Seven Steps to Subnetting

Excerpt from MCSE Guide to Microsoft® Windows 2000® Networking Certification Edition, written by Kelly Caudle, Walter J. Glenn, and James Michael Stewart; published in 2001 by Course Technology. This article has been updated by Beau Sanders to show that the first and last subnets in an incremental range can be used which is a change implemented in subnetting after this original text of this excerpt was published.

Revision Date: November 4, 2018

Creating Class C Subnetting Scheme

Basic subnetting is very easy when performed in seven steps. This example uses the Class C address 211.212.10.0. Using the seven steps provided here, you can create a subnetting scheme that allows you to use this address on your network.

Step 1: Determining Number of Subnets Needed

Determining the number of subnets you need is the very first step in subnetting. The number really depends upon your particular network. In Figure 2-3, the network consists of three routers connected via serial links. Each router also has a single Ethernet network attached.

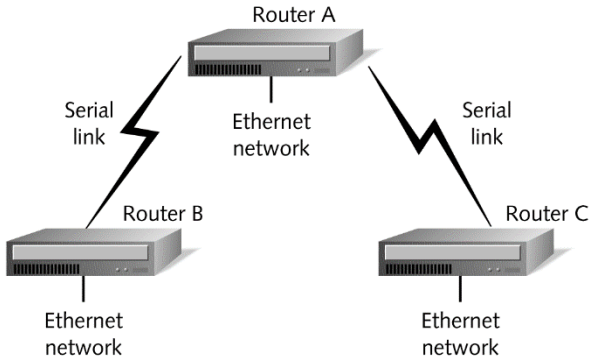

Figure 2-3 Example of network with three routers

Each shared serial link requires one subnet. Therefore, you need two subnets for the serial links between Router A and Routers B and C. You must also have one subnet per Ethernet interface on each router. Since you have three Ethernet networks, you need three subnets. Using this very simple counting method, you find that you need a total of five subnets. Unfortunately, you have been assigned a Class C address. The network address 211.212.10.0 allows for a single network of 254 hosts. You must borrow host ID bits to make this address work for you.

Step 2: Determining Number of Bits You Can Borrow

In Step 2, you must determine the number of bits that you can borrow. This number changes depending on the type of network address you start with. For Class A addresses, you have 24 host ID bits, but you can only borrow up to 22. For Class B addresses, you have 16 host ID bits, but you must have a minimum of two host bits;

therefore, you can borrow 14 bits. Your Class C address (211.212.10.0) has eight total host ID bits, but you can only borrow a maximum of six. The easiest way to determine the number of bits you can borrow is to write the number of octets that contain host ID bits in binary. In the Class C example network 211.212.10.0, you have the following bits to "play" with:

00000000

Step 3: Determining Number of Bits You Must Borrow to Get Needed Number of Subnets

After you determine the number of subnets you need and the number of bits you can borrow, you must calculate the number of host ID bits you must borrow to get the needed number of subnets. The formula for determining the number if bits you must borrow is 2^n= # of subnets. The *n* represents the number of bits you borrow. In other words, raise two to the power of the number of bits you borrow. The result is the number of useable subnets created when you borrow that number of bits. For the example network, you need five subnets. If you borrow three bits, the formula's result is six usable subnets: $2^3 = 8$.

Step 4: Turning On Borrowed Bits and Determining Decimal Value

In Step 4, using the bits you determined were available in Step 2, you turn on (set to 1) the number of bits determined you must borrow in Step 3. You must always begin with the high-order bits (the bits starting on the left of a binary number). Using the number of bits you can work with and the number of bits you must borrow (from Step 3), your result is the following: 11100000. In other words, from the eight total bits from Step 2 (six of which you could borrow), you borrow three host ID bits. In Step 4, you also need to determine the decimal value of the octets from which you borrow host ID bits. In this example, 11100000 equals 224. (128 + 64 + 32 = 224)

Step 5: Determining New Subnet Mask

Step 5 calculates the new subnet mask after you borrow the host ID bits in Step 4. You must add the decimal value from Step 4 to the default subnet mask for the class of address you are subnetting. The example is a Class C address, so the default mask is 255.255.255.0. The new mask after borrowing three bits becomes 255.255.255.224.

Step 6: Finding Host/Subnet Variable

In Step 6, you must find the lowest of the high-order bits (bits starting from the left) turned "on." Step 6 takes you back to the values found in each bit position within the octet. Our example defines the octets when we borrow 11100000 in the fourth octet. The highest order bit turned on represents 2^5, which equals 32. Since 2^5 is the last high-order bit turned on, the Host/Subnet variable you use in Step 7 is 32.

Step 7: Determining Range of Addresses

The final step allows you to take the Host/Subnet variable from Step 6 (32) and create your subnet ranges. Using the Class C network above, the range of subnets when you borrow three bits are:

211.212.10.0	to	211.212.10.31
211.212.10.32	to	211.212.10.63
211.212.10.64	to	211.212.10.95
211.212.10.96	to	211.212.10.127
211.212.10.128	to	211.212.10.159
211.212.10.160	to	211.212.10.191
211.212.10.192	to	211.212.10.223
211.212.10.224	to	211.212.10.255

Tailoring a Class B Address

This example takes a Class B address and tries to fit it within the needs of a network containing 1000 subnets. You are assigned the Class B address 131.107.0.0. Using the following seven steps, you are going to subnet the Class B address to meet your needs.

Step 1: Determining Number of Subnets Needed

Examine your network and determine your needs based on current network configuration and future growth (in this case, 1000 subnets).

Step 2: Determining Number of Bits You Can Borrow

With this Class B network address, you have 16 total bits to work with. You can only borrow up to 14 of these. On your sheet of paper, you should write the number of bits you have in the host ID portion of the address:

00000000.00000000

Step 3: Determining Number of Bits You Must Borrow to Get Number of Subsets Needed

Using the formula 2^n= # of usable subnets, you can easily see that you need to borrow 10 bits. When you plug in 10 borrowed bits, you get the following result:

2^{10} = 1024 useable subnets

Step 4: Turning on Borrowed Bits and Determining Decimal Value

If you turn on 10 bits, you get the following:

11111111.11000000

The decimal values for the octets are 255.192.

Step 5: Determining New Subnet Mask

Your example is a Class B address. In Class B addresses, the default subnet mask is 255.255.0.0. To get your new mask, you add the default mask to the decimal values found in Step 4. The new mask becomes:

255.255.255.192

Step 6: Finding Host/Subnet Variable

In the next-to-last step, you must find the value of the lowest high-order bit turned on in each octet, from which you borrowed host bits. Since this example is a Class B network and you must borrow a great number of bits to get the proper number of subnets, the borrowing crosses an octet boundary. As a result, you have two Host/Subnet variables. In this example, the variable in the third octet is 1, and the variable for the fourth octet is 64. You get these values by looking at the binary numbers in Step 4. The third octet has the final bit position, or the 2^0 bit position, turned on. Since $2^0 = 1$, your variable is 1 in the third octet. In the fourth octet, the second high-order bit or 2^6 is turned on. The variable in this octet is 64.

Step 7: Determining Range of Addresses

Figuring the range of addresses for Class B networks is much harder than for Class C. This is especially true in cases like this scenario in which you must borrow a large number of bits. Using 1 as the variable in the third octet and 64 as the variable in the fourth octet, the range of the first 9 subnets world be:

131.107.0.0	to	131.107.0.63
131.107.0.64	to	131.107.0.127
131.107.0.128	to	131.107.0.191
131.107.0.192	to	131.107.0.255
131.107.1.0	to	131.107.1.63
131.107.1.64.	to	131.107.1.127
131.107.1.128	to	131.107.1.191
131.107.1.192	to	131.107.1.255
131.107.2.0	to	131.107.2.63