12.1.5 Variable Length Subnet Masking (VLSM)

Click one of the buttons to take you to that part of the video.

Variable Length Subnet Masking (VLSM) 0:00-0:08

In this lesson, we're going to learn about Variable Length Subnet Masking (VLSM).

Variable Length Subnet Masking 0:09-7:16

Be aware that you don't actually have to use the default subnet mask.

For example, you could define a subnet mask of 255.255.0.0 for a class A address. By doing this, we reassign one of those octets that was assigned for host addresses to networks, which allows us to create more networks, but with fewer hosts.

In addition, you can also use only part of an octet for a network address if you want to. And this is called partial subnetting or variable length subnet masking, VLSM. Basically, using VLSM, we completely ignore all of the default subnet masks that we just talked about.

We just throw them out the window because we don't care. We completely ignore the default subnet mask boundaries, and then specify in the prefix exactly how many subnet mask bits we want to use. And because of this, VLSM is sometimes called 'classless subnetting' because we're basically ignoring the default class A, B, and C subnet boundaries.

Let's take a look at an example. Let's suppose that we define a subnet mask here of 255.255.252.0 for the hosts on our network. We are taking, essentially, the default class B subnet mask, which includes ones in the first two octets, and then we're adding a whole bunch of extra ones in the third octet.

We're gonna use the first six--1, 2, 3, 4, 5, 6--but not all of them. We're not using all eight, like we would with a class C address. Instead, we're just going to steal six bits from the node portion of the IP address from the third octet, and we're going to use those for the network portion of the address.

This allows us to add additional networks. It also reduces the number of hosts we can have on each network, but when we're dealing with a class B address that's really not an issue, because there are way more than we'll ever use. In this example, because we're using the first six bits of the third octet for the subnet mask, this much is used for the network and this much now is used for the node address.

Let's take a look at another example. Let's suppose we have a network, and it's composed of four separate physical network segments that are connected by routers. This is segment 1, this is segment 2, this is segment 3, this is segment 4. There are additional segments between routers; we're not worrying about those here. We're just concerned about these network segments over here that have hosts, and printers, and servers, and everything like that on them.

Well, this network uses a class A address. The network address is 10.0.0.0. What we want to do is actually divide this one huge network 10.0.0.0 into four separate, smaller subnetworks. We call them subnets.

By default, if we use classful addressing, the default subnet mask for class A says that this much is network and this much is node.

Well, we don't actually need this many addresses. We don't need 16 million IP addresses on each one of these separate subnets. We'll need, you know, 100 or so.

In order to create four additional subnetworks, or subnets, we can steal some bits from the host portion of the address and use them to create subnets. We don't need to create very many, so all we have to do in this situation is grab the first two bits of the host portion of the address from the second octet and use them to create additional networks.

Instead of using the default class A subnet mask of 255.0.0.0, we're instead going to use a subnet mask of 255.192.0.0. Why is that 192? Well, remember that the first bit in an 8-bit octet has a value of 128, and the second bit in an 8-bit octet has a value of 64. If you add those two together, you get 192.

And if we want to use CIDR notation to define the length of the prefix instead of the default subnet mask notation, we would use a /10 here to indicate that we're using 1-2-3-4-5-6-7-8-9-10 bits for the subnet mask, or the prefix.

By doing this, we can actually define four new subnets within the address that we originally started with; remember, the original address was 10.0.0.0. Because we're stealing these two bits from the host portion of the address, we can actually define four subnets, because they are actually four different values that we can define in the decimal version of the IP address, using those two bits in the subnet mask.

We could have 10.0; this would occur if the first two bits in the second octet of the IP address were 00. If they were 01, then we would have 10.64 because we would be using the value of 64 right here from this bit. If it were 10, then it would be 10.128 because we would be using this value from this bit, and not the second one.

If we add ones in both of those first two bits of the second octet, we would have 10.192 because we would add 64 and 128 for a value of 192.

I know this is complicated, but look what we've done. We've defined four separate subnets. We could take this subnet and assign it to this network segment. We could take this subnet and assign it to this one, assign this subnet at this network address to this one, and this network address to this one.

By doing this, we define the four subnets shown here. For the 10.0.0.0 subnet, this is the subnet mask we would use; notice that the subnet mask is the same for all of them. But because we're subnetting, we're stealing bits from the host portion of the address; therefore, the available addresses on that network are going to be much smaller.

For the 10.0.0.0 network, these are the available addresses that we can use, and here's the broadcast address. The subnet address for the second subnet is 10.64.0.0. Here's that subnet mask and here are the addresses that can be used on it, and here's its broadcast address.

I want you to notice that the first available address within a given subnet is just one address higher than the broadcast address for the previous subnet. The last address on the 10.0.0.0 network is 10.63.255.255, so the first available host address on the next one is 10.64.0.1. 0.0, of course, being reserved for the subnet address. And likewise, the first available address on the third subnet, 10.128.0.0, is one higher than the broadcast address of the previous subnet.

Host Communications 7:17-8:29

The important thing that you have to remember when you're working with subnet masks is that for two hosts on the same network segment to communicate directly, they must have exactly the same network address. Which basically means they have to have the same subnet mask.

Consider the three systems shown here. We have one host with an IP address of 192.168.1.1/24, the second host has an IP address of 192.168.1.2/24, and the last one has an IP address of 192.168.1.3/22.

Will these hosts be able to communicate? Yes and no. Well, these two hosts will. Why? Because they have exactly the same network address: 192.168.1.0. We know that because the length of the subnet mask is exactly the same.

This is where things get confusing, because at first glance you might say, "Okay, that's a class C address. 192.168.1.3, so it should be able to communicate with the other hosts, right?"

Wrong, because we have not used the default mask, the 24-bit mask, for this address. We used a 22-bit mask, which means it's on its own network. It can't talk to these other two hosts unless you put a router in between them.

Summary 8:30-8:30

That's it for this lesson. In this lesson we discussed variable length subnet masking.
