Video - Subnetting Across Multiple Octets (6 min)

In classless subnetting, we've seen how we can borrow bits from the host portion of the subnet mask and that the last one that's borrowed is the magic number. Meaning that the last one, if we look at the place value of the 8-bit grouping, the last one, in this case, is in the 32's place so the networks go up by 32 starting with the zero network, we have the 32 network then the 64, 96, 128 so on and so forth but the networks go up by 32. What about when we're subnetting from a class A or class B network address and subnetting generally in the other octets. Let's take a look at another example. This time we have a 10.0.0.0 network address and a classful class A subnet mask and we're going to subnet it so we've taken the eight bits and borrowed three bits, now we have a /11 subnet mask. The last one in the subnet mask is still in the 32's place. I'f we look at this 8-bit grouping in this octet right here, the last one is still in the 32's place. So the networks will go up by 32. The networks are the 10.0, 32, 64, 96 and they're going up by 32 but in the second octet. The only difference is now we have a lot more host addresses, we have 8, 16, 21 zeros for hosts which means there's over two million hosts per subnet.

For instance, if we take a look at this 10.192.0.0 subnet, the next subnet is 10.224 so that means that the host addresses go from 0.1 all the way up to 10.223.255.254 as the last usable host and 255 is the broadcast. The next address is 10.224.0.0 which is the next subnet. All of these subnets are now /11. What about a more difficult scenario? In this scenario, we need to create over 300 equal sized subnets of 20,000 hosts each starting from the 10.0.0/8 network. We need to create a lot more subnets and we need a lot of hosts. Looking at this situation, we know that we're going to need to borrow bits from the host portion of our subnet mask. The question is how many bits will we borrow? If we borrow one bit we have two subnets, now we have four subnets, eight subnets, 16, 32, 64, 128, 256, 512. If we borrow nine bits, we'll have 512 subnets that we create and that will be sufficient to meet the demand for 300 equal-sized subnets. What about the requirement of 20,000 hosts each? We have 15 zeros left for hosts per subnet. Two to the 15th power is 32,768 minus two would be 32,766 hosts per subnet. That meets the requirement of having at least 20,000 hosts in each subnet. So this scenario would work. You can see here that I've now highlighted the borrowed bits from the subnet mask, I've also converted it to decimal here, 255.255.128.0 and I've already ascertained that the last one in the subnet mask is in the 128's place so the magic number is 128. The subnets will still go up by 128 but they'll also need to go up for every number possible in the second octet. If we look below here we can see our subnets. The first subnet will be the 10.0.0.0/17 subnet then the 0.128 subnet then the 1.0 subnet then the 1.128 subnet. As you can see, we're iterating through every possible number in the second octet and then going up by 128 in the third octet. If I was to list out every possible subnet, we'd go all the way up to 10.255.0.0 and the next subnet would be 10.255.128.0/17 is our last subnet for a total of 512 subnets. In a situation like this, there's many more subnets and many more hosts per subnet but the rule of the magic number still applies.