

3

CHAPTER THREE

Concepts in IP Addressing

Terms You'll Need to Understand:

- ✓ Binary
 - ✓ Hexadecimal
 - ✓ Decimal
 - ✓ Octet
 - ✓ IP address
 - ✓ Subnet Mask
 - ✓ Subnet
 - ✓ Host
 - ✓ Increment
-

Techniques You'll Need to Master:

- ✓ Identifying Address Class and Default Mask
- ✓ Determining Host Requirements
- ✓ Determining Subnet Requirements
- ✓ Determining the Increment

Introduction

The CCNA exam(s) require a perfect fluency in subnetting. Success requires speed and accuracy in answering the many questions you will see on this topic. The key to this level of fluency is practice—you must work at your skills until they become second nature.

The following sections discuss binary and hexadecimal numbering systems as compared with the more familiar decimal system. An understanding of binary, in particular, is crucial to success on the test as it is fundamental to computer systems in general, and to topics such as subnetting, access lists, routing, and route summarization.

Binary

Binary is the *language* of digital electronic communication. Binary is another name for Base2 numbering. Our usual numbering system is Base10, in which a single character or column can represent one of 10 values: 0, 1, 2, 3, 4, 5, 6, 7, 8, or 9. The first column indicates how many ones there are in a given value. To represent a value greater than 9, we need another column, which represents how many “tens” there are; if the value we want to represent is greater than 99, we use another column for the “hundreds,” and so on. You might notice that each additional column is ten times greater than the preceding one: ones, tens, hundreds, thousands, and so forth—all “Powers of 10”: 10^1 , 10^2 , 10^3 , and so on. Base10 is easy because most of us have 10 fingers and have known how to count from an early age.

In binary, or Base2, a single character or column can represent one of only two values: 0 or 1. The next column represents how many “twos” there are; the next column how many “fours,” and so on. You’ll notice here that the value of each additional column is two times greater than the previous—all “Powers of 2”: 2^1 , 2^2 , 2^3 , and so on. This is not a coincidence.

Given that a Base2 or binary column can have only two possible values (0 or 1), this makes it easy to represent a binary value as an electrical value: either off (0) or on (1). Computers use binary because it is easily represented as electrical signals in memory or digital values on storage media. The whole system works because computers are quick at computing arithmetic, and as you’ll learn, pretty much all computer operations are really just fast binary math.

Let’s take a look at some Base10 (or decimal) to binary conversions. Take the decimal number 176. Those three digits tell us that we have one 100, plus seven 10s, plus six 1s. Table 3.1 illustrates how decimal numbers represent this distribution of values.

TABLE 3.1 Decimal Values

100,000s	10,000s	1000s	100s	10s	1s
0	0	0	1	7	6

Notice that we have some zeroes in the high-value columns; we can drop those from the beginning if we want to. You will not have to analyze decimal numbers in this way on the exam; we are simply demonstrating how Base10 works so it can be compared to Base2 and Base16 in the same way.

In binary, the columns have different values—the powers of 2. Table 3.2 lists the values of the lowest eight bits in binary.

TABLE 3.2 Binary Values

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

NOTE

The biggest values in a binary string (the ones at the left) are often called the “high-order” bits because they have the highest value. Similarly, the lowest-value bits at the right are referred to as the “low-order” bits.

TIP

You must know the value of each binary bit position! If you have difficulty memorizing them, try starting at 1 and keep doubling as you go to the left.

To represent the decimal number 176 in binary, we need to figure out which columns (or bit positions) are “on” and which are “off.” Now, because this is arithmetic, there are a few different ways to do this.

Start with the decimal number you want to convert:

176

Next, look at the values of each binary bit position and decide if you can subtract the highest column value and end up with a value of 0 or more. Ask yourself: “Can I subtract 128 from 176?” In this case, $176 - 128 = 48$.

Yes, you can subtract 128 from 176 and get a positive value, 48. Because we “used” the 128 column, we put a 1 in that column, as shown in Table 3.3.

TABLE 3.3 Building a Binary String, Part 1

128	64	32	16	8	4	2	1
1							

Now, we try to subtract the next highest column value from the remainder. We get $176 - 128 = 48$. We take the 48 and subtract 64 from it.

Notice that you can't do this without getting a negative number; this is not allowed, so we can't use the 64 column. Therefore, we put a 0 in that column, as shown in Table 3.4.

TABLE 3.4 Building a Binary String, Part 2

128	64	32	16	8	4	2	1
1	0						

Move along and do the math for the rest of the columns: $48 - 32 = 16$. We then subtract 16 from 16 and get 0.

Note that when you get to 0, you are finished—you need to only fill the remaining bit positions with 0s to complete the 8-bit string. So, we used only the 128 column, the 32 column, and the 16 column. Table 3.5 is what we end up with.

TABLE 3.5 Completed Binary Conversion

128	64	32	16	8	4	2	1
1	0	1	1	0	0	0	0

176 decimal = 10110000 binary.

If you add up $128+32+16$, you get 176. That is how you convert from binary to decimal: Simply add up the column values where there is a 1.

EXAM ALERT

You will see several questions on converting from decimal to binary and back, so prepare accordingly.

Hexadecimal

The CCNA exam(s) will ask you a few questions on the conversion of binary to hexadecimal and back, so you need to understand how it works. An understanding of hex is also a useful skill for other areas of networking and computer science.

Binary is Base2; Decimal is Base10; Hexadecimal is Base16. Each column in hex can represent 16 possible values, from 0 through 15. In order to represent a value of 10 through 15 with a single character, hex uses the letters A through F. It is important to understand that the values of 0 through 15 are the possible values of a 4-bit binary number, as shown in Table 3.6.

TABLE 3.6 Decimal, Binary, and Hex Values Compared

Decimal	Binary	Hex
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

EXAM ALERT

You should be able to reproduce Table 3.6 as a quick reference for the exam.

Conversion Between Binary, Hex, and Decimal

The following sections provide an introduction to converting between binary, hex, and decimal. Again, there is more than one mathematical approach to finding the correct answer, but the method shown is simple and reliable.

Decimal to Hexadecimal Conversions

The easiest way to get from decimal to hexadecimal and back is to go through binary. Take the example we used earlier in which we converted 176 decimal to binary:

$$176 = 10110000$$

Given that a single hex character represents 4 binary bits, all we need to do is to break the 8-bit string 10110000 into two 4-bit strings like this:

$$1011 \quad 0000$$

Now, simply match the 4-bit strings to their hex equivalent:

$$1011 = B$$

$$0000 = 0$$

The answer is simply $10110000 = 0xB0$.

The “0x” in front of the answer is an expression that means “the following is in hex.” This is needed because if the hex value was 27, we could not distinguish it from 27 decimal.

Hexadecimal to Decimal Conversions

The reverse of the procedure is easier than it seems, too. Given a hex value of 0xC4, all we need to do is to first convert to binary, and then to decimal.

To convert to binary, take the two hex characters and find their binary value:

$$C = 1100$$

$$0100 = 4$$

Now, make the two 4-bit strings into one 8-bit string:

$$11000100$$

Finally, add the bit values of the columns where you have a 1:

$$128 + 64 + 4 = 196$$

EXAM ALERT

It is critical to polish your skills in binary. You must be confident and quick in conversions, and the better your understanding of binary, the easier subnetting and other advanced IP topics will be for you. Practice, practice, practice!

IP Address Components

CCNA candidates need to be fluent in their understanding of IP addressing concepts. The following sections detail how IP addresses are organized and analyzed, with a view to answering subnetting questions.

Address Class

Early in the development of IP, the IANA (Internet Assigned Numbers Authority) designated five classes of IP address: A, B, C, D, and E. These classes were identified based on the pattern of high-order bits (the high-value bits at the beginning of the first octet). The result is that certain ranges of networks are grouped into classes in a pattern based on the binary values of those high-order bits, as detailed in Table 3.7:

TABLE 3.7 Address Class and Range

Class	High-Order Bits	1st Octet Range
A	0	1–126
B	10	128–191
C	110	192–223
D	1110	224–239
E	11110	240–255

You might notice that 127 is missing. This is because at some point the address 127.0.0.1 was reserved for the loopback (sometimes called “localhost”) IP—this is the IP of the TCP/IP protocol itself on every host machine.

EXAM ALERT

You absolutely must be able to identify the class of an address just by looking at what number is in the first octet. This is critical to answering subnetting questions.

Default Subnet Mask

Each class of address is associated with a default subnet mask, as shown in Table 3.8. An address using its default mask defines a single IP broadcast domain—all the hosts using that same network number and mask can receive each other’s broadcasts and communicate via IP.

TABLE 3.8 Address Class and Default Masks

Class	Default Mask
A	255.0.0.0
B	255.255.0.0
C	255.255.255.0

One of the rules that Cisco devices follow is that a subnet mask must be a contiguous string of 1s followed by a contiguous string of 0s. There are no exceptions to this rule: A valid mask is always a string of 1s, followed by 0s to fill up the rest of the 32 bits. (There is no such rule in the real world, but we will stick to the Cisco rules here—it’s a Cisco exam, after all.)

Therefore, the only possible valid values in any given octet of a subnet mask are 0, 128, 192, 224, 240, 248, 252, 254, and 255. Any other value is invalid.

EXAM ALERT

You should practice associating the correct default subnet mask with any given IP address; this is another critical skill in subnetting.

The Network Field

Every IP address is composed of a network component and a host component. The subnet mask has a single purpose: to identify which part of an IP address is the network component and which part is the host component. Look at a 32-bit IP address expressed in binary, with the subnet mask written right below it. Figure 3.1 shows an example.

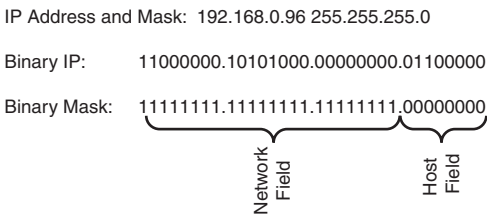


Figure 3.1 IP address and mask in binary, showing network and host fields.

Anywhere you see a binary 1 in the subnet mask, it means “the matching bit in the IP address is part of the network component.” In this example, the network part of the address is 192.168.0.X, and the last octet (X) will be the host component.

Because there are 24 bits in a row in the mask, we can also use a shortcut for the mask notation of /24. These examples show how a dotted decimal mask can be expressed in slash notation:

192.168.1.66 255.255.255.0 = 192.168.1.66 /24

172.16.0.12 255.255.0.0 = 172.16.0.12 /16

10.1.1.1 255.0.0.0 = 10.1.1.1 /8

This slash notation is sometimes called CIDR (Classless Inter-Domain Routing) notation. For some reason, it's a concept that confuses students, but honestly it's the easiest concept of all: The slash notation is simply the number of 1s in a row in the subnet mask. The real reason to use CIDR notation is simply that it is easier to say and especially to type—and it appears interchangeably with dotted decimal throughout the exam. CIDA notation also appears in the output of various IOS commands.

Every IP address has a host component and a network component, and the 1s in the mask tell us which bits in the address identify the network component.

The Host Field

If the 1s in the mask identify the network component of an address, the 0s at the end of the mask identify the host component. In the preceding example, the entire last octet is available for the host IP number.

The number of 0s at the end of the mask mathematically define how many hosts can be on any given network or subnet. The 1s in the mask always identify the network component, and the 0s at the end of the mask always identify the host component of any IP address.

Non-Default Masks

At this point, you should be able to recognize what class an address belongs to, and what its default mask is supposed to be. Here's the big secret: If a mask is longer than it is supposed to be, that network has been subnetted. So it is clearly another critical skill that you be able to spot those non-default masks.

The Subnet Field

Because we have extended the subnet mask past the default boundary into the bits that were previously host bits, we identify the bits we "stole" from the host part as the subnet field. The subnet field is relevant because those bits mathematically define how many subnets we create. Figure 3.2 uses the same IP address from our previous example, but now we have applied a subnetted mask that is longer than the default. Note that this creates the subnet field.

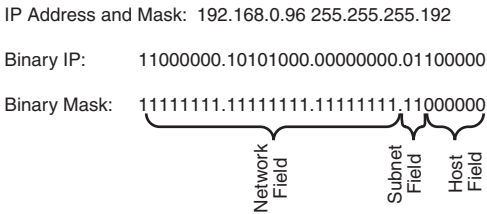


FIGURE 3.2 IP address and non-default mask in binary illustrating the subnet field.

Figure 3.2 identifies the two extra bits past the default boundary as the subnet field—they used to be in the host field, but we subnetted and stole them to become the subnet field.

Subnetting

Subnetting is not as difficult as it initially seems. Because we are dealing with arithmetic, there is definitely more than one way to do this, but the method shown here has worked well. The following sections work through the process of subnetting. Then, we work on some shortcuts to show how you can subnet quickly because CCNA exam candidates often find that they are pressed for time on the exam.

Address Class and Default Mask

Subnetting happens when we extend the subnet mask past the default boundary for the address we are working with. So it's obvious that we first need to be sure of what the default mask is supposed to be for any given address. Previously, we looked at the IANA designations for IP address classes and the number ranges in the first octet that identify those classes. If you didn't pick up on this before, you should memorize those immediately.

When faced with a subnetting question, the first thing to do is decide what class the address belongs to. Here are some examples:

192.168.1.66

The first octet is between 192 and 223: Class C

Default mask for Class C: 255.255.255.0

188.21.21.3

The first octet is between 128 and 191: Class B

Default mask for Class B: 255.255.0.0

24.64.208.5

The first octet is between 1 and 126: Class A

Default mask for Class A: 255.0.0.0

It's important to grasp that if an address uses the correct default mask for its class, it is not subnetted. This means that regardless of how many hosts the 0s at the end of the mask create, all those hosts are on the same network, all in the same broadcast domain. This has some implications for *classful networks* (ones that use the default mask for the address). Take a Class A for example: A Class A network can have 16,777,214 hosts on it. Almost 17 million PCs on one network would never work—there would be so much traffic from broadcasts alone, never mind regular data traffic, that nothing could get through and the network would collapse under its own size. Even a Class B network has 65,534 possible host IPs. This is still too many. So, either we waste a lot of addresses by not using the whole classful A or B network, or we subnet to make the networks smaller.

This is actually one of the most common reasons we subnet: The default or classful networks are too big, causing issues such as excessive broadcast traffic and wasted IP address space. Subnetting creates multiple smaller subnetworks out of one larger classful network, which allows us to make IP networks the “right” size—big or small—for any given situation.

The Increment

By definition, the process of subnetting creates several smaller classless subnets out of one larger classful one. The spacing between these subnets, or how many IP addresses apart they are, is called the *Increment*. Because we are working with binary numbers, a pattern emerges in which the Increment is always one of those powers of 2 again—another good reason to memorize those numbers.

The Increment is really easy to figure out. It is simply the value of the last bit in the subnet mask. Let's look at some examples. Figure 3.3 shows an IP address and subnet mask in binary.

IP Address and Mask: 192.168.21.1 255.255.255.0

Binary IP: 11000000.10101000.00010101.00000001

Binary Mask: 11111111.11111111.11111111.11000000

FIGURE 3.3 IP address and mask in binary.

Note that this is a Class C address, and it uses the correct default mask—so it is not subnetted. This means that there is only one network, so there isn't really an increment to worry about here. It's sufficient at this point to recognize that an address that uses its default mask creates one network (no subnets), so there is no spacing between subnets to consider.

Let's take the same address and subnet it by extending the mask past the default boundary, as shown in Figure 3.4.

IP Address and Mask: 192.168.21.1 255.255.255.192

Binary IP: 11000000.10101000.00010101.00000001

Binary Mask: 11111111.11111111.11111111.11000000

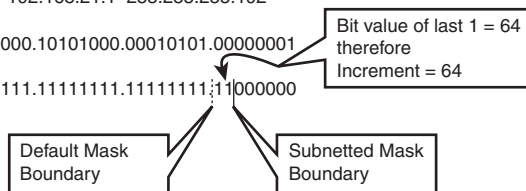


FIGURE 3.4 IP address and subnetted mask.

The very last bit in the subnet mask in the figure is in the bit position worth 64—so the Increment in this case is 64, which means that the subnets we made are evenly spaced at 64 IP addresses apart.

Think about this for a second. We are doing the subnetting in the fourth octet—that is where the mask changes from 1s to 0s. (The octet where this happens is sometimes referred to as the “Interesting” octet.) The lowest possible value in that fourth octet is 0. If the subnets are 64 IP addresses apart, this means that the first subnet starts at 0, the next one starts at 64, the third at 128, and the fourth at 192—all multiples of the Increment. Note that if we add another 64 to that last 192, we get 256—and that is larger than 255, the largest value that is possible in one octet. So this means we only have room for four subnets. Figure 3.5 illustrates this pattern more clearly:

.0	.64	.128	.192
.1	.65	.129	.193
.	.	.	.
.	.	.	.
.	.	.	.
.62	.126	.190	.254
.63	.127	.191	.255

FIGURE 3.5 Subnets created with Increment of 64.

The multiples of the Increment—0, 64, 128, and 192—are the starting addresses of the subnets we created. The subnets are all 64 addresses long, so we have room to make four subnets before we run out of addresses in the fourth octet.

Figure 3.6 shows our IP and subnet mask—note that the value of the last bit in the mask is 16—and the subnets created with that Increment of 16.

192.168.21.0 255.255.255.240

IP: 11000000.10101000.00010101.00000000
Mask: 11111111.11111111.11111111.11110000

Subnets Created with Increment of 16:

.0	.16	.32	.48	.64	.80	.96	.112	.128	.144	.160	.176	.192	.208	.224	.240
.
.
.
.15	.31	.47	.63	.79	.95	.111	.127	.143	.159	.175	.191	.207	.223	.239	.255

FIGURE 3.6 IP address and subnet mask with Increment of 16.

First of all, you should notice that we are subnetting again—the mask extends past the default boundary. The last 1 in the mask is in the bit position worth 16, so our Increment is 16. The multiples of 16 are 0, 16, 32, 48, 64, 80, 96, 112, 128, 144, 160, 176, 192, 208, 224, and 240. Again, we can't make another subnet because $240 + 16 = 256$. Be careful not to start doubling as we did with the binary values; here we are just adding the Increment value each time. It's easy to get confused!

The Increment is really the key to subnetting; if you can determine the Increment, you can see how big your subnets are and how many you have created. Remember, the easy way to find the Increment is to just determine the bit value of the last 1 in the mask.

Number of Hosts

The number of 0s at the end of the mask always defines the number of hosts on any network or subnet. There is a simple mathematical formula that defines how many IP addresses are available to be assigned to hosts.

NOTE

Hosts is another word for computers, router interfaces, printers, or any other network component that can be assigned an IP address.

Now, no one expects you to be a big fan of algebra, but you need to see and understand the formula.

The number of binary bits you have to use determines the maximum number of different values you can express using those bits. If you have three bits, you can make eight different values—0 through 7, or 000 through 111 in binary. Three bits, and $2^3=8$ —this is not a coincidence. The binary values you learned earlier—1, 2, 4, 8, 16, 32, 64, 128—are all powers of 2 and define the maximum number of different values you can create if the mask ends in that bit position. So it should come as no surprise that the formula for the number of hosts on any network or subnet is 2^H-2 , where H is the number of 0s at the end of the mask.

But why do we subtract 2 in the formula? It's pretty straightforward: Every network or subnet has two reserved addresses that cannot be assigned to a host. The rule is that no host can have the IP address in which all the host bits are set to 0, and no host can have the IP address in which all the host bits are set to 1. These addresses are called the Network ID and the Broadcast ID, respectively. They are the first and last IPs in any network or subnet. We lose those two IP addresses from the group of values that could be assigned to hosts.

Think of a network or subnet as a street with houses on it. Each house has a unique address, and the street has a name. The Network ID is like the street name, and all the houses are hosts on a subnet that is known by its Network ID street name. If two hosts have identical network and subnet fields in their addresses, they are on the same network, and can ping each other and exchange data and all that good stuff. If the network and subnet fields are different, even by one bit, they are on different networks and can't communicate until we put a router between them. The routers act like street intersections; you must get to the right intersection (router) before you can get on to the street you want... but we'll save that for later.

In a network where there are no routers, devices running TCP/IP make a decision about whether a particular IP address is on the network by performing a logical AND operation. The AND is a Boolean function that works like this:

1 AND 1 = 1
1 AND 0 = 0
0 AND 0 = 0

This operation applies to IP networking like this: A host does a logical AND between its own IP and its mask. This determines its Network ID. The host can then do an AND between another IP address and its own mask to determine if that second address is on the same network or some other one.

Let's take the IP address and mask of an imaginary host and display them in binary, as shown in Figure 3.7. The AND operation takes each bit in the address and ANDs it with the corresponding bit in the mask below it; the result is the NetID of the host.

```
IP Address and Mask: 192.16.20.12 255.255.255.0  
  
Binary IP:      11000000.00010000.00010100.00001100  
  
Binary Mask:   11111111.11111111.11111111.00000000  
                _____  
AND Result:    11000000.00010000.00010100.00000000  
  
                NetID = 192.16.20.0
```

FIGURE 3.7 The AND operation determines the NetID.

Now the host knows its own NetID and can compare any other host's address to that to see if the other host has the same NetID. If the two NetIDs are different, traffic has to be sent through a router to get to the other network—and if there is no router, the two hosts can't communicate.

EXAM ALERT

Being able to do the AND operation is a useful skill; a lot of test questions center around the NetID, and being able to find it quickly is a big help.

The Broadcast ID

The *Broadcast ID* is the address of everybody on that network or subnet. Sometimes called a directed broadcast, it is the common address of all hosts on that Network ID. This should not be confused with a full IP broadcast to the

address of 255.255.255.255, which hits every IP host that can hear it; the Broadcast ID hits only hosts on a common subnet.

Let's take the previous example of an Increment of 64 and expand on the detail, as shown in Figure 3.8:

Subnets Created with Increment of 64 – NetID and Broadcast ID shown:

.0 N	.64 N	.128 N	.192 N
.1	.65	.129	.193
.	.	.	.
.	.	.	.
.	.	.	.
.62	.126	.190	.254
.63 B	.127 B	.191 B	.255 B

FIGURE 3.8 Subnets from Increment of 64 with NetID and Broadcast ID shown.

Note that all the multiples of the Increment—the numbers that mark the start of each subnet—have been identified by an “N” for Network ID, and the last IP in every subnet is marked with a “B” for Broadcast ID. This leaves us with 62 IPs left over in each subnet, and any of these (but only these) can be assigned to a host.

This leaves us with a range of IP addresses within every network or subnet that can be assigned to hosts. There is an unofficial convention that the gateway or router for a subnet is assigned the first or the last IP address available, but that is entirely arbitrary.

EXAM ALERT

You need to know exactly what the first and last IP addresses are in any subnet; a lot of questions ask for them, and it's fundamental to understanding what is happening when you subnet.

The first valid IP address is defined as

NetID + 1

In Figure 3.8, the first valid host IPs in each subnet are .1, .65, .129, and .193.

The last valid host is defined as

BroadcastID – 1

In Figure 3.8, the last valid host IPs in each subnet are .62, .126, .190, and .254.

TIP

Here are some handy tips to help you keep track of the NetID, first and last hosts, and Broadcast ID:

NetID:	Always Even
First Host:	Always Odd
Last Host:	Always Even
Broadcast ID:	Always Odd

See how the subnetted mask in the previous example has shortened the number of 0s at the end of the mask as compared to the default of 8? We now have only six 0s in the host part, so our formula would be

$$2^6 - 2 = 62$$

Here's something interesting: It doesn't matter what IP address you use with this mask; that mask will always give you 62 hosts on each subnet. You can pick a Class A address, say 22.1.1.0, and that mask would still make 62 hosts per subnet. The number of 0s at the end of the mask always drives how many hosts are on each subnet, regardless of the address.

So, what happened to all the other host IPs we started with? Remember that subnetting takes a classful A, B, or C network and splits it into several equal-sized pieces. It's just like cutting a pie into pieces; the original amount of pie is still there, but each piece is now separate and smaller.

Remember that the number of 0s at the end at the mask always defines how many hosts are on each subnet, regardless of the address in use.

Number of Subnets

Following on with the pie analogy, we know that we slice a classful network into pieces—but how many pieces? There is a simple mathematical relationship to this as well, but it is slightly more complex because of an old rule that we sometimes have to deal with.

The basic formula for the number of subnets is similar to the hosts formula. It is simply 2^S , where S is the number of bits in the subnet field—that means the number of 1s in the mask past the default boundary for that address. If you look at Figure 3.9, you can see how this works.

The default boundary for that Class C address should be at the 24th bit, where the third octet ends and the fourth begins. The subnetted mask extends that by 2 bits into the fourth octet. So, we have stolen 2 bits, and our formula would look like this:

$$\# \text{ of subnets} = 2^S$$

$$S = 2$$

$$2^2 = 4$$

IP Address and Mask: 192.168.21.1 255.255.255.192

Binary IP: 11000000.10101000.00010101.00000001

Binary Mask: 11111111.11111111.11111111.11000000



FIGURE 3.9 Subnetted Class C with Increment of 64.

We made four subnets, as you saw earlier. To figure out how many bits we stole, we first must know where the default boundary is so that we know where to start counting. This is where knowing the address classes and the correct default masks is critical; if you can't figure this out, you will not be able to answer most subnetting questions correctly, and that would be bad.

Now here's where things get tricky. A rule that some older systems use says that the first and last subnets created are invalid and unusable. The rule is known as the Subnet Zero Rule, and obviously if it is in effect, we lose two subnets from the total we create. These two subnets will be referred to from now on as the *zero subnets*. Newer systems do not use the Zero Subnets Rule, including newer Cisco devices. This is confusing and makes things more difficult—but difficult is not something Cisco shies away from on its certification exams. So if you want your CCNA, pay attention to the question and don't complain about how hard it is.

EXAM ALERT

Cisco tests might be difficult and tricky, but they are fair—they will not withhold information you need to answer the question. The test question will always tell you whether somehow the Zero Subnets Rule is in effect; yes, both types of questions are asked.

The Cisco IOS supports the use of the Zero Subnets. The command “ip subnet zero” turns on the ability to use them, so that might be how the question is telling you whether they are in effect. Once you pass your CCNA, you will not likely have to worry about the Zero Subnets Rule again, unless you lose your mind and decide to become a Cisco trainer.

TIP

After you determine whether the zero subnets are available, use the following to get the calculation for the number of subnets right:

Zero subnets not available?

Subtract 2 subnets: formula is $2S-2$

Zero subnets available?

Keep all subnets: formula is $2S$

Working with Subnetting Questions

EXAM ALERT

The approach you need to take to any subnetting question is very simple. After you become fluent in subnetting, you can take some shortcuts; but to build a solid understanding, you need to be methodical.

Every subnetting question you ever see will be about one of three things:

- ▶ Number of hosts
- ▶ Number of subnets
- ▶ The Increment

Your task will be to simply figure out what the question is asking for and solve it without getting confused or distracted.

Determining Host Requirements

There are only two scenarios when determining the host requirements: Either you are given a mask and asked how many hosts per subnet this creates, or you are given a requirement for a certain number of hosts and asked to provide the appropriate mask. Either way, the number of 0s at the end of the mask drives how many hosts per subnet there will be; the address to which that mask is applied is irrelevant. Your task is to put the correct number of 0s at the end of the mask such that 2^H-2 is greater than or equal to the desired number of hosts, or to determine what the value of 2^H-2 actually is. From there, you must choose the correct expression of the mask, either in dotted decimal or CIDR notation.

Determining Subnet Requirements

The scenarios for determining subnet requirements are quite similar to the host questions; either you are told how many subnets you need and asked to provide the appropriate mask, or you are given a mask and asked how many subnets it creates. Note that in both cases (unlike hosts questions), you must know the IP address or at least the class of address you are working with. Creating subnets happens by extending the default mask, so you must know where the mask should

end by default—and for that you need to know the class of address. Once you know where to start, simply extend the mask by the correct number of subnet bits such that $2S-2$ (or possibly just $2S$) gives you the correct number of subnets.

EXAM ALERT

Remember that the zero subnets rule might come into play here; although the majority of questions say that the zero subnets are not valid and therefore the formula should be $2S-2$, some questions—and probably more as time goes on—will clearly state that zero subnets are available. Read the question!

Determining Increment-based Requirements

Increment questions are the most challenging and complex subnetting questions, often requiring you to do a lot of legwork before you can get to the answer.

Increment questions often give you two or more IP addresses and masks, and ask you things such as, “Why can’t Host A ping Host B?” The answer could be that A and B are on different subnets; to determine this, you need to understand where those subnets begin and end, and that depends on the Increment. Another popular question gives you several IP addresses and masks that are applied to PCs, servers, and routers. The system, as it is described, is not working, and you need to determine what device has been incorrectly configured—perhaps two IPs in different subnets, perhaps a host that is using a NetID or BroadcastID as its address.

The key is to first determine what the Increment is or should be; then, carefully plot out the multiples of the Increment—the Network IDs of all the subnets. Then you can add the Broadcast IDs, which are all one less than the next Network ID. Now you have a framework into which you can literally draw the host IP ranges, without risk of “losing the picture” if you do this all in your head.

All of these skills take practice. Everyone goes through the same process in learning subnetting: For quite a while, you will have no idea what is going on—then suddenly, the light goes on and you “get it.” Rest assured that you will get it. It takes longer for some than others, and you do need practice or you will lose the skill.

The Subnetting Chart

So now you should understand concepts and mechanics of subnetting. You can do it and get the right answer almost all of the time, but it takes you a while. This is good—congratulations! If you are not at that point yet, you should practice more before you look at this next section.

What follows is one of many variations of a subnetting chart. This is a good one because it is easy to use under pressure when your brain will behave unpredictably.

CAUTION

You must be able to re-create this chart exactly and correctly before you start your exam. If you make a simple mistake in creating your chart, you could easily get all of your subnetting questions wrong, and that would probably cause you to fail.

The chart represents the last two octets of a subnet mask, and what effect a 1 or a 0 in the different bit positions will have. It lists the Increment, CIDR notation, the mask in decimal, the number of hosts created, and the number of subnets formed from a Class B and C address. Figure 3.10 shows a completed version.

Increment	128	64	32	16	8	4	2	1	128	64	32	16	8	4	2	1
CIDR:	/17	/18	/19	/20	/21	/22	/23	/24	/25	/26	/27	/28	/29	/30	/31	/32
Mask:	128	192	224	240	248	252	254	255	128	192	224	240	248	252	254	255
Hosts:	32,766	16,382	8190	4094	2046	1022	510	254	126	62	30	14	6	2	-	-
B Subnets:	0	2	6	14	30	62	126	254	510	1024	2046	4094	8190	16,382	-	-
C Subnets:	1	1	1	1	1	1	1	1	0	2	6	14	30	62	-	-

FIGURE 3.10 The subnetting chart.

Following are steps to recreate the chart:

1. The first row is simply the binary bit position values—the powers of 2. Start at the right with 1 and keep doubling the value as you go left: 1, 2, 4, 8, 16, 32, 64, 128. Repeat for the third octet.
2. The second row is the CIDR notation—the number of 1s in a row in the mask. Our chart starts at the 17th bit, so number the second row starting at 17, through 32.
3. The third row is the mask in binary. Add consecutive bit values together from left to right to get the valid mask values of 128, 192, 224, 240, 248, 252, 254, and 255. Or you can just memorize them.
4. The fourth row is the number of hosts created. Starting at the right side of the fourth octet, subtract 2 from the increment line (the first line) and enter that value. Do this for the whole fourth octet. When you get to the third octet (the left half of the chart), you will have to change your approach: The value will keep increasing in the same pattern, but subtracting 2 from the top row won't work anymore because the top row resets for the third octet. The simplest approach is to double the last value and add 2. For example, $(126 \times 2) + 2 = 254$; $(254 \times 2) + 2 = 510$; and so on.
5. The fifth row is the number of subnets created from a Class B address. Starting at the left side of the chart (the third octet), repeat the values from the fourth line, but in reverse order. Remember to start with a single 0 instead of two.

CAUTION

Remember that the Subnet Zero Rule will change your answers and how you use your chart. If the Zero Subnets are allowed, add 2 to the values in lines 5 and 6 of your chart in the appropriate octet.

6. The sixth row of the chart is the number of subnets created from a Class C address. Remember, with a Class C, we do not make any subnets (that is, we have only one network) in the third octet, so we have all 1s there. For the fourth octet, the numbers are the same as in Row 5; just start them in the fourth octet instead. The same caution and tactic about the Zero Subnets applies.

Provided you have built it correctly, your chart is a huge help in answering subnetting questions quickly and accurately. All you need to do is determine what the question is asking for, and then look up that value on your chart. All of the answers you need will be in the same column. Practice building and using the chart until it becomes something you can do without thinking. You will need your brain for other more complicated problems.

Exam Prep Questions

1. Which of the following are alternate representations of the decimal number 227?
Choose 2.
 - A. 0x227
 - B. 11100011
 - C. 0x143
 - D. 0xE3
 - E. 11100110
2. Which of the following are alternate representations of 0xB8? Choose two.
 - A. 10110100
 - B. 10111111
 - C. 10111000
 - D. 184
 - E. 0x184
3. You have been asked to create a subnet that supports 16 hosts. What subnet mask should you use?
 - A. 255.255.255.252
 - B. 255.255.255.248
 - C. 255.255.255.240
 - D. 255.255.255.224
4. Given the mask 255.255.254.0, how many hosts per subnet does this create?
 - A. 254
 - B. 256
 - C. 512
 - D. 510
 - E. 2

5. You are a senior network engineer at True North Technologies. Your boss, Mr. Martin, asks you to create a subnet with room for 12 IPs for some new managers. Mr. Martin promises that there will never be more than 12 managers, and he asks you to make sure that you conserve IP address space by providing the minimum number of possible host IPs on the subnet. What subnet mask will best meet these requirements?
- A. 255.255.255.12
 - B. 255.255.255.0
 - C. 255.255.240.0
 - D. 255.255.255.240
 - E. 255.255.255.224
6. Your boss Duncan does not seem to be able to grasp subnetting. He comes out of a management meeting and quietly asks you to help him with a subnetting issue. He needs to divide the Class B address space the company uses into six subnets for the various buildings in the plant, while keeping the subnets as large as possible to allow for future growth. Because the company has not upgraded their Cisco equipment since it was purchased several years ago, none of the routers supports the “ip subnet zero” command. What is the best subnet mask to use in this scenario?
- A. 255.255.0.0
 - B. 255.255.248.0
 - C. 255.255.224.0
 - D. 255.255.240.0
 - E. 255.255.255.224
7. You have purchased several brand-new Cisco routers for your company. Your current address space is 172.16.0.0 /22. Because these new routers support the “ip subnet zero” command, you realize you are about to gain back two subnets that you could not use with the old gear. How many subnets total will be available to you once the upgrades are complete?
- A. 4
 - B. 2
 - C. 32
 - D. 62
 - E. 64

8. Which of the following are true about the following address and mask pair: 10.8.8.0 /24? Choose all that apply.

- A. This is a Class B address.
- B. This is a Class A address.
- C. This is a Class C address.
- D. 16 bits were stolen from the host field.
- E. 24 bits were stolen from the host field.
- F. The default mask for this address is 255.0.0.0.
- G. The mask can also be written as 255.255.255.0.
- H. The mask creates 65,536 subnets total from the default address space.
- I. Each subnet supports 256 valid host IPs.
- J. Each subnet supports 254 valid host IPs.

9. Indy and Greg have configured their own Windows XP PCs and connected them with crossover cables. They can't seem to share their downloaded MP3 files, however. Given their configurations, what could be the problem?

Indy's configuration:

IP: 192.168.0.65

Mask: 255.255.255.192

Greg's configuration:

IP: 192.168.0.62

Mask: 255.255.255.192

- A. Indy is using a Broadcast ID for his IP.
 - B. Greg is using an invalid mask.
 - C. Indy's IP is in one of the Zero Subnets.
 - D. Greg and Indy are using IPs in different subnets.
10. You are given an old router to practice for your CCNA. Your boss Dave has spent a lot of time teaching you subnetting. Now he challenges you to apply your knowledge. He hands you a note that says:

"Given the subnetted address space of 192.168.1.0 /29, give the E0 interface the first valid IP in the eighth subnet. Give the S0 interface the last valid IP in the twelfth subnet. The Zero Subnets are available. You have 10 minutes. Go."

Which two of the following are the correct IP and Mask configurations? Choose 2.

- A. E0: 192.168.1.1 255.255.255.0
- B. E0: 192.168.1.56 255.255.255.248
- C. E0: 192.168.1.57 255.255.255.248
- D. S0: 192.168.1.254 255.255.255.0
- E. S0: 192.168.1.95 255.255.255.248
- F. S0: 192.168.1.94 255.255.255.248

11. The following questions are part of a Subnetting SuperChallenge. This monster question will stretch your subnetting skills, especially if you give yourself a time limit. Start with 10 minutes and see if you can get down to 5.

The Vancouver Sailing Company has four locations: a head office and three branch offices. Each of the branches is connected to the head office by a point-to-point T1 circuit. The branches have one or more LANs connected to their routers. The routers are called Main, Jib, Genoa, and Spinnaker. The company has been assigned the 172.16.0.0/20 address space to work within.

Your task will be to choose the correct IP address and mask for each interface, based on the information provided. Remember that no IP address may overlap with any address in another subnet, and that the required number of hosts for each subnet will affect your decision as to which address to use.

Here are the known IP configurations for the routers:

Main:

S0/0:172.16.0.1 /30

S0/1:172.16.0.5 /30

S0/2:172.16.0.9 /30

Fa1/0:172.16.4.1 /23

Fa1/1:172.16.6.1 /23

Jib:

S0/0: Connects to Main S0/0.

Fa1/0: 172.16.8.33/27.

Fa1/1: 30 hosts needed.

Genoa:

S0/0: Connects to Main S0/1.

Fa1/0: 172.16.8.129/26.

Fa1/1: 100 hosts needed.

Fa2/0: 100 hosts needed.

Fa2/1: 172.16.13.0/24.

Spinnaker:

S0/0: connects to Main S0/2.

Fa1/0: 500 Hosts needed.

Choose the correct IP and mask assignments for each router:

- A.** Jib Fa1/1: 172.16.8.62/27
- B.** Jib Fa1/1: 172.16.8.64/27
- C.** Jib Fa1/1: 172.16.8.65/28
- D.** Jib Fa1/0: 172.16.8.65/27
- E.** Jib Fa1/1: 172.16.8.65/27
- F.** Genoa S0/0: 172.16.0.2/30
- G.** Genoa S0/0:172.16.0.6/30
- H.** Genoa Fa1/1:172.16.12.1/26
- I.** Genoa Fa1/1:172.16.12.1/25
- J.** Genoa Fa2/0:172.16.12.129/24
- K.** Genoa Fa2/0:172.16.12.129/25
- L.** Genoa Fa2/1:172.16.12.193/25
- M.** Genoa Fa2/1:172.16.13.1/25
- N.** Genoa Fa0/2:172.16.13.1/24
- O.** Spinnaker S0/0: 172.16.0.10/30
- P.** Spinnaker S0/0: 172.16.0.12/30
- Q.** Spinnaker Fa1/0: 172.16.13.0/23
- R.** Spinnaker Fa1/0: 172.16.14.0/23
- S.** Spinnaker Fa1/0: 172.16.14.1/23

Answers to Exam Prep Questions

1. Answers B and D are correct. Answer A in decimal would be 551. Answer C in decimal would be 323. Answer E in decimal is 230.
2. Answers C and D are correct. Answer A in hex is 0xB4. Answer B in hex is 0xBF. Answer E is simply an attempt to trick you—the correct decimal answer is incorrectly expressed as a hex value.
3. Answer D is correct. A will only support 2 hosts; B only 6, and C only 14. Watch out for the minus 2 in the host calculation! Answer C creates 16 hosts on the subnet, but we lose 2—one for the Net ID and one for the Broadcast ID.
4. Answer D is correct. The mask 255.255.254.0 gives us nine 0s at the end of the mask; $2^9 - 2 = 510$. Answer A is checking to see if you missed the 254 in the third octet because you are used to seeing 255. Answer B does the same thing plus tries to catch you on not subtracting 2 from the host calculation. Answer C tries to catch you on not subtracting 2, and Answer E is the Increment of the given mask that you might pick if you were really off track.
5. The correct answer is D. Disregarding for the moment the possibility that Mr. Martin might be wrong, let's look at the requirements. He says make room for 12 managers, and make the subnets as small as possible while doing so. You need to find the mask that has sufficient host IP space without making it bigger than necessary. Answer A is invalid; 12 is not a valid mask value. Remember, a mask is a continuous string of 1s followed by a continuous string of 0s. In answer B, the mask is valid, but it is not correct. This mask has eight 0s at the end, which, when we apply the formula $2^8 - 2$ gives us 254 hosts. That makes more than enough room for the 12 managers, but does not meet the “as small as possible” requirement. Answer C has the correct mask value in the wrong octet. That mask gives us eight 0s in the fourth octet, plus another four in the third octet; that would give us 4094 hosts on the subnet. Answer E gives us 30 hosts per subnet, but that only meets half the requirement. This mask does not provide the minimum number of hosts.
6. The correct answer is C. The default mask for a Class B is 255.255.0.0. Answer C extends that mask by three bits, creating eight subnets ($2^3 = 8$). The Zero Subnets are lost because the routers cannot use them, so we are left with six subnets. Answer A is incorrect because it is the default mask for a Class B and not subnetted at all. Answers B and D are incorrect because although they create sufficient subnets, they do not maximize the number of hosts per subnet and so are not the best answer. Answer E uses the correct mask in the wrong octet.
7. Answer E is correct. With “ip subnet zero” enabled, all 64 subnets created by the mask in use become available. Answers A, B, and C are not even close and are simply distracters. Answer D wants to catch you by subtracting the zero subnets.
8. The correct answers are B, D, F, G, H, and J. Answers A and C are incorrect because this is a Class A address. Answer E is incorrect because only 16 bits were stolen. Answer I is incorrect because it does not subtract the two IPs for the NetID and Broadcast ID.

9. Answer D is correct. With that mask, the Increment is 64. Greg is in the first subnet, and Indy is in the second. Without a router between them, their PCs will not be able to communicate above layer 2. Answer A is incorrect; the Broadcast ID for Indy would be .63. Answer B is incorrect; nothing is wrong with the mask. Answer C is incorrect; the Zero Subnets are the first and last created, and Indy is in the second subnet. The question does not mention the Zero Subnets, and in any case Windows XP fully supports them.
10. The correct answers are C and F. This is an Increment question. The Increment here is 8, so you should start by jotting down the multiples of 8 (those are all the NetIDs), and then noting what 1 less than each of the NetIDs is (those are the Broadcast IDs). From there, it is easy to find what the first and last IPs in each subnet are. (Remember that Dave says we can use the Zero Subnets.) Answers A and D are incorrect because they do not use the subnetted address space Dave requested. Answer B is incorrect because it is a NetID. Answer E is incorrect because it is a Broadcast ID.
11. SuperChallenge answers:
- A. Incorrect (same subnet as Fa1/0)
 - B. Incorrect (Network ID)
 - C. Incorrect (Not enough Hosts)
 - D. Incorrect (Fa1/0 IP already assigned)
 - E. Correct
 - F. Incorrect (Wrong subnet—not on the same network as the connected interface on Main)
 - G. Correct
 - H. Incorrect (Not enough hosts)
 - I. Correct
 - J. Incorrect (Overlaps with Fa1/1)
 - K. Correct
 - L. Incorrect (Overlaps with Fa2/0)
 - M. Incorrect (Not enough hosts)
 - N. Incorrect (There is no Fa0/2 interface on Genoa)
 - O. Correct
 - P. Incorrect (Network ID)
 - Q. Incorrect (Overlaps with Genoa)
 - R. Incorrect (Network ID)
 - S. Correct