by S. Lee Henry

DAVE RIDLEY

Variable-Length Subnet Masking

Several months ago, I described subnet masking (see "Demystifying Netmasks," May 1998, Page 43). I looked at the way a mask defines which bits in an IP address are used to specify a single host on a particular subnet and how it determines the total number of hosts in a subnet. To review by example, the subnet mask `255.255.255.240` (in which the rightmost 4 bits are set to zero) defines a subnet with up to 16 (2^4) available addresses, two of which are reserved for the subnet address and the subnet broadcast address.

Dividing a network into a number of same-size subnets is relatively easy, but the concept of a mask is often misunderstood. The exercise of selecting the correct number of bits to use is as easy as running up a list of powers of two to determine the size of subnet required. Subnets for a class C network, for example, will each have from 2 (2^2 - 2) to 126 (2^7 - 2) available hosts addresses.

The appropriate subnet size depends entirely on your organization. Because subnet masks are used to effect proper routing with respect to the individual subnets, the size of each subnet (with standard subnet masking) is determined by the size of the largest group to which you want to route. Not that all the subnets will reserve the same number of possible addresses. For instance, if you were an Internet service provider (ISP) using standard subnet masking, you might offer customers subnets capable of supporting up to 16 addresses each. However, consider the address space that is wasted if most of them have fewer than six systems.

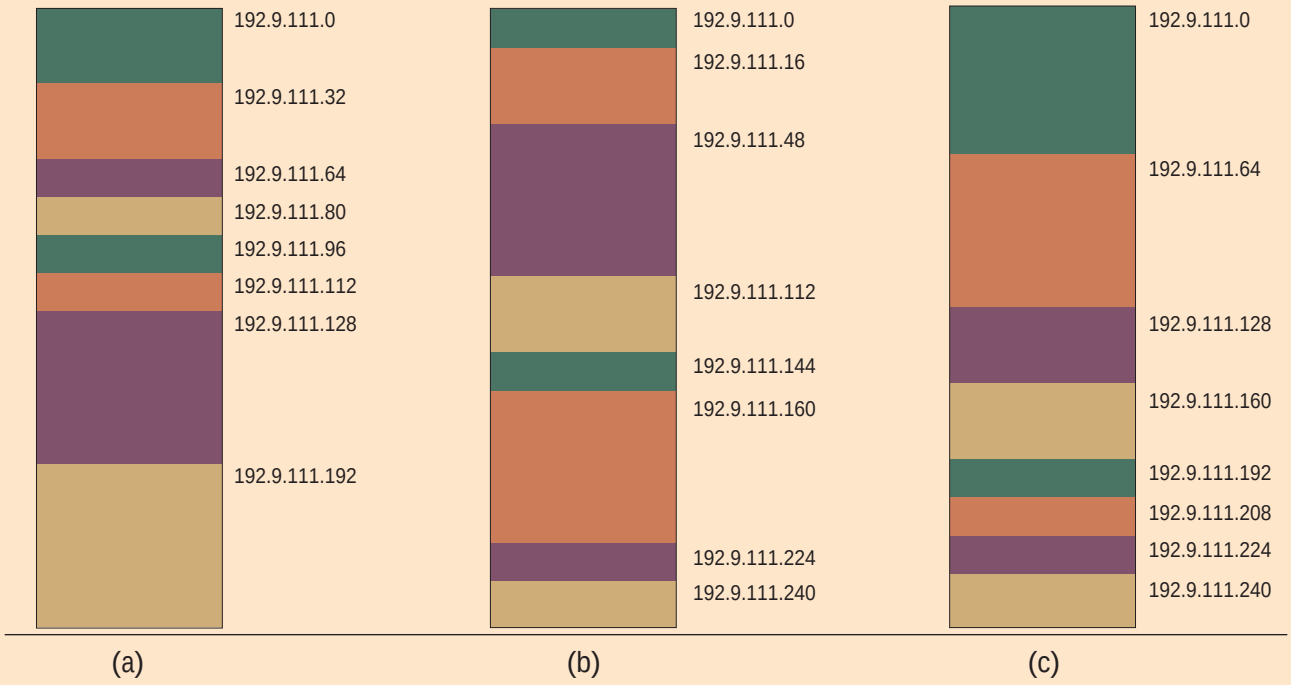
Introducing VLSM

To conserve address space, making it possible to define subnets of varying sizes, Variable-Length Subnet Masking, or VLSM, was introduced. VLSM is, in concept, a simple extension of standard subnet masking. With VLSM, a number of subnet masks are defined for a single network and used as needed to create different-size subnets. A class C network might, for example, be broken into subnets containing 16, 32 and 64 host addresses. What are the steps?

The first step is to define the individual subnet masks. The subnet mask `255.255.255.240`, as we've mentioned, reserves the rightmost 4 bits for host addresses and defines subnets with up to 16 available addresses, 14 of which can be used for the host. The subnet mask `255.255.255.224` reserves 5 bits for host addresses and, thereby, defines subnets with 32 addresses and up to 30 hosts each. The third subnet mask we need requires one more bit of address space to accommodate up to 64 addresses and 62 hosts; that subnet mask is `255.255.255.192`.

With these three subnet masks (`255.255.255.240`, `255.255.255.224` and `255.255.255.192`), we can now define subnets of all three sizes. The only "trick" left is to determine how to break off each subnet and associate it with the correct subnet mask for its size.

Figure 1. Possible Arrangements of Subnet Addresses



Breaking the network address space into variable-size chunks means that we must define nonoverlapping address spaces. The address space of a class C network allows for up to 254 (256 - 2) hosts. If we break this into subnets, we lose a small portion of the address space in order to accomodate the additional subnet and broadcast addresses. If we break this space into two subnets of 64 addresses each, two of 32 each and four of 16 each, then 16 addresses are reserved for the subnet and subnet broadcast addresses, 14 more than normal.

The subnets can be arranged in any way that is convenient to you. Any of the arrangements shown in Figure 1, or many others for that matter, are correct. The most important factor is what is easiest to manage.

Listing 1 shows us what the first subnet for each arrangement in Figure 1 would be if our class C network address was `192.9.111.0`.

The addresses `192.9.111.31`, `192.9.111.15` and `192.9.111.63` might not *look* like broadcast addresses, because we are used to seeing addresses like `192.9.111.255` used for this purpose, but they are. Keep in mind that the rightmost bits of each address are an appropriately-sized (for the subnet) string of `1`'s.

Listing 2 shows the subnets in Figure 1 (b) alone.

I think I got that right! If not, keep in mind that the process of calculating subnet addresses is tedious even if it isn't terribly difficult, and tedium is the mother of mistake. Be sure to double-check your addresses or build a script to calculate them for you. If you're dividing a class B network into subnets, instead of a class C as we've shown here, then there will be many more opportunities for error.

Listing 1. First Subnet for Each Arrangement in Figure 1

```
     First Subnet   Subnet Mask       Hosts (from-to)
(a)  192.9.111.0    255.255.255.224   192.9.111.1-192.9.111.30
(b)  192.9.111.0    255.255.255.240   192.9.111.1-192.9.111.14
(c)  192.9.111.0    255.255.255.192   192.9.111.1-192.9.111.62
```

Listing 2. Subnets for Arrangement (b)

```
     Subnet          Subnet Mask      Hosts (from-to)     Broadcast
(1)  192.9.111.0     255.255.255.240  192.9.111.1-14      192.9.111.15
(2)  192.9.111.16    255.255.255.224  192.9.111.17-46     192.9.111.47
(3)  192.9.111.48    255.255.255.192  192.9.111.49-110    192.9.111.111
(4)  192.9.111.112   255.255.255.224  192.9.111.113-142   192.9.111.143
(5)  192.9.111.144   255.255.255.240  192.9.111.145-158   192.9.111.159
(6)  192.9.111.160   255.255.255.192  192.9.111.161-222   192.9.111.223
(7)  192.9.111.224   255.255.255.240  192.9.111.225-238   192.9.111.239
(8)  192.9.111.240   255.255.255.240  192.9.111.241-254   192.9.111.255
```

S. Lee Henry *has been performing UNIX parlor tricks for approximately 15 years. She currently lives with her stepfamily on a sailboat in Marin County, CA. You can reach her at* `slee@cpg.com`.