CLASSLESS ADDRESSING

Subnetting and supernetting in classful addressing did not really solve the address depletion problem and made the distribution of addresses and the routing process more difficult. With the growth of the Internet, it was clear that a larger address space was needed as a long-term solution. The larger address space, however, requires that the length of IP addresses to be increased, which means the format of the IP packets needs to be changed. Although the long-range solution has already been devised and is called IPv6 (see Chapters 26 to 28), a short-term solution was also devised to use the same address space but to change the distribution of addresses to provide a fair share to each organization. The short-term solution still uses IPv4 addresses, but it is called **classless addressing**.

There was another motivation for classless addressing. During the 1990s, Internet service providers (ISPs) came into prominence. An ISP is an organization that provides Internet access for individuals, small businesses, and midsize organizations that do not want to create an Internet site and become involved in providing Internet services (such as e-mail services) for their employees. An ISP can provide these services. An ISP is granted a large range of addresses and then subdivides the addresses (in groups of 1, 2, 4, 8, 16, and so on), giving a range of addresses to a household or a small business. The customers are connected via a dial-up modem, DSL, or cable modem to the ISP. However, each customer needs some IPv4 addresses. In classless addressing, variable-length blocks are used that belong to no classes. We can have a block of 1 address, 2 addresses, 4 addresses, 128 addresses, and so on.

Variable-Length Blocks

In classless addressing, the whole address space is divided into variable length blocks. Theoretically, we can have a block of 2^0, 2^1, 2^2, . . . , 2^{32} addresses. The only restriction, as we discuss later, is that the number of addresses in a block needs to be a power of 2. An organization can be granted one block of addresses. Figure 5.27 shows the division of the whole address space into nonoverlapping blocks.

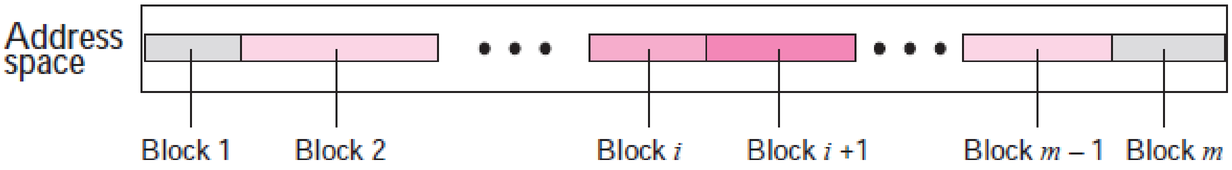


Figure 5.27 *Variable-length blocks in classless addressing*

Two-Level Addressing

In classful addressing, two-level addressing was provided by dividing an address into *netid* and *hostid*. The netid defined the network; the hostid defined the host in the network. The same idea can be applied in classless addressing. When an organization is granted a block of addresses, the block is actually divided into two parts, the **prefix** and the **suffix.** The prefix plays the same role as the netid; the suffix plays the same role as the hostid. All addresses in the block have the same prefix; each address has a different suffix. Figure 5.28 shows the prefix and suffix in a classless block.

> **In classless addressing, the prefix defines the network and the suffix defines the host.**

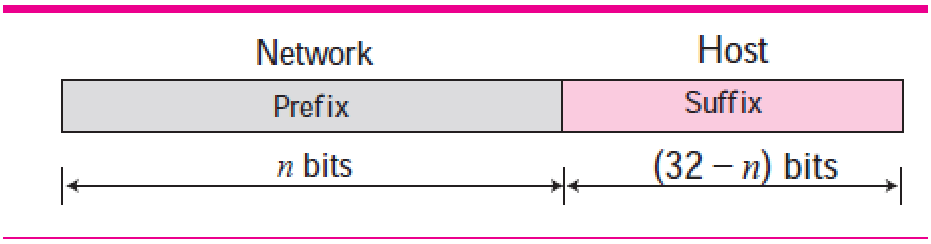


Figure 5.28 *Prefix and suffix*

In classful addressing, the length of the netid, *n*, depends on the class of the address; it can be only 8, 16, or 24. In classless addressing, the length of the prefix, *n,* depends on the size of the block; it can be 0, 1, 2, 3, . . . , 32. In classless addressing, the value of *n* is referred to as **prefix length;** the value of 32 - *n* is referred to as **suffix length.**

The prefix length in classless addressing can be 1 to 32.

Example 5.22

What is the prefix length and suffix length if the whole Internet is considered as one single block with 4,294,967,296 addresses?

Solution

In this case, the prefix length is 0 and the suffix length is 32. All 32 bits vary to define 2^{32} = 4,294,967,296 hosts in this single block.

Example 5.23

What is the prefix length and suffix length if the Internet is divided into 4,294,967,296 blocks and each block has one single address?

Solution

In this case, the prefix length for each block is 32 and the suffix length is 0. All 32 bits are needed to define 232 = 4,294,967,296 blocks. The only address in each block is defined by the block itself.

Example 5.24

The number of addresses in a block is inversely related to the value of the prefix length, *n*. A small *n* means a larger block; a large *n* means a small block.

Slash Notation

The netid length in classful addressing or the prefix length in classless addressing play a very important role when we need to extract the information about the block from a given address in the block. However, there is a difference here in classful and classless addressing.

❑ In classful addressing, the netid length is inherent in the address. Given an address, we know the class of the address that allows us to find the netid length (8, 16, or 24).

❑ In classless addressing, the prefix length cannot be found if we are given only an address in the block. The given address can belong to a block with any prefix length.

In classless addressing, we need to include the prefix length to each address if we need to find the block of the address. In this case, the prefix length, *n*, is added to the address separated by a slash. The notation is informally referred to as **slash notation.** An address in classless addressing can then be represented as shown in Figure 5.29.

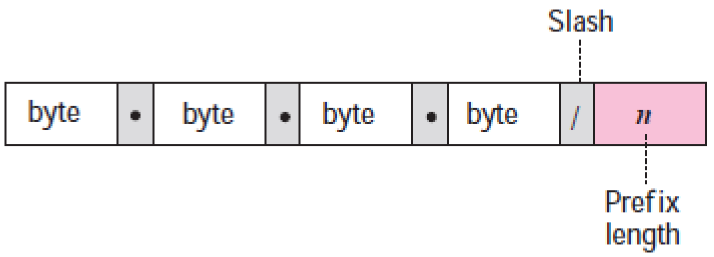


Figure 5.29 *Slash notation*

The slash notation is formally referred to as **classless interdomain routing** or **CIDR** (pronounced cider) notation.

> In classless addressing, we need to know one of the addresses in the block and the prefix length to define the block.

Network Mask

The idea of network mask in classless addressing is the same as the one in classful addressing. A network mask is a 32-bit number with the *n* leftmost bits all set to 0s and the rest of the bits all set to 1s.

Example 5.26

The following addresses are defined using slash notations.

a. In the address 12.23.24.78/8, the network mask is 255.0.0.0. The mask has eight 1s and twenty-four 0s. The prefix length is 8; the suffix length is 24.

b. In the address 130.11.232.156/16, the network mask is 255.255.0.0. The mask has sixteen 1s and sixteen 0s.The prefix length is 16; the suffix length is 16.

c. In the address 167.199.170.82/27, the network mask is 255.255.255.224. The mask has twenty-seven 1s and five 0s. The prefix length is 27; the suffix length is 5.

Extracting Block Information

An address in slash notation (CIDR) contains all information we need about the block: the first address (network address), the number of addresses, and the last address. These three pieces of information can be found as follows:

❑ The number of addresses in the block can be found as: $\boldsymbol{N} = \mathbf{2}^{32-n}$ in which *n* is the prefix length and *N* is the number of addresses in the block.

❑ The first address (network address) in the block can be found by ANDing the address with the network mask:

First address = **(any address) AND (network mask)**

Alternatively, we can keep the *n* leftmost bits of any address in the block and set the 32- *n* bits to 0s to find the first address.

❑ The last address in the block can be found by either adding the first address with the number of addresses or, directly, by ORing the address with the complement (NOTing) of the network mask:

Last address = **(any address) OR [NOT (network mask)]**

Alternatively, we can keep the *n* leftmost bits of any address in the block and set the 32 - *n* bits to 1s to find the last address.

Example 5.27

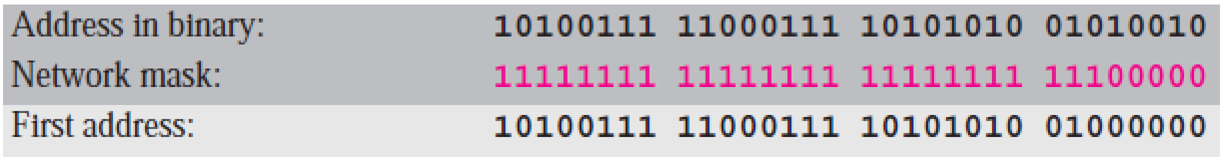
One of the addresses in a block is 167.199.170.82/27. Find the number of addresses in the network, the first address, and the last address.

Solution

The value of *n* is 27. The network mask has twenty-seven 1s and five 0s. It is 255.255.255.240.

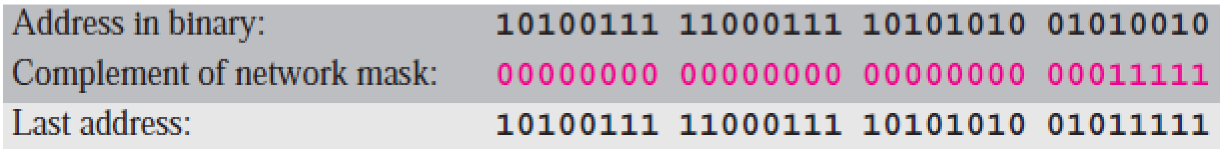
a. The number of addresses in the network is 232 − *n* = 232 − *n* = 25 = 32.

b. We use the AND operation to find the first address (network address).The first address is 167.199.170.64/27.



Address in binary:	10100111	11000111	10101010	01010010
Network mask:	11111111	11111111	11111111	11100000
First address:	10100111	11000111	10101010	01000000

c. To find the last address, we first find the complement of the network mask and then OR it with the given address: The last address is 167.199.170.95/27.



Address in binary:	10100111	11000111	10101010	01010010
Complement of network mask:	00000000	00000000	00000000	00011111
Last address:	10100111	11000111	10101010	01011111

Example 5.28

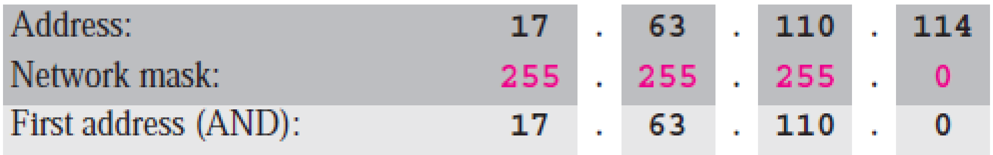
One of the addresses in a block is 17.63.110.114/24. Find the number of addresses, the first address, and the last address in the block.

Solution

The network mask is 255.255.255.0.

a. The number of addresses in the network is $2^{32 - 24}$ = 256.

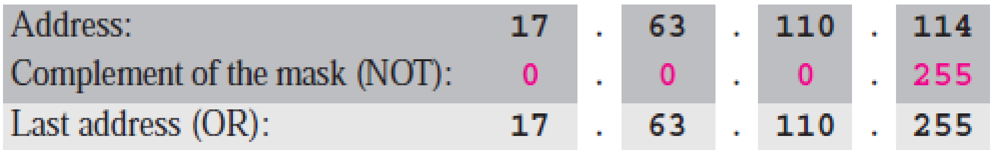
b. To find the first address, we use the short cut methods discussed early in the chapter.



Address:	17	.	63	.	110	.	114
Network mask:	255	.	255	.	255	.	0
First address (AND):	17	.	63	.	110	.	0

The first address is 17.63.110.0/24.

c. To find the last address, we use the complement of the network mask and the first short cut method we discussed before. The last address is 17.63.110.255/24.



Address:	17	.	63	.	110	.	114
Complement of the mask (NOT):	0	.	0	.	0	.	255
Last address (OR):	17	.	63	.	110	.	255

Example 5.29

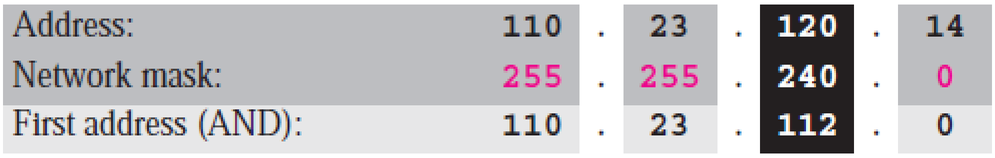
One of the addresses in a block is 110.23.120.14/20. Find the number of addresses, the first address, and the last address in the block.

Solution

The network mask is 255.255.240.0.

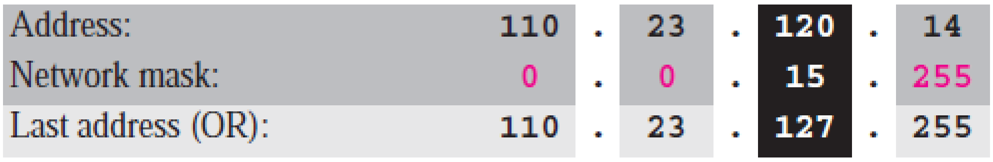
a. The number of addresses in the network is $2^{32 -20}$ = 4096.

b. To find the first address, we apply the first short cut to bytes 1, 2, and 4 and the second short cut to byte 3. The first address is 110.23.112.0/20.



Address:	110	.	23	.	120	.	14
Network mask:	255	.	255	.	240	.	0
First address (AND):	110	.	23	.	112	.	0

c. To find the last address, we apply the first short cut to bytes 1, 2, and 4 and the second short cut to byte 3. The OR operation is applied to the complement of the mask. The last address is 110.23.127.255/20.



Address:	110	.	23	.	120	.	14
Network mask:	0	.	0	.	15	.	255
Last address (OR):	110	.	23	.	127	.	255

Block Allocation

The next issue in classless addressing is block allocation. How are the blocks allocated? The ultimate responsibility of block allocation is given to a global authority called the Internet Corporation for Assigned Names and Addresses (ICANN). However, ICANN does not normally allocate addresses to individual Internet users. It assigns a large block of addresses to an ISP (or a larger organization that is considered an ISP in this case). For the proper operation of the CIDR, three restrictions need to be applied to the allocated block.

1. The number of requested addresses, *N*, needs to be a power of 2. This is needed to provide an integer value for the prefix length, *n* (see the second restriction). The number of addresses can be 1, 2, 4, 8, 16, and so on.

2. The value of prefix length can be found from the number of addresses in the block. Since $N = 2^{32 - n}$, then $n = \log_2 (2^{32}/N) = 32 - \log_2 N$. That is the reason why *N* needs to be a power of 2.

3. The requested block needs to be allocated where there are a contiguous number of unallocated addresses in the address space. However, there is a restriction on choosing the beginning addresses of the block. The beginning address needs to be divisible by the number of addresses in the block. To see this restriction, we can show that the beginning address can be calculated as ($X \times 2^{n - 32}$) in which *X* is the decimal value of the prefix. In other words, the beginning address is $X \times N$.

Example 5.30

An ISP has requested a block of 1000 addresses. The following block is granted.

a. Since 1000 is not a power of 2, 1024 addresses are granted (1024 = 210).

b. The prefix length for the block is calculated as $n = 32 - \log_2 1024 = 22$.

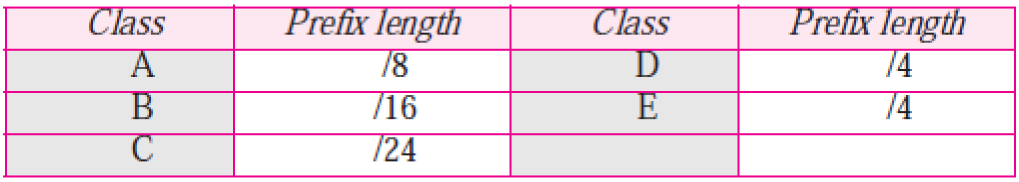
c. The beginning address is chosen as 18.14.12.0 (which is divisible by 1024).

The granted block is 18.14.12.0/22. The first address is 18.14.12.0/22 and the last address is 18.14.15.255/22.

Relation to Classful Addressing

All issues discussed for classless addressing can be applied to classful addressing. As a matter of fact, classful addressing is a special case of the classless addressing in which the blocks in class A, B, and C have the prefix length $n_A = 8$, $n_B = 16$, and $n_C = 24$. A block in classful addressing can be easily changed to a block in class addressing if we use the prefix length defined in Table 5.1.

Table 5.1 *Prefix length for classful addressing*



Class	*Prefix length*	*Class*	*Prefix length*
A	/8	D	/4
B	/16	E	/4
C	/24		

Example 5.31

Assume an organization has given a class A block as 73.0.0.0 in the past. If the block is not revoked by the authority, the classless architecture assumes that the organization has a block 73.0.0.0/8 in classless addressing.

Subnetting

Three levels of hierarchy can be created using subnetting. An organization (or an ISP) that is granted a range of addresses may divide the range into several subranges and assign each subrange to a **subnetwork** (or **subnet**). The concept is the same as we discussed for classful addressing. Note that nothing stops the organization from creating more levels. A subnetwork can be divided into several sub-subnetworks. A sub-subnetwork can be divided into several sub-sub-subnetworks. And so on.

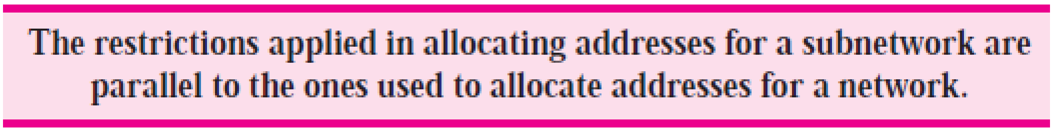
Designing Subnets

The subnetworks in a network should be carefully designed to enable the routing of packets. We assume the total number of addresses granted to the organization is *N*, the prefix length is *n,* the assigned number of addresses to each subnetwork is *N*sub, the prefix length for each subnetwork is *n*sub, and the total number of subnetworks is *s.* Then, the following steps need to be carefully followed to guarantee the proper operation of the subnetworks.

1. The number of addresses in each subnetwork should be a power of 2.
2. The prefix length for each subnetwork should be found using the following formula:

$$n_{sub} = n + \log_2 (N/N_{sub})$$

3. The starting address in each subnetwork should be divisible by the number of addresses in that subnetwork. This can be achieved if we first assign addresses to larger networks.



The restrictions applied in allocating addresses for a subnetwork are parallel to the ones used to allocate addresses for a network.

Finding Information about Each Subnetwork

After designing the subnetworks, the information about each subnetwork, such as first and last address, can be found using the process we described to find the information about each network in the Internet.

Example 5.32

An organization is granted the block 130.34.12.64/26. The organization needs four subnetworks, each with an equal number of hosts. Design the subnetworks and find the information about each network.

Solution

The number of addresses for the whole network can be found as $N = 2^{32 - 26} = 64$. Using the process described in the previous section, the first address in the network is 130.34.12.64/26 and the last address is 130.34.12.127/26. We now design the subnetworks:

1. We grant 16 addresses for each subnetwork to meet the first requirement (64/16 is a power of 2).

2. The **subnetwork** mask for each subnetwork is:

$$n_1 = n_2 = n_3 = n_4 = n + \log_2 (N/N_i) = 26 + \log_2 4 = 28$$

3. We grant 16 addresses to each subnet starting from the first available address. Figure 5.30 shows the subblock for each subnet. Note that the starting address in each subnetwork is divisible by the number of addresses in that subnetwork.

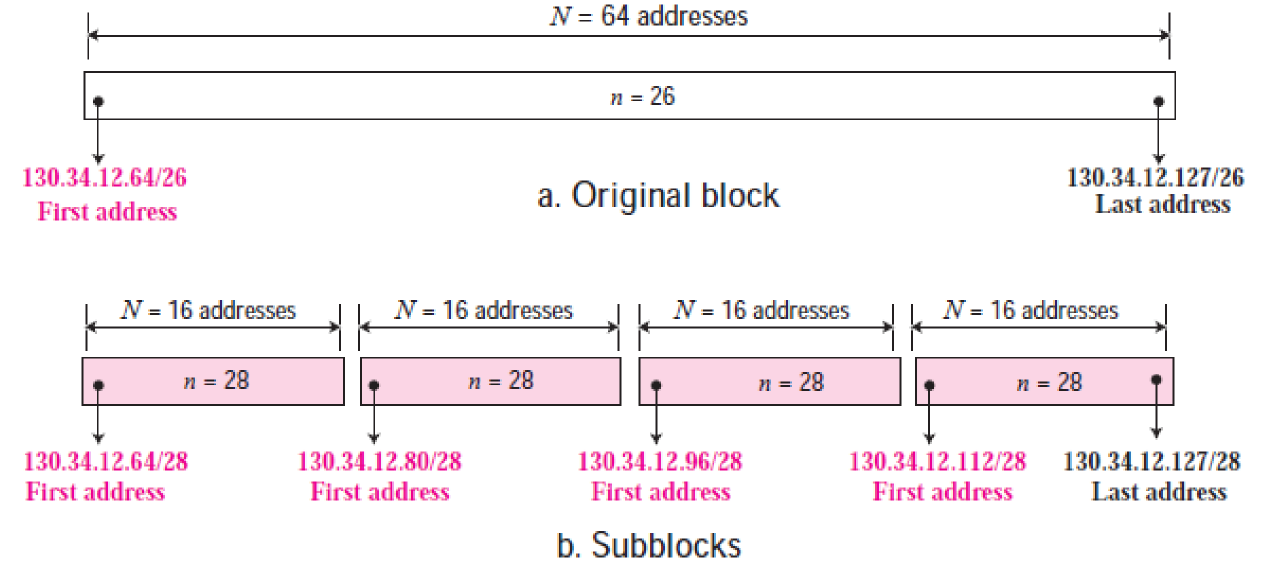


Figure 5.30 *Solution to Example 5.32*

Example 5.33

An organization is granted a block of addresses with the beginning address 14.24.74.0/24. The organization needs to have 3 subblocks of addresses to use in its three subnets as shown below:

- ❑ One subblock of 120 addresses.
- ❑ One subblock of 60 addresses.
- ❑ One subblock of 10 addresses.

Solution

There are $2^{32 - 24} = 256$ addresses in this block. The first address is 14.24.74.0/24; the last address is 14.24.74.255/24.

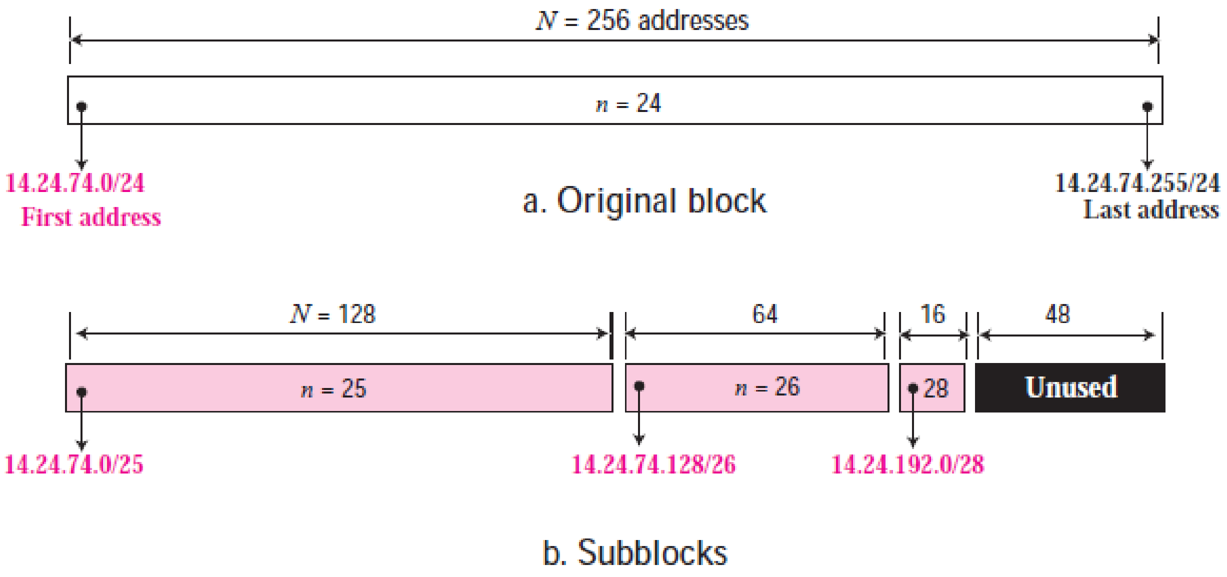
a. The number of addresses in the first subblock is not a power of 2. We allocate 128 addresses. The first can be used as network address and the last as the special address. There are still 126 addresses available. The subnet mask for this subnet can be found as $n1 = 24 + \log_2 (256/128) = 25$. The first address in this block is 14.24.74.0/25; the last address is 14.24.74.127/25.

b. The number of addresses in the second subblock is not a power of 2 either. We allocate 64 addresses. The first can be used as network address and the last as the special address. There are still 62 addresses available. The subnet mask for this subnet can be found as $n1 = 24 + \log_2 (256/64) = 26$. The first address in this block is 14.24.74.128/26; the last address is 14.24.74.191/26.

c. The number of addresses in the third subblock is not a power of 2 either. We allocate 16 addresses. The first can be used as network address and the last as the special address. There are still 14 addresses available. The subnet mask for this subnet can be found as $n1 = 24 + \log_2 (256/16) = 28$. The first address in this block is 14.24.74.192/28; the last address is 14.24.74.207/28.

d. If we add all addresses in the previous subblocks, the result is 208 addresses, which means 48 addresses are left in reserve. The first address in this range is 14.24.74.209. The last address is 14.24.74.255. We don't know about the prefix length yet.

e. Figure 5.31 shows the configuration of blocks. We have shown the first address in each block.



a. Original block

b. Subblocks

Figure 5.31 *Solution to Example 5.33*

Example 5.34

Assume a company has three offices: Central, East, and West. The Central office is connected to the East and West offices via private, point-to-point WAN lines. The company is granted a block of 64 addresses with the beginning address 70.12.100.128/26. The management has decided to allocate 32 addresses for the Central office and divides the rest of addresses between the two other offices.

1. The number of addresses are assigned as follows:

Central office $N_c = 32$ East office $N_e = 16$ West office $N_w = 16$

2. We can find the prefix length for each subnetwork:

$n_c = n + \log_2(64/32) = 27$ $n_e = n + \log_2(64/16) = 28$ $n_w = n + \log_2(64/16) = 28$

3. Figure 5.32 shows the configuration designed by the management. The Central office uses addresses 70.12.100.128/27 to 70.12.100.159/27. The company has used three of these addresses for the routers and has reserved the last address in the subblock. The East office uses the addresses 70.12.100.160/28 to 70.12.100.175/28. One of these addresses is used for the router and the company has reserved the last address in the subblock. The West office uses the addresses 70.12.100.160/28 to 70.12.100.175/28. One of these addresses is used for the router and the company has reserved the last address in the subblock. The company uses no address for the point-to-point connections in WANs.

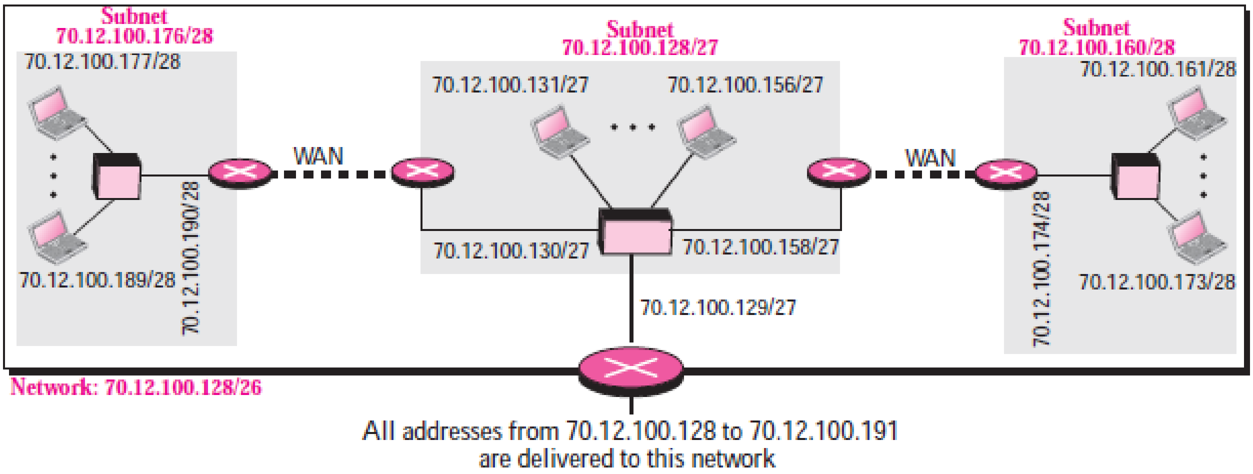


Figure 5.32 *Example 14*

Address Aggregation

One of the advantages of CIDR architecture is **address aggregation.** ICANN assigns a large **block of addresses** to an ISP. Each ISP in turn divides its assigned block into smaller subblocks and grants the subblocks to its customers; many blocks of addresses are aggregated in one block and granted to one ISP.

Example 5.35

An ISP is granted a block of addresses starting with 190.100.0.0/16 (65,536 addresses). The ISP needs to distribute these addresses to three groups of customers as follows:

❑ The first group has 64 customers; each needs approximately 256 addresses.

❑ The second group has 128 customers; each needs approximately 128 addresses.

❑ The third group has 128 customers; each needs approximately 64 addresses.

We design the subblocks and find out how many addresses are still available after these allocations.

Solution

Let us solve the problem in two steps. In the first step, we allocate a subblock of addresses to each group. The total number of addresses allocated to each group and the prefix length for each subblock can found as

Group 1: $64 \times 256 = 16,384$ $n_1 = 16 + \log_2 (65536/16384) = 18$

Group 2: $128 \times 128 = 16,384$ $n_2 = 16 + \log_2 (65536/16384) = 18$

Group 3: $128 \times 64 = 8192$ $n_3 = 16 + \log_2 (65536/8192) = 19$

Figure 5.33 shows the design for the first hierarchical level.

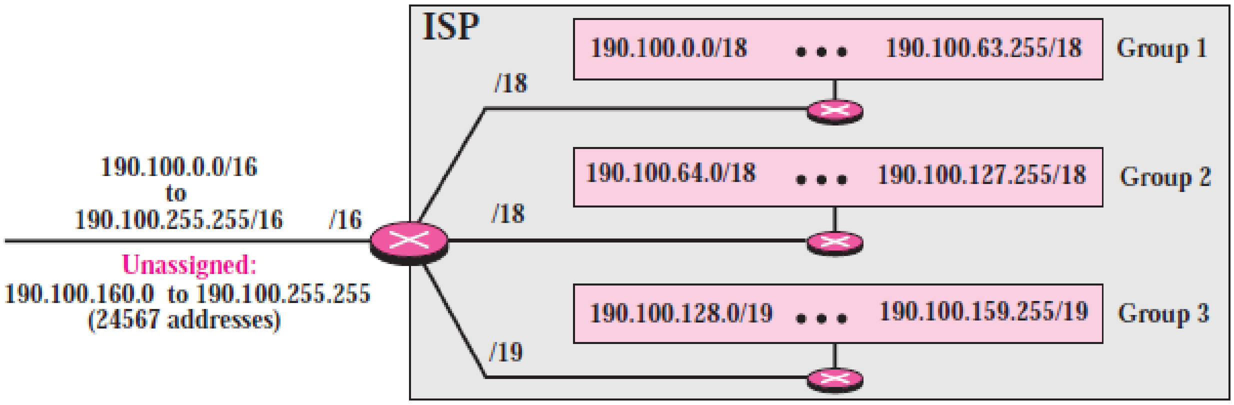


Figure 5.33 *Solution to Example 5.35: first step*

Now we can think about each group. The prefix length changes for the networks in each group depending on the number of addresses used in each network. Figure 5.34 shows the second level of the hierarchy. Note that we have used the first address for each customer as the subnet address and have reserved the last address as a special address.

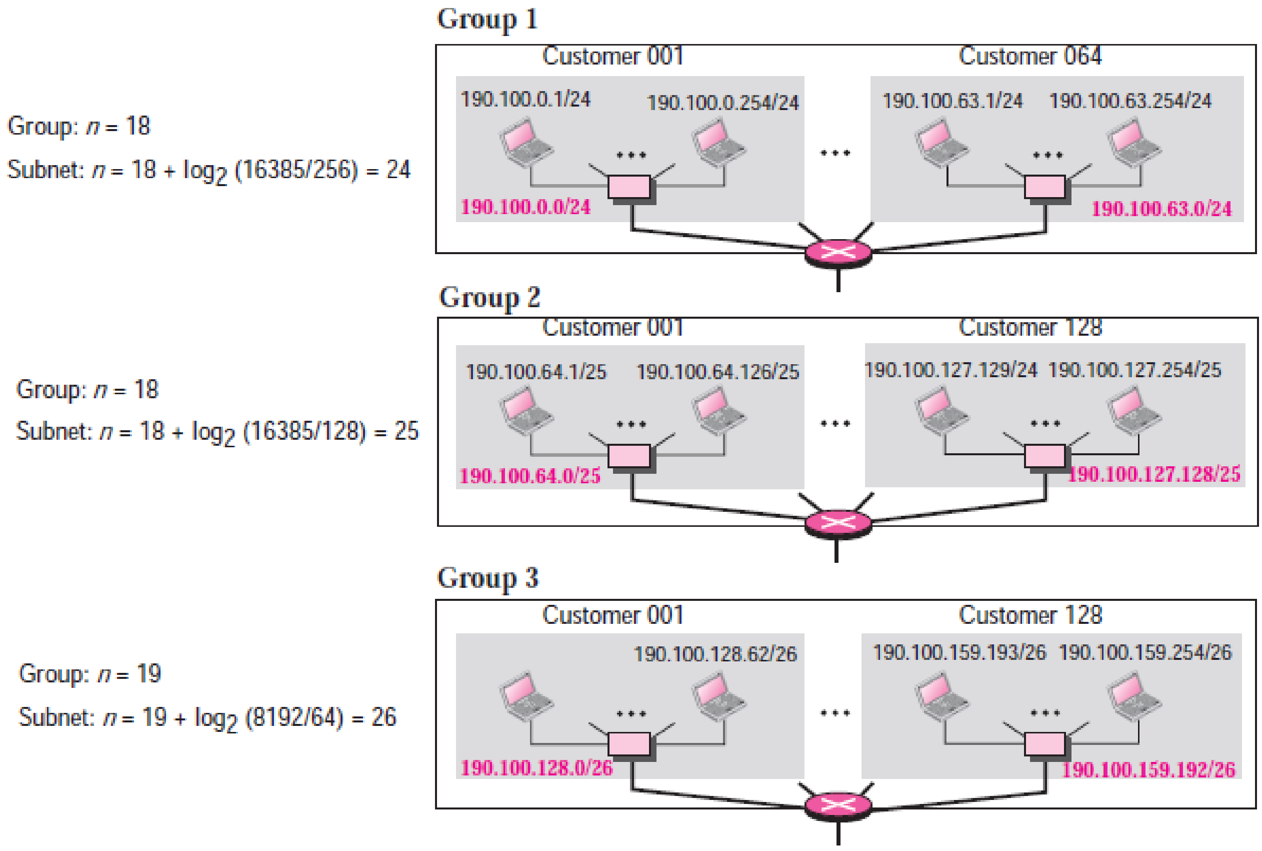


Figure 5.34 *Solution to Example 5.35: second step*