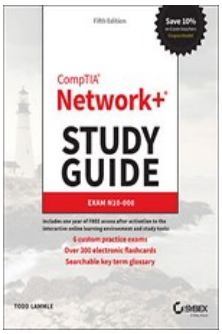

CompTIA Network+ Study Guide: Exam N10-008, 5th Edition

by Todd Lammle

Chapter 7: IP Addressing

The following Comptia Network+ Exam Objectives are Covered in This Chapter

- **1.4 Given a scenario, configure a subnet and use appropriate IP addressing schemes.**
  - Public vs. private
    - RFC1918
    - Network address translation (NAT)
  - IPv4 vs. IPv6
    - Automatic Private IP Addressing (APIPA)
    - Extended unique identifier (EUI-64)
    - Multicast
    - Unicast
    - Anycast
    - Broadcast
    - Link local
    - Loopback
    - Default gateway
  - IPv4 subnetting
    - Classless (variable-length subnet mask**)**
    - Classful
      - A
      - B
      - C
      - D
      - E
    - Classless Inter-Domain Routing (CIDR) notation
  - IPv6 concepts
    - Tunneling
    - Dual stack
    - Shorthand notation
    - Router advertisement
    - Stateless address autoconfiguration (SLAAC)
  - Virtual IP (VIP)

- Subinterfaces

One of the most important topics in any discussion of TCP/IP is IP addressing. An IP address is a numeric identifier assigned to each machine on an IP network. It designates the specific location of a device on the network.

An IP address is a logical address, not a hardware address—the latter is hard-coded on a network interface card (NIC) and used for finding hosts on a local network. IP addressing was designed to allow hosts on one network to communicate with a host on a different network regardless of the type of LANs the hosts are participating in.

Before we get into the more complicated aspects of IP addressing, you need to understand some of the basics. First, I'm going to explain some of the fundamentals of IP addressing and its terminology. Then you'll learn about the hierarchical IP addressing scheme and private IP addresses.

I'll define unicast, multicast, and broadcast addresses and then finish the chapter with a discussion on IPv6. And I promise to make it all as painless as possible.

The reason that we would even discuss IPv6 (besides to cover the objectives, of course) is the lack of IPv4 addresses available for use in future networks, which we need to keep our corporate and private networks and even the Internet running. Basically, we're running out of addresses for all our new hosts! IPv6 will fix this for us.

Note To find Todd Lammle CompTIA videos and practice questions, please see `www.lammle.com`.

IP Terminology

Throughout this chapter, you'll learn several important terms vital to your understanding of the Internet Protocol. Here are a few to get you started:

- **Bit** A *bit* is one binary digit, either a 1 or a 0.
- **Byte** A *byte* is 7 or 8 bits, depending on whether parity is used. For the rest of this chapter, always assume a byte is 8 bits.
- **Octet** An octet, made up of 8 bits, is just an ordinary 8-bit binary number. In this chapter, the terms *byte* and *octet* are completely interchangeable, and they are typically displayed in decimal up to 255.
- **Network Address** This is the designation used in routing to send packets to a remote network—for example, 10.0.0.0, 172.16.0.0, and 192.168.10.0.
- **IP Address** A logical address used to define a single host; however, IP addresses can be used to reference many or all hosts as well. If you see something written as just IP, it is referring to IPv4. IPv6 will always be written as IPv6.
- **Broadcast Address** The *broadcast address* is used by applications and hosts to send information to all hosts on a network. Examples include 255.255.255.255, which designates all networks and all hosts; 172.16.255.255, which specifies all subnets and hosts on network 172.16.0.0; and 10.255.255.255, which broadcasts to all subnets and hosts on network 10.0.0.0.

The Hierarchical IP Addressing Scheme

An IP address consists of 32 bits of information. These bits are divided into four sections, referred to as octets or bytes, and four octets sum up to 32 bits (8 × 4 = 32). You can depict an IP address using one of three methods:

- Dotted-decimal, as in 172.16.30.56
- Binary, as in 10101100.00010000.00011110.00111000
- Hexadecimal, as in AC.10.1E.38

Each of these examples validly represents the same IP address. Hexadecimal is used with IPv6, and IP addressing uses dotted-decimal or binary, but you still might find an IP address stored in hexadecimal in some programs. Windows is a good example of a program that stores a machine's IP address in hex. Windows 10 (and all other Windows versions) store the IP addresses in hexadecimal subkeys in HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces.

The 32-bit IP address is known as a structured, or hierarchical, address as opposed to a flat, or nonhierarchical, address.

Although either type of addressing scheme can be used, *hierarchical addressing* has been chosen for a very important reason. The major advantage of this scheme is that it can handle a large number of addresses, namely, 4.3 billion (a 32-bit address space with two possible values for each position—either 0 or 1—gives you 2^{32}, or 4,294,967,296). The disadvantage of the flat-addressing scheme, and the reason it's not used for IP addressing, relates to routing. If every address were unique, all routers on the Internet would need to store the address of each and every machine on the Internet. This would make efficient routing impossible, even if only a fraction of all possible addresses were used.

The solution to this problem is to use a two- or three-level hierarchical addressing scheme that is structured by network and host or by network, subnet, and host.

This two- or three-level scheme is comparable to a telephone number. The first section, the area code, designates a very large area. The second section, the prefix, narrows the scope to a local calling area. The final segment, the customer number, zooms in on the specific connection. IP addresses use the same type of layered structure. Rather than all 32 bits being treated as a unique identifier, as in flat addressing, a part of the address is designated as the network address and the other part is designated as either the subnet and host or just the host address.

Next, I'm going to cover IP network addressing and the different classes of addresses used for our networks.

Network Addressing

The *network address*—also called the network number—uniquely identifies each network. Every machine on the same network shares that network address as part of its IP address. In the IP address 172.16.30.56, for example, 172.16 is the network address (and in just a minute I'll show you how this is true).

The *host address* is assigned to and uniquely identifies each machine on a network. This part of the address must be unique because it identifies a particular machine—an individual—as opposed to a network, which is a group. So in the sample IP address 172.16.30.56, the 30.56 is the host address.

The designers of the Internet decided to create classes of networks based on network size. For the small number of networks possessing a very large number of hosts, they created the rank *Class A network*. At the other extreme is the *Class C network*, which is reserved for the numerous networks with a small number of hosts. The class distinction for networks between very large and very small is predictably the *Class B network*.

Subdividing an IP address into a network and host address is determined by the class designation of your network. Figure 7.1 summarizes the classes of networks—a subject I'll explain in greater detail throughout this chapter.

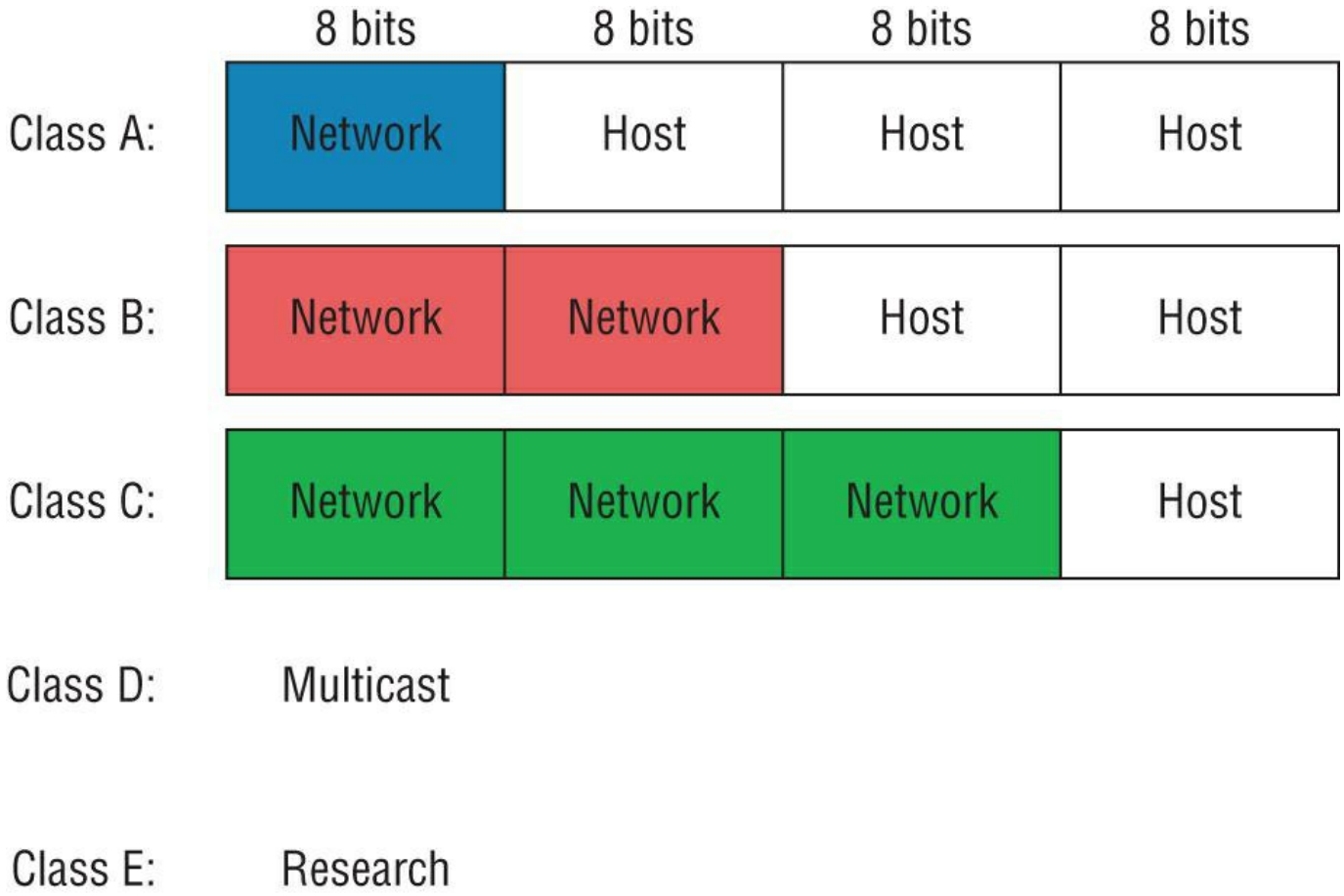

Figure 7.1: Summary of the three classes of networks

To ensure efficient routing, Internet designers defined a mandate for the leading-bits section of the address for each different network class. For example, since a router knows that a Class A network address always starts with a 0, the router might be able to speed a packet on its way after reading only the first bit of its address. This is where the address schemes define the difference between a Class A, a Class B, and a Class C address. Coming up, I'll discuss the differences between these three classes followed by a discussion of the Class D and Class E addresses. For now, know that Classes A, B, and C are the only ranges that are used to address hosts in our networks.

Class A Addresses

In a Class A network address, the first byte is assigned to the network address, and the three remaining bytes are used for the host addresses. The Class A format is as follows:

```
network.host.host.host
```

For example, in the IP address 49.22.102.70, the 49 is the network address and 22.102.70 is the host address. Every machine on this particular network would begin with the distinctive network address of 49.

Class A network addresses are 1 byte long, with the first bit of that byte reserved and the 7 remaining bits available for manipulation or addressing. As a result, the theoretical maximum number of Class A networks that can be created is 128. Why? Well, each of the 7-bit positions can be either a 0 or a 1 and 2^7 gives you 128.

The designers of the IP address scheme said that the first bit of the first byte in a Class A network address must always be off, or 0. This means a Class A address must be between 0 and 127 in the first byte, inclusive.

Consider the following network address:

```
0xxxxxxx
```

If we turn the other 7 bits all off and then turn them all on, we'll find the Class A range of network addresses:

```
00000000 = 0
```

```
01111111 = 127
```

So, a Class A network is defined in the first octet between 0 and 127, and it can't be less or more.

To complicate matters further, the network address of all 0s (0000 0000) is reserved to designate the default route (see Table 7.1). Additionally, the address 127, which is reserved for diagnostics, can't be used either. This means that you can really only use the numbers 1 to 126 to designate Class A network addresses. This means the actual number of usable Class A network addresses is 128 minus 2, or 126.

Table 7.1: Reserved IP addresses

Address	Function
Network address of all 0s	Interpreted to mean "this network or segment."
Network address of all 1s	Interpreted to mean "all networks."
Network 127.0.0.1	Reserved for loopback tests. Designates the local host and allows that host to send a test packet to itself without generating network traffic.
Host address of all 0s	Interpreted to mean "network address" or any host on specified network.
Host address of all 1s	Interpreted to mean "all hosts" on the specified network; for example, 126.255.255.255 means "all hosts" on network 126 (Class A address).
Entire IP address set to all 0s	Used by Cisco routers to designate the default route. Could also mean "any network."
Entire IP address set to all 1s (same as 255.255.255.255)	Broadcast to all hosts on the current network; sometimes called an "all 1s broadcast" or limited broadcast.

Each Class A address has 3 bytes (24 bit positions) for the host address of a machine. This means there are 2^{24}—or 16,777,216—unique combinations and, therefore, precisely that many potential unique host addresses for each Class A network. Because host addresses with the two patterns of all 0s and all 1s are reserved, the actual maximum usable number of hosts for a Class A network is 2^{24} minus 2, which equals 16,777,214. Either way, you can see that's a seriously huge number of hosts to have on a network segment!

Here's an example of how to figure out the valid host IDs in a Class A network address:

- All host bits off is the network address: 10.0.0.0.
- All host bits on is the broadcast address: 10.255.255.255.

The valid hosts are the numbers in between the network address and the broadcast address: 10.0.0.1 through 10.255.255.254. Notice that 0s and 255s can be valid host IDs. All you need to remember when trying to find valid host addresses is that the host bits can't ever be all turned off or all turned on at the same time.

Class B Addresses

In a Class B network address, the first 2 bytes are assigned to the network address and the remaining 2 bytes are used for host addresses. The format is as follows:

```
network.network.host.host
```

For example, in the IP address 172.16.30.56, the network address is 172.16 and the host address is 30.56.

With a network address being 2 bytes (8 bits each), we're left with 2^{16} unique combinations. But the Internet designers decided that all Class B network addresses should start with the binary digit 1, then 0. This leaves 14 bit positions available to manipulate, so in reality, we get 16,384 (that is, 2^{14}) unique Class B network addresses.

In a Class B network, the Request For Comments (RFCs) state that the first bit of the first byte must always be turned on but the second bit must always be turned off. If we turn the other 6 bits all off and then all on, we will find the range for a Class B network:

```
10000000 = 128
10111111 = 191
```

As you can see, a Class B network is defined when the first byte is configured from 128 to 191.

A Class B address uses 2 bytes for host addresses. This is 2^{16} minus the two reserved patterns (all 0s and all 1s), for a total of

65,534 possible host addresses for each Class B network.

Here's an example of how to find the valid hosts in a Class B network:

- All host bits turned off is the network address: 172.16.0.0.
- All host bits turned on is the broadcast address: 172.16.255.255.

The valid hosts would be the numbers in between the network address and the broadcast address: 172.16.0.1 through 172.16.255.254.

Class C Addresses

The first 3 bytes of a Class C network address are dedicated to the network portion of the address, with only 1 measly byte remaining for the host address. Here's the format:

```
network.network.network.host
```

Using the example IP address 192.168.100.102, the network address is 192.168.100 and the host address is 102.

In a Class C network address, the first 3 bit positions are always the binary 110. The calculation is as follows: 3 bytes, or 24 bits, minus 3 reserved positions leaves 21 positions. Hence, there are 2^{21}, or 2,097,152, possible Class C networks.

For Class C networks, the RFCs define the first 2 bits of the first octet as always turned on, but the third bit can never be on. Following the same process as the previous classes, convert from binary to decimal to find the range. Here's the range for a Class C network:

```
11000000 = 192
11011111 = 223
```

So, if you see an IP address with a range from 192 up to 223, you'll know it's a Class C IP address.

Each unique Class C network has 1 byte to use for host addresses. This gets us to 2^8, or 256, minus the two reserved patterns of all 0s and all 1s for a total of 254 available host addresses for each Class C network.

Here's an example of how to find a valid host ID in a Class C network:

- All host bits turned off is the network ID: 192.168.100.0.
- All host bits turned on is the broadcast address: 192.168.100.255.

The valid hosts would be the numbers in between the network address and the broadcast address: 192.168.100.1 through 192.168.100.254.

Class D and E Addresses

Addresses with the first octet of 224 to 255 are reserved for Class D and E networks. Class D (224–239) is used for multicast addresses and Class E (240–255) for scientific purposes. You do need to remember that the multicast range is from 224.0.0.0 through 239.255.255.255. Multicasts will be covered later in this chapter.

Special Purposes of Network Addresses

Some IP addresses are reserved for special purposes, so network administrators can't ever assign them to hosts. Table 7.1 listed the members of this exclusive little club and the reasons they're included in it.

Private IP Addresses (RFC 1918)

The people who created the IP addressing scheme also created what we call *private IP addresses*. These addresses can be used on a private network, but they're not routable through the Internet. This is designed for the purpose of creating a measure of much-needed security, but it also conveniently saves valuable IP address space.

If every host on every network had to have real routable IP addresses, we would have run out of available IP addresses to hand out years ago. But by using private IP addresses, ISPs, corporations, and home users need only a relatively tiny group of bona fide IP addresses to connect their networks to the Internet. This is economical because they can use private IP addresses on their inside networks and get along just fine.

To accomplish this task, the ISP and the corporation—the end users, no matter who they are—need to use something called network address translation (NAT), which basically takes a private IP address and converts it for use on the Internet. NAT provides security in that these IP addresses cannot be seen by external users. External users will only be able to see the public IP address to which the private IP address has been mapped. Moreover, multiple devices in the same private network can use the same, real IP address to transmit out onto the Internet. Doing things this way saves megatons of address space—a very good thing for us all!

Table 7.2 lists the RFC 1918 reserved private addresses.

Real World Scenario: So, What Private IP Address Should I Use?

That's a really great question: Should you use Class A, Class B, or even Class C private addressing when setting up your network? Let's take Acme Corporation in San Francisco as an example. This company is moving into a new building and needs a whole new network (what a treat this is!). It has 14 departments, with about 70 users in each. You could probably squeeze three or four Class C addresses to use, or maybe you could use a Class B, or even a Class A just for fun.

The rule of thumb in the consulting world is that when you're setting up a corporate network—regardless of how small it is—you should use a Class A network address because it gives you the most flexibility and growth options. For example, if you used the 10.0.0.0 network address with a /24 mask, then you'd have 65,536 networks, each with 254 hosts. Lots of room for growth with this network design! You would then subnet this network address space using Classless Inter-Domain Routing (CIDR, also referred to as variable-length subnet mask, or VLSM), which provides only the needed number of hosts to each department or building without wasting IP addresses. (A /24 tells you that a subnet mask has 24 bits out of 32 bits turned on for subnetting a network. This will be covered, as well as CIDR, in more detail in Chapter 8.)

But if you're setting up a home network, you'd opt for a Class C address because it is the easiest for people to understand and configure. Using the default Class C mask gives you one network with 254 hosts—plenty for a home network.

With the Acme Corporation, a nice 10.1.*x*.0 with a /24 mask (the *x* is the subnet for each department) makes this easy to design, install, and troubleshoot.

Table 7.2: Reserved RFC 1918 IP address space

Address Class	Reserved Address Space
Class A	10.0.0.0 through 10.255.255.255 (prefix /8)
Class B	172.16.0.0 through 172.31.255.255 (prefix /12)
Class C	192.168.0.0 through 192.168.255.255 (prefix /16)

Virtual IP (VIP)

When a public IP address is substituted for the actual private IP address that has been assigned to the network interface of the device, the public IP address becomes an example of what is called a *virtual IP address*. This means it doesn't correspond to an actual physical network interface. A well-used example is a subinterface configured on a physical router interface, which allows you to create multiple IPs or subnets on one interface.

There are other examples of such virtual IP addresses. For example, when a web proxy server substitutes its IP address for the sender's IP address before sending a packet to the Internet, it is another example of creating a virtual IP address.

APIPA

I discussed this in Chapter 6, "Introduction to the Internet Protocol," but it is worth repeating here. What happens if you have a few hosts connected together with a switch or hub and you don't have a DHCP server? You can add static IP information to a host or you can use what is called Automatic Private IP Addressing (APIPA). I don't recommend this, but APIPA is a "feature," so you do need to remember it, hence mentioning it two chapters in a row!

With APIPA, clients can automatically self-configure an IP address and subnet mask, which is the minimum information needed

for hosts to communicate when a DHCP server isn't available. In this way, it could be thought of as a DHCP failover scheme. If all of the hosts set themselves with an APIPA address, they could communicate with one another but unfortunately not with any addresses that were statically configured, such as default gateways!

The IP address range for APIPA is 169.254.0.1 through 169.254.255.254. The client also configures itself with a default Class B subnet mask of 255.255.0.0.

However, when you're in your corporate network and you're running a DHCP server and your host displays that it is using this IP address range, this means that either your DHCP client on the host is not working or the DHCP server is down or can't be reached because of a network issue. For example, if you plug a DHCP client into a port that is disabled, the host will receive an APIPA address. I don't know anyone who has seen a host in the APIPA address range and been happy about it! If users cannot connect to the Internet and their IP addresses fall into the APIPA address range, the DHCP server is most likely the problem.

IPv4 Address Types

Most people use *broadcast* as a generic term, and most of the time, we understand what they mean. But not always. For example, you might say, "The host broadcasted through a router to a DHCP server," but, well, it's pretty unlikely that this would ever really happen. What you probably mean—using the correct technical jargon—is, "The DHCP client broadcasted for an IP address; a router then forwarded this as a unicast packet to the DHCP server." Oh, and remember that with IPv4, broadcasts are pretty important, but with IPv6, there aren't any broadcasts sent at all—as you'll see in a bit!

Okay, I've referred to broadcast addresses throughout earlier chapters, and even showed you some examples of various IP addresses. But I really haven't gone into the different terms and uses associated with them yet, and it's about time I did. So here are the four IPv4 address types that I'd like to define for you:

- **Layer 2 Broadcasts** These are sent to all nodes on a LAN.
- **Broadcasts (Layer 3)** These are sent to all nodes on the network.
- **Unicast** This is an address for a single interface, and these are used to send packets to a single destination host.
- **Multicast** These are packets sent from a single source and transmitted to many devices on different networks. This is referred to as *one-to-many*.

Layer 2 Broadcasts

First, understand that layer 2 broadcasts are also known as hardware broadcasts—they only go out on a LAN, and they don't go past the LAN boundary (router).

The typical hardware address is 6 bytes (48 bits) and looks something like 0c.43.a4.f3.12.c2. The broadcast would be all 1s in binary, which would be all *F*s in hexadecimal, as in FF.FF.FF.FF.FF.FF.

Layer 3 Broadcasts

Then there are the plain old broadcast addresses at layer 3. Broadcast messages are meant to reach all hosts on a broadcast domain. These are the network broadcasts that have all host bits on.

Here's an example that you're already familiar with: The network address of 172.16.0.0 would have a broadcast address of 172.16.255.255—all host bits on. Broadcasts can also be "any network and all hosts," as indicated by 255.255.255.255.

A good example of a broadcast message is an Address Resolution Protocol (ARP) request. When a host has a packet, it knows the logical address (IP) of the destination. To get the packet to the destination, the host needs to forward the packet to a default gateway if the destination resides on a different IP network. If the destination is on the local network, the source will forward the packet directly to the destination. Because the source doesn't have the MAC address to which it needs to forward the frame, it sends out a broadcast, something that every device in the local broadcast domain will listen to. This broadcast says, in essence, "If you are the owner of IP address 192.168.2.3, please forward your MAC address to me," with the source giving the appropriate information.

Unicast Address

A unicast address is assigned to a single interface, and this term is used in both IPv4 and IPv6 to describe your host interface IP address.

Multicast Address (Class D)

Multicast is a different beast entirely. At first glance, it appears to be a hybrid of unicast and broadcast communication, but that isn't quite the case. Multicast does allow point-to-multipoint communication, which is similar to broadcasts, but it happens in a different manner. The crux of *multicast* is that it enables multiple recipients to receive messages without flooding the messages to all hosts on a broadcast domain. However, this is not the default behavior—it's what we can do with multicasting if it's configured correctly!

Multicast works by sending messages or data to IP multicast group addresses. Routers then forward copies (unlike broadcasts, which are not forwarded) of the packet out to every interface that has hosts subscribed to a particular group address. This is where multicast differs from broadcast messages—with multicast communication, copies of packets, in theory, are sent only to subscribed hosts. When I say in theory, this means that the hosts will receive, for example, a multicast packet destined for 224.0.0.10 (this is an EIGRP packet and only a router running the EIGRP protocol will read it). All hosts on the broadcast LAN (Ethernet is a broadcast multi-access LAN technology) will pick up the frame, read the destination address, and immediately discard the frame, unless they are in the multicast group. This saves PC processing, not LAN bandwidth. Multicasting can cause severe LAN congestion, in some instances, if not implemented carefully.

There are several different groups that users or applications can subscribe to. The range of multicast addresses starts with 224.0.0.0 and goes through 239.255.255.255. As you can see, this range of addresses falls within IP Class D address space based on classful IP assignment.

Internet Protocol Version 6 (IPv6)

People refer to IPv6 as "the next-generation Internet protocol," and it was originally created as the answer to IPv4's inevitable, looming address-exhaustion crisis. Though you've probably heard a thing or two about IPv6 already, it has been improved even further in the quest to bring us the flexibility, efficiency, capability, and optimized functionality that can truly meet our ever-increasing needs. The capacity of its predecessor, IPv4, pales in comparison—and that's the reason it will eventually fade into history completely.

The IPv6 header and address structure has been completely overhauled, and many of the features that were basically just afterthoughts and addendums in IPv4 are now included as full-blown standards in IPv6. It's well equipped, poised, and ready to manage the mind-blowing demands of the Internet to come.

Why Do We Need IPv6?

Well, the short answer is because we need to communicate and our current system isn't really cutting it anymore—kind of like how the Pony Express couldn't compete with airmail. Just look at how much time and effort we've invested in coming up with slick new ways to conserve bandwidth and IP addresses.

It's reality: the number of people and devices that connect to networks increases each and every day. That's not a bad thing at all—we're finding new and exciting ways to communicate with more people all the time, something that's become integral to our culture today. In fact, it's now pretty much a basic human need. But the forecast isn't exactly blue skies and sunshine because, as I alluded to in this chapter's introduction, IPv4, upon which our ability to communicate is presently dependent, is going to run out of addresses for us to use. IPv4 has only about 4.3 billion addresses available—in theory—and we know that we don't even get to use all of those. There really are only about 250 million addresses that can be assigned to devices. Sure, the use of Classless Inter-Domain Routing (CIDR, also referred to as variable-length subnet mask, or VLSM) and NAT/PAT has helped to delay the inevitable dearth of addresses, but the truth is we will run out of them, and it's going to happen within a few years. China is barely online, compared to their huge population, and corporations there surely want to be. There are a lot of reports that give us all kinds of numbers, but all you really need to think about to convince yourself that I'm not just being an alarmist is the fact that there are about 7.8 billion people in the world today, and it's estimated that just over 59 percent of the population is connected to the Internet—wow! IPv6 to the rescue!

That statistic is basically screaming at us the ugly truth that, based on IPv4's capacity, every person can't have a single computer with an IP address—let alone all the other devices we use with them. I have more than one computer, and it's pretty likely you do, too. And I'm not even including in the mix phones, laptops, game consoles, fax machines, routers, switches, and a mother lode of other devices we use every day! So I think I've made it pretty clear that we've got to do something before we run out of addresses and lose the ability to connect with each other as we know it. And that "something" just happens to be implementing IPv6.

The Benefits of and Uses for IPv6

What's so fabulous about IPv6? Is it really the answer to our coming dilemma? Is it really worth it to upgrade from IPv4? All good questions—you may even think of a few more. Of course, there's going to be that group of people with the time-tested and well-known "resistance-to-change syndrome," but don't listen to them. If we had done that years ago, we'd still be waiting weeks, even months for our mail to arrive via horseback. Instead, just know that the answer is a resounding YES! Not only does IPv6 give us lots of addresses (3.4×10^{38} = definitely enough), but there are many other features built into this version that make it well worth the cost, time, and effort required to migrate to it.

Today's networks, as well as the Internet, have a ton of unforeseen requirements that simply were not considerations when IPv4 was created. We've tried to compensate with a collection of add-ons that can actually make implementing them more difficult than mandating them by a standard. By default, IPv6 has improved upon and included many of those features as standard and mandatory. One of these sweet new standards is IPSec—a feature that provides end-to-end security. Another little beauty is known as *mobility*, and as its name suggests, it allows a device to roam from one network to another without dropping connections.

But it's the efficiency features that are really going to rock the house! For starters, the header in an IPv6 packet has half the fields, and they are aligned to 64 bits, which gives us some seriously souped-up processing speed—compared to IPv4, lookups happen at light speed. Most of the information that used to be bound into the IPv4 header was taken out, and now you can choose to put it, or parts of it, back into the header in the form of optional extension headers that follow the basic header fields.

And of course there's that whole new universe of addresses (3.4×10^{38}) we talked about already. But where did we get them? Did that *Criss Angel Mindfreak* dude just show up and, blammo, they all materialized? The obvious answer is no, but that huge proliferation of addresses had to come from somewhere, right? Well, it just so happens that IPv6 gives us a substantially larger address space, meaning the address is a whole lot bigger—four times bigger, as a matter of fact! An IPv6 address is actually 128 bits in length, and no worries—I'm going to break down the address piece by piece and show you exactly what it looks like coming up in the next section, "IPv6 Addressing and Expressions." For now, let me just say that all that additional room permits more levels of hierarchy inside the address space and a more flexible address architecture. It also makes routing much more efficient and scalable because the addresses can be aggregated a lot more effectively. And IPv6 also allows multiple addresses for hosts and networks. Plus, the new version of IP now includes an expanded use of multicast communication (one device sending to many hosts or to a select group), which will also join in to boost efficiency on networks because communications will be more specific.

IPv4 uses broadcasts very prolifically, causing a bunch of problems, the worst of which is, of course, the dreaded broadcast storm—an uncontrolled deluge of forwarded broadcast traffic that can bring an entire network to its knees and devour every last bit of bandwidth. Another nasty thing about broadcast traffic is that it interrupts each and every device on the network. When a broadcast is sent out, every machine has to stop what it's doing and analyze the traffic, whether the broadcast is meant for it or not.

But smile, everyone: There is no such thing as a broadcast in IPv6 because it uses multicast traffic instead. And there are two other types of communication as well: unicast, which is the same as it is in IPv4, and a new type called *anycast*. Anycast communication allows the same address to be placed on more than one device so that when traffic is sent to one device addressed in this way, it is routed to the nearest host that shares the same address. This is just the beginning—we'll get more into the various types of communication later in this chapter in the section "Address Types."

IPv6 Addressing and Expressions

Just as understanding how IP addresses are structured and used is critical with IPv4 addressing, it's also vital when it comes to IPv6. You've already read about the fact that at 128 bits, an IPv6 address is much larger than an IPv4 address. Because of this, as well as because of the new ways the addresses can be used, you've probably guessed that IPv6 will be more complicated to manage. But no worries! As I said, I'll break it down into the basics and show you what the address looks like, how you can write it, and what many of its common uses are. It's going to be a little weird at first, but before you know it, you'll have it nailed.

So let's take a look at Figure 7.2, which has a sample IPv6 address broken down into sections.

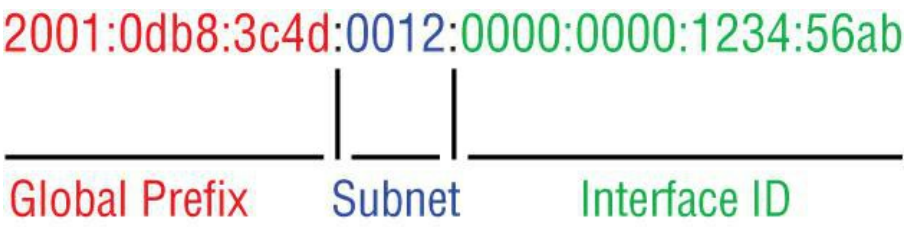

Figure 7.2: IPv6 address example

As you can now see, the address is truly much larger—but what else is different? Well, first, notice that it has eight groups of numbers instead of four, and also that those groups are separated by colons instead of periods. And hey, wait a second…there are letters in that address! Yep, the address is expressed in hexadecimal just like a MAC address is, so you could say this address has eight 16-bit hexadecimal colon-delimited blocks. That's already quite a mouthful, and you probably haven't even tried to say the address out loud yet.

One other thing I want to point out is for when you set up your test network to play with IPv6, because I know you're going to want to do that. When you use a web browser to make an HTTPS connection to an IPv6 device, you have to type the address into the browser with brackets around the literal address. Why? Well, a colon is already being used by the browser for specifying a port number. So basically, if you don't enclose the address in brackets, the browser will have no way to identify the information.

Here's an example of how this looks:

```
https://[2001:0db8:3c4d:0012:0000:0000:1234:56ab]/default.html
```

Now obviously, if you could, you would rather use names to specify a destination (like `www.lammle.com`); but even though it's definitely going to be a pain in the rear, you just have to accept the fact that sometimes you have to bite the bullet and type in the address number. It should be pretty clear that DNS is extremely important when implementing IPv6.

Shortened Expression

The good news is there are a few tricks to help rescue you when you're writing these monster addresses. For one thing, you can actually leave out parts of the address to abbreviate it, but to get away with doing that you have to follow a couple of rules. First, you can drop any leading zeros in each of the individual blocks. After you do that, the sample address from earlier would then look like this:

```
2001:db8:3c4d:12:0:0:1234:56ab
```

That's a definite improvement—at least you don't have to write all of those extra zeros! But what about whole blocks that don't have anything in them except zeros? Well, you can kind of lose those, too—at least some of them. Again, referring to our sample address, you can remove the two blocks of zeros by replacing them with double colons, like this:

```
2001:db8:3c4d:12::1234:56ab
```

Cool—you replaced the blocks of all zeros with double colons. The rule you have to follow to get away with this is that you can only replace one contiguous block of zeros in an address. So if my address has four blocks of zeros and each of them is separated, I don't get to replace them all. Check out this example:

```
2001:0000:0000:0012:0000:0000:1234:56ab
```

And just know that you *can't* use double colons twice, like this:

```
2001::12::1234:56ab
```

Instead, this is the best that you can do:

```
2001::12:0:0:1234:56ab
```

The reason why this example is your best shot is that if you remove two sets of zeros, the device looking at the address will have no way of knowing where the zeros go back in. Basically, the router would look at the incorrect address and say, "Well, do I place two blocks into the first set of double colons and two into the second set, or do I place three blocks into the first set and one block into the second set?" And on and on it would go because the information the router needs just isn't there.

Address Types

We're all familiar with IPv4's unicast, broadcast, and multicast addresses, which basically define who or at least how many other devices we're talking to. But as I mentioned, IPv6 introduces the anycast address type. Broadcasts, as we know them, have been eliminated in IPv6 because of their cumbersome inefficiency.

Since a single interface can have multiple types of IPv6 addresses assigned for various purposes, let's find out what each of these types of IPv6 addresses are and the communication methods of each:

- **Unicast** Packets addressed to a unicast address are delivered to a single interface, same as in IPv4. For load balancing, multiple interfaces can use the same address.
- **Global Unicast Addresses** These are your typical publicly routable addresses, and they're used the same way globally unique addresses are in IPv4.
- **Link-Local Addresses** These are like the APIPA addresses in IPv4 in that they're not meant to be routed and are unique for each link (LAN). Think of them as a handy tool that gives you the ability to throw a temporary LAN together for meetings or for creating a small LAN that's not going to be routed but still needs to share and access files and services locally. However, link-local is used on every LAN that connects to a router interface as well.
- **Unique Local Addresses** These addresses are also intended for nonrouting purposes, but they are nearly globally unique, so it's unlikely you'll ever have one of them overlap with any other address. Unique local addresses were designed to replace site-local addresses, so they basically do almost exactly what IPv4 private addresses do—allow communication throughout a site while being routable to multiple local networks. The difference between link-local and unique local is that unique local can be routed within your organization or company.
- **Multicast** Again, as in IPv4, packets addressed to a multicast address are delivered to all interfaces identified by the multicast address. Sometimes people call them *one-to-many addresses*. It's really easy to spot multicast addresses in IPv6 because they always start with *FF*.
- **Anycast** Like multicast addresses, an anycast address identifies multiple interfaces, but there's a big difference: The anycast packet is delivered to only one address—actually, to the first IPv6 address it finds defined in terms of routing distance. And again, this address is special because you can apply a single address to more than one interface. You could call them one-to-one-of-many addresses, but just saying anycast is a lot easier. This is also referred to as one-to-nearest addressing.

You're probably wondering if there are any special, reserved addresses in IPv6 because you know they're there in IPv4. Well, there are—plenty of them! Let's go over them now.

Special Addresses

I'm going to list some of the addresses and address ranges that you should definitely make a point to remember in Table 7.3 because you'll eventually use them. They're all special or reserved for specific use, but unlike IPv4, IPv6 gives us a galaxy of addresses, so reserving a few here and there doesn't hurt a thing.

Table 7.3: Special IPv6 addresses

Address	Meaning
0:0:0:0:0:0:0:0	Equals ::. This is the equivalent of IPv4's 0.0.0.0 and is typically the source address of a host before the host receives an IP address when you're using DHCP-driven stateful configuration.
0:0:0:0:0:0:0:1	Equals ::1. The equivalent of 127.0.0.1 in IPv4.
0::FFFF:192.168.100.1	This is how an IPv4 address would be written in a mixed IPv6/IPv4 network environment.
2000::/3	The global unicast address range allocated for Internet access.
FC00::/7	The unique local unicast range.
FE80::/10	The link-local unicast range.
FF00::/8	The multicast range.
3FFF:FFFF::/32	Reserved for examples and documentation.
2001:0DB8::/32	Also reserved for examples and documentation.
2002::/16	Used with 6to4 tunneling, which is an IPv4-to-IPv6 transition system. The structure allows IPv6 packets to be transmitted over an IPv4 network without the need to configure explicit tunnels.

Stateless Address Autoconfiguration (SLAAC)

Autoconfiguration is an especially useful solution because it allows devices on a network to address themselves with a link-local unicast address as well as with a global unicast address. This process happens through first learning the prefix information from the router and then appending the device's own interface address as the interface ID. But where does it get that interface ID? Well, you know every device on an Ethernet network has a physical MAC address, which is exactly what's used for the interface ID. But since the interface ID in an IPv6 address is 64 bits in length and a MAC address is only 48 bits, where do the extra 16 bits come from? The MAC address is padded in the middle with the extra bits—it's padded with FF:FE.

For example, let's say I have a device with a MAC address that looks like this: 0060:d673:1987. After it's been padded, it would look like this: 0260:d6FF:FE73:1987. Figure 7.3 illustrates what an EUI-64 address looks like.

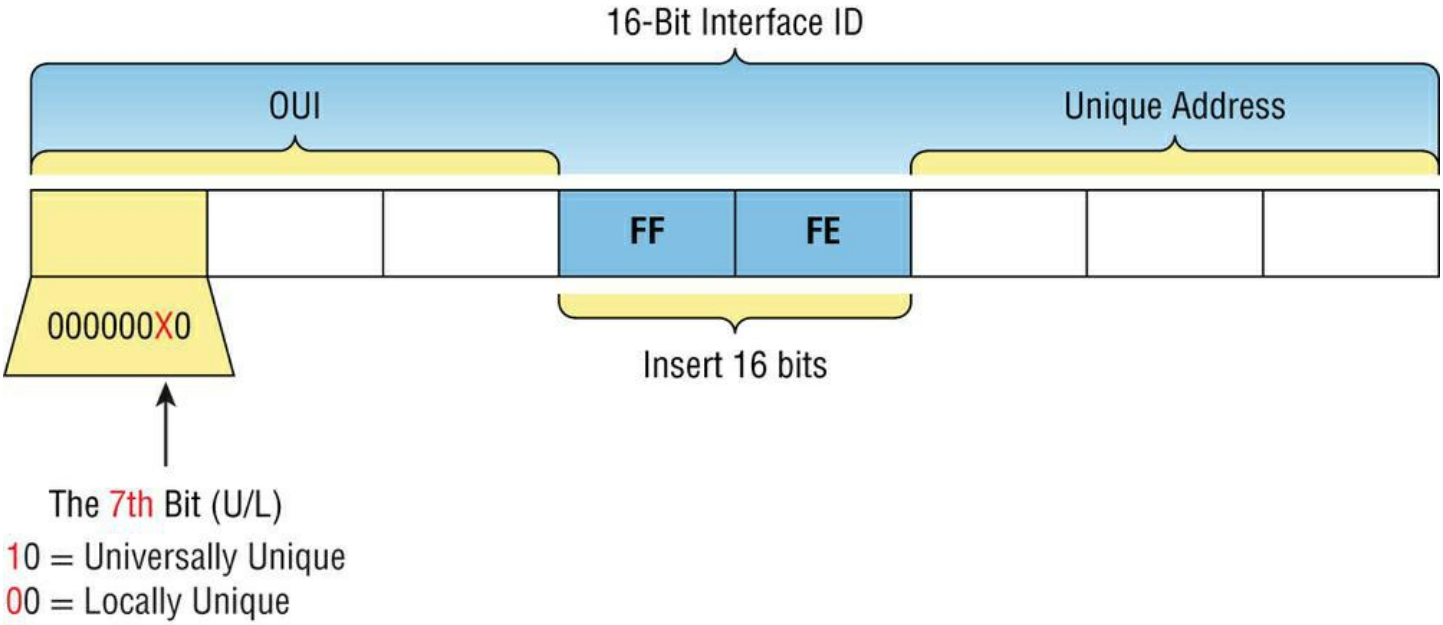

Figure 7.3: EUI-64 interface ID assignment

So where did that 2 in the beginning of the address come from? Another good question. You see that part of the process of padding, called modified EUI-64 format, changes the Universal/Local (U/L) bit to specify if the address is locally unique or globally unique. And the bit that gets changed is the 7th bit in the address.

The reason for modifying the U/L bit is that, when using manually assigned addresses on an interface, you can simply assign the address 2001:db8:1:9::1/64 instead of the much longer 2001:db8:1:9:0200::1/64. Also, if you are going to manually assign link-local addresses, you can assign the short address fe80::1 instead of the long fe80::0200:0:0:1 or fe80:0:0:0:0200::1. So, even though at first glance it seems the IETF made this harder for you to simply understand IPv6 addressing by flipping the 7th bit, in reality this made addressing much simpler. Also, since most people don't typically override the burned-in address, the U/L bit is by default a 0, which means that you'll see this inverted to a 1 most of the time. But because you're studying the exam objectives, you'll need to look at inverting it both ways.

Here are a few examples:

- MAC address 0**0**90:2716:fd0f
- IPv6 EUI-64 address: 2001:0db8:0:1:0**2**90:27ff:fe16:fd0f

That one was easy! Too easy for the exam objectives, so let's do another:

- MAC address a**a**12:bcbc:1234
- IPv6 EUI-64 address: 2001:0db8:0:1:a**8**12:bcff:febc:1234

1010101**1**0 represents the first 8 bits of the MAC address (aa), which when inverting the 7th bit becomes 1010101**0**0. The answer becomes a8. I can't tell you how important this is for you to understand, so bear with me and work through a couple more!

- MAC address 0**c**0c:dede:1234
- IPv6 EUI-64 address: 2001:0db8:0:1:0**e**0c:deff:fede:1234

0c is 00001100 in the first 8 bits of the MAC address, which then becomes 00001110 when flipping the 7th bit. The answer is then 0e. Let's practice one more:

- MAC address 0**b**34:ba12:1234
- IPv6 EUI-64 address: 2001:0db8:0:1:0**9**34:baff:fe12:1234

0b in binary is 00001011, the first 8 bits of the MAC address, which then becomes 00001001. The answer is 09.

Note Pay extra-special attention to this EUI-64 address assignment and be able to convert the 7th bit based on the EUI-64 rules!

DHCPv6 (Stateful)

DHCPv6 works pretty much the same way DHCP does in v4, with the obvious difference that it supports IPv6's new addressing scheme. And it might come as a surprise, but there are a couple of other options that DHCP still provides for us that autoconfiguration doesn't. And no, I'm not kidding—in autoconfiguration, there's absolutely no mention of DNS servers, domain names, or many of the other options that DHCP has always generously provided for us via IPv4. This is a big reason that the odds favor DHCP's continued use in IPv6 into the future at least partially—maybe even most of the time!

This means that you're definitely going to need another server around to supply and dispense all the additional, required information—maybe to even manage the address assignment, if needed!

Migrating to IPv6

We certainly have talked a lot about how IPv6 works and how we can configure it to work on our networks, but what is doing that going to cost us? And how much work is it really going to take? Good questions for sure, but the answers to them won't be the same for everyone. This is because how much you are going to end up having to pony up is highly dependent upon what you've got going on already in terms of your infrastructure. Obviously, if you've been making your really old routers and switches "last" and therefore have to upgrade every one of them so that they're IPv6 compliant, that could very well turn out to be a good-sized chunk of change! Oh, and that sum doesn't even include server and computer operating systems (OSs) and the blood, sweat, and maybe even tears spent on making all your applications compliant. So, my friend, it could cost you quite a bit! The good news is that unless you've really let things go, many OSs and network devices have been IPv6 compliant for a few years—we just haven't been using all their features until now.

Then there's that other question about the amount of work and time. Straight up—this one could still be pretty intense. No matter what, it's going to take you some time to get all of your systems moved over and make sure that things are working correctly. And if you're talking about a huge network with tons of devices, well, it could take a really long time! But don't panic—that's why migration strategies have been created, to allow for a gradual integration. I'm going to show you three of the primary transition strategies available to us. The first is called dual stacking, which allows a device to have both the IPv4 and IPv6 protocol stacks running so it's capable of continuing on with its existing communications and simultaneously running newer IPv6 communications as they're implemented. The next strategy is the 6to4 tunneling approach; this is your choice if you have an all-IPv6 network that must communicate over an IPv4 network to reach another IPv6 network. I'll surprise you with the third one just for fun!

Dual Stacking

This is the most common type of migration strategy because, well, it's the easiest on us—it allows our devices to communicate using either IPv4 or IPv6. Dual stacking lets you upgrade your devices and applications on the network one at a time. As more and more hosts and devices on the network are upgraded, more of your communication will happen over IPv6. Once your migration is complete, everything's running on IPv6 and you get to remove all the old IPv4 protocol stacks you no longer need.

6to4 Tunneling

6to4 tunneling is really useful for carrying IPv6 packets over a network that's still running IPv4. It's quite possible that you'll have IPv6 subnets or other portions of your network that are all IPv6, and those networks will have to communicate with each other. Not so complicated, but when you consider that you might find this happening over a WAN or some other network that you don't control, well, that could be a bit ugly. So what do we do about this if we don't control the whole tamale? Create a tunnel that will carry the IPv6 traffic for us across the IPv4 network, that's what.

The whole idea of tunneling isn't a difficult concept, and creating tunnels really isn't as hard as you might think. All it really comes down to is snatching the IPv6 packet that's happily traveling across the network and sticking an IPv4 header onto the front of it. This is kind of like catch-and-release fishing, except for the fish doesn't get something plastered on its face before being thrown back into the stream.

To get a picture of this, take a look at Figure 7.4.

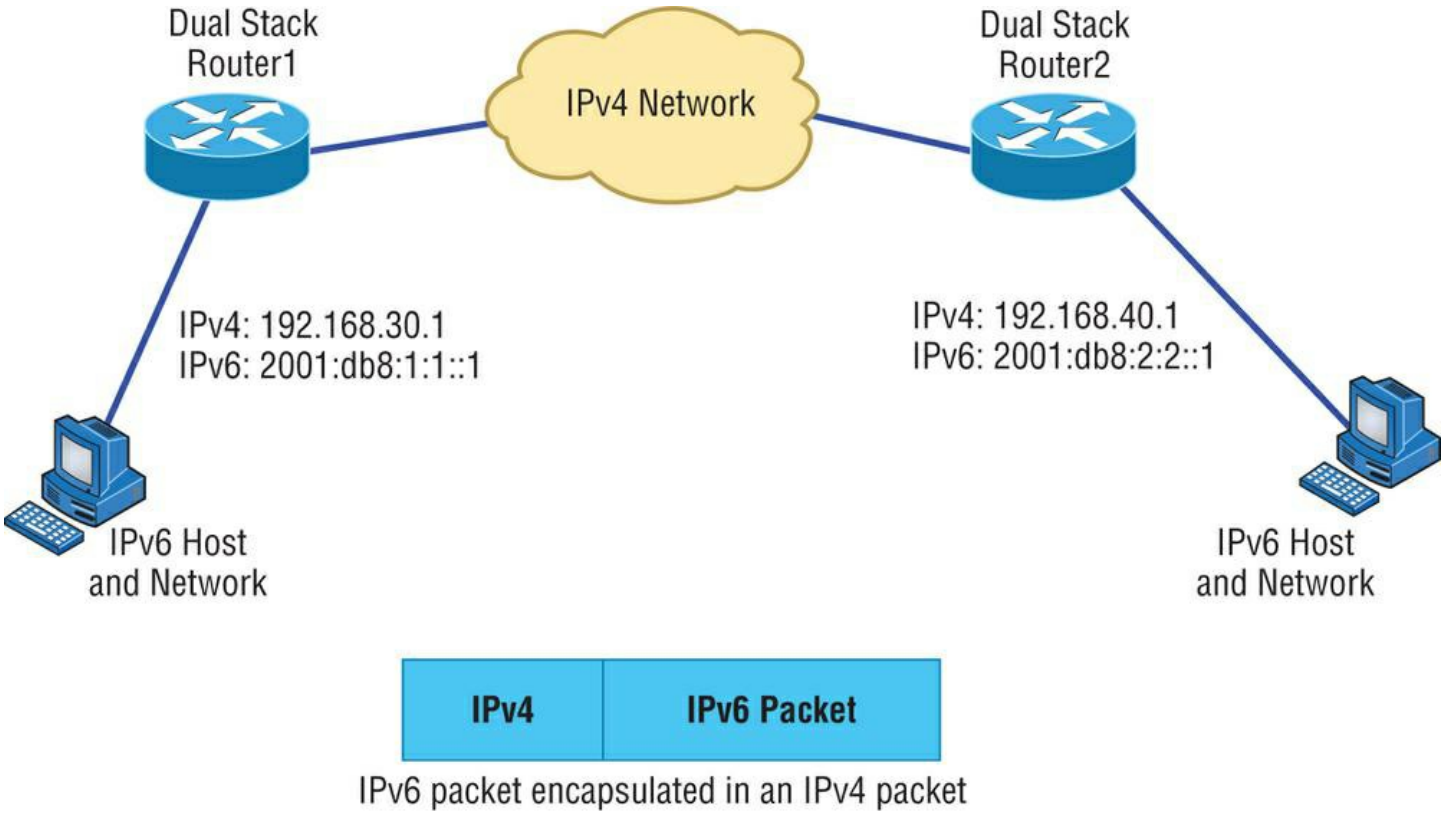

Figure 7.4: A 6to4 tunnel

Nice—but to make this happen, we're going to need a couple of dual-stacked routers, which I just demonstrated for you, so you should be good to go. Now we have to add a little configuration to place a tunnel between those routers. Tunnels are pretty simple—we just have to tell each router where the tunnel begins and where we want it to end up. The opposite of this would be a *4to6 tunnel*, which is rare to find because this means your whole business network is IPv4 (okay, this sounds normal so far) but you're traversing an IPv6-only Internet to get to another IPv4 network. That's not so common at the time of this writing.

One important note here—if the IPv4 network that you're traversing in this 6to4 situation has a NAT translation point, it would absolutely break the tunnel encapsulation we've just created! Over the years, NAT/PAT has been upgraded a lot so that it can handle specific protocols and dynamic connections, and without one of these upgrades, NAT likes to demolish most connections. And since this transition strategy isn't present in most NAT implementations, that means trouble.

But there is a way around this little problem (the third strategy I told you about), and it's called *Teredo*, which allows all your tunnel traffic to be placed in UDP packets. NAT doesn't blast away at UDP packets, so they won't get broken as other protocol packets do. So with Teredo in place and your packets disguised under their UDP cloak, the packets will easily slip by NAT alive and well!

Miredo is a tunneling technique used on native IPv6 Linux and BSD Unix machines to communicate on the IPv4 Internet directly without a dual-stack router or 6to4 tunnel. This is rarely used.

Summary

In this chapter, I covered the very basics of both IPv4 and IPv6 and how they work in an internetwork (remember that if the acronym *IP* is used alone, it is referring to just IPv4). As you now know by reading this chapter, even when discussing and configuring the basics, there is a lot to understand—and we just scratched the surface. We also covered RFC 1918, APIPA addresses, address Classes A-D, NAT, EUI-64, tunneling, dual-stack and virtual IP addressing. But trust me when I say this—you now know more than you'll need to meet the Network+ objectives.

I discussed in detail the difference between each class of address and how to find a network address, broadcast address, and valid host range.

I explained why we need IPv6 and the benefits associated with it. I followed that up by covering addressing with IPv6 as well as how to use the shortened expressions. And during the discussion on addressing with IPv6, I showed you the different address types, plus the special addresses reserved in IPv6.

The next chapter is very important, but it's one that some people find rather challenging, so take a break and get ready for a really fun but long chapter on IP subnetting. I promise not to torture you too much!

Exam Essentials

Remember the Class A range. The IP range for a Class A network is 1 through 126. This provides 8 bits of network addressing and 24 bits of host addressing by default.

Remember the Class B range. The IP range for a Class B network is 128 through 191. Class B addressing provides 16 bits of network addressing and 16 bits of host addressing by default.

Remember the Class C range. The IP range for a Class C network is 192 through 223. Class C addressing provides 24 bits of network addressing and 8 bits of host addressing by default.

Remember the private IP ranges. The Class A private address range is 10.0.0.0 through 10.255.255.255.

The Class B private address range is 172.16.0.0 through 172.31.255.255.

The Class C private address range is 192.168.0.0 through 192.168.255.255.

Remember the APIPA range. The IP address range for APIPA is 169.254.0.1 through 169.254.255.254. The client also configures itself with a default Class B subnet mask of 255.255.0.0.

Understand why we need IPv6. Without IPv6, the world would soon be depleted of IP addresses.

Understand link-local. Link-local addresses are like an IPv4 APIPA IP address, but they can't be routed at all, not even in your organization.

Understand unique local. Similar to link-local, this is like a private IP address in IPv4 and cannot be routed to the Internet. However, the difference between link-local and unique local is that unique local can be routed within your organization or company.

Remember IPv6 addressing. IPv6 addressing is not like IPv4 addressing. IPv6 addressing has much more address space and the address is 128 bits long, represented in hexadecimal, unlike an IPv4 address, which is only 32 bits long and represented in decimal.

Understand and be able to read an EUI-64 address with the 7th bit inverted. Hosts can use autoconfiguration to obtain an IPv6 address, and one of the ways is through what is called EUI-64. This takes the unique MAC address of a host and inserts FF:FE in the middle of the address to change a 48-bit MAC address to a 64-bit interface ID. In addition to the 16 bits being inserted into the interface ID, the 7th bit of the first byte is inverted, typically from a 0 to a 1.

Written Labs

You can find the answers to the written labs in Appendix A.

Written Lab 7.1

Provide the answers to the following questions:

1. What is the valid range used for a Class C private IP address? ?
2. Name some of the benefits of IPv6 over IPv4. ?
3. What is the term for the autoconfiguration technology responsible for addresses that start with 169.254? ?
4. What defines a unicast address? ?
5. What defines a multicast address? ?
6. What is the name for a 48-bit (6-byte) numerical address physically assigned to a network interface, such as a NIC? ?
7. IPv6 has how many more bits, compared to addresses in IPv4? ?
8. What is the private address range for Class B networks? ?
9. What is the Class C range of values for the first octet in decimal and in binary? ?
10. What is the 127.0.0.1 address used for? ?

Answers

1. The class C private range is 192.168.0.0 through 192.168.255.255.
2. IPv6 has the following characteristics, among others, that make it preferable to IPv4: more available addresses, simpler header, options for authentication, and other security.
3. Automatic Private IP Addressing (APIPA) is the technology that results in hosts automatically configuring themselves with addresses that begin with 169.254.
4. An IP address assigned to an interface, considered a one-to-one communication.
5. One-to-many address
6. A MAC address, sometimes called a hardware address or even a burned-in address
7. IPv6 has 128-bit (16-octet) addresses, compared to IPv4's 32-bit (4-octet) addresses, so 96 more bits than IPv4.
8. 172.16.0.0 through 172.31.255.255
9. 192–223, 110*xxxxx*
10. Loopback or diagnostics. Actually, the full range of 127.0.0.1 through 127.255.255.254 is referred to as the loopback address.

Written Lab 7.2

In this lab, write the answers to the following IPv6 questions:

1. Which type of packet is addressed and delivered to only a single interface? ?
2. Which type of address is used just like a regular public routable address in IPv4? ?
3. Which type of address is not meant to be routed? ?
4. Which type of address is not meant to be routed to the Internet but is still globally unique? ?
5. Which type of address is meant to be delivered to multiple interfaces? ?
6. Which type of address identifies multiple interfaces, but packets are delivered only to the first address it finds? ?
7. Which addressing type is also referred to as one-to-nearest? ?
8. IPv4 had a loopback address of 127.0.0.1. What is the IPv6 loopback address? ?
9. What does a link-local address always start with? ?
10. What does a unique local unicast range start with? ?

Answers

1. Unicast
2. Global unicast
3. Link-local
4. Unique local (used to be called site-local)
5. Multicast
6. Anycast
7. Anycast
8. ::1
9. FE80::/10
10. FC00::/7

Review Questions

You can find the answers to the review questions in Appendix B.

1. Which of the following addresses is not allowed on the Internet? ?

   A. 191.192.168.1

   B. 191.168.169.254

   C. 172.32.255.0

   D. 172.31.12.251

2. A host automatically configured with an address from which of the following ranges indicates an inability to contact a DHCP server? ?

   A. 169.254.0.*x* with a mask of 255.255.255.0

   B. 169.254.*x*.*x* with a mask of 255.255.0.0

C. 169.254.*x.x* with a mask of 255.255.255.0

D. 169.255.*x.x* with a mask of 255.255.0.0

3. Which statement regarding private IP addresses is most accurate? ?

A. Private addresses cannot be used in intranets that require routing.

B. Private addresses must be assigned by a registrar or ISP.

C. A remote host across the Internet cannot ping your host if it has a private address.

D. Private addresses can only be used by a single administrative domain.

4. Which of the following is a valid Class A address? ?

A. 191.10.0.1

B. 127.10.0.1

C. 128.10.0.1

D. 126.10.0.1

5. Which of the following is a valid Class B address? ?

A. 10.1.1.1

B. 126.1.1.1

C. 129.1.1.1

D. 192.168.1.1

6. Which of the following describes a broadcast address? ?

A. All network bits are on (1s).

B. All host bits are on (1s).

C. All network bits are off (0s).

D. All host bits are off (0s).

7. Which of the following is a layer 2 broadcast? ?

A. FF.FF.FF.EE.EE.EE

B. FF.FF.FF.FF.FF.FF

C. 255.255.255.255

D. 255.0.0.0

8. In a Class C IP address, how long is the network address? ?

A. 8 bits

B. 16 bits

C. 24 bits

D. 32 bits

9. Which of the following is true when describing a unicast address? ?

A. Packets addressed to a unicast address are delivered to a single interface.

B. These are your typical publicly routable addresses, just like regular publicly routable addresses in IPv4.

C. These are like private addresses in IPv4 in that they are not meant to be routed.

D. These addresses are meant for nonrouting purposes, but they are almost globally unique, so it is unlikely they will have an address overlap.

10. A host is rebooted and you view the IP address that it was assigned. The address is 169.123.13.34. Which of the following happened?

A. The host received an APIPA address.

B. The host received a multicast address.

C. The host received a public address.

D. The host received a private address.

11. An IPv4 address uses 32 bits. How many bits is an IPv6 address?

A. 64

B. 128

C. 192

D. 255

12. Which of the following is true when describing a multicast address?

A. Packets addressed to a unicast address from a multicast address are delivered to a single interface.

B. Packets are delivered to all interfaces identified by the address. This is also called a one-to-many address.

C. It identifies multiple interfaces and is delivered to only one address. This address can also be called one-to-one-of-many.

D. These addresses are meant for nonrouting purposes, but they are almost globally unique so it is unlikely they will have an address overlap.

13. Which of the following is true when describing an anycast address?

A. Packets addressed to a unicast address from an anycast address are delivered to a single interface.

B. Packets are delivered to all interfaces identified by the address. This is also called a one-to-many address.

C. This address identifies multiple interfaces, and the anycast packet is delivered to only one address: the closest one. This address can also be called one-to-nearest.

D. These addresses are meant for nonrouting purposes, but they are almost globally unique so it is unlikely they will have an address overlap.

14. You want to ping the loopback address of your local host. Which two addresses could you type? (Choose two.)

A. `ping 127.0.0.1`

B. `ping 0.0.0.0`

C. `ping ::1`

D. `trace 0.0.::1`

15. What two statements about IPv6 addresses are true? (Choose two.)

A. Leading zeros are required.

B. Two colons (::) are used to represent successive hexadecimal fields of zeros.

C. Two colons (::) are used to separate fields.

D. A single interface will have multiple IPv6 addresses of different types.

16. What two statements about IPv4 and IPv6 addresses are true? (Choose two.)

A. An IPv6 address is 32 bits long, represented in hexadecimal.

B. An IPv6 address is 128 bits long, represented in decimal.

C. An IPv4 address is 32 bits long, represented in decimal.

D. An IPv6 address is 128 bits long, represented in hexadecimal.

17. Which of the following is a Class C network address?

A. 10.10.10.0

B. 127.0.0.1

C. 128.0.0.0

D. 192.255.254.0

18. Which of the following are private IP addresses? (Choose two.)

A. 12.0.0.1

B. 168.172.19.39

C. 172.20.14.36

D. 172.33.194.30

E. 192.168.24.43

19. IPv6 unicast routing is running on the Corp router. Which of the following addresses would be used as the EUI-64 address?

```
Corp#sh int f0/0
FastEthernet0/0 is up, line protocol is up
Hardware is AmdFE, address is 000d.bd3b.0d80 (bia 000d.bd3b.0d80)
[output cut]
```

A. FF02::3c3d:0d:bdff:fe3b:0d80

B. FE80::3c3d:2d:bdff:fe3b:0d80

C. FE80::3c3d:0d:bdff:fe3b:0d80

D. FE80::3c3d:2d:ffbd:3bfe:0d80

20. Which of the following is an invalid IP address for a host?

A. 10.0.0.1

B. 128.0.0.1

C. 224.0.0.1

D. 172.0.0.1

Answers

1. D. The addresses in the range 172.16.0.0 through 172.31.255.255 are all considered private, based on RFC 1918. Use of these addresses on the Internet is prohibited so that they can be used simultaneously in different administrative domains without concern for conflict. Some experts in the industry believe these addresses are not routable, which is not true.
2. B. APIPA uses the link-local private address range of 169.254.0.0 through 169.254.255.255 and a subnet mask of 255.255.0.0.

   APIPA addresses are used by DHCP clients that cannot contact a DHCP server and have no static alternate configuration. These addresses are not Internet routable and cannot, by default, be used across routers on an internetwork.
3. C. Private IP addresses are not routable over the Internet, as either source or destination addresses. Because of that fact, any entity that wishes to use such addresses internally can do so without causing conflicts with other entities and without asking permission of any registrar or service provider. Despite not being allowed on the Internet, private IP addresses are fully routable on private intranets.
4. D. The Class A range is 1 through 126 in the first octet/byte, so only option D is a valid Class A address.
5. C. The Class B range is 128 through 191 in the first octet/byte. Only option C is a valid Class B address.
6. B. If you turned on all host bits (all of the host bits are 1s), this would be a broadcast address for that network.
7. B. A Layer 2 broadcast is also referred to as a MAC address broadcast, which is in hexadecimal and is FF.FF.FF.FF.FF.FF.
8. C. A default Class C subnet mask is 255.255.255.0, which means that the first three octets, or first 24 bits, are the network number.
9. A. Packets addressed to a unicast address are delivered to a single interface. For load balancing, multiple interfaces can use the same address.
10. C. I wonder how many of you picked APIPA address as your answer? An APIPA address is 169.254.*x.x*. The host address in this question is a public address. Somewhat of a tricky question if you did not read carefully.
11. B. An IPv6 address is 128 bits in size.
12. B. Packets addressed to a multicast address are delivered to all interfaces identified by the multicast address, the same as in IPv4. A multicast address is also called a one-to-many address. You can tell multicast addresses in IPv6 because they always start with *FF*.
13. C. Anycast addresses identify multiple interfaces, which is the same as multicast; however, the big difference is that the anycast packet is delivered to only one address: the first one it finds defined in terms of routing distance. This address can also be called one-to-one-of-many or

one-to-nearest.

14. A, C. The loopback address with IPv4 is 127.0.0.1. With IPv6, that address is ::1.
15. B, D. In order to shorten the written length of an IPv6 address, successive fields of zeros may be replaced by double colons. In trying to shorten the address further, leading zeros may also be removed. Just as with IPv4, a single device's interface can have more than one address; with IPv6 there are more types of addresses and the same rule applies. There can be link-local, global unicast, and multicast addresses all assigned to the same interface.
16. C, D. IPv4 addresses are 32 bits long and are represented in decimal format. IPv6 addresses are 128 bits long and represented in hexadecimal format.
17. D. Only option D is in the Class C range of 192 through 224. It might look wrong because there is a 255 in the address, but this is not wrong—you can have a 255 in a network address, just not in the first octet.
18. C, E. The Class A private address range is 10.0.0.0 through 10.255.255.255. The Class B private address range is 172.16.0.0 through 172.31.255.255, and the Class C private address range is 192.168.0.0 through 192.168.255.255.
19. B. This can be a hard question if you don't remember to invert the 7th bit! Always look for the 7th bit when studying for the exam. The EUI-64 autoconfiguration inserts an FF:FE in the middle of the 48-bit MAC address to create a unique IPv6 address.
20. C. Option C is a multicast address and cannot be used to address hosts.