IP Subnetting Made Easy

IP subnetting seems to trip up quite a few people in the networking world. I've known experienced consultants who have worked in the industry for 15 years and still resort to subnet calculators.

IP subnetting can be a daunting subject for those who are not familiar with it. The principles of subnetting are based on binary (and some mathematical) principles such as eXclusive-OR (XOR), which, for many people, are foreign concepts or aspects of college courses long since forgotten.

There are really only two times when your average networking people need to know how subnet math truly works: when they study for their first networking certification (the CCNA) and when they study for their last (the CCIE). In fact, I don't really think you need the math for the CCNA coursework, although Cisco makes you learn it. The CCIE exam does make you do bizarre things with subnet masks that require a full understanding of the math behind the topic. However, these things are rarely, if ever, seen in the real world, and are not a topic for this book.

If you want to be able to do IP subnetting in your head, there are a couple of things you will need to understand. First, Cisco, along with every other major manufacturer out there, wants you to learn subnetting its way. Why? So its tests are harder to pass and so everyone who's certified speaks the same language.

Cisco, in particular, seems to want people to think in a way that makes it harder to figure out what's really going on. As an example, in Cisco parlance, a native Class C network with a subnet mask of 255.255.255.224 is said to consist of 6 networks with 30 hosts in each network.

Call me pedantic, but that's incorrect. The subnet mask actually results in 8 networks with 32 hosts each. Cisco's point is that, using classful networking rules, there are only 6 *available* networks, with 30 *available* hosts in each network. While this is a valid concept, I believe it causes confusion, especially given our previous discussion showing how the all-zero and all-ones networks are usable.

The reason Cisco's method causes confusion is simple: there is no easy way to prove the answer. Using the method I will outline here, however, you will always have an easy way to prove your answer. In this case, 8 * 32 = 256.

Everything having to do with subnet masks has something to do with the number 256. In fact, this will be our first rule:

> Every result will either produce 256 or be divisible by 256.

Looking at a subnet mask, the maximum value for an octet is 255. Remember, though, that the values start with 0, so 255 is really the 256th number possible. This is a very important concept, because everything else is predicated on it.

The second rule astounds many people. In fact, it astounded me when I discovered it while writing my own subnet calculator program:

> Only nine values are possible for any octet in a subnet mask. They are: 0, 128, 192, 224, 240, 248, 252, 254, and 255.

Subnet masks are, by their nature, inclusive. That is to say, you can only add or subtract bits from the mask in bit order, and they must be added from left to right—you cannot skip a bit in a subnet mask. Figure 36-10 shows bits validly and invalidly used, with their resulting decimal octets.

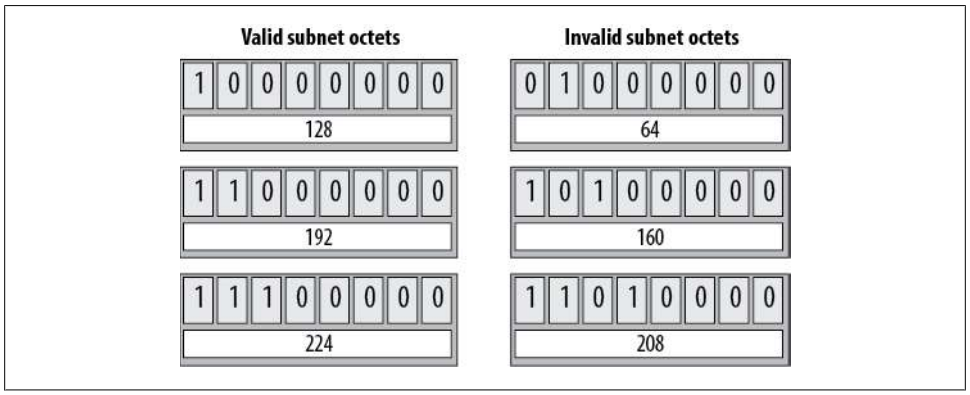

Figure 36-10. Valid and invalid subnet octets

The CCIE exam requires knowledge of manipulating subnet masks in unnatural ways, but you should never see this in the real world. While it is possible to use subnet masks in nonobvious ways, you should never do it outside a lab environment.

Because there are only eight bits in an octet, if you limit yourself to only allowing bits to be added or subtracted, there can only be a finite number of values represented in binary. Those values are shown in Figure 36-11.

Notice the column titled Ratio. This is something you won't see in the manufacturers' texts, which is a shame, because I believe it is the very essence of subnet masks in the real world.

In Figure 36-12, I've laid out the ratios as they apply to any octet in any position in the subnet mask. You don't need to memorize any of this, as it will all become clear shortly. Notice the patterns at work here. The ratio you will learn to use for a Class C network works for any class network. Simply by moving the ratio to the proper position in the subnet mask, you can figure out what the octet should be.

Mask	Binary	Ratio
0	0000 0000	1:256
128	1000 0000	2:128
192	1100 0000	4:64
224	1110 0000	8:32
240	1111 0000	16:16
248	1111 1000	32:8
252	1111 1100	64:4
254	1111 1110	128:2
255	1111 1111	256:1

Figure 36-11. Possible subnet octet values

Mask	First octet		Second octet		Third octet		Fourth octet	
0	1	256 * 256 * 256 * 256	256 * 1	256 * 256 * 256	256 * 256 * 1	256 * 256	256 * 256 * 256 * 1	256
128	2	128 * 256 * 256 * 256	256 * 2	128 * 256 * 256	256 * 256 * 2	128 * 256	256 * 256 * 256 * 2	128
192	4	64 * 256 * 256 * 256	256 * 4	64 * 256 * 256	256 * 256 * 4	64 * 256	256 * 256 * 256 * 4	64
224	8	32 * 256 * 256 * 256	256 * 8	32 * 256 * 256	256 * 256 * 8	32 * 256	256 * 256 * 256 * 8	32
240	16	16 * 256 * 256 * 256	256 * 16	16 * 256 * 256	256 * 256 * 16	16 * 256	256 * 256 * 256 * 16	16
248	32	8 * 256 * 256 * 256	256 * 32	8 * 256 * 256	256 * 256 * 32	8 * 256	256 * 256 * 256 * 32	8
252	64	4 * 256 * 256 * 256	256 * 64	4 * 256 * 256	256 * 256 * 64	4 * 256	256 * 256 * 256 * 64	4
254	128	2 * 256 * 256 * 256	256 * 128	2 * 256 * 256	256 * 256 * 128	2 * 256	256 * 256 * 256 * 128	2
255	256	1 * 256 * 256 * 256	256 * 256	1 * 256 * 256	256 * 256 * 256	1 * 256	256 * 256 * 256 * 256	1

Figure 36-12. Subnet octet ratios

Using the Class C network 192.168.1.0 255.255.255.0 as an example, if you apply a subnet mask of 255.255.255.224, the result is 8 subnets with 32 hosts in each. Look at Figure 36-11, and you'll see that the ratio for the subnet octet 224 is 8:32.

The ratios happen to correlate with the number of subnets and hosts per subnet in a native class C network. If you look at the other columns, you'll notice that the ratio is the same, but it is in a different position in the equation.

Let's look at a single example. Figure 36-13 shows the ratios for the subnet octet of 224.

Mask	First octet		Second octet		Third octet		Fourth octet	
224	8	32 * 256 * 256 * 256	256 * 8	32 * 256 * 256	256 * 256 * 8	32 * 256	256 * 256 * 256 * 8	32

Figure 36-13. Ratios for 224

In practice, here's what the Figure 36-13 is saying, using the network 10.0.0.0:

- 10.0.0.0 224.0.0.0 = 8 subnets of 536,870,912 (32 * 256 * 256 * 256) hosts
- 10.0.0.0 255.224.0.0 = 2,048 (256 * 8) subnets of 2,097,156 (32 * 256 * 256) hosts
- 10.0.0.0 255.255.224.0 = 524,288 (256 * 256 * 8) subnets of 8,192 (32 * 256) hosts
- 10.0.0.0 255.255.255.224 = 134,217,728 (256 * 256 * 256 * 8) subnets of 32 hosts

Notice that in each case, the ratio of 8:32 appears somewhere in the equation.

That's all well and good, but it still looks like a lot of icky math to me, so let's make it even simpler. This leads us to our third rule:

> When you know these rules, all you need to be able to do is double or halve numbers.

Subnet masks, like all things computer-related, are based on binary. Because binary is based on powers of two, all you really need to be able to do is to double a number or cut it in half.

Memorize the nine possible values for subnet masks (two are easy—0 and 255), and write them down on a sheet of paper (as shown in Figure 36-14). Once you have those down, you've successfully done all the memorization you need.

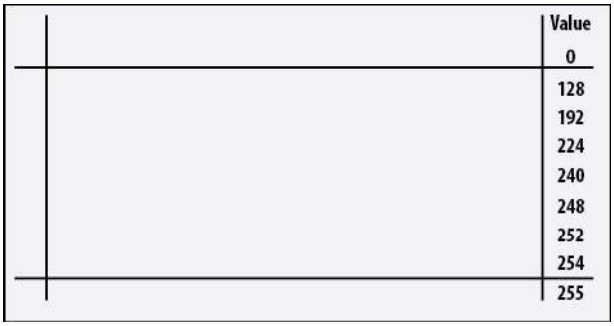

Figure 36-14. Subnet worksheet step #1

You'll fill in the rest of the sheet by using the simplest math possible: halving and doubling. Write the numbers 1:256 next to the 0 at the top. This is easy to remember. How many natural networks are there in a Class C? 1. How many hosts are there in a Class C? 256. Remember that everything regarding subnets comes back to the number 256.

Now, take the number to the left of the colon, and double it (1 * 2 = 2), and take the number to the right of the colon, and cut it in half (256 / 2 = 128). Repeat this process until you get to the bottom of the sheet. You've just figured out the network:host ratios for a native Class C network! Your paper should look like the example in Figure 36-15.

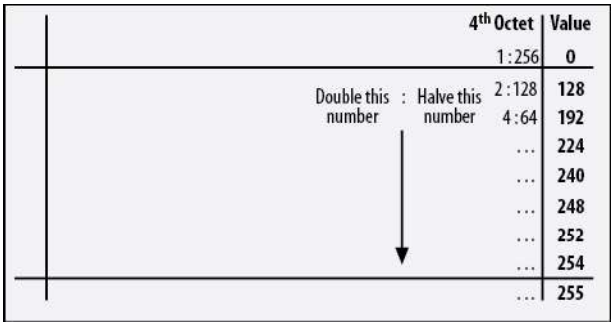

Figure 36-15. Subnet worksheet step #2

This trick is so simple that when I take mentally exhausting exams like the CCIE lab, I actually write out a sheet like this so I don't have to think at all during the lab. Now that you have all the ratios, you're set for working on a Class C network—but what about other networks? Well, you have two choices: always carry this book with you (which certainly would appeal to me) or apply some simple logic to what you've already learned.

As an example, let's use the private network 172.16.0.0 255.255.0.0, which is /16. People often get confused when working with ranges other than /24 because these ranges aren't usually seen in small companies. In practice, when you're using this method, there is no difference. We'll divide the network into eight pieces. Using your subnet worksheet, which octet value has an 8 on the left of the colon? Replace the leftmost zero in the native subnet mask with this number, and you have your answer: the subnet mask you need is 255.255.224.0. It's that simple.

Now let's try a more complex problem, like one you might see on an exam. With a network of 172.16.0.0 255.255.0.0, what subnet mask would you need to allow a network with 1,200 hosts in it? Here's the easy way to figure this one out. A /24 network has 256 hosts in it. Double this number until you get a number larger than 1,200:

256 × 2 = 512
512 × 2 = 1,024
1,024 × 2 = 2,048

Let's fill in the sheet some more. Draw a line under the top entry and another above the bottom entry, as shown in Figure 36-16. The entries between the lines are subnet octet values that fall between classful boundaries. This third column of numbers in Figure 36-16 indicates a further binary progression.

These numbers relate to the number of hosts in a subnet. You can keep the progression going as long as you need. When you get to the top of the sheet, start a new column to the left. These values will be the number of hosts for your second and first octets.

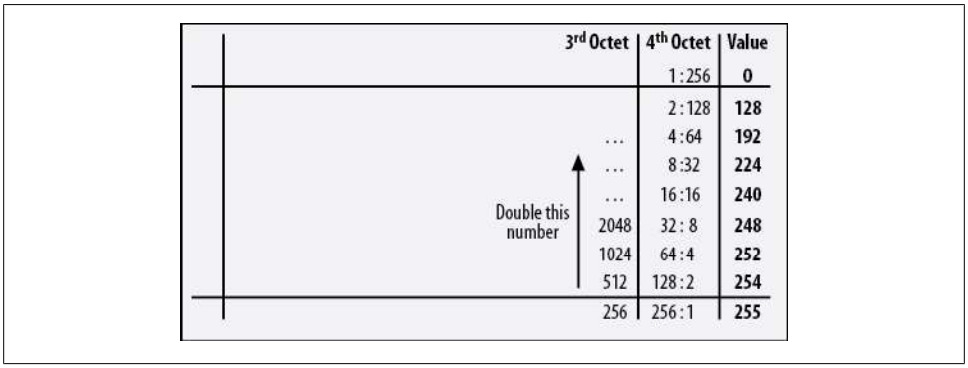

Figure 36-16. Subnet worksheet step #3

Looking at this worksheet reveals that the required subnet mask is 255.255.248.0. The network 172.16.0.0 with a subnet mask of 255.255.248.0 will result in 32 subnets with 2,048 hosts in each. To validate the math, perform this equation: 8 * 256 = 2,048.

Beware of a trick question you may see on tests. When using this method, you must remember that the first and last subnet may not be used given classical subnetting rules, and the first and last host may not be used. Therefore, if you encounter the question, "What would the subnet mask be for a network that required 1,024 hosts to be on the network?" the answer would not be 252, as it might appear, because two of those hosts are unusable (network and broadcast). If you require 1,024 hosts to be live devices on the network, you must have a network capable of supporting 2,048 hosts. This is one of the limitations of using a binary system.

Similarly, if you need 8 usable hosts on the network, you must provide 16 host addresses, not 8.

Figure 36-17 shows the same information as that presented in the preceding worksheet, but in a different format. Some people respond better to seeing this information in a horizontal rather than a vertical format. I like this model, because it is laid out the same way the bits are laid out in a subnet mask. Use whichever format works for you. There are no tests in this book.

Remember, a proper subnet mask can have only one of nine values in any single octet. If you start with a value of 1 on the bottom left and a value of 256 on the top right, then double those numbers under each subsequent value, you can quickly figure out what your subnet mask needs to be based on how many hosts or subnets it needs to have.

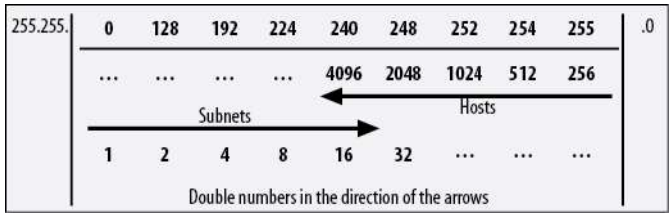

Figure 36-17. Horizontal format of the subnet worksheet

Once you have memorized the nine subnet octet values (remember, two are easy!) and can double and halve the rest in your head, you'll find that the numbers become obvious. You'll see all of these numbers over and over if you're around computers for any length of time. They're all powers of two, and when you start to recognize the patterns I've outlined for you, calculating any subnet mask will be only a matter of doing some quick, very simple math in your head.