Two-Level Addressing

The whole purpose of IPv4 addressing is to define a destination for an Internet packet (at the network layer). When classful addressing was designed, it was assumed that the whole Internet is divided into many networks and each network connects many hosts. In other words, the Internet was seen as a network of networks. A network was normally created by an organization that wanted to be connected to the Internet. The Internet authorities allocated a block of addresses to the organization (in class A, B, or C).

> The range of addresses allocated to an organization in classful addressing was a block of addresses in Class A, B, or C.

Since all addresses in a network belonged to a single block, each address in classful addressing contains two parts: netid and hostid. The netid defines the network; the hostid defines a particular host connected to that network. Figure 11 shows an IPv4 address in classful addressing. If n bits in the class defines the net, then 32 – n bits defines the host. However, the value of n depends on the class the block belongs to. The value of n can be 8, 16 or 24 corresponding to classes A, B, and C respectively.

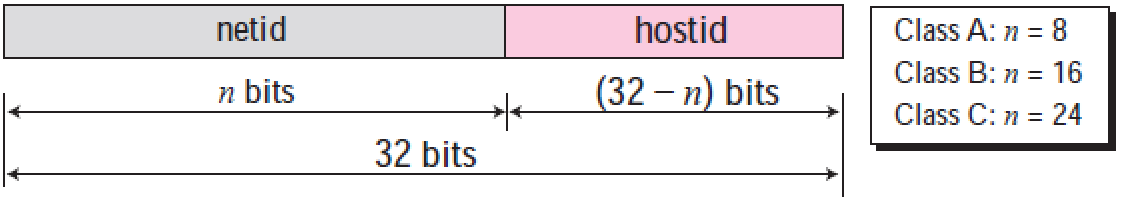


Figure 11 *Two-level addressing in classful addressing*

Example 1

Two-level addressing can be found in other communication systems. For example, a telephone system inside the United States can be thought of as two parts: area code and local part. The area code defines the area, the local part defines a particular telephone subscriber in that area.

(626) 3581301

The area code, 626, can be compared with the netid, the local part, 3581301, can be compared to the hostid.

Extracting Information in a Block

A block is a range of addresses. Given any address in the block, we normally like to know three pieces of information about the block: the number of addresses, the first address, and the last address. Before we can extract these pieces of information, we need to know the class of the address, which we showed how to find in the previous section. After the class of the block is found, we know the value of n, the length of netid in bits. We can now find these three pieces of information as shown in Figure 12.

1. The number of addresses in the block, N, can be found using N = 232- n.
2. To find the first address, we keep the n leftmost bits and set the (32 - n) rightmost bits all to 0s.
3. To find the last address, we keep the n leftmost bits and set the (32 - n) rightmost bits all to 1s.

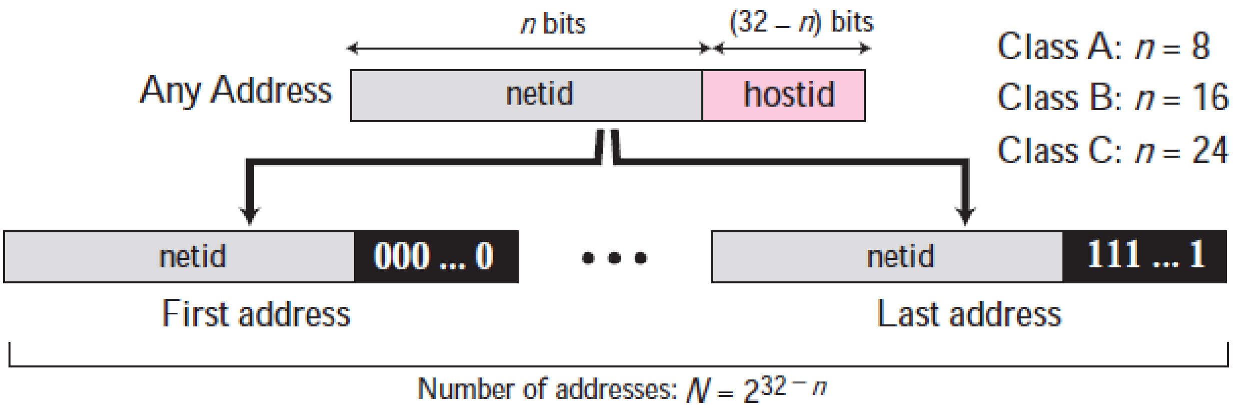


Figure 12 *Information extraction in classful addressing*

Example 2

An address in a block is given as 73.22.17.25. Find the number of addresses in the block, the first address, and the last address.

Solution

Since 73 is between 0 and 127, the class of the address is A. The value of *n* for class A is 8. Figure 13 shows a possible configuration of the network that uses this block. Note that we show the value of *n* in the network address after a slash. Although this was not a common practice in classful addressing, it helps to make it a habit in classless addressing in the next section.

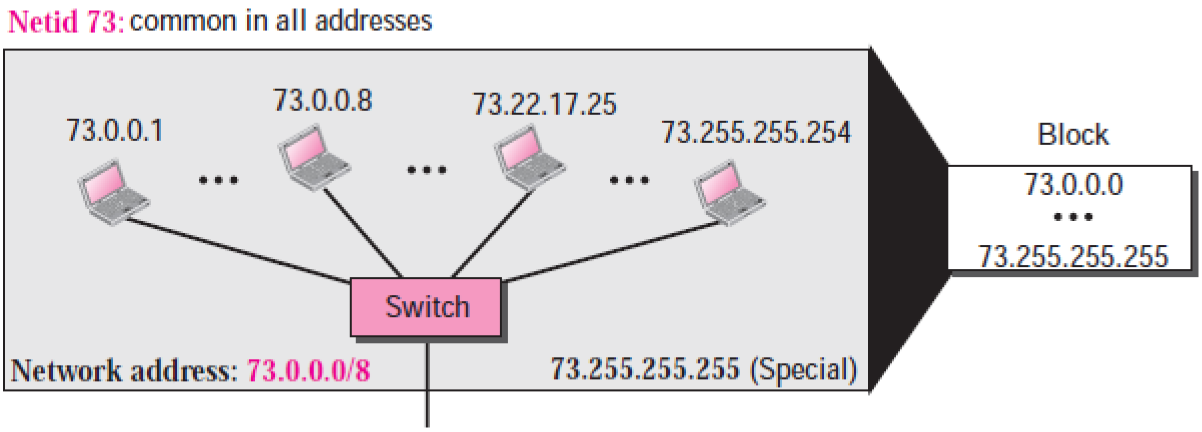


Figure 13 *Solution to Example 2*

1. The number of addresses in this block is $N = 2^{32-n} = 2^{24} = 16,777,216$.

2. To find the first address, we keep the leftmost 8 bits and set the rightmost 24 bits all to 0s. The first address is 73.0.0.0/8 in which 8 is the value of *n*. The first address is called the *network address* and is not assigned to any host. It is used to define the network.

3. To find the last address, we keep the leftmost 8 bits and set the rightmost 24 bits all to 1s. The last address is 73.255.255.255. The last address is normally used for a special purpose, as discussed later in the chapter.

Example 3

An address in a block is given as 180.8.17.9. Find the number of addresses in the block, the first address, and the last address.

Solution

Since 180 is between 128 and 191, the class of the address is B. The value of *n* for class B is 16. Figure 14 shows a possible configuration of the network that uses this block.

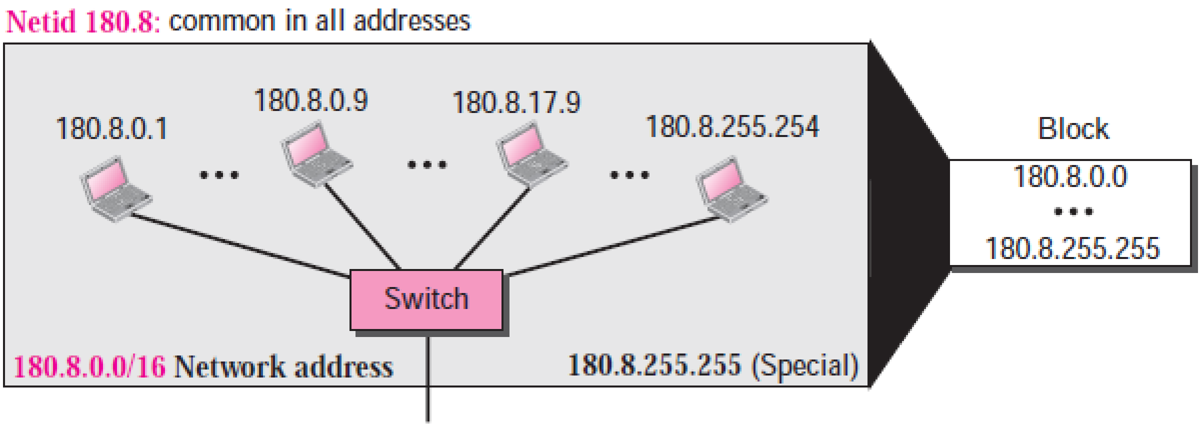


Figure 14 *Solution to Example 3*

1. The number of addresses in this block is $N = 2^{32-n} = 2^{16} = 65{,}536$.

2. To find the first address, we keep the leftmost 16 bits and set the rightmost 16 bits all to 0s. The first address (network address) is 18.8.0.0/16, in which 16 is the value of *n*.

3. To find the last address, we keep the leftmost 16 bits and set the rightmost 16 bits all to 1s. The last address is 18.8.255.255.

Example 4

An address in a block is given as 200.11.8.45. Find the number of addresses in the block, the first address, and the last address.

Solution

Since 200 is between 192 and 223, the class of the address is C. The value of *n* for class C is 24. Figure 5.18 shows a possible configuration of the network that uses this block.

1. The number of addresses in this block is $N=2^{32-n} = 2^8 = 256$.

2. To find the first address, we keep the leftmost 24 bits and set the rightmost 8 bits all to 0s. The first address is 200.11.8.0/24. The first address is called the network address.

3. To find the last address, we keep the leftmost 24 bits and set the rightmost 8 bits all to 1s. The last address is 200.11.8.255.

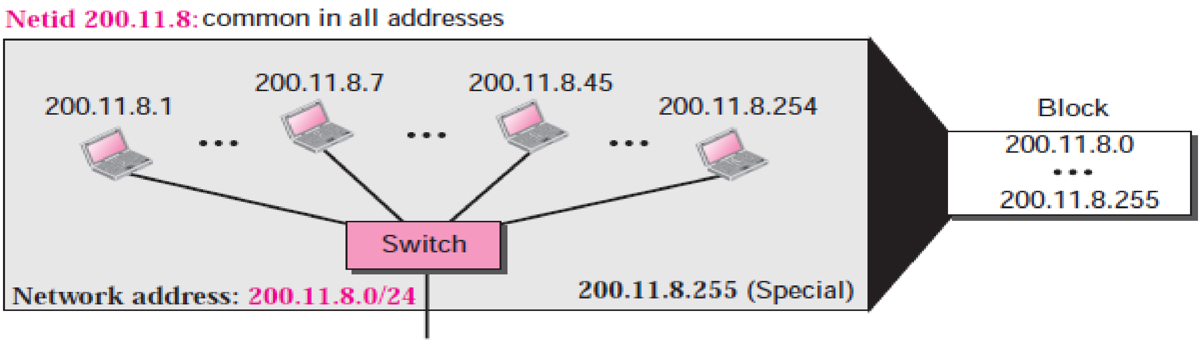


Figure 15 *Solution to Example 4*

An Example

Figure 16 shows a hypothetical part of an internet with three networks.

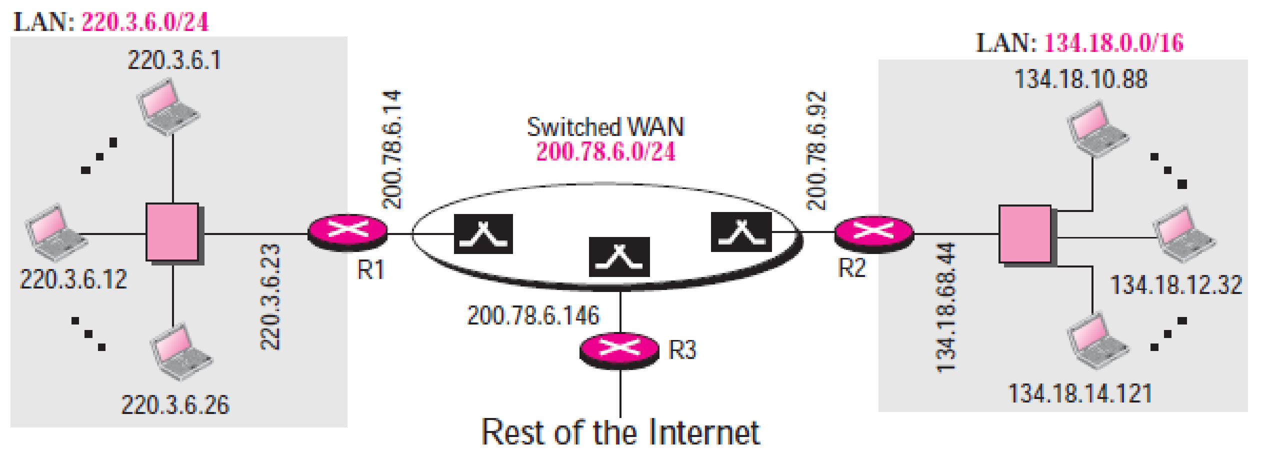


Figure 16 *Sample internet*

We have

1. A LAN with the network address 220.3.6.0 (class C).
2. A LAN with the network address 134.18.0.0 (class B).
3. A switched WAN (class C), such as Frame Relay or ATM, that can be connected to many routers. We have shown three. One router connects the WAN to the left LAN, one connects the WAN to the right LAN, and one connects the WAN to the rest of the internet.

Network Address

The above three examples show that, given any address, we can find all information about the block. The first address, **network address,** is particularly important because it is used in routing a packet to its destination network. For the moment, let us assume that an internet is made of *m* networks and a router with *m* interfaces. When a packet arrives at the router from any source host, the router needs to know to which network the packet should be sent; the router needs to know from which interface the packet should be sent out. When the packet arrives at the network, it reaches its destination host using another strategy that we discuss in later chapters. Figure 17 shows the idea. After the network address has been found, the router consults its routing table to find the corresponding interface from which the packet should be sent out. The network address is actually the identifier of the network; each network is identified by its network address.

> The network address is the identifier of a network.

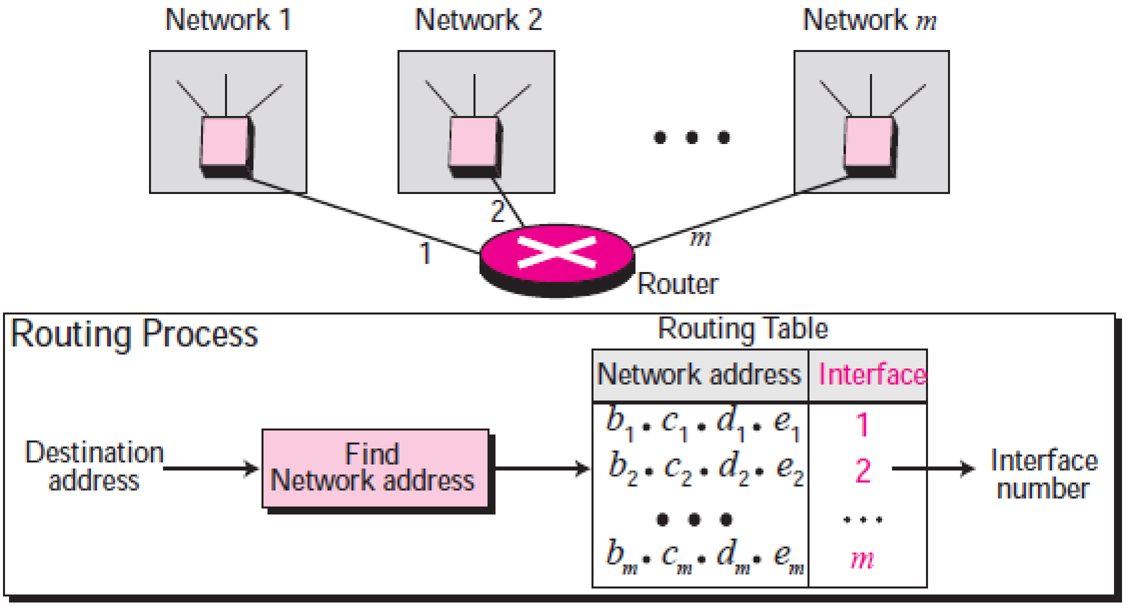


Figure 17 *Network address*

Network Mask

The methods we described previously for extracting the network address are mostly used to show the concept. The routers in the Internet normally use an algorithm to extract the network address from the destination address of a packet. To do this, we need a network mask. A **network mask** or a **default mask** in classful addressing is a 32-bit number with n leftmost bits all set to 1s and (32 - n) rightmost bits all set to 0s. Since n is different for each class in classful addressing, we have three default masks in classful addressing as shown in Figure 18.

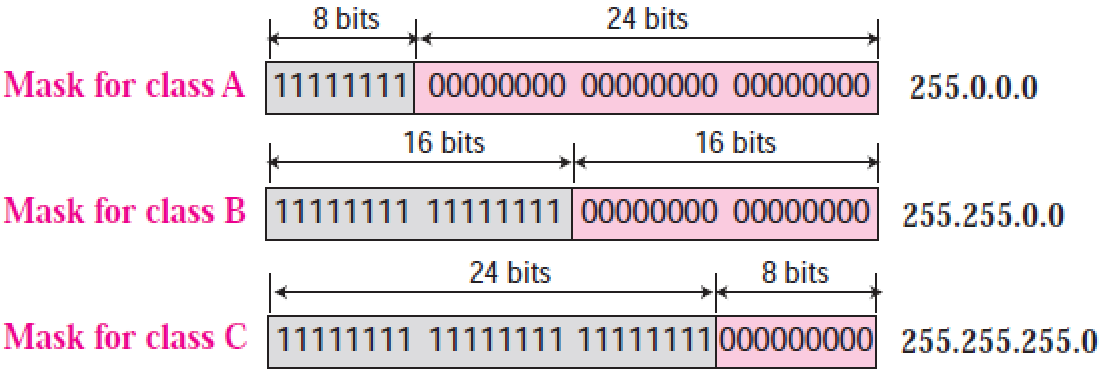


Figure 18 *Network mask*

To extract the network address from the destination address of a packet, a router uses the AND operation described in the previous section. When the destination address (or any address in the block) is ANDed with the default mask, the result is the network address (Figure 19). The router applies the AND operation on the binary (or hexadecimal representation) of the address and the mask, but when we show an example, we use the short cut discussed before and apply the mask on the dotted-decimal notation. The default mask can also be used to find the number of addresses in the block and the last address in the block, but we discuss these applications in classless addressing.

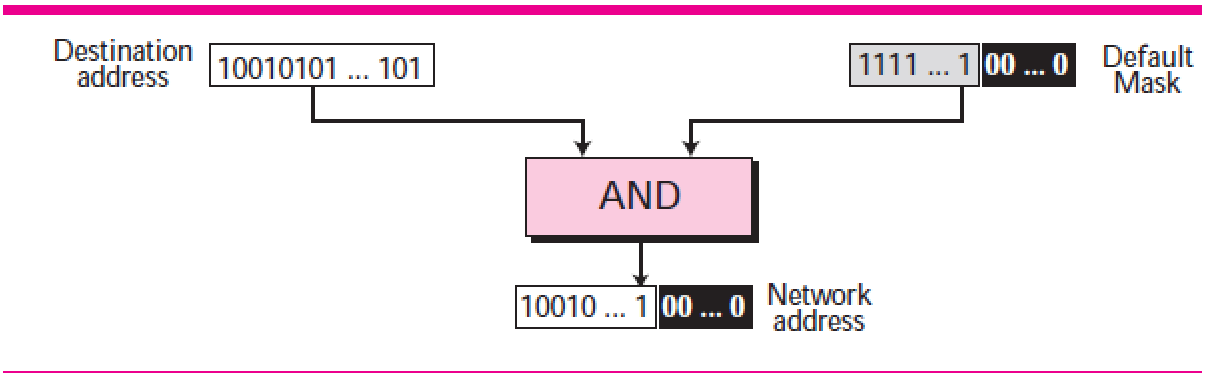


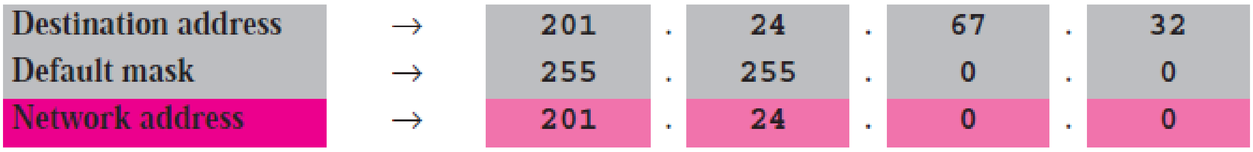
Figure 19 *Finding a network address using the default mask*

Example 5

A router receives a packet with the destination address 201.24.67.32. Show how the router finds the network address of the packet.

Solution

We assume that the router first finds the class of the address and then uses the corresponding default mask on the destination address, but we need to know that a router uses another strategy as we will discuss in the next chapter. Since the class of the address is B, we assume that the router applies the default mask for class B, 255.255.0.0 to find the network address.



Destination address	→	201	.	24	.	67	.	32
Default mask	→	255	.	255	.	0	.	0
Network address	→	201	.	24	.	0	.	0

We have used the first short cut as described in the previous section. The network address is 201.24.0.0 as expected.

Three-Level Addressing: Subnetting

As we discussed before, the IP addresses were originally designed with two levels of addressing. To reach a host on the Internet, we must first reach the network and then the host. It soon became clear that we need more than two hierarchical levels, for two reasons. First, an organization that was granted a block in class A or B needed to divide its large network into several subnetworks for better security and management. Second, since the blocks in class A and B were almost depleted and the blocks in class C were smaller than the needs of most organizations, an organization that has been granted a block in class A or B could divide the block into smaller subblocks and share them with other organizations. The idea of splitting a block to smaller blocks is referred to as subnetting. In **subnetting,** a network is divided into several smaller subnetworks (subnets) with each subnetwork having its own subnetwork address.

Example 6

Three-level addressing can be found in the telephone system if we think about the local part of a telephone number as an exchange and a subscriber connection:

(626) 358 - 1301

in which 626 is the area code, 358 is the exchange, and 1301 is the subscriber connection.

Example 7

Figure 20 shows a network using class B addresses before subnetting. We have just one network with almost 216 hosts. The whole network is connected, through one single connection, to one of the routers in the Internet. Note that we have shown /16 to show the length of the netid (class B).

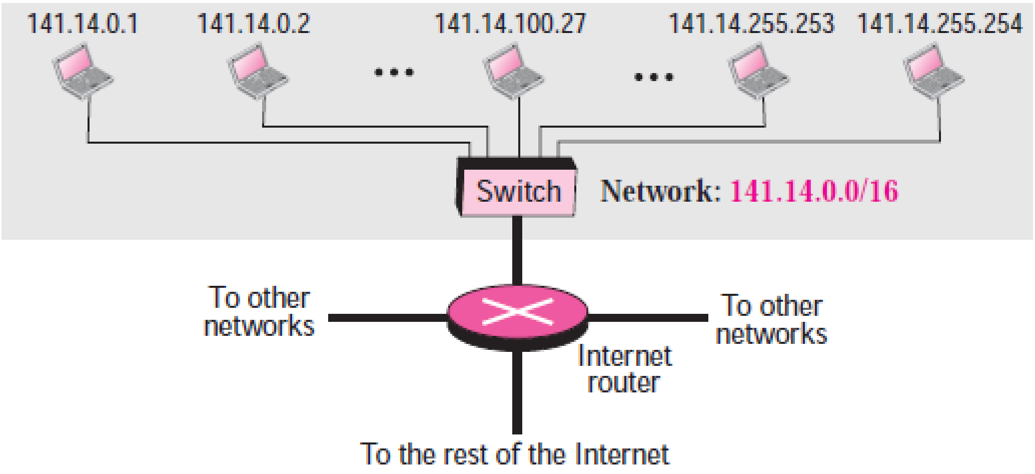


Figure 20 *Example 7*

Example 8

Figure 21 shows the same network in Figure 20 after subnetting. The whole network is still connected to the Internet through the same router. However, the network has used a private router to divide the network into four subnetworks. The rest of the Internet still sees only one network; internally the network is made of four subnetworks. Each subnetwork can now have almost 214 hosts. The network can belong to a university campus with four different schools (buildings). After subnetting, each school has its own subnetworks, but still the whole campus is one network for the rest of the Internet. Note that /16 and /18 show the length of the netid and subnetids.

Subnet Mask

We discussed the network mask (default mask) before. The network mask is used when a network is not subnetted. When we divide a network to several subnetworks, we need to create a subnetwork mask (or subnet mask) for each subnetwork. A subnetwork has subnetid and hostid as shown in Figure 22.

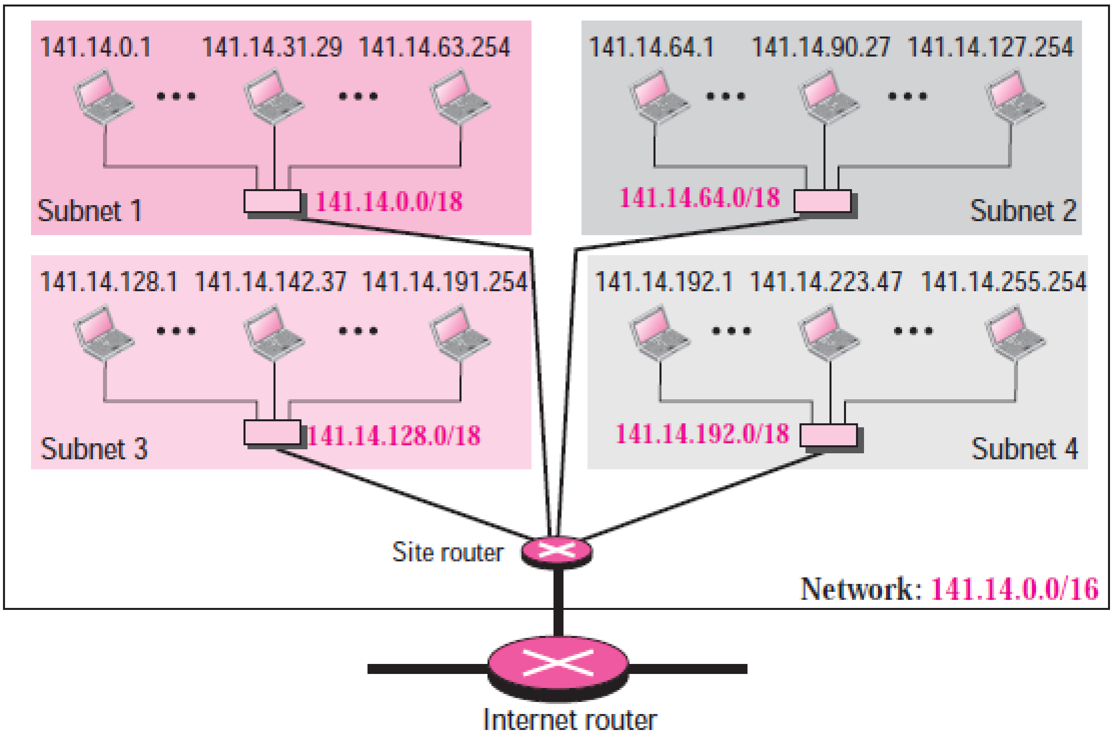


Figure 21 *Example 8*

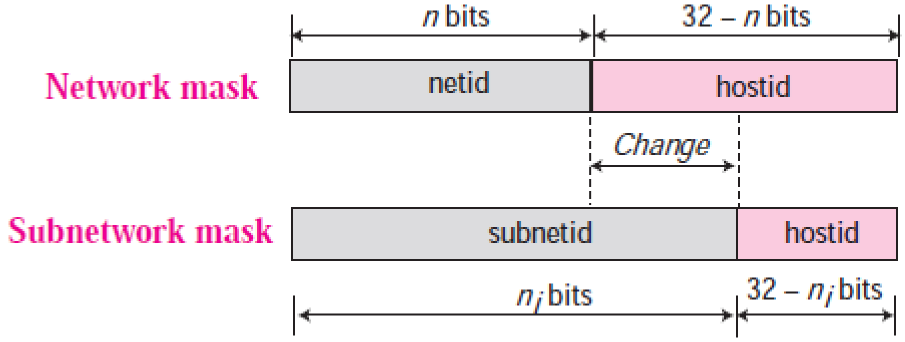


Figure 22 *Network mask and subnetwork mask*

Subnetting increases the length of the netid and decreases the length of hostid. When we divide a network to *s* number of subnetworks, each of equal numbers of hosts, we can calculate the subnetid for each subnetwork as

$$n_{\text{sub}} = n + \log_2 s$$

in which *n* is the length of netid, n_{sub} is the length of each subnetid, and *s* is the number of subnets which must be a power of 2.

Example 9

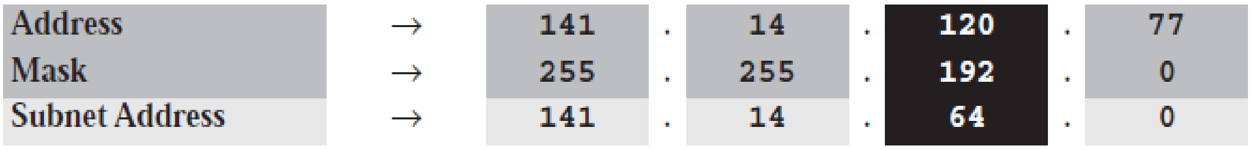
In Example 8, we divided a class B network into four subnetworks. The value of *n*= 16 and the value of *n*1= *n*2= *n*3= *n*4= 16 + $\log_2 4$ = 18. This means that the subnet mask has eighteen 1s and fourteen 0s. In other words, the subnet mask is 255.255.192.0 which is different from the network mask for class B (255.255.0.0).

Subnet Address

When a network is subnetted, the first address in the subnet is the identifier of the subnet and is used by the router to route the packets destined for that subnetwork. Given any address in the subnet, the router can find the subnet mask using the same procedure we discussed to find the network mask: ANDing the given address with the subnet mask. The short cuts we discussed in the previous section can be used to find the subnet address.

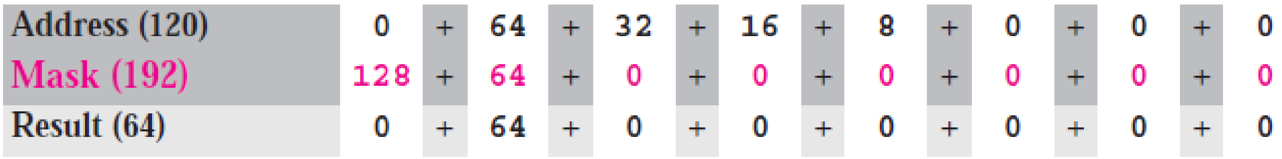
Example 10

In Example 8, we show that a network is divided into four subnets. Since one of the addresses in subnet 2 is 141.14.120.77, we can find the subnet address as:



Address	→	141	.	14	.	120	.	77
Mask	→	255	.	255	.	192	.	0
Subnet Address	→	141	.	14	.	64	.	0

The values of the first, second, and fourth bytes are calculated using the first short cut for AND operation. The value of the third byte is calculated using the second short cut for the AND operation.



Address (120)	0	+	64	+	32	+	16	+	8	+	0	+	0	+	0
Mask (192)	128	+	64	+	0	+	0	+	0	+	0	+	0	+	0
Result (64)	0	+	64	+	0	+	0	+	0	+	0	+	0	+	0

Designing Subnets

We show how to design a subnet when we discuss classless addressing. Since classful addressing is a special case of classless addressing, what is discussed later can also be applied to classful addressing.

Supernetting

Subnetting could not completely solve address depletion problems in classful addressing because most organizations did not want to share their granted blocks with others. Since class C blocks were still available but the size of the block did not meet the requirement of new organizations that wanted to join the Internet, one solution was **supernetting.** In supernetting, an organization can combine several class C blocks to create a larger range of addresses. In other words, several networks are combined to create a supernetwork. By doing this, an organization can apply for several class C blocks instead of just one. For example, an organization that needs 1000 addresses can be granted four class C blocks.

Supernet Mask

A **supernet mask** is the reverse of a subnet mask. A subnet mask for class C has more 1s than the default mask for this class. A supernet mask for class C has less 1s than the default mask for this class. Figure 23 shows the difference between a subnet mask and a supernet mask. A subnet mask that divides a block into eight subblocks has three more 1s ($2^3 = 8$) than the default mask; a supernet mask that combines eight blocks into one superblock has three less 1s than the default mask.

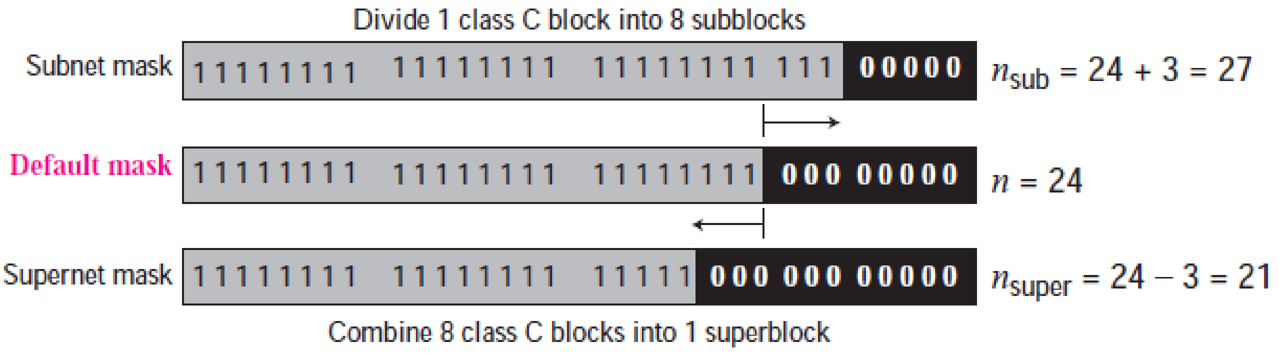


Figure 23 *Comparison of subnet, default, and supernet masks*

In supernetting, the number of class C addresses that can be combined to make a supernet needs to be a power of 2. The length of the supernetid can be found using the formula

$$n_{super} = n - \log_2 c$$

in which n_{super} defines the length of the supernetid in bits and c defines the number of class C blocks that are combined. Unfortunately, supernetting provided two new problems: First, the number of blocks to combine needs to be a power of 2, which means an organization that needed seven blocks should be granted at least eight blocks (address wasting). Second, supernetting and subnetting really complicated the routing of packets in the Internet.