

# VLSM: Variable Length Subnet Masks

*Professor Don Colton*

Brigham Young University Hawaii

## 1 Introduction

IPv4 is currently the main addressing method on the Internet. Students who plan to use networking skills in their future careers should have some technical skill with this addressing method.

We assume you already know how to do classless addressing.

This tutorial introduces Variable Length Subnet Masks, and explains in detail how to design them.

VLSM is a big win in situations where there is large variation in the sizes of the LANs, and you are not at liberty to use private addressing solutions.

## 2 Scenarios

In these types of problems, you are always given a scenario. The scenario tells you how many work groups are involved, and how many routers, and how they are to be connected. It also tells you what range of addresses you have to work with.

Typically we are working with four or more work groups, four or more routers, and one class C network.

## 3 Users and Hosts

Each work group is given a name and a number of “users.” From this you need to determine the size of the subnet that will be required.

The simple math is to take the number of users, add two, and round up to a power of two. We add two because the all-zeroes host address and the all-ones host address are reserved. All-zeroes is reserved for the subnet itself, and all-ones is reserved for broadcast.

The word “user” can be misleading. In the context of these problems, a user is any device that uses the subnet. As such, it includes the routers that will be attached. It also includes any devices such

as network printers that will be attached. In other words, don’t just count the number of people.

In defined problems, you just take the number of users and add two.

In real life, you take the *estimated* number of users and add two. When possible, allow for some growth. You know it’s going to happen. Might as well build it in from the start. That way you can put off a network redesign for as long as possible.

## 4 Router to Router LANs

Besides the basic LANs that are described in the scenario, there are other LANs that are implicit. I am speaking of the very small LANs that connect the pairs of routers.

Each router-to-router connection is a very small LAN that only has two users, being the two routers themselves.

These LANs also need to be included in your list of LANs. And each of them will require a /30 address space. That gives us two bits for host addressing, yielding four total IP addresses, where we can use two IPs for the two routers, and two IPs for the reserved addresses (subnet and broadcast).

## 5 Powers of Two

Powers of two are fundamental to this entire subnetting task. Each subnet will occupy a number of IP addresses. That number is required to be a power of two, AND to start of a boundary that is a multiple of that number.

The first few powers of two are:  $2^0 = 1$ ,  $2^1 = 2$ ,  $2^2 = 4$ ,  $2^3 = 8$ ,  $2^4 = 16$ ,  $2^5 = 32$ ,  $2^6 = 64$ ,  $2^7 = 128$ ,  $2^8 = 256$ ,  $2^9 = 512$ , and  $2^{10} = 1024$ .

Each power is constructed by multiplying together some number of twos. For example,  $2^9$  is equal to nine twos multiplied together.  $2 \times 2 = 512$ .

## 6 Space Available

Often you will be asked to lay out your LANs within a class C address space. To be more general, we will give you a part of a class C address space by taking away some addresses that have already been assigned.

Let's say that instead of you having all the IP addresses from 0 to 255 available to you, that some higher entity has reserved the first 8 IP addresses for their own purposes, and you cannot use them. Or the last 8 IP addresses. Or both. Or some other number.

We have to pack our desired LANs into the available space, while carefully observing the alignments.

## 7 Alignment

The most restricted LANs to place are the big ones. So start with your biggest LAN. For proper alignment, we must position it into a space that starts on a multiple of the LAN's block size.

After assigning your biggest LAN, continue with your next biggest LAN, and so on until all the LANs have spaces assigned.

In a class C space, if your LAN size is 128, you only have two choices. Your LAN can go in the first 128 slots or in the last 128 slots.

If one of those slots has already been taken over by the other powers that be, then your choice is simple. If both are available, flip a coin.

In a class C space, if your LAN size is 64, you have four choices: 0, 64, 128, and 192. Rule out any that have been colonized by the powers that be. Then flip a coin and pick among those that are left.

## 8 Rules of Thumb

Although you can choose to place each LAN in any random space that is big enough, a good rule of thumb is to start at (a) the front, (b) the back, or (c) the middle. Then work out from there, without leaving any empty spaces.

These rules always work because blocks that are the same or smaller can always be positioned right next to a legally placed block that is the same size or larger. So just position the first block carefully and then pack the rest in snugly against it.

## 9 Sample Problem

Enough about the rules. Let's do a sample problem.

Given: You have four work groups. We will call them San Francisco (SF), Salt Lake (SL), New York (NY), and Dallas (D).

### 9.1 Identify the LANs

Each work group has a single router. Each router is connected to all three other routers. Specifically, we have the following six router pairs: SF-SL, SF-NY, SF-D, SL-NY, SL-D, NY-D.

How many users per work group? Let's say SF has 15, SL has 40, NY has 8, and D has 5.

SF will require a 32-address space.  $15 + 2 = 17$ . 17 rounded up is 32. 32 uses 5 bits, making the subnet mask /27. ( $32 - 5 = 27$ .)

SL will require a 64-address space.  $40 + 2 = 42$ . 42 rounded up is 64. 64 uses 6 bits, making the subnet mask /26.

NY will require a 16-address space.  $8 + 2 = 10$ . 10 rounded up is 16. The subnet mask will be /28.

D will require an 8-address space.  $5 + 2 = 7$ . 7 rounded up is 8. The subnet mask will be /29.

That gives us the following LANs, sorted by decreasing size: SL=64 (/26), SF=32 (/27), NY=16 (/28), D=8 (/29), SF-SL=4 (/30), SF-NY=4, SF-D=4, SL-NY=4, SL-D=4, NY-D=4.

### 9.2 Identify the Space Available

Next, assume the first and last 8 IP addresses have been claimed by some higher power. All we have left is 8 through 247.

### 9.3 Assign the Largest LAN

SL is largest. SL must be assigned on a 64 boundary. That means 0, 64, 128, or 192. Because of prior claims on the first and last 8 IP addresses, we cannot use the first or last block. That leaves 64 and 128. Both of these are in the middle, so we may as well use our rule of thumb (c) and assign from the middle outward.

We choose to assign SL=64 to slot x.x.x.64/26. It will use the IP addresses from x.x.x.64 up through x.x.x.127.

(We say x.x.x.64 here because we do not wish to bother with the specifics of our class C address space. It might be 200.200.200.0/24 or any other class C

space. The x.x.x just stands for whatever our actual class C network is.)

## 9.4 Assign the Other LANs

We proceed from largest to smallest, positioning each LAN right after the one before.

Next is SF=32 with a /27 subnet mask. It can be positioned adjacent to the SL block because every multiple of 64 (the SL size) is also automatically a multiple of 32 (the SF size).

We choose to assign SF=32 to slot x.x.x.128/27. It will use the IP addresses from x.x.x.128 up through x.x.x.159.

Next is NY=16 with a /28 subnet mask. It can be positioned adjacent to either the SL block or the SF block.

We choose to assign NY=16 to slot x.x.x.160/28. It will use the IP addresses from x.x.x.160 up through x.x.x.175. Notice that 160 is clearly a multiple of 16.

We choose to assign D=8 to slot x.x.x.176/29. It will use the IP addresses from x.x.x.176 up through x.x.x.183. Notice that 176 is a multiple of 8, but it's a bit harder to do the math. We are safe, however, because we positioned it next to a larger block that was correctly aligned.

We choose to assign SF-SL=4 to slot x.x.x.184/30. It will use the IP addresses from x.x.x.184 up through x.x.x.187.

We choose to assign SF-NY=4 to slot x.x.x.188/30. It will use the IP addresses from x.x.x.188 up through x.x.x.191.

We choose to assign SF-D=4 to slot x.x.x.192/30. It will use the IP addresses from x.x.x.192 up through x.x.x.195.

We choose to assign SL-NY=4 to slot x.x.x.196/30. It will use the IP addresses from x.x.x.196 up through x.x.x.199.

We choose to assign SL-D=4 to slot x.x.x.200/30. It will use the IP addresses from x.x.x.200 up through x.x.x.203.

We choose to assign NY-D=4 to slot x.x.x.204/30. It will use the IP addresses from x.x.x.204 up through x.x.x.207.

We are done. Every LAN has an assigned slot. Every slot starts on a correct boundary. None of the slots overlap each other. We are done.

## 9.5 Results

Given a scenario, your task would be to properly assign each of the subnets. The following table is an example of that result.

Notice that the subnets are not all the same size. That is the V in VLSM: Variable.

Notice also that we have ten subnets and two reserved spaces that are beyond our control. If we did *not* use VLSM, even with just the ten subnets, we would end up with 16 equally-sized LANs. In our class-C space, each would be 16 IPs wide. There is no way that SL or SF could even fit.

VLSM is a big win in a situation where there is large variation in the sizes of the LANs.

LAN	size	start/mask
res	8	x.x.x.0/29
SL	64	x.x.x.64/26
SF	32	x.x.x.128/27
NY	16	x.x.x.160/28
D	8	x.x.x.176/29
SF-SL	4	x.x.x.184/30
SF-NY	4	x.x.x.188/30
SF-D	4	x.x.x.192/30
SL-NY	4	x.x.x.196/30
SL-D	4	x.x.x.200/30
NY-D	4	x.x.x.204/30
res	8	x.x.x.248/29

## 9.6 Sample Problem Improvements

We used a simple rule of thumb to specify the locations where things were assigned.

Instead of assigning from the middle out, we might want to assign the router-to-router LANs adjacent to the space claimed by those higher powers. There are already 8 IPs reserved at the front and the back. We might as well expand those to 16 by adding four of the r-r LANs. It is nice when we can leave empty chunks that are the largest possible powers of two.