By George Ou

Introduction

IP subnetting is a fundamental subject that's critical for any IP network engineer to understand, yet students have traditionally had a difficult time understanding it. Over the years, I've watched students needlessly struggle through school and in practice when dealing with subnetting because it was never explained to them in an easy-to-understand way. I've helped countless individuals learn what subnetting is all about using my own graphical approach and calculator shortcuts, and I've put all that experience into this article.

IP addresses and subnets

Although IP stands for Internet Protocol, it's a communications protocol used from the smallest private network to the massive global Internet. An IP address is a unique identifier given to a single device on an IP network. The IP address consists of a 32-bit number that ranges from 0 to 4294967295. This means that theoretically, the Internet can contain approximately 4.3 billion unique objects. But to make such a large address block easier to handle, it was chopped up into four 8-bit numbers, or "octets," separated by a period. Instead of 32 binary base-2 digits, which would be too long to read, it's converted to four base-256 digits. Octets are made up of numbers ranging from 0 to 255. The numbers below show how IP addresses increment.

0.0.0.0
0.0.0.1
…increment 252 hosts...
0.0.0.254
0.0.0.255
0.0.1.0
0.0.1.1
...increment 252 hosts...
0.0.1.254
0.0.1.255
0.0.2.0
0.0.2.1
...increment 4+ billion hosts…
255.255.255.255

The word *subnet* is short for *sub network*--a smaller network within a larger one. The smallest subnet that has no more subdivisions within it is considered a single "broadcast domain," which directly correlates to a single LAN (local area network) segment on an Ethernet switch. The broadcast domain serves an important function because this is where devices on a network communicate directly with each other's MAC addresses, which don't route across multiple subnets, let alone the entire Internet. MAC address communications are limited to a smaller network because they rely on ARP broadcasting to find their way around, and broadcasting can be scaled only so much before the amount of broadcast traffic brings down the entire network with sheer broadcast noise. For this reason, the most common smallest subnet is 8 bits, or precisely a single octet, although it can be smaller or slightly larger.

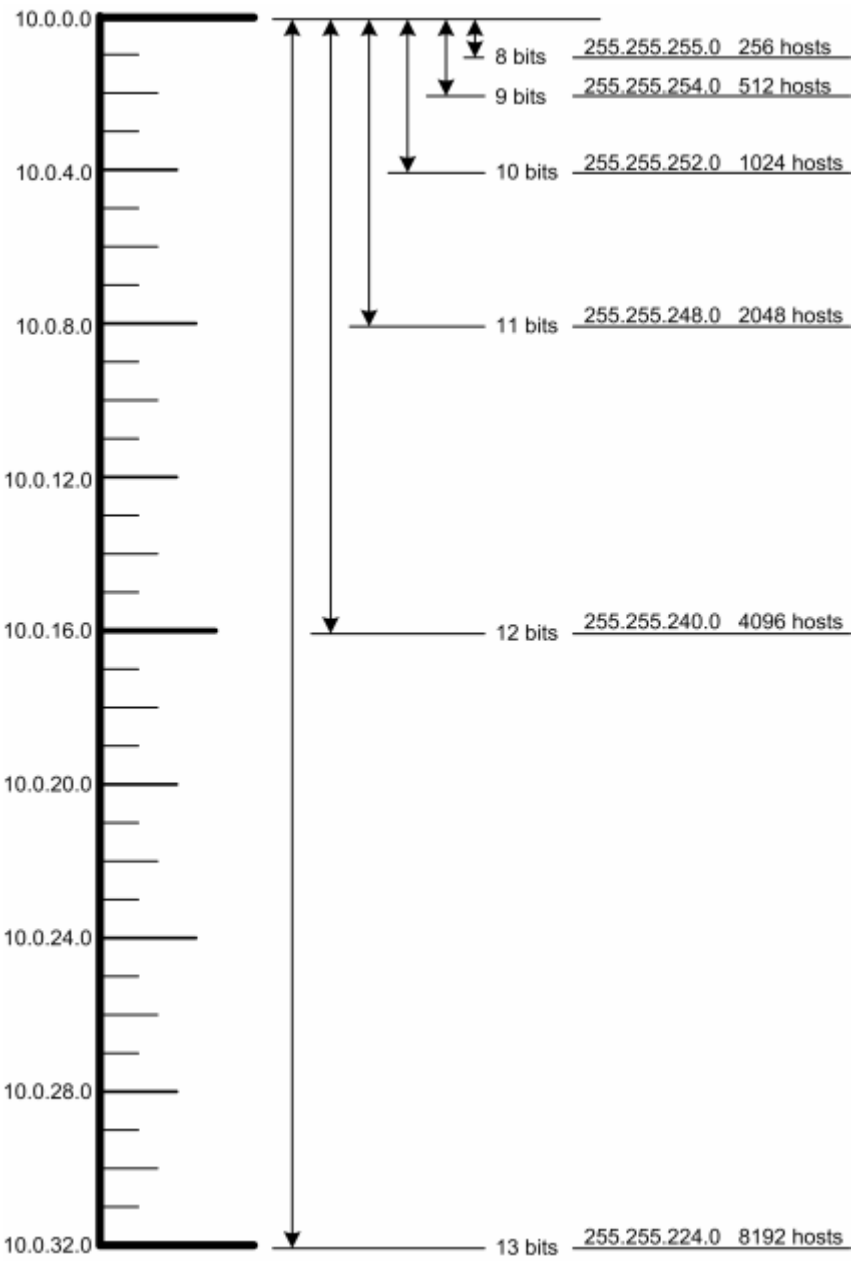
Subnets have a beginning and an ending, and the beginning number is always even and the ending number is always odd. The beginning number is the "Network ID" and the ending number is the "Broadcast ID." You're not allowed to use these numbers because they both have special meaning with special purposes. The Network ID is the official designation for a particular subnet, and the ending number is the broadcast address that every device on a subnet listens to. Anytime you want to refer to a subnet, you point to its Network ID and its subnet mask, which defines its size. Anytime you want to send data to everyone on the subnet (such as a multicast), you send it

to the Broadcast ID. Later in this article, I'll show you an easy mathematical and graphical way to determine the Network and Broadcast IDs.

The graphical subnet ruler

Over the years, as I watched people struggle with the subject of IP subnetting, I wanted a better way to teach the subject. I soon realized that many students in IT lacked the necessary background in mathematics and had a hard time with the concept of binary numbers. To help close this gap, I came up with the graphical method of illustrating subnets shown in **Figure A**. In this example, we're looking at a range of IP addresses from 10.0.0.0 up to 10.0.32.0. Note that the ending IP of 10.0.32.0 itself is actually the beginning of the next subnet. This network range ends at the number right before it, which is 10.0.31.255

Figure A

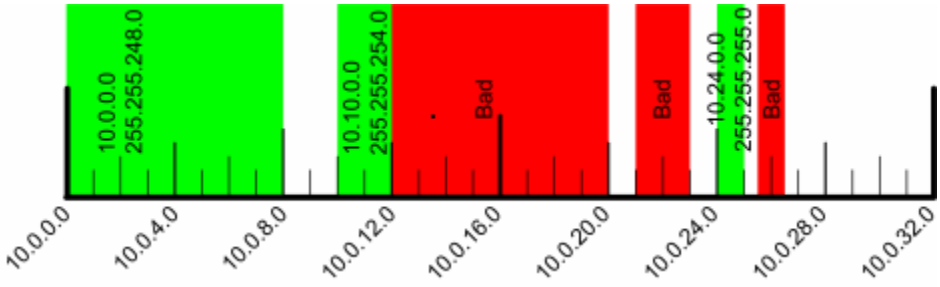


Note that for every bit increase, the size of the subnet doubles in length, along with the number of hosts. The smallest tick mark represents 8 bits, which contains a subnet with 256 hosts--but since you can't use the first and last IP addresses, there are actually only 254 usable hosts on the network. The easiest way to compute how many usable hosts are in a subnet is to raise 2 to the power of the bit size minus 2. Go up to 9 bits ,and we're up to 510 usable hosts, because 2 to the 9th is 512, and we don't count the beginning and ending. Keep on going all the way up to 13 bits, and we're up to 8,190 usable hosts for the entire ruler shown above.

Learning to properly chop subnets

Subnets can be subdivided into smaller subnets and even smaller ones still. The most important thing to know about chopping up a network is that you can't arbitrarily pick the beginning and ending. The chopping must be along clean binary divisions. The best way to learn this is to look at my subnet ruler and see what's a valid subnet. In **Figure B**, green subnets are valid and red subnets are not.

Figure B



The ruler was constructed like any other ruler, where we mark it down the middle and bisect it. Then, we bisect the remaining sections, with shrinking markers every time we start a new round of bisecting. In the sample above, there were five rounds of bisections. If you look carefully at the edge of any valid (green) subnet blocks, you'll notice that none of the markers contained within the subnet is higher than the edge's markers. There is a mathematical reason for this, which we'll illustrate later, but seeing it graphically will make the math easier to understand.

The role of the subnet mask

The subnet mask plays a crucial role in defining the size of a subnet. Take a look at **Figure C**. Notice the pattern and pay special attention to the numbers in red. Whenever you're dealing with subnets, it will come in handy to remember eight special numbers that reoccur when dealing with subnet masks. They are **255**, **254**, **252**, **248**, **240**, **224**, **192**, and **128**. You'll see these numbers over and over again in IP networking, and memorizing them will make your life much easier.

I've included three class sizes. You'll see the first two classes, with host bit length from 0 to 16, most often. It's common for DSL and T1 IP blocks to be in the 0- to 8-bit range. Private networks typically work in the 8- to 24-bit range.

Note how the binary mask has all those zeros growing from right to left.

Figure C

Subnet mask quick reference							
Host Bit length	math	Max hosts	Subnet mask	Mask octet	Binary mask	Mask length	Subnet length
0	2^0=	1	255.255.255.255	4	11111111	32	0
1	2^1=	2	255.255.255.254	4	11111110	31	1
2	2^2=	4	255.255.255.252	4	11111100	30	2
3	2^3=	8	255.255.255.248	4	11111000	29	3
4	2^4=	16	255.255.255.240	4	11110000	28	4
5	2^5=	32	255.255.255.224	4	11100000	27	5
6	2^6=	64	255.255.255.192	4	11000000	26	6
7	2^7=	128	255.255.255.128	4	10000000	25	7
8	2^8=	256	255.255.255.0	3	11111111	24	8
9	2^9=	512	255.255.254.0	3	11111110	23	9
10	2^10=	1024	255.255.252.0	3	11111100	22	10
11	2^11=	2048	255.255.248.0	3	11111000	21	11
12	2^12=	4096	255.255.240.0	3	11110000	20	12
13	2^13=	8192	255.255.224.0	3	11100000	19	13
14	2^14=	16384	255.255.192.0	3	11000000	18	14
15	2^15=	32768	255.255.128.0	3	10000000	17	15
16	2^16=	65536	255.255.0.0	2	11111111	16	16
17	2^17=	131072	255.254.0.0	2	11111110	15	17
18	2^18=	262144	255.252.0.0	2	11111100	14	18
19	2^19=	524288	255.248.0.0	2	11111000	13	19
20	2^20=	1048576	255.240.0.0	2	11110000	12	20
21	2^21=	2097152	255.224.0.0	2	11100000	11	21
22	2^22=	4194304	255.192.0.0	2	11000000	10	22
23	2^23=	8388608	255.128.0.0	2	10000000	9	23
24	2^24=	16777216	255.0.0.0	1	11111111	8	24

The subnet mask in binary form always has all ones to the left and all zeros to the right. The number of zeros is identical to the *subnet length*. I showed only the portion of the binary subnet in the octet that's interesting, since all octets to the right consist of zeros and all octets to the left consist of ones. So if we look at the subnet mask where the subnet length is 11 bits long, the full binary subnet mask is 11111111.11111111.11111000.00000000. As you can see under *mask octet*, the subnet mask transitions from 1 to 0 in the third octet. The particular binary subnet mask translates directly to base-256 form as 255.255.248.0.

The "mask" in subnet mask

The subnet mask not only determines the size of a subnet, but it can also help you pinpoint where the end points on the subnet are if you're given any IP address within that subnet. The reason it's called a subnet "mask" is that it literally masks out the host bits and leaves only the Network ID that begins the subnet. Once you know the beginning of the subnet and how big it is, you can determine the end of the subnet, which is the Broadcast ID.

To calculate the Network ID, you simply take any IP address within that subnet and run the AND operator on the subnet mask. Let's take an IP address of 10.20.237.15 and a subnet mask of 255.255.248.0. Note that this can be and often is written in shorthand as **10.20.237.15/21** because the subnet mask length is 21. **Figure D** and **Figure E** show the Decimal and Binary versions of the AND operation.

Figure D

	10	20	237	15
AND	255	255	248	0
	10	20	232	0

Decimal math

Figure E

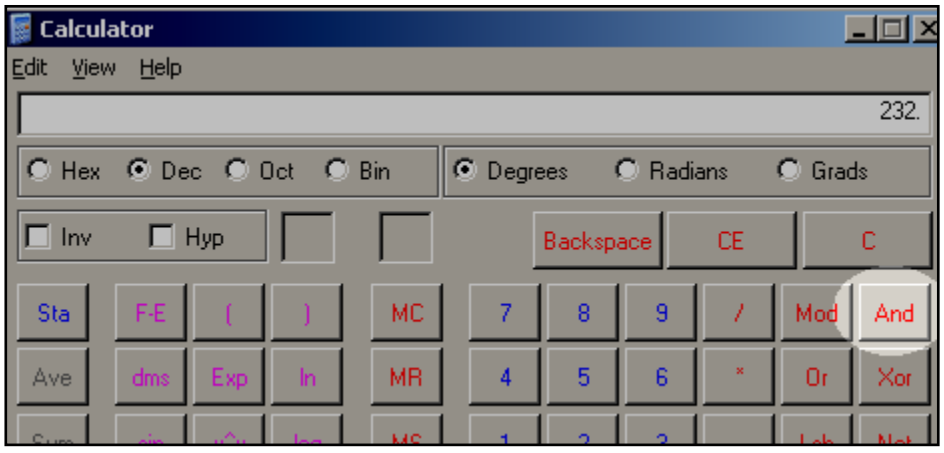
				Mask
	00001010	00010100	11101101	00001111
AND	11111111	11111111	11111000	00000000
	00001010	00010100	11101000	00000000

Binary math

The binary version shows how the **0**s act as a mask on the IP address on top. Inside the masking box, the **0**s convert all numbers on top into zeros, no matter what the number is. When you take the resultant binary Network ID and convert it to decimal, you get 10.20.232.0 as the Network ID.

One thing that's always bothered me about the way subnetting is taught is that students are not shown a simple trick to bypass the need for binary conversions when doing AND operations. I even see IT people in the field using this slow and cumbersome technique to convert everything to binary, run the AND operation, and then convert back to decimal using the Windows Calculator. But there's a really simple shortcut using the Windows Calculator, since the AND operator works directly on decimal numbers. Simply punch in 237, hit the AND operator, and then 248 and [Enter] to instantly get 232, as shown in **Figure F**. I'll never understand why this isn't explained to students, because it makes mask calculations a lot easier.

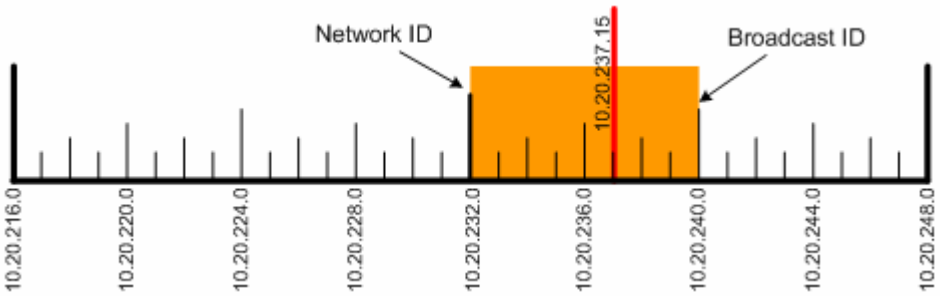
Figure F



Since there are 11 zeros in the subnet mask, the subnet is 11 bits long. This means there are 2^11, or 2,048, maximum hosts in the subnet and the last IP in this subnet is 10.20.239.255. You could compute this quickly by seeing there are three zeros in the third octet, which means the third octet of the IP address can

have a variance of 2^3, or 8. So the next subnet starts at 10.20.232+8.0, which is 10.20.240.0. If we decrease that by 1, we have 10.20.239.255, which is where this subnet ends. To help you visualize this, **Figure G** shows it on my subnet ruler.

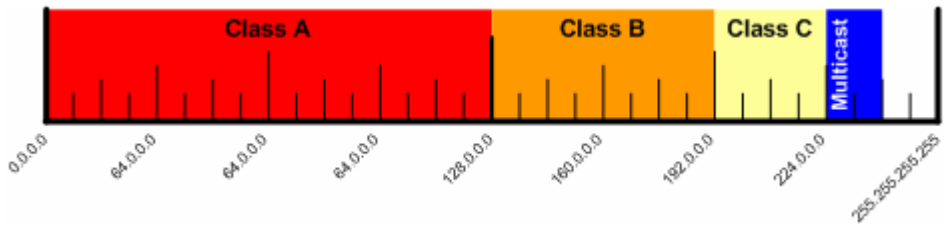
Figure G



IP classes made simple

For an arbitrary classification of IP subnets, the creators of the Internet chose to break the Internet into multiple classes. Note that these aren't important as far as your subnet calculations are concerned; this is just how the Internet is "laid out." The Internet is laid out as Class A, B, C, D, and E. Class A uses up the first half of the entire Internet, Class B uses half of the remaining half, Class C uses the remaining half again, Class D (Multicasting) uses up the remaining half again, and whatever is left over is reserved for Class E. I've had students tell me that they struggled with the memorization of IP classes for weeks until they saw this simple table shown in **Figure H**. This is because you don't actually need to memorize anything, you just learn the technique for constructing the ruler using half of what's available.

Figure H



Remember that all subnets start with EVEN numbers and all subnet endings are ODD. Note that 0.0.0.0/8 (0.0.0.0 to 0.255.255.255) isn't used and 127.0.0.0/8 (127.0.0.0 to 127.255.255.255) is reserved for loopback addresses.

All Class A addresses have their first octet between 1 to 126 because 0 and 127 are reserved. Class A subnets are all 24 bits long, which means the subnet mask is only 8 bits long. For example, we have the entire 3.0.0.0/8 subnet owned by GE, since GE was lucky enough to get in early to be assigned 16.8 million addresses. The U.S. Army owns 6.0.0.0/8. Level 3 Communications owns 8.0.0.0/8. IBM owns 9.0.0.0/8. AT&T owns 12.0.0.0/8. Xerox owns 13.0.0.0/8. HP owns 15.0.0.0/8 and 16.0.0.0/8. Apple owns 17.0.0.0/8.

All Class B addresses have their first octet between 128 and 191. Class B subnets are all 16 bits long, which means the subnet masks are 16 bits long. For example, BBN Communications owns 128.1.0.0/16, which is 128.1.0.0 to 128.1.255.255. Carnegie Mellon University owns 128.2.0.0/16.

All Class C addresses have their first octet between 192 and 223. Class C subnets are all 8 bits long, so the subnet mask is only 24 bits long. Note that ARIN (the organization that assigns Internet addresses) will sell blocks of four Class C addresses only to individual companies and you have to really justify why you need 1,024 Public IP addresses. If you need to run BGP so you can use multiple ISPs for redundancy, you have to have your own block of IP addresses. Also note that this isn't the old days, where blocks of 16.8 million Class A addresses were handed out for basically nothing. You have to pay an annual fee for your block of 1,024 addresses with a subnet mask of /22, or 255.255.252.0.

The concept of subnet classes can cause harm in actual practice. I've actually seen people forget to turn classes off in their old Cisco router and watch large subnet routes get hijacked on a large WAN configured for dynamic

routing whenever some routes were added. This is because a Cisco router will assume the subnet mask is the full /8 or /16 or /24 even if you define something in between. All newer Cisco IOS software versions turn off the concept of subnet classes and uses classless routing by default. This is done with the default command "IP Classless."

Public versus private IP addresses

Besides the reserved IP addresses (0.0.0.0/8 and 127.0.0.0/8) mentioned above, there are other addresses not used on the public Internet. These *private subnets* consist of private IP addresses and are usually behind a firewall or router that performs NAT (network address translation). NAT is needed because private IP addresses are nonroutable on the public Internet, so they must be translated into public IP addresses before they touch the Internet. Private IPs are never routed because no one really owns them. And since anyone can use them, there's no right place to point a private IP address to on the public Internet. Private IP addresses are used in most LAN and WAN environments, unless you're lucky enough to own a Class A or at least a Class B block of addresses, in which case you might have enough IPs to assign internal and external IP addresses.

The following blocks of IP addresses are allocated for private networks:

- 10.0.0.0/8 (10.0.0.0 to 10.255.255.255)
- 172.16.0.0/12 (172.16.0.0 to 172.31.255.255)
- 192.168.0.0/16 (192.168.0.0 to 192.168.255.255)
- 169.254.0.0/16 (169.254.0.0 to 169.254.255.255)*

*Note that 169.254.0.0/16 is a block of private IP addresses used for random self IP assignment where DHCP servers are not available.

10.0.0.0/8 is normally used for larger networks, since there are approximately 16.8 million IP addresses available within that block. They chop it up into lots of smaller groups of subnets for each geographic location, which are then subdivided into even smaller subnets. Smaller companies typically use the 172.16.0.0/12 range, chopped up into smaller subnets, although there's no reason they can't use 10.0.0.0/8 if they want to. Home networks typically use a /24 subnet within the 192.168.0.0/16 subnet.

The use of private IP addresses and NAT has prolonged the life of IPv4 for the foreseeable future because it effectively allows a single public IP address to represent thousands of private IP addresses. At the current rate that IPv4 addresses are handed out, we have enough IPv4 addresses for approximately 17 years. ARIN is much more stingy now about handing them out, and small blocks of IP addresses are relatively expensive compared to the old days, when companies like Apple were simply handed a block of 16.8 million addresses. The next version of IP addresses, called IPv6, is 128 bits long--and there are more than 79 thousand trillion trillion times more IP addresses than IPv4. Even if you assigned 4.3 billion people on the planet with 4.3 billion IP addresses each, you would still have more than 18 million trillion IPv6 addresses left!

Additional resources

- TechRepublic's Downloads RSS Feed XML
- Sign up for TechRepublic's Downloads Weekly Update newsletter
- Sign up for our Network Administration NetNote
- Check out all of TechRepublic's free newsletters
- "Cisco IP subnetting 101: An introduction to supernetting" (TechRepublic article)
- "Cisco IP subnetting 101: Learn more about all 1s and all 0s subnet masks" (TechRepublic article)
- "Cisco IP subnetting 101: Five more things you should know" (TechRepublic article)

Version history

Version: 1.0
Published: June 28, 2006

Tell us what you think

TechRepublic downloads are designed to help you get your job done as painlessly and effectively as possible. Because we're continually looking for ways to improve the usefulness of these tools, we need your feedback. Please take a minute to drop us a line and tell us how well this download worked for you and offer your suggestions for improvement.

Thanks!

—The TechRepublic Downloads Team