Subnetting

IP address stands for internet protocol **address**; it is an identifying number that is associated with a specific computer or computer network. When connected to the internet, the **IP address** allows the computers dto send and receive information.

Private IP Address and Public IP Address are used to uniquely identify a machine on the internet. Private IP address is used with a local network and public IP address is used outside the network. Public IP address is provided by ISP, Internet Service Provider.

Following are the important differences between Private IP Address and Public IP Address.

Sr. No.	Key	Private IP Address	Public IP Address
1	Scope	Private IP address scope is local to present network.	Public IP address scope is global.
2	Communication	Private IP Address is used to communicate within the network.	Public IP Address is used to communicate outside the network.
3	Format	Private IP Addresses differ in a uniform manner.	Public IP Addresses differ in varying range.
4	Provider	Local Network Operator creates private IP addresses using network operating system.	ISP, Internet Service Provider controls the public IP address.
5	Cost	Private IP Addresses are free of cost.	Public IP Address comes with a cost.
6	Locate	Private IP Address can be located using ipconfig command.	Public IP Address needs to be searched on search engine like

Sr. No.	Key	Private IP Address	Public IP Address
			google.

7	Range	Private IP Address range:
		`10.0.0.0 – 10.255.255.255,`
		`172.16.0.0 – 172.31.255.255,`
		`192.168.0.0 – 192.168.255.255`
			Except private IP Addresses, rest IP addresses are public.

8	Example	Private IP Address is like 192.168.11.50.	Public IP Address is like 17.5.7.8.

Understanding IP Addressing:

Everything You Ever Wanted To Know

The Internet continues to grow at a phenomenal rate. This is reflected in the tremendous popularity of the World Wide Web (WWW), the opportunities that businesses see in reaching customers from virtual storefronts, and the emergence of new ways of doing business. It is clear that expanding business and public awareness will continue to increase demand for access to resources on the Internet.

Internet Scaling Problems

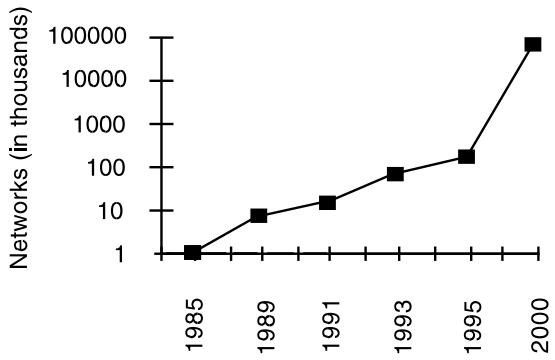
Over the past few years, the Internet has experienced two major scaling issues as it has struggled to provide continuous and uninterrupted growth:

- The eventual exhaustion of IP version 4 (IPv4) address space
- The need to route traffic between the ever increasing number of networks that comprise the Internet

The first problem is concerned with the eventual depletion of the IP address space. IPv4 defines a 32-bit address which means that there are only 232 (4,294,967,296) IPv4 addresses available. As the Internet continues to grow, this finite number of IP addresses will eventually be exhausted.

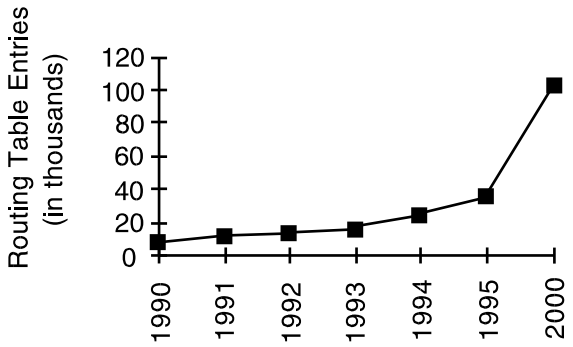
The address shortage problem is aggravated by the fact that portions of the IP address space have not been efficiently allocated. Also, the traditional model of classful addressing does not allow the address space to be used to its maximum potential. The Address Lifetime Expectancy (ALE) Working Group of the Internet Engineering Task Force (IETF) has expressed concerns that if the current address allocation policies are not modified, the Internet will experience a near to medium term exhaustion of its unallocated address pool. If the Internet's address supply problem is not solved, new users may be unable to connect to the global Internet. More than half of all possible IPv4 addresses have been assigned to ISPs, corporations, and government agencies, but only an estimated 69 million addresses are actually in use.

FIG URE 1 . Netw ork Number G row th



The second problem is caused by the rapid growth in the size of the Internet routing tables. Internet backbone routers are required to maintain complete routing information for the Internet. Over recent years, routing tables have experienced exponential growth as increasing numbers of organizations connect to the Internet. In December 1990 there were 2,190 routes, in December 1995 there were more than 30,000 routes, and in December 2000 more than 100,000 routes.

FIG URE 2 . G row th of Internet Routing Tables



Unfortunately, the routing problem cannot be solved by simply installing more router memory and increasing the size of the routing tables. Other factors related to the capacity problem include the growing demand for CPU horsepower to compute routing table/topology changes, the increasingly dynamic nature of WWW connections and their effect on router forwarding caches, and the sheer volume of information that needs to be managed by people and machines. If the number of entries in the global routing table is allowed to increase without bounds, core routers will be forced to drop routes and portions of the Internet will become unreachable.

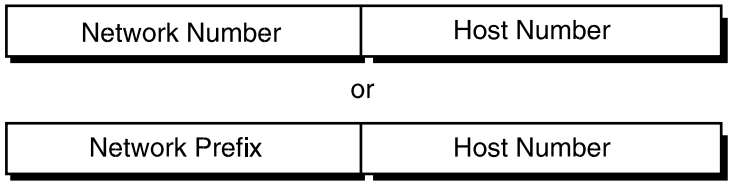
The long-term solution to these problems can be found in the widespread deployment of IP Next Generation (IPng or IPv6). Currently, IPv6 is being tested and implemented on the 6Bone network, which is an informal collaborative project covering North America, Europe, and Japan. 6Bone supports the routing of IPv6 packets, since that function has not yet been integrated into many production routers. Until IPv6 can be deployed worldwide, IPv4

patches will need to be used and modified to continue to provide the universal connectivity users have come to expect.

Classful IP Addressing

When IP was first standardized in September 1981, the specification required that each system attached to an IP-based Internet be assigned a unique, 32-bit Internet address value. Systems that have interfaces to more than one network require a unique IP address for each network interface. The first part of an Internet address identifies the network on which the host resides, while the second part identifies the particular host on the given network. This creates the two-level addressing hierarchy that is illustrated in Figure 3.

FIG URE 3 . Two-Level Internet A ddress S tructure

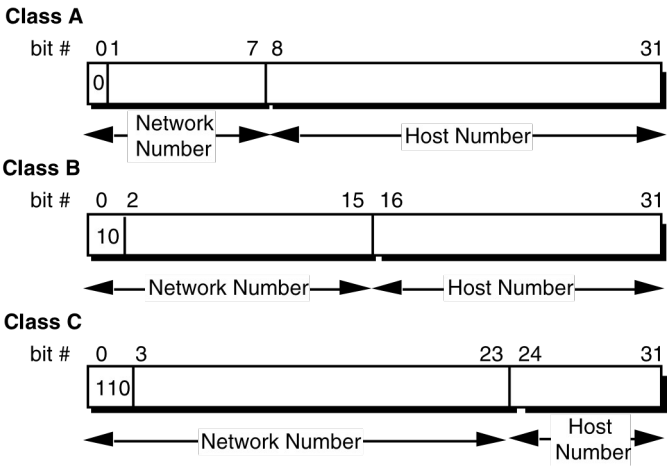


In recent years, the network number field has been referred to as the network prefix because the leading portion of each IP address identifies the network number. All hosts on a given network share the same network prefix but must have a unique host number. Similarly, any two hosts on different networks must have different network prefixes but may have the same host number.

Primary Address Classes

To provide the flexibility required to support networks of varying sizes, the Internet designers decided that the IP address space should be divided into three address classes-Class A, Class B, and Class C. This is often referred to as classful addressing. Each class fixes the boundary between the network prefix and the host number at a different point within the 32-bit address. The formats of the fundamental address classes are illustrated in Figure 4.

FIGURE 4. Principle Classful IP Address Formats



One of the fundamental features of classful IP addressing is that each address contains a self-encoding key that identifies the dividing point between the network prefix and the host number. For example, if the first two bits of an IP address are 1-0, the dividing point falls between the 15th and 16th bits. This simplified the routing system during the early years of the Internet because the original routing protocols did not supply a deciphering key or mask with each route to identify the length of the network prefix.

Class A Networks (/8 Prefixes)

Each Class A network address has an 8-bit network prefix, with the highest order bit set to 0 (zero) and a 7-bit network number, followed by a 24-bit host number. Today, Class A networks are referred to as “/8s” (pronounced “slash eight” or just “eights”) since they have an 8bit network prefix.

Network host

[0-------] . [-------- . -------- . --------]

A maximum of 126 (27 -2) /8 networks can be defined. The calculation subtracts two because the /8 network 0.0.0.0 is reserved for use as the default route and the /8 network 127.0.0.0 (also written 127/8 or 127.0.0.0/8) is reserved for the “loopback” function. Each /8 supports a maximum of 224 -2

(16,777,214) hosts per network. The host calculation subtracts two because the all-0s (all zeros or “this network”) and all-1s (all ones or “broadcast”) host numbers may not be assigned to individual hosts.

Since the /8 address block contains 231 (2,147,483,648) individual addresses and the IPv4 address space contains a maximum of 232 (4,294,967,296) addresses, the /8 address space is 50 percent of the total IPv4 unicast address space.

Class B Networks (/16 Prefixes)

Each Class B network address has a 16-bit network prefix, with the two highest order bits set to 1-0 and a 14-bit network number, followed by a 16-bit host number. Class B networks are now referred to as "/16s" since they have a 16-bit network prefix.

Network Host

[10------ . --------] .[-------- . --------]

A maximum of 16,384 (214) /16 networks can be defined with up to 65,534 (216-2) hosts per network. Since the entire /16 address block contains 230 (1,073,741,824) addresses, it represents 25 percent of the total IPv4 unicast address space.

Class C Networks (/24 Prefixes)

Each Class C network address has a 24-bit network prefix, with the three highest order bits set to 1-1-0 and a 21-bit network number, followed by an 8-bit host number. Class C networks are now referred to as "/24s" since they have a 24-bit network prefix.

Network host

[110----- . -------- . --------] .[--------]

A maximum of 2,097,152 (221) /24 networks can be defined with up to 254 (28-2) hosts per network. Since the entire /24 address block contains 229 (536,870,912) addresses, it represents 12.5 percent (or oneeighth) of the total IPv4 unicast address space.

Other Classes

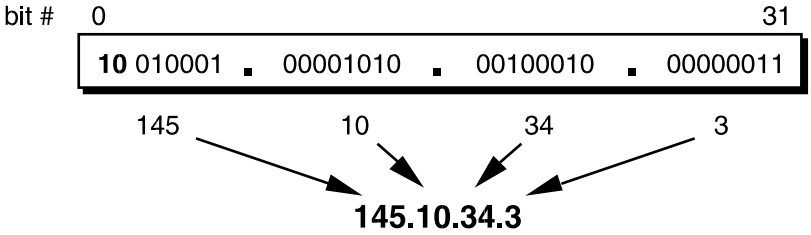
In addition to the three most popular classes, there are two additional classes. Class D addresses have their leading four bits set to 1-1-1-0 and are used to support IP Multicasting. Class E addresses have their leading four bits set to 1-1-1-1 and are reserved for experimental use.

Dotted-Decimal Notation

To make Internet addresses easier for people to read and write, IP addresses are often expressed as four decimal numbers, each separated by a dot. This format is called "dotted-decimal notation."

Dotted-decimal notation divides the 32-bit Internet address into four 8bit fields and specifies the value of each field independently as a decimal number with the fields separated by dots. Figure 5 shows how a typical / 16 (Class B) Internet address can be expressed in dotted-decimal notation.

FIG URE 5 . D otted D ecimal Notation



IP Address Classes

Class B	128.1.0.1 to 191.255.255.254	Supports 65,000 hosts on each of 16,000 netw
Class C	192.0.1.1 to 223.255.254.254	Supports 254 hosts on each of 2 million networ
Class D	224.0.0.0 to 239.255.255.255	Reserved for multicast groups.
Class E	240.0.0.0 to 254.255.255.254	Reserved for future use, or research and develc purposes.

Private Address Space

10.0.0.0 to 10.255.255.255

172.16.0.0 to 172.31.255.255

192.168.0.0 to 192.168.255.255

Default Subnet Masks

255.0.0.0 Class A

255.255.0.0 Class B

255.255.255.0 Class C

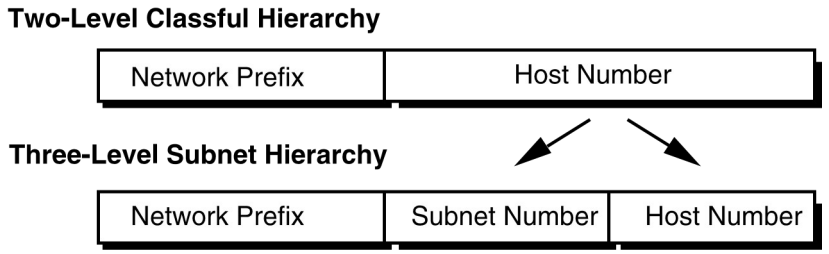
Subnetting

In 1985, RFC 950 defined a standard procedure to support the subnetting, or division, of a single Class A, B, or C network number into smaller pieces. Subnetting was introduced to overcome some of the problems that parts of the Internet were beginning to experience with the classful two-level addressing hierarchy, such as:

- Internet routing tables were beginning to grow.
- Local administrators had to request another network number from the Internet before a new network could be installed at their site.

Both of these problems were attacked by adding another level of hierarchy to the IP addressing structure. Instead of the classful two-level hierarchy, subnetting supports a three-level hierarchy. Figure 7 illustrates the basic idea of subnetting, which is to divide the standard classful host number field into two parts-the subnet number and the host number on that subnet.

FIG URE 7 . S ubnet A ddress H ierarchy

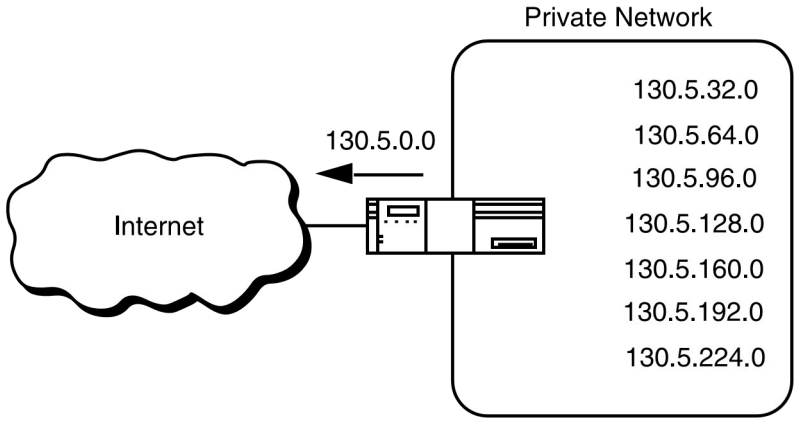


Subnetting attacked the expanding routing table problem by ensuring that the subnet structure of a network is never visible outside of the organization's private network. The route from the Internet to any subnet of a given IP address is the same, no matter which subnet the destination host is on. This 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.

Subnetting overcame the registered number issue by assigning each organization one (or at most a few) network numbers from the IPv4 address space. The organization was then free to assign a distinct

subnetwork number for each of its internal networks. This allowed the organization to deploy additional subnets without obtaining a new network number from the Internet.

FIG URE 8 . S ubnetting the Routing Requirements of the Internet



In Figure 8, a site with several logical networks uses subnet addressing 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 subnetworks based on the third octet of the classful address. The deployment of subnetting 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 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 (that is, the rapid changing of routes) within the private network does not affect the Internet routing table since Internet routers do not know about the reachability of the individual subnetsthey just know about the reachability of the parent network number.

Extended Network Prefix

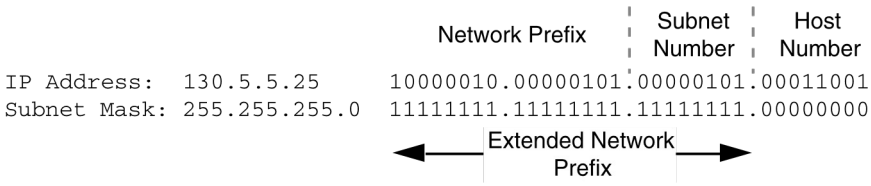
Internet routers use only the network prefix of the destination address to route traffic to a subnetted environment. Routers within the subnetted environment use the extended network prefix to route traffic between the individual subnets. The extended network prefix is composed of the classful network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

FIG URE 9 . Extended Netw ork Prefix

The extended network prefix has traditionally been identified by the subnet mask. For example, if an administrator has the /16 address of 130.5.0.0 and wants to use the entire third octet to represent the subnet number, the administrator must specify a subnet mask of 255.255.255.0.

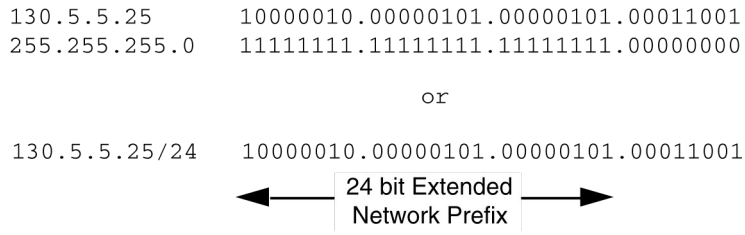
The bits in the subnet mask and the Internet address have a one to one correspondence. The bits of the subnet mask are set to 1 (one) if the system examining the address should treat the corresponding bit in the IP address as part of the extended network prefix. The bits in the mask are set to 0 (zero) if the system should treat the bit as part of the host number. This numbering is illustrated in Figure 10.

FIG URE 1 0 . S ubnet Mask

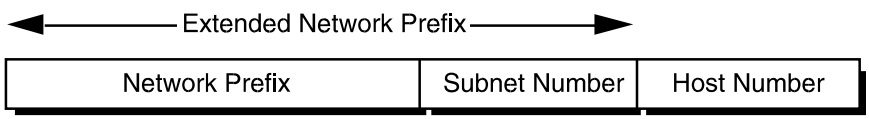


The standards describing modern routing protocols often refer to the extended network prefix length rather than the subnet mask. The prefix length is equal to the number of contiguous one-bits in the traditional subnet mask. This means that specifying the network address 130.5.5.25 with a subnet mask of 255.255.255.0 can also be expressed as 130.5.5.25/24. The /<prefix length> notation is more compact and easier to understand than writing out the mask in its traditional dotteddecimal format. This is illustrated in Figure 11.

FIG URE 1 1 . Extended Netw ork Prefix Leng th



Note that modern routing protocols still carry the subnet mask. None of the Internet standard routing protocols have a 1-byte field in the header



that contains the number of bits in the extended network prefix. Each routing protocol is still required to carry the complete four-octet subnet mask.

Subnet Design Considerations

The deployment of an addressing plan requires careful thought. Four key questions that must be answered before any design should be undertaken are:

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 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 an organization needs nine subnets, 23 (or 8) will not provide enough subnet addressing space, so the network administrator will need to round up to 24 (or 16).

The network administrator must always allow adequate room for growth. For example, although 14 subnets are required today, 16 subnets might not be enough in two years when the 17th subnet needs to be deployed. In this case, it would be wise to select 25 (or 32) as the maximum number of subnets.

The second step is to ensure that there are enough host addresses for the organization's largest subnet. If the largest subnet needs to support 50 host addresses today, 25 (or 32) will not provide enough host address space so the network administrator will need to round up to 26 (or 64).

The final step is to make sure that the organization's address allocation provides enough bits to deploy the required subnet addressing plan. For example, if the organization has a single /16, it could easily deploy 4 bits for the subnet number and 6 bits for the host number. However, if the organization has several /24s and it needs to deploy nine subnets, it may have to subnet each of its /24s into four subnets (using 2 bits) and then build the network by combining the subnets of three /24 network numbers.

An alternative solution would be to deploy network numbers from the private address space (RFC 1918) for internal connectivity and use a Network Address Translator (NAT) to provide external Internet access.

Subnet Example #1

Given

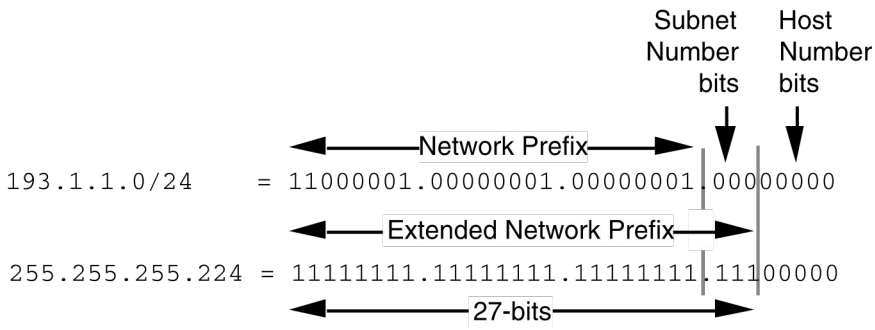
An organization is assigned the network number 193.1.1.0/24 and it needs to define six subnets. The largest subnet is required to support 25 hosts.

Defining the Subnet Mask / Extended Prefix Length

The first step in defining the subnet mask 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 (21), 4 (22), 8 (23), 16 (24), and so on]. 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 (23) and have two unused subnets that can be reserved for future growth.

Since 8 = 23, three bits are required to enumerate the eight subnets in the block. In this example, the organization is subnetting 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 notation is illustrated in Figure 12.

FIG URE 1 2 . Example # 1 -D efining the S ubnet Mask /Extended Prefix Leng th



A 27-bit extended network prefix leaves 5 bits to define host addresses on each subnet. This means that each subnetwork with a 27-bit prefix represents a contiguous block of 25 (32) individual IP addresses. However, since the all-0s and all-1s host addresses cannot be allocated, there are 30 (25-2) assignable host addresses on each subnet.

Defining the Subnet Numbers

The eight subnets will be numbered 0 through 7. Throughout the remainder of this paper, the XXX notation indicates the binary representation of the number. The 3-bit binary representation of the decimal values 0 through 7 are: 0 (000), 1 (001), 2 (010), 3 (011), 4 (100), 5 (101), 6 (110), and 7 (111).

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 (110) into the 3 bits of the subnet number field.
The eight subnet numbers for this example are listed in the following code sample. 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

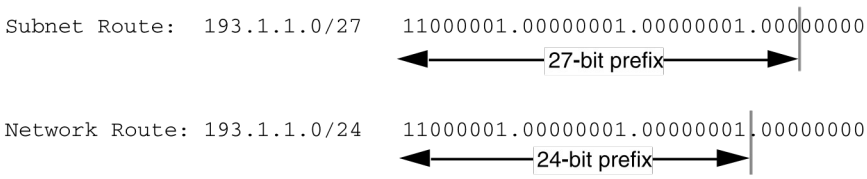
An easy way to verify that the subnets are correct is to ensure that they are all multiples of the Subnet #1 address. In this example, all subnets are multiples of 32: 0, 32, 64, 96, and so on.

The All-0s Subnet and All-1s Subnet

When subnetting was first defined in RFC 950, it prohibited the use of the all-0s and the all-1s subnets. The reason for this restriction was to eliminate situations that could potentially confuse a classful router. Today a router can be both classless and classful at the same time-it could be running RIP-1 (classful protocol) and BGP-4 (Border Gateway Protocol Version 4-a classless protocol) at the same time.

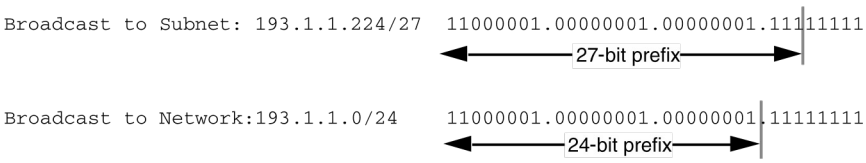
With respect to the all-0s subnet, a router requires that each routing table update include the route/<prefix length> pair to differentiate between a route to the all-0s subnet and a route to the entire network. For example, when using RIP-1which does not supply a mask or prefix length with each route, the routing advertisements for subnet 193.1.1.0/27 and for network 193.1.1.0/24 are identical-193.1.1.0. Without somehow knowing the prefix length or mask, a router cannot tell the difference between a route to the all-0s subnet and the route to the entire network. This example is illustrated in Figure 13.

FIG URE 1 3 . D ifferentiating Betw een a Route to the A ll-0 s S ubnet and the Entire Netw ork



Regarding the all-1s subnet, a router requires that each routing table entry include the prefix length so that it can determine whether a broadcast (directed or all-subnets) should be sent only to the all-1s subnet or to the entire network. For example, when the routing table does not contain a mask or prefix length for each route, confusion can occur because the same broadcast address (193.1.1.255) is used for both the entire network 193.1.1.0/24 and the all-1s subnet 193.1.1.224/27. This issue is illustrated in Figure 14.

FIG URE 1 4 . Identifying a Broadcast to the A ll 1 s S ubnet and the Entire Netw ork



Defining Host Addresses for Each Subnet

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

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 (25 -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 (011112) into the 5-bits of Subnet #2's host number field.
The valid host addresses for Subnet #2 in this example are listed in the following sample code. The underlined portion of each address identifies the extended network prefix, while the bold digits identify the 5bit 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 listed in the following sample code. 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-1s 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-1s 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 is assigned the network number 140.25.0.0/16 and it must 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 addresses 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 (26-2) host addresses. However, this choice would only provide two unused host addresses on each subnet for future growth, which is not likely to support additional growth. The network administrator must define a block of 126 (27-2) host addresses with 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 notation is illustrated in Figure 15.

FIG URE 1 5 . Example # 2 -D efining the S ubnet Mask /Extended Prefix Leng th

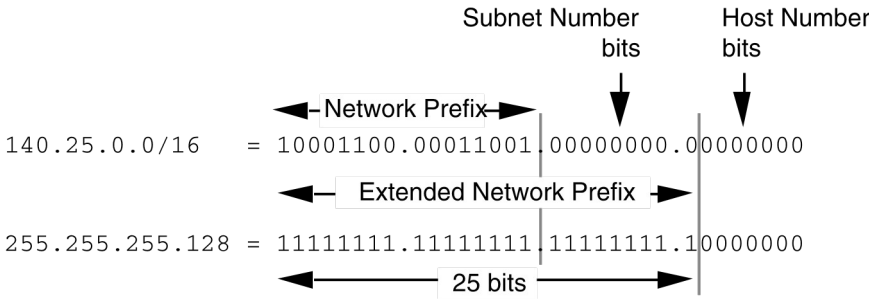


Figure 15 shows that the 25-bit extended prefix assigns 9 bits to the subnet number field. Since 29 = 512, nine bits allow the definition of 512 subnets. Depending on the organization's requirements, the network administrator could have elected to assign additional bits to the host number field (allowing 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 illustrates what happens to the dotted- decimal representation of a subnet address when the subnet number bits extend across an octet boundary. Note that the same type of confusion can 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 (0000000002), 1 (0000000012), 2 (0000000102), 3 (0000000112), ..., 511 (1111111112). To define Subnet #3, the network administrator places the binary representation of 3 (0000000112) into the 9 bits of the subnet number field. The 512 subnet numbers for this example are listed in the

following sample code. 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

Note that the sequential subnet numbers are not sequential when expressed in dotted-decimal notation. This can be confusing to people who expect dotted-decimal notation to make IP addressing easier. In this example, the dotted-decimal notation obscures the subnet numbering scheme.

Defining Host Addresses for Each Subnet

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

The valid host addresses for Subnet #3 are listed in the following sample code. 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-1s host address or:

10001100.00011001.00000001.1 **1111111** = 140.25.1.255

The broadcast address for Subnet #3 is exactly one less than the base address for Subnet #4 (140.25.2.0).

Network Address 192.10.10.0

Address class C

Default subnet mask 255 . 255 . 255 . 0

Custom subnet mask 255 . 255 . 255 . 240

Total number of subnets 16

Total number of host addresses 16

Number of usable addresses 14

Number of bits borrowed 4

Show your work for Problem 1 in the space below.

	256	128	64	32	16	8	4	2	– Number of Hosts
Number of Subnets –	2	4	8	16	32	64	128	256	
	128	64	32	16	8	4	2	1	– Binary values
192 . 10 . 10 .	0	0	0	0	0	0	0	0	

Add the binary value numbers to the left of the line to create the custom subnet mask.

128
64
32
+16
240

16 Observe the total number of hosts.
−2 Subtract 2 for the number of usable hosts.
14

Address class D

Default subnet mask 255 . 255 . 0 . 0

Custom subnet mask 255 . 255 . 255 . 192

Total number of subnets 1,024

Total number of host addresses 64

Number of usable addresses 62

Number of bits borrowed 10

Show your work for <u>Problem 2</u> in the space below.

	65,536	32,768	16,384	8,192	4,096	2,048	1,024	512	. 256	128	64	32	16	8	4	2
Number of Hosts =																65,536
Number of Subnets =	2	4	8	16	32	64	128	256.	512	1024	2048	4,096	8,192	16,384	32,768	
Binary values =	128	64	32	16	8	4	2	1 ,	128	64	32	16	8	4	2	1
165 . 100 .	0	0	0	0	0	0	0	0 .	0	0	0	0	0	0	0	0

128 128
64 +64
32 192
16
8
4
2
+1
255

Add the binary value numbers to the left of the line to create the custom subnet mask.

64 Observe the total number of hosts.
-2 Subtract 2 for the number of usable hosts.
62