IPv6 Fundamentals

A Straightforward Approach to Understanding IPv6

Second Edition

ciscopress.com

Rick Graziani

IPv6 Fundamentals: A Straightforward Approach to Understanding IPv6

Second Edition

Rick Graziani

Cisco Press

800 East 96th Street

Indianapolis, IN 46240

The **ipconfig /all** command in Example 6-4 shows the link-local addresses and the MAC address for WinPC in the topology. Windows hosts generated their own link-local addresses using fe80::/64 and a 64-bit random number for the Interface ID. Two factors that indicate the link-local Interface ID was not generated using EUI-64:

- The link-local address does not contain the Ethernet MAC address (physical address) in the Interface ID.
- The link-local address does not contain fffe in the middle of the Interface ID.

Example 6-4 *IPv6 Configuration on WinPC*

```
WinPC> ipconfig /all
Windows IP Configuration
<output omitted for brevity>

Ethernet adapter Local Area Connection:
   Connection-specific DNS Suffix  . :
   Description: Intel<R> PRO/1000 MT Network Connection
   Physical Address: 00-50-56-AF-97-68
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled:. . . . : Yes
   IPv6 Address. . . . . . . . . . . : 2001:db8:cafe:1::100
   Link-local IPv6 Address . . . . . : fe80::d0f8:9ff6:4201:7086%11
<output omitted for brevity>
```

Zone ID (%) on Link-Local Interfaces

You might have noticed in Example 6-4 that %11 follows the link-local address. This is known as the Zone ID, or the Scope ID or Interface Scope. Operating systems such as Windows, Linux, and Mac OS use the Zone ID to associate a link-local address with a specific interface. The Zone ID helps determine which interface to use when sending packets destined for a link-local address. This is particularly important when a device has multiple interfaces, each on a separate link (subnet). The Zone ID is only locally significant.

Remember that link-local addresses have to be unique only on that link (subnet). A computer with two interfaces will have two link-local addresses. When sending a packet to a link-local address, the device needs to know on which interface to send that packet. This is where the Zone ID is used to specify the proper exit interface.

Example 6-5 illustrates the use of the Zone ID on a Windows host. This Windows host has two interfaces, an Ethernet NIC and a wireless NIC, both enabled for IPv6. The **ipconfig** command displays the Windows host's two link-local addresses in two separate zones:

- **%11:** The Zone ID for the Ethernet LAN
- **%12:** The Zone ID for the wireless LAN

Notice that both interfaces are on the same link (subnet) and have the same default gateway address. Although interfaces are on the same link (subnet), their Zone IDs differ. The **netsh interface ipv6 show interfaces** command shows the internal index used as the Zone ID. Again, notice the Zone ID of 11 for the Ethernet Local Area Connection and 12 for the wireless network.

Example 6-5 *Windows Host Link-Local Address and Zone ID*

```
Windows-Host> ipconfig

Windows IP Configuration

Wireless LAN adapter Wireless Network Connection:

   Connection-specific DNS Suffix  . :
   IPv6 Address. . . . . . . . . . . : 2001:db8:face:1::aaaa
   Link-local IPv6 Address . . . . . : fe80::6c51:4f86:ff70:67f5%12
   Default Gateway . . . . . . . . . : fe80::481d:70ff:fe6f:9503%12

Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . :
   IPv6 Address. . . . . . . . . . . : 2001:db8:face:1::bbbb
   Link-local IPv6 Address . . . . . : fe80::9d23:50de:14ce:c8ab%11
   Default Gateway . . . . . . . . . : fe80::481d:70ff:fe6f:9503%11

Windows-Host> netsh interface ipv6 show interfaces

Idx     Met         MTU          State                Name
---  ----------  ----------  ------------  ---------------------------
  1          50  4294967295  connected     Loopback Pseudo-Interface 1
 12          10        1500  connected     Wireless Network Connection
 25          50        1280  disconnected  isatap
 11          10        1500  connected     Local Area Connection
 16          50        1280  disconnected  Teredo Tunneling Pseudo-Interface
Windows-Host>
```

Note Linux and Mac OS X devices use %eth (Linux) and %en (Mac) followed by the Zone ID or Interface Scope value.

Example 6-6 shows examples of this same Windows host pinging the link-local address of the default gateway. Notice that the first two examples are successful pings using the Zone IDs %11 and %12 appended to the link-local address of the destination, the default gateway. The third example is a successful ping to the same address but without including the Zone ID. The fourth example is a ping to the default gateway's link-local address, but the pings are unsuccessful because of a wrong Zone ID.

Note Linux and Mac OS X both require the use of the Zone ID when pinging a link-local address. This is demonstrated later in this chapter.

Example 6-6 *Windows Host Pinging the Default Gateway Using the Zone ID*

```
Windows-Host> ping fe80::481d:70ff:fe6f:9503%11

Pinging fe80::481d:70ff:fe6f:9503%11 with 32 bytes of data:
Reply from fe80::481d:70ff:fe6f:9503%11: time=2ms
Reply from fe80::481d:70ff:fe6f:9503%11: time=1ms
<output omitted for brevity>

Windows-Host> ping fe80::481d:70ff:fe6f:9503%12

Pinging fe80::481d:70ff:fe6f:9503%12 with 32 bytes of data:
Reply from fe80::481d:70ff:fe6f:9503%12: time=13ms
Reply from fe80::481d:70ff:fe6f:9503%12: time=4ms
<output omitted for brevity>

Windows-Host> ping fe80::481d:70ff:fe6f:9503

Pinging fe80::481d:70ff:fe6f:9503 with 32 bytes of data:
Reply from fe80::481d:70ff:fe6f:9503: time=4ms
Reply from fe80::481d:70ff:fe6f:9503: time=4ms
<output omitted for brevity>

Windows-Host> ping fe80::481d:70ff:fe6f:9503%16

Pinging fe80::481d:70ff:fe6f:9503%16 with 32 bytes of data:
Request timed out.
Request timed out.
<output omitted for brevity>
```

So, when do you need to use the Zone ID? Only when there are multiple interfaces and you are communicating to a link-local address. It also depends on your operating system.

Note For more information regarding the Zone ID, see RFC 4007, *IPv6 Scoped Address Architecture*. A Zone ID is not necessary when there are multiple global unicast address because of the default source address selection process, which is discussed in Chapter 9.

Later in this chapter, you will see an example of pinging a link-local address from a Cisco IOS router. You will see that both the link-local address and exit interface are needed, much like the Zone ID with Windows. You will also see examples of pinging a link-local address from a Windows, Linux, and Mac OS host.

Manual Configuration of a Link-Local Address

Dynamically assigned link-local addresses are ideal for most devices, such as hosts, including clients and servers. So in the majority of cases, there is no need to change the default behavior of having the link-local address automatically created. It might be important to some to use a randomly generated Interface ID rather than EUI-64 for privacy reasons, but there is typically no need to be able to remember or identify the link-local address for end systems such as PCs, servers, temperature sensors, etc.

However, there are certain devices, such as routers, where it is helpful to identify the link-local address to a device. Dynamic routing protocols such as EIGRP for IPv6 and OSPFv3 use the link-local address as the next-hop address in the routing table, for establishing adjacencies, and for other messages. The disadvantage of an automatically generated link-local address is that the long Interface ID (up to 16 hexadecimal digits) is difficult to recognize. It's much easier to use a simpler, manually configured link-local address that is easier to identify.

The command for configuring a static link-local unicast address is as follows:

```
Router(config-if)# ipv6 address ipv6-address link-local
```

Table 6-2 shows the command for configuring a static link-local unicast address. Notice that the **link-local** parameter is required after the *ipv6-address*.

Table 6-2 *Configuring a Static Link-Local Unicast Address*

Command	Description
Router(config)# **interface** *interface-type interface-number*	Specifies the interface type and interface number.
Router(config-if)# **ipv6 address** *ipv6-address* **link-local**	Specifies the IPv6 link-local address. The **link-local** parameter is required.

Figure 6-7 shows the topology with static link-local addresses assigned to each interface. Example 6-7 shows the configuration of these addresses on routers R1, R2, and R3. For each router, the same link-local address is configured for each interface. This makes it easy to recognize the link-local addresses for that router. As mentioned earlier, this is done to keep things simple. It is a practical method on routers with point-to-point links, but there are other options for routers with client-facing links.

In this simple topology, R1 has the Interface ID ::1 on all its interfaces, R2 has the Interface ID ::2 on all its interfaces, and Router R3 has the Interface ID ::3 on all its interfaces. Remember that a link-local address has to be unique only for that link because it is not routable off the link.

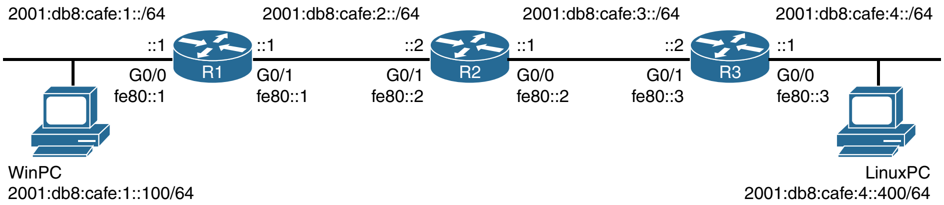

Figure 6-7 *Topology Used to Configure Static Link-Local Addresses*

Example 6-7 *Configuring Static Link-Local Unicast Addresses on R1, R2, and R3*

```
R1(config)# interface gigabitethernet 0/0
R1(config-if)# ipv6 address fe80::1 ?
  link-local  Use link-local address

R1(config-if)# ipv6 address fe80::1 link-local
R1(config-if)# exit
R1(config)# interface gigabitethernet 0/1
R1(config-if)# ipv6 address fe80::1 link-local
------------------------------------------
R2(config)# interface gigabitethernet 0/0
R2(config-if)# ipv6 address fe80::2 link-local
R2(config-if)# exit
R2(config)# interface gigabitethernet 0/1
R2(config-if)# ipv6 address fe80::2 link-local
------------------------------------------
R3(config)# interface gigabitethernet 0/0
R3(config-if)# ipv6 address fe80::3 link-local
R3(config-if)# exit
R3(config)# interface gigabitethernet 0/1
R3(config-if)# ipv6 address fe80::3 link-local
```

In Example 6-8, the configuration is verified using the **show ipv6 interface brief** command. Again, notice that for each router, the same link-local address is configured on each interface.

Example 6-8 *Verifying the Static Link-Local Unicast Addresses on R1, R2, and R3*

```
R1# show ipv6 interface brief
GigabitEthernet0/0     [up/up]
    FE80::1
    2001:DB8:CAFE:1::1
GigabitEthernet0/1     [up/up]
    FE80::1
    2001:DB8:CAFE:2::1
```

```
R1#
--------------------------------------------
R2# show ipv6 interface brief
GigabitEthernet0/0      [up/up]
    FE80::2
    2001:DB8:CAFE:3::1
GigabitEthernet0/1      [up/up]
    FE80::2
    2001:DB8:CAFE:2::2
R2#
--------------------------------------------
R3# show ipv6 interface brief
GigabitEthernet0/0      [up/up]
    FE80::3
    2001:DB8:CAFE:4::1
GigabitEthernet0/1      [up/up]
    FE80::3
    2001:DB8:CAFE:3::2
R3#
```

Manual configuration makes it easier to recognize the link-local address and identify the appropriate device. Example 6-9 shows a sample IPv6 routing table. Notice how the manually configured link-local addresses make it easy to identify the next-hop routers fe80::1 for router R1 and fe80::3 for router R3.

Manual configuration of any address moves this address space into the managed address area. These addresses are assigned and need to be tracked by the organization, including ensuring that the addresses don't overlap with addresses that are dynamically assigned.

Note OSPFv3 is used here only to demonstrate the advantage of recognizable link-local addresses. OSPFv3 is discussed in Chapter 16, "OSPFv3."

Example 6-9 *R1's IPv6 Routing Table*

```
R2# show ipv6 route ospf
IPv6 Routing Table - default - 7 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
<output omitted for brevity>
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

O   2001:DB8:CAFE:1::/64 [110/2]
     via FE80::1, GigabitEthernet0/1
O   2001:DB8:CAFE:4::/64 [110/2]
     via FE80::3, GigabitEthernet0/0
R2#
```