Chapter three

Internetworking with TCP/IP

The IP protocol and IP addresses

One of the most important protocols in the TCP/IP suite is the IP protocol. This is used at the Internet layer of TCP/IP (i.e. the Network layer in the OSI model) and is used to attach network addresses to packets. The IP protocol provides *best effort delivery* between network stations.

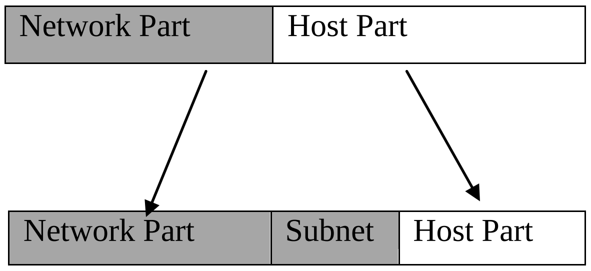
IP routes packets on the network by using *IP addresses*. An IP address consists of 4 numbers between 0 and 255 separated by dots. For example, 146.23.0.194 is a valid IP address. When you surf the Internet and type a URL into your browser (e.g. www.yahoo.com) you are actually using IP addresses. Every Internet domain such as Yahoo or Google has at least one IP address on the Internet.

Subnets and Subnet Masks (Classless Inter-domain routing)

It is possible to divide a large network in to smaller networks using network masks. All smaller networks are called subnets. The net mask used for these subnets is called the subnet mask.

The figure below shows how the host part in the network address is divided in to the subnet and a smaller host part to form multiple smaller networks.

In the lower part of the figure, both of the shaded areas (network part and subnet) are used as a larger network part for smaller networks.



Instead of the class-full two-level hierarchy, sub-netting supports a three-level hierarchy. Figure below illustrates the basic idea of sub-netting which is to divide the standard class-full host-number field into two parts - the subnet-number and the host number on that subnet.

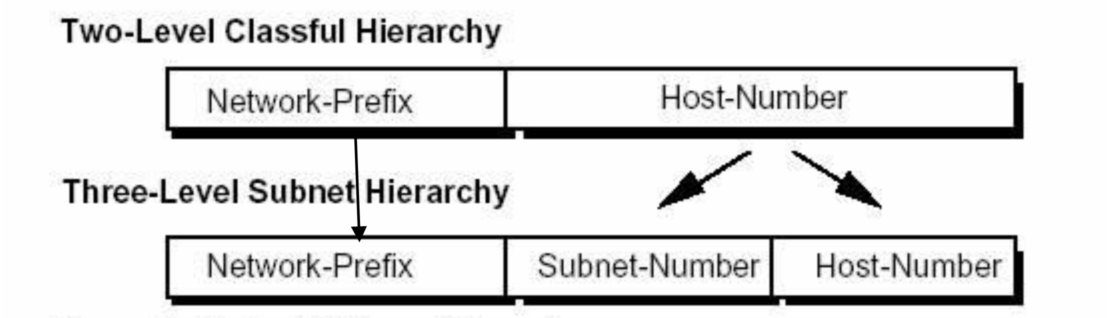


Figure 6: Subnet Address Hierarchy

Sub-netting defend the expanding routing table problem by ensuring that the subnet structure of a network is never visible outside of the organization's private network. The route from the Internet to any subnet of a given IP address is the same, no matter which subnet the destination host is on. This is because all subnets of a given network number use the same network-prefix but different subnet numbers. The routers within the private organization need to differentiate between the individual subnets, but as far as the Internet routers are concerned, all of the subnets in the organization are collected into a single routing table entry. This allows the local administrator to introduce arbitrary complexity into the private network without affecting the size of the Internet's routing tables.

Sub-netting overcame the registered number issue by assigning each organization one (or at most a few) network number(s) from the IPv4 address space. The organization was then free to assign a distinct sub-network number for each of its internal networks. This allows the organization to deploy additional subnets without needing to obtain a new network number from the Internet.

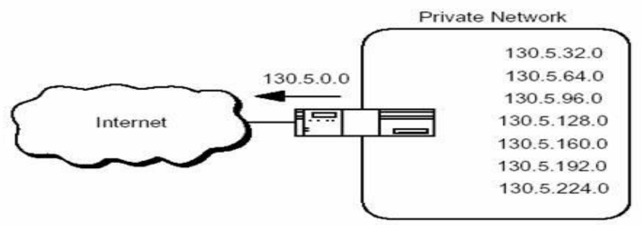


Figure 7: Subnetting Reduces the Routing Requirements of the Internet

In Figure 7, a site with several logical networks uses subnet addressing to cover them with a single /16 (Class B) network address. The router accepts all traffic from the Internet addressed to network 130.5.0.0, and forwards traffic to the interior sub-networks based on the third octet of the class-full address.

The deployment of sub-netting within the private network provides several benefits:

- The size of the global Internet routing table does not grow because the site administrator does not need to obtain additional address space and the routing advertisements for all of the subnets. The subnets are combined into a single routing table entry.
- The local administrator has the flexibility to deploy additional subnets without obtaining a new network number from the Internet.
- Route flapping (i.e., the rapid changing of routes) within the private network does not affect the Internet routing table since Internet routers do not know about the reach-ability of the individual subnets - they just know about the reach-ability of the parent network number.

Extended-Network-Prefix

Internet routers use only the network-prefix of the destination address to route traffic to a sub-netted environment. Routers within the sub-netted environment use the extended network- prefix to route traffic between the individual subnets. The extended-network-prefix is composed of the class-full network-prefix and the subnet-number.

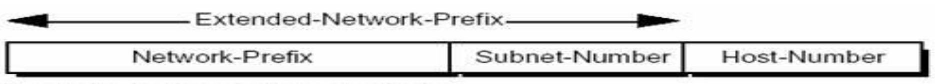
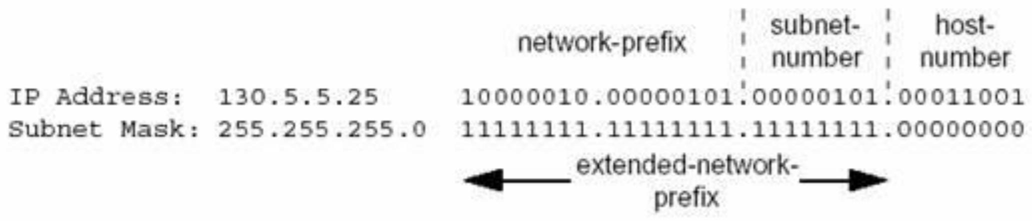


Figure 8: Extended-Network-Prefix

The extended-network-prefix has traditionally been identified by the subnet mask. For example, if you have the /16 address of 130.5.0.0 and you want to use the entire third octet to represent the subnet-number, you need to specify a subnet mask of 255.255.255.0. The bits of the subnet mask are set to 1 if the system examining the address should treat the corresponding bit in the IP address as part of the extended network-prefix. The bits in the mask are set to 0 if the system should treat the bit as part of the host-number. This is illustrated if Figure 9.



```
IP Address:  130.5.5.25      10000010.00000101.00000101.00011001
Subnet Mask: 255.255.255.0   11111111.11111111.11111111.00000000
```

Figure 9: Subnet Mask

Subnet Design Considerations

The deployment of an addressing plan requires careful thought on the part of the network administrator. There are four key questions that must be answered before any design should be undertaken:

1) How many total subnets does the organization need today?

2) How many total subnets will the organization need in the future?

3) How many hosts are there on the organization's largest subnet today?

4) How many hosts will there be on the organization's largest subnet in the future?

The first step in the planning process is to take the maximum number of subnets required and round up to the nearest power of two. For example, if a organization needs 9 subnets, 2^3 (or 8) will not provide enough subnet addressing space, so the network administrator will need to round up to 2^4 (or 16). When performing this assessment, it is critical that the network administrator always allow adequate room for future growth. For example, if 14 subnets are required today, then 16 subnets might not be enough in two years when the 17th subnet needs to be deployed. In this case, it might be wise to allow for more growth and select 2^5 (or 32) as the maximum number of subnets.

The second step is to make sure that there are enough host addresses for the organization's largest subnet. If the largest subnet needs to support 50 host addresses today, 2^5 (or 32) will not provide enough host address space so the network administrator will need to round up to 2^6 (or 64).

Subnet Example #1

Given

An organization has been assigned the network number 193.1.1.0/24 and it needs to define six subnets. The largest subnet is required to support 25 hosts.

Defining the Subnet Mask / Extended-Prefix Length

The first step is to determine the number of bits required to define the six subnets. Since a network address can only be subnetted along binary boundaries, subnets must be created in blocks of powers of two [2 (2^1), 4 (2^2), 8 (2^3), 16 (2^4), etc.]. Thus, it is impossible to define an IP address block such that it contains exactly six subnets. For this example, the network administrator must define a block of 8 (2^3) and have two unused subnets that can be reserved for future growth.

Since 8 = 2^3, three bits are required to enumerate the eight subnets in the block. In this example, the organization is sub-netting a /24 so it will need three more bits, or a /27, as the extended-network-prefix. A 27-bit extended-network-prefix can be expressed in dotted-decimal notation as 255.255.255.224. This is illustrated in Figure 11.

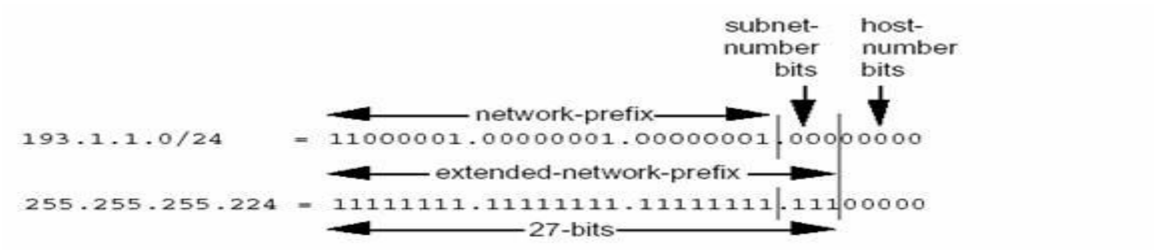


Figure 11: Example #1 - Defining the Subnet Mask/Extended-Prefix Length

A 27-bit extended-network-prefix leaves 5 bits to define host addresses on each subnet. This means that each sub-network with a 27-bit prefix represents a contiguous block of 2^5 (32) individual IP addresses. However, since the all-0s and all-1s host addresses cannot be allocated, there are 30 (2^5 -2) assignable host addresses on each subnet.

Defining Each of the Subnet Numbers

The eight subnets will be numbered 0 through 7. In this paper, the XXX2 notation indicates the binary representation of the number. The 3-bit binary representation of the decimal values 0 through 7 are: 0 (0002), 1 (0012), 2 (0102), 3 (0112), 4 (1002), 5 (1012), 6 (1102), and 7 (1112).

In general, to define Subnet #n, the network administrator places the binary representation of n into the bits of the subnet-number field. For example, to define Subnet #6, the network administrator simply places the binary representation of 6 (1102) into the 3-bits of the subnet-number field.

The eight subnet numbers for this example are given below. The underlined portion of each address identifies the extended-network-prefix, while the **bold** digits identify the 3- bits representing the subnet-number field:

```
 Base Net: 11000001.00000001.00000001 .00000000 = 193.1.1.0/24
Subnet #0: 11000001.00000001.00000001. 000 00000 = 193.1.1.0/27
```

Subnet #1: 11000001.00000001.00000001. **001** 00000 = 193.1.1.32/27

Subnet #2: 11000001.00000001.00000001. **010** 00000 = 193.1.1.64/27

Subnet #3: 11000001.00000001.00000001. **011** 00000 = 193.1.1.96/27

Subnet #4: 11000001.00000001.00000001. **100** 00000 = 193.1.1.128/27

Subnet #5: 11000001.00000001.00000001. **101** 00000 = 193.1.1.160/27

Subnet #6: 11000001.00000001.00000001. **110** 00000 = 193.1.1.192/27

Subnet #7: 11000001.00000001.00000001. **111** 00000 = 193.1.1.224/27

An easy way to check if the subnets are correct is to ensure that they are all multiples of the Subnet #1 address. In this case, all subnets are multiples of 32: 0, 32, 64, 96, etc.

Defining Host Addresses for Each Subnet

According to Internet practices, the host-number field of an IP address cannot contain all 0-bits or 1-bits. The all-0s host-number identifies the base network (or sub-network) number, while the all-1s host-number represents the broadcast address for the network (or sub-network).

In our current example, there are 5 bits in the host-number field of each subnet address. This means that each subnet represents a block of 30 host addresses (2^5-2 = 30, note that the 2 is subtracted because the all-0s and the all-1s host addresses cannot be used). The hosts on each subnet are numbered 1 through 30.

In general, to define the address assigned to Host #n of a particular subnet, the network administrator places the binary representation of n into the subnet's host-number field. For example, to define the address assigned to Host #15 on Subnet #2, the network administrator simply places the binary representation of 15 (01111_2) into the 5-bits of Subnet #2's host-number field.

The valid host addresses for Subnet #2 in our example are given below. The underlined portion of each address identifies the extended-network-prefix, while the **bold** digits identify the 5-bit host-number field:

Subnet #2: 11000001.00000001.00000001.010 **00000** = 193.1.1.64/27

Host #1: 11000001.00000001.00000001.010 **00001 =** 193.1.1.65/27

Host #2: 11000001.00000001.00000001.010 **00010 =** 193.1.1.66/27

Host #3: 11000001.00000001.00000001.010 **00011 =** 193.1.1.67/27

Host #4: 11000001.00000001.00000001.010 **00100 =** 193.1.1.68/27

Host #5: 11000001.00000001.00000001.010 **00101 =** 193.1.1.69/27

.

.

Host #15: 11000001.00000001.00000001.010 **01111 =** 193.1.1.79/27

Host #16: 11000001.00000001.00000001.010 **10000 =** 193.1.1.80/27

.

.

Host #27: 11000001.00000001.00000001.010 **11011 =** 193.1.1.91/27

Host #28: 11000001.00000001.00000001.010 **11100 =** 193.1.1.92/27

Host #29: 11000001.00000001.00000001.010 **11101 =** 193.1.1.93/27

Host #30: 11000001.00000001.00000001.010 **11110 =** 193.1.1.94/27

The valid host addresses for Subnet #6 are given below. The underlined portion of each address identifies the extended-network-prefix, while the **bold** digits identify the 5-bit host-number field:

Subnet #6: 11000001.00000001.00000001.110 **00000** = 193.1.1.192/27

Host #1: 11000001.00000001.00000001.110 **00001** = 193.1.1.193/27

Host #2: 11000001.00000001.00000001.110 **00010** = 193.1.1.194/27

Host #3: 11000001.00000001.00000001.110 **00011** = 193.1.1.195/27

Host #4: 11000001.00000001.00000001.110 **00100** = 193.1.1.196/27

Host #5: 11000001.00000001.00000001.110 **00101** = 193.1.1.197/27

.

.

Host #15: 11000001.00000001.00000001.110 **01111** = 193.1.1.207/27

Host #16: 11000001.00000001.00000001.110 **10000** = 193.1.1.208/27

.

.

Host #27: 11000001.00000001.00000001.110 **11011** = 193.1.1.219/27

Host #28: 11000001.00000001.00000001.110 **11100** = 193.1.1.220/27

Host #29: 11000001.00000001.00000001.110 **11101** = 193.1.1.221/27

Host #30: 11000001.00000001.00000001.110 **11110** = 193.1.1.222/27

Defining the Broadcast Address for Each Subnet

The broadcast address for Subnet #2 is the all 1's host address or:

11000001.00000001.00000001.010 **11111** = 193.1.1.95

Note that the broadcast address for Subnet #2 is exactly one less than the base address
for Subnet #3 (193.1.1.96). This is always the case - the broadcast address for Subnet
#n is one less than the base address for Subnet #(n+1).

The broadcast address for Subnet #6 is simply the all 1's host address or:

11000001.00000001.00000001.110 **11111** = 193.1.1.223

Again, the broadcast address for Subnet #6 is exactly one less than the base address for Subnet #7 (193.1.1.224).

Subnet Example #2

Given

An organization has been assigned the network number 140.25.0.0/16 and it needs to create a set of subnets that supports up to 60 hosts on each subnet.

Defining the Subnet Mask / Extended-Prefix Length

The first step is to determine the number of bits required to define 60 hosts on each subnet. Since a block of host address can only be assigned along binary boundaries, host address blocks can only be created in powers of two. This means that it is impossible to create a block that contains exactly 60 host addresses. To support 60 hosts, the network administrator must define a minimum address block of 62 (2^6-2) host addresses.

However, this choice would only provide two unused host addresses on each subnet for future growth. Since this does not appear to be adequate to support additional growth, the network administrator elects to define a block of 126 (2^7-2) host addresses and has 66 addresses on each subnet for future growth. A block of 126 host addresses requires 7-bits in the host-number field.

The next step is to determine the subnet mask/extended-prefix length. Since 7-bits of the 32-bit IP address are required for the host-number field, the extended-prefix must be a /25 (25 = 32-7). A 25-bit extended-network-prefix can be expressed in dotted-decimal notation as 255.255.255.128. This is illustrated in Figure 14.

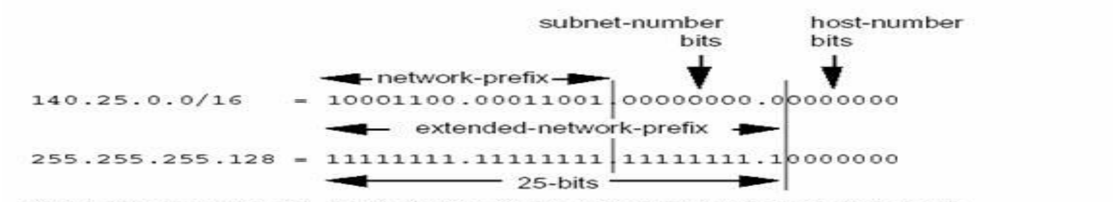


Figure 14: Example #2 - Defining the Subnet Mask/Extended-Prefix Length

Figure 14 show that the 25-bit extended-prefix assigns 9-bits to the subnet number field. Since 2^9 = 512, nine bits allow the definition of 512 subnets. Depending on the organization's requirements, the network administrator could have elected to assign additional bits to the host-number field (allowing more hosts on each subnet) and reduce the number of bits in the subnet-number field (decreasing the total number of subnets that can be defined).

Although this example creates a rather large number of subnets, it provides an interesting example because it illustrates what happens to the dotted-decimal representation of a subnet address when the subnet-number bits extend across an octet boundary. It should be mentioned that the same type of confusion can also occur when the host-number bits extend across an octet boundary.

Defining Each of the Subnet Numbers

The 512 subnets will be numbered 0 through 511. The 9-bit binary representation of the decimal values 0 through 511 are: 0 (000000000_2), 1 (000000001_2), 2 (000000010_2), 3 (000000011_2), ..., 511 (111111111_2). To define subnet #3, the network administrator places the binary representation of 3 (000000011_2) into the 9-bits of the subnet-number field.

The 512 subnet numbers for this example are given below. The underlined portion of each address identifies the extended-network-prefix, while the **bold** digits identify the 9- bits representing the subnet-number field:

```
 Base Net: 10001100.00011001 .00000000.00000000 = 140.25.0.0/16
Subnet #0: 10001100.00011001. 00000000. 0 0000000 = 140.25.0.0/25
Subnet #1: 10001100.00011001. 00000000. 1 0000000 = 140.25.0.128/25
Subnet #2: 10001100.00011001. 00000001. 0 0000000 = 140.25.1.0/25
Subnet #3: 10001100.00011001. 00000001. 1 0000000 = 140.25.1.128/25
Subnet #4: 10001100.00011001. 00000010. 0 0000000 = 140.25.2.0/25
Subnet #5: 10001100.00011001. 00000010. 1 0000000 = 140.25.2.128/25
```

Subnet #6: 10001100.00011001. **00000011. 0** 0000000 = 140.25.3.0/25

Subnet #7: 10001100.00011001. **00000011. 1** 0000000 = 140.25.3.128/25

Subnet #8: 10001100.00011001. **00000100. 0** 0000000 = 140.25.4.0/25

Subnet #9: 10001100.00011001. **00000100. 1** 0000000 = 140.25.4.128/25

.

.

Subnet #510: 10001100.00011001. **11111111. 0** 0000000 = 140.25.255.0/25

Subnet #511: 10001100.00011001. **11111111. 1** 0000000 = 140.25.255.128/25

Notice how sequential subnet numbers do not appear to be sequential when expressed in dotted-decimal notation. This can cause a great deal of misunderstanding and confusion since everyone believes that dotted-decimal notation makes it much easier for human users to understand IP addressing. In this example, the dotted-decimal notation obscures rather than clarifies the subnet numbering scheme!

Defining Hosts Addresses for Each Subnet

In this example there are 7 bits in the host-number field of each subnet address. As discussed earlier, this means that each subnet represents a block of 126 host addresses. The hosts on each subnet will be numbered 1 through 126.

The valid host addresses for Subnet #3 are given below. The underlined portion of each address identifies the extended-network-prefix, while the **bold** digits identify the 7-bit host-number field:

Subnet #3: 10001100.00011001.00000001.1 **0000000** = 140.25.1.128/25

Host #1: 10001100.00011001.00000001.1 **0000001** = 140.25.1.129/25

Host #2: 10001100.00011001.00000001.1 **0000010** = 140.25.1.130/25

Host #3: 10001100.00011001.00000001.1 **0000011** = 140.25.1.131/25

Host #4: 10001100.00011001.00000001.1 **0000100** = 140.25.1.132/25

Host #5: 10001100.00011001.00000001.1 **0000101** = 140.25.1.133/25

Host #6: 10001100.00011001.00000001.1 **0000110** = 140.25.1.134/25

.

.

Host #62: 10001100.00011001.00000001.1 **0111110** = 140.25.1.190/25

Host #63: 10001100.00011001.00000001.1 **0111111** = 140.25.1.191/25

Host #64: 10001100.00011001.00000001.1 **1000000** = 140.25.1.192/25

Host #65: 10001100.00011001.00000001.1 **1000001** = 140.25.1.193/25

.

.

Host #123: 10001100.00011001.00000001.1 **1111011** = 140.25.1.251/25

Host #124: 10001100.00011001.00000001.1 **1111100** = 140.25.1.252/25

Host #125: 10001100.00011001.00000001.1 **1111101** = 140.25.1.253/25

Host #126: 10001100.00011001.00000001.1 **1111110** = 140.25.1.254/25

Defining the Broadcast Address for Each Subnet

The broadcast address for Subnet #3 is the all 1's host address or:

`10001100.00011001.00000001.1` **1111111** `= 140.25.1.255`

As is true in general, the broadcast address for Subnet #3 is exactly one less than the base address for Subnet #4 (`140.25.2.0`).

Super-Netting

Super-netting is the reverse process of sub-netting .Class A and class B network addresses have been exhausted now. If one organization assigned 8 class C networks, like below:

```
11000000 10101000 00000000 00000000
11000000 10101000 00000001 00000000
11000000 10101000 00000010 00000000
11000000 10101000 00000011 00000000
11000000 10101000 00000100 00000000
11000000 10101000 00000101 00000000
11000000 10101000 00000110 00000000
11000000 10101000 00000111 00000000
```

Where, the last eight digits are the host part.
To avoid large numbers of routing entries in routing tables on the internet, these networks re- combined into one network using the super-netting method.

Suppose for example that Class C networks from 192.168.0.0 to 192.168.7.0 are allocated to an organization, these networks are listed in binary notation as shown above.

Now you can see that only the right most 3 bits in the network part of all these networks are changing, If you add these 3 bits to the host part, you can have a single network, as shown below

```
11000000 10101000 00000     000 00000000
```

So the network address for this network is 192.168.0.0, and the subnet mask is 255.255.248.0 rather than 255.255.255.0

So using super netting (in certain situations it can be applied) routing tables on the internet are reduced to a greater extent. Instead of creating eight routing entries for individual class C networks; for example above, a single routing entry for the super-net mentioned previously can suffice.

These super-netting and sub-netting technique can also be helpful in reducing the routing and Congestion problems.