Video - Subnetting with the Magic Number (15 min)

This video focuses on subnetting and how to find your subnetworks easily and quickly using the magic number trick. Let's say we're subnetting a 192.168.1.0 network, /24. I have the IP address and subnet mask in decimal here and I have them listed in binary here. The network portion of the subnet mask and the host portion of the subnet mask. To subnet this network, what I can do is, is I can borrow one bit from the host portion of the subnet mask and now I have one subnet bit. This changes the subnet mask from /24 to /25. If I borrow one bit from the host portion, I have one bit or two to the first power makes two possible subnetworks. The subnetworks I've created are the 192.168.1.0 subnet and the 192.168.1.128 subnet both /25. The host on each network are 126 usable hosts. We have seven zeros here and two to the seventh power is 128 minus two makes 126 usable host per subnet. If I decide to borrow two subnetwork bits from the host portion of the original 192.168.1.0 /24 network, I now have /26. I've borrowed two bits so now I have two to the second power of subnet bits, so I have effectively four subnetworks. The four subnetworks are listed here. For host bits, I have six zeros in the host portion of the address and two to the sixth power is 64 minus two for each subnet makes 62 usable host per subnet. The subnetworks that are created are the 192.168.1.0 subnet. The 64 subnet, the 128 subnet and the 192 subnet.

Notice how the number of borrowed bits and the number of host bits determines a lot about the subnet. If I borrow three subnetwork bits, I now have a /27 so it goes from /24 to /27. I have three subnet bits. Two to the third power is eight. Two times two times two is eight and you can see that I now have eight subnets or eight subnetworks created from the original 192.168.1.0 network. Each subnet is now /27. Notice that there are now 32 hosts or 30 usable host per subnetwork. We could go up to /28 and now I have four subnetwork bits. Notice that the subnet mask has changed from 255.255.255.0 when it was /24 to now 240. I now have 16 subnetworks or two to the fourth power and there's two to the fourth power of host per network. So there are 16 or 14 usable host addresses per subnet. Borrowing five bits looks like this and borrowing six bits looks like this. I've now borrowed six bits from the original eight bit host portion of the original /24 network and I now have two to the sixth power of possible subnetworks. Meaning I have 64 subnets. The host portion of the address is now only two bits or two to the second power, meaning there's only four host addresses per subnet.

Now if you have four host addresses, you have to leave room for the network address and the broadcast address. So four minus two leaves only two usable host per subnet. You cannot go higher than a /30 subnet mask otherwise you'll not have enough room for usable host addresses. In other words, we can't borrow anymore bits from the host portion or we won't have enough addresses for hosts. You might be asking yourself how I was able to discern the different subnetworks from the subnet mask or subnet bits that were borrowed. There are many different ways to derive the subnetworks from the subnet mask but my favorite is the magic number technique. The magic number is simply the place value of the last one in the subnet mask. In this situation we have a /25 subnet mask, we have 25 ones and the last one is the one here all the way on the right highlighted in red. This one if we look at it according to this octet is in the 128ths place. If we think about the place values of eight bits, this is the 128ths place. So the magic number in this situation is 128. The magic number tells us where to find the networks. It tells us that the networks will go up by 128. So in this case, the first network which is always zero, so 192.168.1.0 /25 and from there the networks go up by 128 so the next network is 192.168.1.128.

Let's see how this works borrowing two bits from the host portion. Now I have borrowed two bits from the host portion of the subnet mask. The magic number is now the last binary one which in this case is in the 64th place. So the magic number is 64. The magic number tells us how to find our networks and lets us know that the networks will be going up in increments of 64. For instance the first subnet is always zero, so 192.168.1.0 /26, the next subnet is 64. 64 plus 64 is 128 and 128 plus 64 is 192. You can see how the subnets are going up by 64. There's only four subnets because we've borrowed two subnet bits. Also we can go no higher than 192 because 64 plus 192 is 256 and that's too large a number.

Let's see how it works with three bits. With three bits, we now have a slash 25, 26, 27 subnet mask, /27 subnet mask. In decimal, 255.255.255.224. The last one in the subnet mask is our magic number. In this case it's in the 32s place in the binary conversion table. So the magic number is 32. If we look at the networks, we'll see that they go up by 32. The first network being zero, 1921681.0 /27 then the 32 subnet /27, 64, 96, 128, 160, 192 and 224. The subnets go up in increments of 32 following the magic number. Now that we see the pattern and how the last borrowed bit affects the location of the subnetworks, let's try it with a /28 subnet mask. Now I have the 192.168.1.0 /24 network. I wanna subnet it, changing it to /28. /28 means I borrow four bits in the subnet mask. So I now have 28 ones counted from left to right. 255.255.255.240 subnet mask, 128 plus 64 plus 32 plus 16 is 240. The last one is in the 16ths place. This is the 128ths place, the 64ths place, the 32s place, the 16ths place. So the magic number is 16. We'll see that the networks go up by 16. The first network is always zero so 192.168.1.0 /28. I'll copy this and paste it and the next network will be the 16th network. Why? Because the magic number is 16, because the last one is in the place value of 16. The next network will be 32 and the following network will be 48 and after 48, 64 and so on and so forth. The networks will go up by 16. In each network, we have the host bits, four bits for host which is also two to the fourth power of 16. So the network will go from zero up to 15 with zero being the subnetwork address or network address and 15 being the broadcast address and the next subnet is the 16 subnetwork. Now let's try creating /29 subnets. We can see that that involves borrowing five bits from the host portion of the subnet mask and that the last one is in the eighths place. The magic number is eight. The networks will go up in increments of eight. So starting from zero, we'll have the eight network and the 16 network and the 24 network and so on and so forth all subnets being /29. For the /30, we see that the last one is in the four's place and that the networks will go up by four. So similarly the networks will go up in increments of four.

So the first network is the zero network then the four subnetwork then the eight subnetwork. Each subnetwork will have four host addresses. In this case zero to three. Zero will be the network address. Three, the broadcast address and the addresses between, one and two, will be the usable host addresses. Only two usable host addresses in a /30 subnetted network. You might ask yourself will this technique, the magic number technique of looking at the place value of the last one in the subnet mask work in other subnetted octets? In other words if we're starting from a class B /16 network address or a /eight, class A network address, would this technique work, and it does. For instance if we take a 172.16.0.0 network /16, you can see in this scenario I've subnetted it to a /23 subnet mask. In other words I've taken the original 16 network bits and borrowed one, two, three, four, five, six, seven ones from the host portion of the subnet mask. In this situation, the last borrowed one is in the two's place. So the magic number in this situation is two. This means that the subnets will go up by two in this octet here.

So in other words our first subnet will be 172.16.0.0 /23 and the next network will be 172.16.2.0 /23. The only difference is now instead of 16 host bits like we would have had in a /16 network, we now have nine host bits. So two to the ninth power meaning we have 512 minus two would be 510 host addresses that we can assign. In other words the network goes from 0.0 all the way up to 172.16.1.255. So the first usable host address would be 0.1 and the last usable host address would be 1.254. The next subnet starts at 2.0. After that, the networks continue to go up by two. In other words the next subnet will be the four subnet and then the six subnet and the eight subnet. Notice that the networks are going up by two but in the third octet, not in the fourth octet. The only difference not to be confused on is the larger number of host addresses that you'll experience when you subnet in this octet or in the second octet. You can see how the magic number technique can help you quickly locate the networks or subnets.

