and uses the Neighbor Discovery Protocol. An IPv6 device sends a neighbor solicitation (NS) multicast with a prefix of FE02. The neighbor responds with a neighbor advertisement (NA) message listing its MAC address. As with ARP, these mappings have a set lifetime called the *reachable time*, so an NS can also be sent periodically to verify that a neighbor is still reachable.

To add a static entry to the Neighbor Discovery table, use the command **ipv6 neighbor** *ipv6-address interface-type interface-number hardware-address*. A static address does not age out of the table.

Display the mappings with the **show ipv6 neighbors** command.