IP Subnet Masking

The *network address* is obtained by ANDing the address of any host on the network with the subnet mask:

Example: What is the network address for host 172.0.0.125 given the subnet mask 255.255.255.0?

```
Host Address: 172.0.0.125       AC.00.00.7D
Subnet mask:  255.255.255.0     FF.FF.FF.00
                    (AND)    & ___________
           = Network Address    AC.00.00.00 = 172.0.0.0
```

The *broadcast address* is obtained by ORing the network address with the 1s complement of the subnet mask:

Example: What is the broadcast address for 172.0.0.0/24?

First, note that "/24" is a shorthand way of giving the subnet mask, which is always a group of 1s. This notation says that the subnet mask has 24 1s and 8 zeros, which looks like this:

```
1111 1111 1111 1111 1111 1111 0000 0000

 F    F    F    F    F    F    0    0
 └──┘ └──┘ └──┘ └──┘ └──┘ └──┘ └──┘ └──┘
  255   .   255   .   255   .   0
```

The broadcast address is obtained:

```
1s comp subnet mask: 0000 0000 0000 0000 0000 0000 1111 1111 (00.00.00.FF)
Network address:     1010 1100 0000 0000 0000 0000 0000 0000 (AC.00.00.00)
               (OR)+ ____________________________________________
                     1010 1100 0000 0000 0000 0000 1111 1111 (AC.00.00.FF)
                                                      = 172.0.0.255
```

The *number of hosts on any network* is the <u>number of available network addresses</u> minus 2. This is because all networks use two addresses from those available - the network address (all zeroes) and the broadcast address (all ones).

Example: Divide the network above (172.0.0.0/24) into two individual and equal-sized parcels. For each new network, state the subnet mask, network address, broadcast address, and number of hosts.

First, the subnet mask will be modified to /25:

```
Mask = 1111 1111 1111 1111 1111 1111 1000 0000 ( 255.255.255.128 )
```

Note how the network portion of the address has increased in size by 1 bit, and the host portion of the address has become one bit smaller. The underlined "1" in the mask is the new one we've added to split the network. This bit will be toggled in the final addressing scheme to determine the new network addresses.

To get the new network addresses, simply toggle the added subnet bit(s) through all possible states.

```
Network 1 Address = 1010 1100 0000 0000 0000 0000 0000 0000
                  =  AC.00.00.00 = 172.0.0.0
```

The underlined bit is the one we added to the subnet mask.

```
1s comp subnet mask: 0000 0000 0000 0000 0000 0000 0111 1111 (00.00.00.80)
Network address:     1010 1100 0000 0000 0000 0000 0000 0000 (AC.00.00.00)
               (OR)+ _____________________________________________
                     1010 1100 0000 0000 0000 0000 0111 1111 (AC.00.00.7F)

Network 1 Broadcast address = 172.0.0.127
```

Network 1 has addresses 172.0.0.0 to 172.0.0.127, and host addresses 172.0.0.1 to 172.0.0.126 (126 hosts).

To find the remaining networks, toggle the added subnet mask bit(s) through all possible states. Since we only added one bit, there are two states (networks).

```
Network 2 Address = 1010 1100 0000 0000 0000 0000 1000 0000
                  =  AC.00.00.80 = 172.0.0.128
```

The underlined bit is the one we added to the subnet mask.

```
1s comp subnet mask: 0000 0000 0000 0000 0000 0000 0111 1111 (00.00.00.80)
Network address:     1010 1100 0000 0000 0000 0000 1000 0000 (AC.00.00.80)
               (OR)+ _____________________________________________
                     1010 1100 0000 0000 0000 0000 1111 1111 (AC.00.00.FF)

Network 2 Broadcast address = 172.0.0.255
```

Network 2 has addresses 172.0.0.128 to 172.0.0.255, and host addresses 172.0.0.129 to 172.0.0.254 (126 hosts).