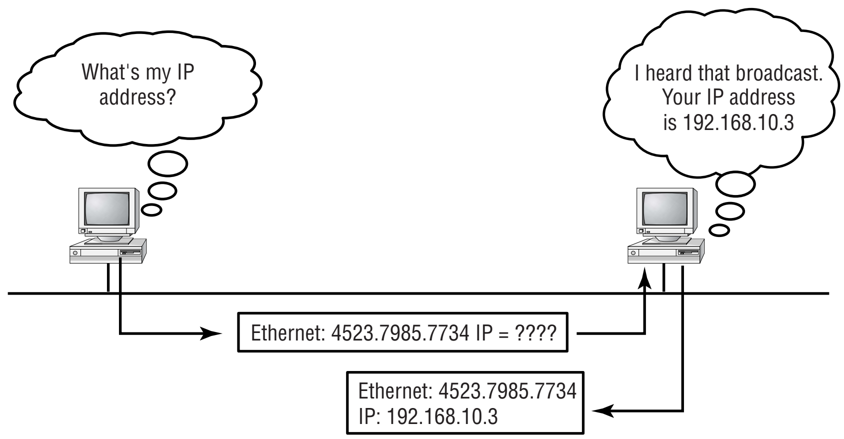
machine, called a RARP server, responds with the answer, and the identity crisis is over. RARP uses the information it does know about the machine's MAC address to learn its IP address and complete the machine's ID portrait.

NOTE: RARP resolves Ethernet addresses to IP addresses.

Figure 3.9 shows a diskless workstation asking for its IP address with a RARP broadcast.

FIGURE 3.9 RARP broadcast example



IP Addressing

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 location of a device on the network. An IP address is a software 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 a host 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. In this section you will learn about some of the fundamentals of IP addressing and its terminology. Later on, you will learn about the hierarchical IP addressing scheme and subnetting.

NOTE: To understand IP addressing and subnetting, it's important to have already mastered binary-to-decimal conversion and the powers of 2. If you need to review these topics, see the upcoming sidebars covering these issues.

IP Terminology

Throughout this chapter you will learn several terms that are critical to understanding the Internet Protocol. To start, here are a few of the most important:

Bit One digit; either a 1 or a 0.

Byte 7 or 8 bits, depending on whether parity is used. For the rest of this chapter, always assume a byte is 8 bits.

Octet Always 8 bits. Base-8 addressing scheme.

Network address 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.

Broadcast address Used by applications and hosts to send information to all nodes on a network. Examples include 255.255.255.255, which is all networks, all nodes; 172.16.255.255, which is all subnets and hosts on network 17.16.0.0; 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, each containing 1 byte (8 bits). 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 `82 39 1E 38`

All these examples represent the same IP address. Although hexadecimal is not used as often as dotted-decimal or binary when IP addressing is discussed, you still might find an IP address stored in hexadecimal in some programs; for example, the Windows Registry stores a machine's IP address in hex.

The 32-bit IP address is a structured or hierarchical address, as opposed to a flat or nonhierarchical, address. Although either type of addressing scheme could have been used, the hierarchical variety was chosen for a good reason. The 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 approximately 4.3 billion). The disadvantage of this 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 the possible addresses were used.

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

Network Addressing

The *network address* 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.

The *node 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. This number can also be referred to as a *host address*. In the sample IP address `172.16.30.56`, .30.56 is the node 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 nodes, 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 nodes. The class distinction for networks between very large and very small is predictably called the *Class B network*.

Subdividing an IP address into a network and node address is determined by the class designation of one's network. Figure 3.10 summarizes the three classes of networks, which will be described in much more detail throughout this chapter.

FIGURE 3.10 Summary of the three classes of networks

	8 bits	8 bits	8 bits	8 bits
Class A:	Network	Host	Host	Host
Class B:	Network	Network	Host	Host
Class C:	Network	Network	Network	Host
Class D:	Multicast			
Class E:	Research			

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, Class B, and Class C address.

Network Address Range: Class A

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.

Here is how those numbers are defined:

> 0xxxxxxx: If we turn the other 7 bits all off and then turn them all on, we will find your Class A range of network addresses.
>
> **0**0000000=0
> **0**1111111=127

So, a Class A network is defined in the first octet between 0 and 127. It can't be less or more. (We'll talk about illegal addresses in a minute.)

NOTE: If you are having any difficulty with the binary-to-decimal conversions, please read the "Binary-to-Decimal Conversion Review" sidebar.

Binary-to-Decimal Conversion Review

Prior to learning about IP addressing, you must have a fundamental understanding of binary-to-decimal conversions. Here is how it works: Binary numbers use 8 bits to define a decimal number. These bits are weighted from right to left in an increment that doubles in value.

Here is an example of 8 bits and the value assigned to each bit:

```
128    64    32    16    8    4    2    1
```

Here is an example of binary-to-decimal conversion:

```
128    64    32    16    8    4    2    1    Binary value
0      0     1     0     0    1    1    0    Byte in binary
```

Add the value of the bits that are turned on:

```
32
4
2
=38
```

Any time you find a bit turned on (a one), you add the values of each bit position. Let's practice on a few more:

```
01010101=85
64
16
  4
  1
=85
```

Try a few on your own:

```
00001111=15
10001100=140
11001100=204
```

You will need to memorize the binary-to-decimal conversions in the following list. You will use this information when you practice subnetting later in this chapter:

```
00000000=0
10000000=128
11000000=192
11100000=224
11110000=240
11111000=248
11111100=252
11111110=254
11111111=255
```

Network Address Range: Class B

In a Class B network, the 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 you turn the other six bits all off and then all on, you will find the range for a Class B network:

10000000=128
10111111=191

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

Network Address Range: Class C

For Class C networks, the RFCs define the first two bits of the first octet 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 is the range for a Class C network:

11000000=192
11011111=223

So, if you see an IP address that starts at 192 and goes to 223, you'll know it is a Class C IP address.

Network Address Ranges: Classes D and E

The addresses between 224 and 255 are reserved for Class D and E networks. Class D is used for multicast addresses and Class E for scientific purposes. We will not discuss Class D and E addresses in this book.

Network Addresses: Special Purpose

Some IP addresses are reserved for special purposes, and network administrators shouldn't assign these addresses to nodes. Table 3.2 lists the members of this exclusive little club and why they're included in it.

TABLE 3.2 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 node and allows that node to send a test packet to itself without generating network traffic.
Node address of all 0s	Interpreted to mean "this node."
Node address of all 1s	Interpreted to mean "all nodes" on the specified network; for example, `128.2.255.255` means "all nodes" on network 128.2 (Class B address).
Entire IP address set to all 0s	Used by Cisco routers to designate the default route.
Entire IP address set to all 1s (same as `255.255.255.255`)	Broadcast to all nodes on the current network; sometimes called an "all 1s broadcast."

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 node addresses. The Class A format is

`Network.Node.Node.Node`

For example, in the IP address `49.22.102.70`, 49 is the network address, and 22.102.70 is the node address. Every machine on this particular network would have the distinctive network address of 49.

Class A addresses are one byte long, with the first bit of that byte reserved and the seven remaining bits available for manipulation. As a result, the maximum number of Class A networks that can be created is 128. Why? Because each of the seven bit positions can either be a 0 or a 1, thus 2^7 or 128.

To complicate matters further, the network address of all 0s (0000 0000) is reserved to designate the default route (see Table 3.2 in the previous section). Additionally, the address 127, which is reserved for diagnostics, can't be used either, which means that you can 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. Got it?

Each Class A address has three bytes (24-bit positions) for the node address of a machine. Thus, there are 2^{24}—or 16,777,216—unique combinations and, therefore, precisely that many possible unique node addresses for each Class A network. Because addresses with the two patterns of all 0s and all 1s are reserved, the actual maximum usable number of nodes for a Class A network is 2^{24} minus 2, which equals 16,777,214.

Class A Valid Host IDs

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

`10.0.0.0` All host bits off is the network address.

`10.255.255.255` All host bits on is the broadcast address.

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

Class B Addresses

In a Class B network address, the first two bytes are assigned to the network address, and the remaining two bytes are used for node addresses. The format is

`Network.Network.Node.Node`

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

With a network address being two bytes (eight bits each), there would be 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 to manipulate, therefore 16,384 (2^{14}) unique Class B network addresses.

A Class B address uses two bytes for node addresses. This is 2^{16} minus the two reserved patterns (all 0s and all 1s), for a total of 65,534 possible node addresses for each Class B network.

Class B Valid Host IDs

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

`172.16.0.0` All host bits turned off is the network address.

`172.16.255.255` All host bits turned on is the broadcast address.

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 three bytes of a Class C network address are dedicated to the network portion of the address, with only one measly byte remaining for the node address. The format is

`Network.Network.Network.Node`

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

In a Class C network address, the first three bit positions are always the binary 110. The calculation is such: 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.

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

Class C Valid Host IDs

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

`192.168.100.0` All host bits turned off is the network ID.

`192.168.100.255` All host bits turned on is the broadcast address.

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`.

Subnetting

In the previous section, you learned how to define and find the valid host ranges used in a Class A, Class B, and Class C network address by turning the host bits all off and then all on. However, you were defining only one network. What happens if you wanted to take one network address and create six networks from it? You would have to perform what is called *subnetting*, which allows you to take one larger network and break it into many smaller networks.

There are many reasons to perform subnetting. Some of the benefits of subnetting include the following:

Reduced network traffic We all appreciate less traffic of any kind. Networks are no different. Without trusty routers, packet traffic could grind the entire network down to a near standstill. With routers, most traffic will stay on the local network; only packets destined for other networks will pass through the router. Routers create broadcast domains. The smaller broadcast domains you create, the less network traffic on that network segment.

Optimized network performance This is a result of reduced network traffic.

Simplified management It's easier to identify and isolate network problems in a group of smaller connected networks than within one gigantic network.

Facilitated spanning of large geographical distances Because WAN links are considerably slower and more expensive than LAN links, a single large network that spans long distances can create problems in every arena listed above. Connecting multiple smaller networks makes the system more efficient.

To create subnetworks, you take bits from the host portion of the IP address and reserve them to define the subnet address. This means fewer bits for hosts, so the more subnets, the fewer bits available for defining hosts.

In this section you will learn how to create subnets, starting with Class C addresses. However, before you implement subnetting, you need to determine your current requirements and plan for future conditions. Follow these steps:

1. Determine the number of required network IDs.
   - **A.** One for each subnet
   - **B.** One for each wide area network connection
2. Determine the number of required host IDs per subnet.
   - **A.** One for each TCP/IP host
   - **B.** One for each router interface
3. Based on the above requirement, create the following:
   - **A.** One subnet mask for your entire network
   - **B.** A unique subnet ID for each physical segment
   - **C.** A range of host IDs for each subnet

Understanding the Powers of 2

Powers of 2 are important to understand and memorize for use with IP subnetting. To review powers of 2, remember that when you see a number with another number to its upper right, this means you should multiply the number by itself as many times as the upper number specifies. For example, 2^3 is 2x2x2, which equals 8. Here is the list of powers of 2 that you should memorize:

$2^1=2$

$2^2=4$

$2^3=8$

$2^4=16$

$2^5=32$

$2^6=64$

$2^7=128$

$2^8=256$

Subnet Masks

For the subnet address scheme to work, every machine on the network must know which part of the host address will be used as the subnet address. This is accomplished by assigning a *subnet mask* to each machine. This is a 32-bit value that allows the recipient of IP packets to distinguish the network ID portion of the IP address from the host ID portion of the IP address.

The network administrator creates a 32-bit subnet mask composed of 1s and 0s. The 1s in the subnet mask represent the positions that refer to the network or subnet addresses.

Not all networks need subnets, meaning they use the default subnet mask. This is basically the same as saying that a network doesn't have a subnet address. Table 3.3 shows the default subnet masks for Classes A, B, and C. These cannot change. In other words, you cannot make a Class B subnet mask read 255.0.0.0. The host will read such an address as invalid and typically won't even let you type it in. For a Class A network, you cannot change the first byte in a subnet mask; it must read 255.0.0.0 at a minimum. Similarly, you cannot assign 255.255.255.255, as this is all 1s and a broadcast address. A Class B address must start with 255.255.0.0, and a Class C must start with 255.255.255.0.

TABLE 3.3 Default Subnet Mask

Class	Format	Default Subnet Mask
A	Net.Node.Node.Node	255.0.0.0
B	Net.Net.Node.Node	255.255.0.0
C	Net.Net.Net.Node	255.255.255.0

Subnetting Class C Addresses

There are many different ways to subnet a network. The right way is the way that works best for you. First you will learn to use the binary method, and then we'll look at an easier way to do the same thing.

In a Class C address, only 8 bits is available for defining the hosts. Remember that subnet bits start at the left and go to the right, without skipping bits. This means that subnet masks can be

```
10000000=128
11000000=192
11100000=224
11110000=240
11111000=248
11111100=252
11111110=254
```

Now, the RFCs state that you cannot have only one bit for subnetting, since that would mean that the bit would always be either off or on, which would be illegal. So, the first subnet mask you can legally use is 192, and the last one is 252, since you need at least two bits for defining hosts.

The Binary Method: Subnetting a Class C Address

In this section you will learn how to subnet a Class C address using the binary method. We will take the first subnet mask available with a Class C address, which borrows two bits from subnetting. For this example, we are using 255.255.255.192.

192=11000000 Two bits for subnetting, 6 bits for defining the hosts in each subnet. What are the subnets? Since the subnet bits can't be both off or on at the same time, the only two valid subnets are

- **01**000000=64 (all host bits off)

or

- **10**000000=128 (all host bits off)

The valid hosts would be defined as the numbers between the subnets, minus the all host bits off and all host bits on.

To find the hosts, first find your subnet by turning all the host bits off, then turn all the host bits on to find your broadcast address for the subnet. The valid hosts must be between those two numbers. Table 3.4 shows the 64 subnet, valid host range, and broadcast address.

TABLE 3.4 Subnet 64

Subnet	Host	Meaning
01	000000=64	The network (do this first)
01	000001=65	The first valid host
01	111110=126	The last valid host
01	111111=127	The broadcast address (do this second)

Table 3.5 shows the 128 subnet, valid host range, and broadcast address.

TABLE 3.5 Subnet 128

Subnet	Host	Meaning
10	000000=128	The subnet address
10	000001=129	The first valid host
10	111110=190	The last valid host
10	111111=191	The broadcast address

That wasn't all that hard. Hopefully you understood what I was trying to show you. However, the example I presented only used two subnet bits. What if you had to subnet using 9, 10, or even 20 subnet bits? Let's learn an alternate method of subnetting that makes it easier to subnet larger numbers.

The Alternate Method: Subnetting a Class C Address

When you have a subnet mask and need to determine the amount of subnets, valid hosts, and broadcast addresses that the mask provides, all you need to do is answer five simple questions:

1. How many subnets does the subnet mask produce?
2. How many valid hosts per subnet?

3. What are the valid subnets?
4. What are the valid hosts in each subnet?
5. What is the broadcast address of each subnet?

It is important at this point that you understand your powers of 2. Please refer to the sidebar earlier in this chapter if you need help. Here is how you determine the answers to the five questions:

1. How many subnets? 2^x–2=amount of subnets. X is the amount of masked bits, or the 1s. For example, 11000000 is 2^2–2. In this example, there are 2 subnets.
2. How many hosts per subnet? 2^x–2=amount of hosts per subnet. X is the amount of unmasked bits, or the 0s. For example, 11000000 is 2^6–2. In this example, there are 62 hosts per subnet.
3. What are the valid subnets? 256–subnet mask=base number. For example, 256–192=64.
4. What are the valid hosts? Valid hosts are the numbers between the subnets, minus all 0s and all 1s.
5. What is the broadcast address for each subnet? Broadcast address is all host bits turned on, which is the number immediately preceding the next subnet.

Now, because this can seem confusing, I need to assure you that it is easier than it looks. Just try a few with me and see for yourself.

Subnetting Practice Examples: Class C Addresses

This section will give you an opportunity to practice subnetting Class C addresses using the method I just described. We're going to start with the first Class C subnet mask and work through every subnet that we can using a Class C address. When we're done, I'll show you how easy this is with Class A and B networks as well.

Practice Example 1: 255.255.255.192

Let's use the Class C subnet address from the preceding example, `255.255.255.192`, to see how much simpler this method is than writing out the binary numbers. In this example, you will subnet the network address 192.168.10.0 and subnet mask 255.255.255.192.

192.168.10.0=Network address

255.255.255.192=Subnet mask

Now, answer the five questions:

1. How many subnets? Since 192 is two bits on (11000000), the answer would be 2^2–2=2. (The minus 2 is the subnet bits all on or all off, which is not valid by default.)
2. How many hosts per subnet? We have 6 host bits off (11000000), so the equation would be 2^6–2=62 hosts.
3. What are the valid subnets? 256–192=64, which is the first subnet and our base number or variable. Keep adding the variable to itself until you reach the subnet mask. 64+64=128. 128+64=192, which is invalid because it is the subnet mask (all subnet bits turned on). Our two valid subnets are, then, 64 and 128.
4. What are the valid hosts? These are the numbers between the subnets. The easiest way to find the hosts is to write out the subnet address and the broadcast address. This way the valid hosts are obvious.
5. What is the broadcast address for each subnet? The number right before the next subnet is all host bits turned on and is the broadcast address. Table 3.6 shows the 64 and 128 subnets, the valid host ranges of each, and the broadcast address of both subnets.

TABLE 3.6 The 64 and 128 Subnet Ranges

First Subnet	Second Subnet	Meaning
64	128	The subnets (do this first)
65	129	Our first host (perform host addressing last)
126	190	Our last host
127	191	The broadcast address (do this second)

Notice that we came up with the same answers as when we did it the binary way. This is a much easier way to do it because you never have to do any binary-to-decimal conversions. However, you might be thinking that it is not easier than the first method I showed you. For the first subnet with only two subnet bits, you're right, it isn't that much easier. Remember, we're going for the big one: being able to subnet in your head. You need to practice this approach to be able to perform subnetting in your head.

Practice Example 2: 255.255.255.224

In this example, you will subnet the network address 192.168.10.0 and subnet mask 255.255.255.224.

`192.168.10.0`=Network address

`255.255.255.224`=Subnet mask

1. How many subnets? 224 is 11100000, so our equation would be 2^3–2=6.
2. How many hosts? 2^5–2=30.
3. What are the valid subnets? 256–224=32. 32+32=64. 64+32=96. 96+32=128. 128+32=160. 160+32=192. 192+64=224, which is invalid because it is our subnet mask (all subnet bits on). Our subnets are 32, 64, 96, 128, 160, and 192.
4. What are the valid hosts?
5. What is the broadcast address for each subnet?

To answer questions 4 and 5, first just write out the subnets, then write out the broadcast addresses, which is the number right before the next subnet. Last, fill in the host addresses. Table 3.7 shows all the subnets for the 255.255.255.224 Class C subnet mask.

TABLE 3.7 The Class C 255.255.255.224 Mask

Subnet 1	Subnet 2	Subnet 3	Subnet 4	Subnet 5	Subnet 6	Meaning
32	64	96	128	160	192	The subnet address
33	65	97	129	161	193	The first valid host
62	94	126	158	190	222	Our last valid host
63	95	127	159	191	223	The broadcast address

Practice Example 3: 255.255.255.240

Let's practice on another one:

`192.168.10.0`=Network number

`255.255.255.240`=Subnet mask

1. 240 is 11110000 in binary. 2^4–2=14 subnets.
2. Four host bits, or 2^4–2=14.
3. 256–240=16. 16+16=32. 32+16=48. 48+16=64. 64+16=80. 80+16=96. 96+16=112. 112+16=128. 128+16=144. 144+16=160. 160+16=176. 176+16=192. 192+16=208. 208+16=224. 224+16=240, which is our subnet mask and therefore invalid. So, our valid subnets are 16, 32, 48, 64, 80, 96, 112, 128, 144, 160, 176, 192, 208, and 224.
4. What are the valid hosts?
5. What is the broadcast address for each subnet?

To answer questions 4 and 5, view the following table, which shows the subnets, valid hosts, and broadcast addresses for each subnet. First, find the broadcast address of each subnet, then fill in the host addresses.

Subnet	16	32	48	64	80	96	112	128	144	160	176	192	208	224
First Host	17	33	49	65	81	97	113	129	145	161	177	193	209	225
Last Host	30	46	62	78	94	110	126	142	158	174	190	206	222	238
Broadcast	31	47	63	79	95	111	127	143	159	175	191	207	223	239

Practice Example 4: 255.255.255.248

Let's keep practicing:

`192.168.10.0`=Network address

`255.255.255.248`=Subnet mask

1. 248 in binary=11111000. 2^5–2=30 subnets.

2. $2^3-2=6$ hosts.
3. 256–248=8, 16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, 120, 128, 136, 144, 152, 160, 168, 176, 184, 192, 200, 208, 216, 224, 232, and 240.
4. First find the broadcast addresses in step 5, then come back and perform step 4 by filling in the host addresses.
5. Find the broadcast address of each subnet, which is always the number right before the next subnet.

Take a look at the following table, which shows the subnets (first three and last three only), valid hosts, and broadcast addresses for the Class C 255.255.255.248 mask.

Subnet	8	16	24	224	232	240
First Host	9	17	25	225	233	241
Last Host	14	22	30	230	238	246
Broadcast	15	23	31	231	239	247

Practice Example 5: 255.255.255.252

`192.168.10.0`=Network number

`255.255.255.252`=Subnet mask

1. 62.
2. 2.
3. 4, 8, 12, etc., all the way to 248.
4. First find the broadcast addresses in step 5, then come back and perform step 4 by filling in the host addresses.
5. Find the broadcast address of each subnet, which is always the number right before the next subnet.

The following table shows you the subnet, valid host, and broadcast address of the first three and last three subnets in the 255.255.255.252 Class C subnet.

Subnet	4	8	12	240	244	248
First Host	5	9	13	241	245	249
Last Host	6	10	14	242	246	250
Broadcast	7	11	15	243	247	251

Practice Example 6: 255.255.255.128

OK, we told you that using only one subnet bit was illegal and not to use it. But aren't all rules meant to be broken? This mask can be used when you need two subnets, each with 126 hosts. The standard five questions don't work here, and we'll just explain how to use it. First, use the global configuration command `ip subnet-zero` to tell your router to break the rules and use a 1-bit subnet mask.

Since 128 is 1000000 in binary, there is only one bit for subnetting. Since this bit can be either off or on, the two available subnets are 0 and 128. You can determine the subnet value by looking at the decimal value of the fourth octet. The following table will show you the two subnets, valid host range, and broadcast address for the Class C 255.255.255.128 mask.

Subnet	0	128
First Host	1	129
Last Host	126	254
Broadcast	127	255

So, if you have an IP address of `192.168.10.5` using the 255.255.255 .128-subnet mask, you know it is in the range of the 0 subnet and the 128-bit must be off. If you have an IP address of `192.168.10.189`, then the 128 must be on, and the host is considered to be in the 128 subnet. You'll see this again in a minute.

Subnetting in Your Head: Class C Addresses

It is possible to perform subnetting in your head. Don't you believe me? I'll show you how; it's relatively easy. Take the following example:

`192.168.10.33`=Network address

`255.255.255.224`=Subnet mask

First, determine the subnet and broadcast address of the above IP address. You can do this by answering question 3 in the five-question process. 256–224=32. 32+32=64. Bingo. The address falls between the two subnets and must be part of the 192.168.10.32 subnet. The next subnet is 64, so the broadcast address is 63. (Remember that the broadcast address of a subnet is always the number right before the next subnet.) The valid host range is 10.33–10.62. This is too easy.

Let's try another one. Here, you will subnet another Class C address:

`192.168.10.33`=Network address

`255.255.255.240`=Subnet mask

What subnet and broadcast address is the above IP address a member of? 256–240=16. 16+16=32. 32+16=48. Bingo, the host address is between the 32 and 48 subnets. The subnet is 192.168.10.32, and the broadcast address is 47. The valid host range is 33–46.

Now that we have completed all the Class C subnets, what should we do next? Class B subnetting, did you say? Sounds good to me.

Subnetting Class B Addresses

Since we went through all the possible Class C subnets, let's take a look at subnetting a Class B network. First, let's look at all the possible Class B subnet masks. Notice that we have a lot more possible subnets than we do with a Class C network address.

```
255.255.128.0
255.255.192.0
255.255.224.0
255.255.240.0
255.255.248.0
255.255.252.0
```

```
255.255.254.0
255.255.255.0
255.255.255.128
255.255.255.192
255.255.255.224
255.255.255.240
255.255.255.248
255.255.255.252
```

The Class B network address has 16 bits available for hosts addressing. This means we can use up to 14 bits for subnetting since we must leave at least two bits for host addressing.

Do you notice a pattern in the subnet values? This is why we had you memorize the binary-to-decimal numbers at the beginning of this section. Since subnet mask bits start on the left, move to the right, and cannot skip bits, the numbers are always the same. Memorize this pattern.

The process of subnetting a Class B network is the same as for a Class C, except you just have more host bits. Use the same subnet numbers you used with Class C, but add a zero to the network portion and a 255 to the broadcast section in the fourth octet. The following table shows you a host range of two subnets used in a Class B subnet.

16.0	32.0
16.255	32.255

Just add the valid hosts between the numbers, and you're set.

Subnetting Practice Examples: Class B Addresses

This section will give you an opportunity to practice subnetting Class B addresses.

Practice Example 1: 255.255.192.0

`172.16.0.0`=Network address

`255.255.192.0`=Subnet mask

1. $2^2-2=2$.
2. $2^{14}-2=16{,}382$.

3. 256–192=64. 64+64=128.
4. First find the broadcast addresses in step 5, then come back and perform step 4 by filling in the host addresses.
5. Find the broadcast address of each subnet, which is always the number right before the next subnet.

The following table shows the two subnets available, the valid host range, and the broadcast address of each.

Subnet	64.0	128.0
First Host	64.1	128.1
Last Host	127.254	191.254
Broadcast	127.255	191.255

Notice we just added the fourth octet's lowest and highest values and came up with the answers. Again, it is the same answer as for a Class C subnet, but we just added the fourth octet.

Practice Example 2: 255.255.240.0

`172.16.0.0`=Network address

`255.255.240.0`=Subnet address

1. 2^4–2=14.
2. 2^{12}–2=4094.
3. 256–240=16, 32, 48, etc., up to 224. Notice these are the same numbers as a Class C 240 mask.
4. First find the broadcast addresses in step 5, then come back and perform step 4 by filling in the host addresses.
5. Find the broadcast address of each subnet, which is always the number right before the next subnet.

The following table shows the first three subnets, valid hosts, and broadcast addresses in a Class B 255.255.240.0 mask.

Subnet	16.0	32.0	48.0
First Host	16.1	32.1	48.1
Last Host	31.254	47.254	63.254
Broadcast	31.255	47.255	63.255

Practice Example 3: 255.255.254.0

1. 2^7–2=126.
2. 2^9–2=510.
3. 256–254=2, 4, 6, 8, etc., up to 252.
4. First find the broadcast addresses in step 5, then come back and perform step 4 by filling in the host addresses.
5. Find the broadcast address of each subnet, which is always the number right before the next subnet.

The following table shows the first four subnets, valid hosts, and broadcast addresses in a Class B 255.255.254.0 mask.

Subnet	2.0	4.0	6.0	8.0
First Host	2.1	4.1	6.1	8.1
Last Host	3.254	4.254	7.254	9.254
Broadcast	3.255	5.255	7.255	9.255

Practice Example 4: 255.255.255.0

Contrary to popular belief, 255.255.255.0 is not a Class C subnet mask. It is amazing how many people see this mask used in a Class B network and think it is a Class C subnet mask. This is a Class B subnet mask with 8 bits of subnetting—it is considerably different from a Class C mask. Subnetting this address is fairly simple:

1. 2^8–2=254.
2. 2^8–2=254.

3. 256–255=1, 2, 3, etc. all the way to 254.
4. First find the broadcast addresses in step 5, then come back and perform step 4 by filling in the host addresses.
5. Find the broadcast address of each subnet, which is always the number right before the next subnet.

The following table shows the first three subnets and the last one, valid hosts, and broadcast addresses in a Class B 255.255.255.0 mask.

Subnet	1.0	2.0	3.0	254.0
First Host	1.1	2.1	3.1	254.1
Last Host	1.254	2.254	3.254	254.254
Broadcast	1.255	2.255	3.255	254.255

Practice Example 5: 255.255.255.128

This must be illegal! What type of mask is this? Don't you wish it were illegal? This is one of the hardest subnet masks you can play with. It is actually a good subnet to use in production, as it creates over 500 subnets with 126 hosts for each subnet. That's a nice mixture.

1. 2^9–2=510.
2. 2^7–2=126.
3. This is the tricky part. 256–255=1, 2, 3, etc., for the third octet. However, you can't forget the one subnet bit used in the fourth octet. Remember when we showed you how to figure one subnet bit with a Class C mask? You figure this the same way. (Now you know why we showed you the 1-bit subnet mask in the Class C section—to make this part easier.) You actually get two subnets for each third octet value, hence the 510 subnets. For example, if the third octet was showing subnet 3, the two subnets would actually be 3.0 and 3.128.
4. First find the broadcast addresses in step 5, then come back and perform step 4 by filling in the host addresses.
5. Find the broadcast address of each subnet, which is always the number right before the next subnet.

The following table shows how you can create subnets, valid hosts, and broadcast addresses using the Class B 255.255.255.128 subnet mask.

Subnet	0.128	1.0	1.128	2.0	2.128	3.0	3.128
First Host	0.129	1.1	1.129	2.1	2.129	3.1	3.129
Last Host	0.254	1.126	1.254	2.126	2.254	3.126	3.254
Broadcast	0.255	1.127	1.255	2.127	2.255	3.127	3.255

Practice Example 6: 255.255.255.192

This one gets just a little tricky. Both the 0 subnet as well as the 192 subnet could be valid in the fourth octet. It just depends on what the third octet is doing.

1. 2^{10}–2=1022 subnets.
2. 2^{6}–2=62 hosts.
3. 256–192=64 and 128. However, as long as all the subnet bits on the third are not all off, then subnet 0 in the fourth octet is valid. Also, as long as all the subnet bits in the third octet are not all on, 192 is valid in the fourth octet as a subnet.
4. First find the broadcast addresses in step 5, then come back and perform step 4 by filling in the host addresses.
5. Find the broadcast address of each subnet, which is always the number right before the next subnet.

The following table shows the first two subnet ranges, valid hosts, and broadcast addresses.

Subnet	0.64	0.128	0.192	1.0	1.64	1.128	1.192
First Host	0.65	0.129	0.193	1.1	1.65	1.129	1.193
Last Host	0.126	0.190	0.254	1.62	1.126	1.190	1.254
Broadcast	0.127	0.191	0.255	1.63	1.127	1.191	1.255

Notice that for each subnet value in the third octet, you get subnets 0, 64, 128, and 192 in the fourth octet. This is true for every subnet in the third octet except 0 and 255. We just demonstrated the 0-subnet value in the third octet.

Notice, however, that for the 1 subnet in the third octet, the fourth octet has four subnets, 0, 64, 128, and 192.

Practice Example 7: 255.255.255.224

This is done the same way as the preceding subnet mask; however, we just have more subnets and fewer hosts per subnet available.

1. 2^{11}–2=2046 subnets.
2. 2^{5}–2=30 hosts.
3. 256–224=32, 64, 96, 128, 160, 192. However, as demonstrated above, both the 0 and 224 subnets can be used as long as the third octet does not show a value of 0 or 255. Here is an example of having no subnet bits in the third octet.
4. First find the broadcast addresses in step 5, then come back and perform step 4 by filling in the host addresses.
5. Find the broadcast address of each subnet, which is always the number right before the next subnet.

The following table shows the first range of subnets.

Subnet	0.32	0.64	0.96	0.128	0.160	0.192	0.224
First Host	0.33	0.65	0.97	0.129	0.161	0.193	0.225
Last Host	0.62	0.94	0.126	0.158	0.190	0.222	0.254
Broadcast	0.63	0.95	0.127	0.159	0.191	0.223	0.255

Let's take a look at a situation where a subnet bit is turned on in the third octet. The following table shows the full range of subnets available in the fourth octet.

Subnet	1.0	1.32	1.64	1.128	1.160	1.192	1.224
First Host	1.1	1.33	1.65	1.129	1.161	1.193	1.225
Last Host	1.30	1.62	1.126	1.158	1.190	1.222	1.254
Broadcast	1.31	1.63	1.127	1.159	1.191	1.223	1.255

This next table shows the last subnet.

Subnet	255.0	255.32	255.64	255.128	255.160	255.192
First Host	255.1	255.33	255.65	255.129	255.161	255.193
Last Host	255.62	255.62	255.126	255.158	255.190	255.222
Broadcast	255.63	255.63	255.127	255.159	255.191	255.223

Subnetting in Your Head: Class B Addresses

Can we subnet Class B addresses in our heads? I know what you are thinking: "Are you nuts?" It's actually easier than writing it out. I'll show you how:

Question: What subnet and broadcast address is the IP address `172.16.10.33 255.255.255.224` a member of?

Answer: 256–224=32. 32+32=64. Bingo—33 is between 32 and 64. However, remember that the third octet is considered part of the subnet, so the answer would be the 10.32 subnet. The broadcast is 10.63, since 10.64 is the next subnet.

Let's try four more:

Question: What subnet and broadcast address is the IP address `172.16.90.66 255.255.255.192` a member of?

Answer: 256–192=64. 64+64=128. The subnet is 172.16.90.64. The broadcast must be 172.16.90.127, since 90.128 is the next subnet.

Question: What subnet and broadcast address is the IP address `172.16.50.97 255.255.255.224` a member of?

Answer: 256–224=32, 64, 96, 128. The subnet is 172.16.50.96, and the broadcast must be 172.16.50.127 since 50.128 is the next subnet.

Question: What subnet and broadcast address is the IP address `172.16.10.10 255.255.255.192` a member of?

Answer: 256–192=64. This address must be in the 172.16.10.0 subnet, and the broadcast must be 172.16.10.63.

Question: What subnet and broadcast address is the IP address `172.16.10.10 255.255.255.224` a member of?

Answer: 256–224=32. The subnet is 172.16.10.0, with a broadcast of 172.16.10.31.

Subnetting Class A Addresses

Class A subnetting is not performed any differently from Classes B and C, though there are 24 bits to play with instead of the 16 in a Class B address and the eight bits in a Class C address.

Let's start by listing all the Class A subnets:

```
255.128.0.0
255.192.0.0
255.224.0.0
255.240.0.0
255.248.0.0
255.252.0.0
255.254.0.0
255.255.0.0
255.255.128.0
255.255.192.0
255.255.224.0
255.255.240.0
255.255.248.0
255.255.252.0
255.255.254.0
255.255.255.0
255.255.255.128
255.255.255.192
255.255.255.224
255.255.255.240
255.255.255.248
255.255.255.252
```

That's it. You must leave at least two bits for defining hosts. We hope you can see the pattern by now. Remember, we're going to do this the same way as a Class B or C subnet, but we just have more host bits.

Subnetting Practice Examples: Class A Addresses

When you look at an IP address and a subnet mask, you must be able to determine the bits used for subnets and the bits used for determining hosts. This is imperative. If you are still struggling with this concept, please reread the preceding "IP Addressing" section, which shows you how to determine the difference between the subnet and host bits.

Practice Example 1: 255.255.0.0

Class A addresses use a default mask of 255.0.0.0, which leaves 22 bits for subnetting since you must leave two bits for host addressing. The 255.255.0.0 mask with a Class A address is using eight subnet bits.

1. 2^8–2=254.
2. 2^{16}–2=65, 534.
3. 256–255=1, 2, 3, etc. (all in the second octet). The subnets would be 10.1.0.0, 10.2.0.0, 10.3.0.0, etc., up to 10.254.0.0.
4. First find the broadcast addresses in step 5, then come back and perform step 4 by filling in the host addresses.
5. Find the broadcast address of each subnet, which is always the number right before the next subnet.

Table 3.8 shows the first and last subnet, valid host range, and broadcast addresses.

TABLE 3.8 The First and Last Subnet

	First Subnet	Last Subnet
Subnet	10.1.0.0	10.254.0.0
First Host	10.1.0.1	10.254.0.1
Last Host	10.1.255.254	10.254.255.254
Broadcast	10.1.255.255	10.254.255.255

Practice Example 2: 255.255.240.0

255.255.240.0 gives us 12 bits of subnetting and leaves us 12 bits for host addressing.

1. 2^{12}–2=4094.
2. 2^{12}–2=4094.

3. 256–240=16. However, since the second octet is 255, or all subnet bits on, we can start the third octet with 0 as long as a subnet bit is turned on in the second octet. So the subnets become 10.1.0.0, 10.1.16.0, 10.1.32.0, and 10.1.48.0, all the way to 10.1.240.0. The next set of subnets would be 10.2.0.0, 10.2.16.0, 10.2.32.0, 10.2.48.0, all the way to 10.2.240.0. Notice that we can use 240 in the third octet as long as all the subnet bits in the second octet are not on. In other words, 10.255.240.0 is invalid because all subnet bits are turned on. The last valid subnet would be 10.255.224.0.
4. First find the broadcast addresses in step 5, then come back and perform step 4 by filling in the host addresses.
5. Find the broadcast address of each subnet, which is always the number right before the next subnet.

Table 3.9 shows some examples of the host ranges.

TABLE 3.9 Valid Host Ranges for a Class A 255.255.240.0 Mask

	First Subnet	Second Subnet	Last Subnet
Subnet	10.1.0.0	10.1.16.0	10.255.224.0
First Host	10.1.0.1	10.1.16.1	10.255.224.1
Last Host	10.1.15.254	10.1.31.254	10.255.239.254
Broadcast	10.1.15.255	10.1.31.255	10.255.239.255

Practice Example 3: 255.255.255.192

Let's do one more example using the second, third, and fourth octets for subnetting.

1. 2^{18}–2=262,142 subnets.
2. 2^{6}–2=62 hosts.
3. Now, we need to add subnet numbers from the second, third, and fourth octets. In the second and third, they can range from 1 to 255, as long as all subnet bits in the second, third, and fourth octets are not

all on at the same time. For the fourth octet, it will be 256–192=64. However, 0 will be valid as long as at least one other subnet bit is turned on in the second or third octet. Also, 192 will be valid as long as all the bits in the second and third octets are not turned on.

4. First find the broadcast addresses in step 5, then come back and perform step 4 by filling in the host addresses.

5. Find the broadcast address of each subnet, which is always the number right before the next subnet.

The following table will show the first few subnets and find the valid hosts and broadcast addresses in the Class A 255.255.255.192 mask.

Subnet	10.1.0.0	10.1.0.64	10.1.0.128	10.1.0.192
First Host	10.1.0.1	10.1.0.65	10.1.0.129	10.1.0.193
Last Host	10.1.0.62	10.1.0.126	10.1.0.190	10.1.0.254
Broadcast	10.1.0.63	10.1.0.127	10.1.0.192	10.1.0.255

The following table will show the last three subnets and find the valid hosts used in the Class A 255.255.255.192 mask.

Subnet	10.255.255.0	10.255.255.64	10.255.255.128
First Host	10.255.255.1	10.255.255.65	10.255.255.129
Last Host	10.255.255.62	10.255.255.126	10.255.255.190
Broadcast	10.255.255.63	10.255.255.127	10.255.255.191

Summary

Wow, if you made it this far and understood everything the first time through, I am very impressed! We covered a lot of ground in this chapter, and it is the largest chapter in the book. We discussed the Internet Protocol stack, as well as IP addressing and subnetting. This information is important to understand for the CCNA exam, of course, but also in any networking job or production environment you will be building or troubleshooting. It wouldn't hurt to read this chapter more than once and to practice subnetting