Subject Name: Computer Networks
Subject Code: CSE302, MCA406
CSE 6th and MCA 8th Semester
Contact period: 4L
Courtesy: B. A. Forouzan

Network Layer: Logical Addressing

By
Dr. Zeenat Rehena
Asst. Professor
CSE Dept. AU

Introduction

- Communication at network layer is host-to-host.
- The packet transmitted by the sending computer may pass through several LANs and WANs before reaching the destination computer.
- For this type of communication, we need global addressing
- We use the term **IP address** to mean a logical address in the network layer of the TCP/IP protocol suite.

IPv4 ADDRESSES

An IPv4 address is a 32-bit address that uniquely and universally defines the connection of a device (for example, a computer or a router) to the Internet.

IPv4 addresses are **unique** in the sense that each address defines one and only one, connection to the Internet.

IPv4 addresses are **universal** in the sense that the addressing system must be accepted by any host that wants to be connected to the Internet.

Address Space

A protocol such as IPv4 that defines addresses has an **address space**.
An address space is the total number of addresses used by the protocol.

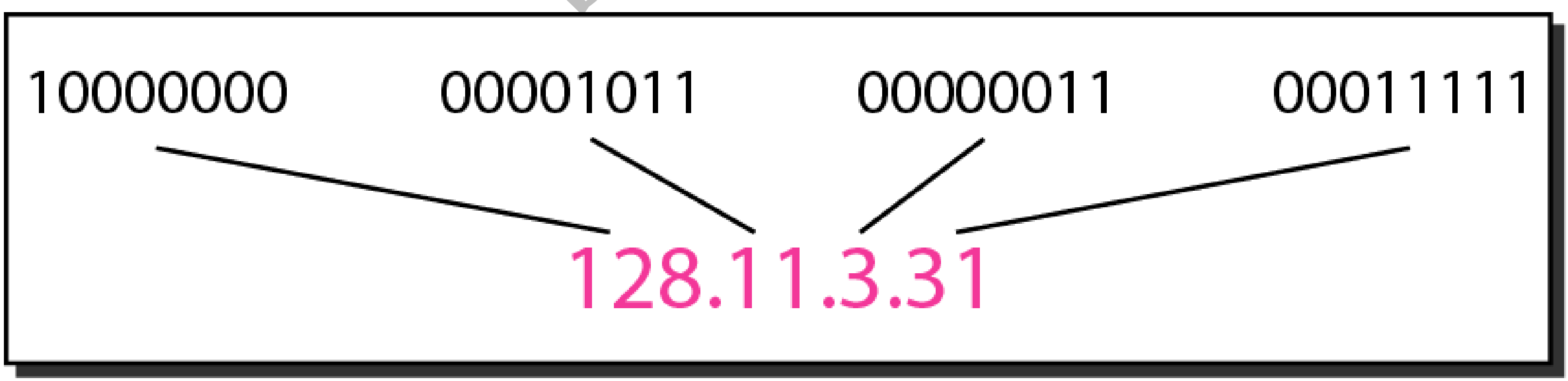
IPv4 uses 32-bit addresses, therefore,
The address space of IPv4 is 2^{32} *or 4,294,967,296.*

Dotted-decimal notation and binary notation for an IPv4 address

There are two types of notations used to show an IPv4 address:

1. Binary notation – each octet is often referred to as byte
2. Dotted decimal notation – written in decimal form with a decimal point (dot) separating the bytes

An example is given below



Example 1

Change the following IPv4 addresses from binary notation to dotted-decimal notation.

a. 10000001 00001011 00001011 11101111

b. 11000001 10000011 00011011 11111111

Solution

We replace each group of 8 bits with its equivalent decimal number (see Appendix B) and add dots for separation.

a. 129.11.11.239

b. 193.131.27.255

Example 2

Change the following IPv4 addresses from dotted-decimal notation to binary notation.

a. 111.56.45.78
b. 221.34.7.82

Solution

We replace each decimal number with its binary equivalent (see Appendix B).

a. 01101111 00111000 00101101 01001110
b. 11011101 00100010 00000111 01010010

Example 3

Find the error, if any, in the following IPv4 addresses.

a. 111.56.045.78

b. 221.34.7.8.20

c. 75.45.301.14

d. 11100010.23.14.67

Solution

a. There must be no leading zero (045).

b. There can be no more than four numbers.

c. Each number needs to be less than or equal to 255.

d. A mixture of binary notation and dotted-decimal notation is not allowed.

Classful Addressing

In classful addressing, the address space is divided into five classes: A, B, C, D, and E.

Example of classes in binary and dotted-decimal notation

	First byte	Second byte	Third byte	Fourth byte
Class A	0			
Class B	10			
Class C	110			
Class D	1110			
Class E	1111			

a. Binary notation

	First byte	Second byte	Third byte	Fourth byte
Class A	0–127			
Class B	128–191			
Class C	192–223			
Class D	224–239			
Class E	240–255			

b. Dotted-decimal notation

Example 4

Assignment -1

Find the class of each address.

a. 00000001 00001011 00001011 11101111

b. 11000001 10000011 00011011 11111111

c. 14.23.120.8

d. 252.5.15.111

Classes and Blocks

One problem with classful addressing is that each class is divided into a fixed number of blocks with each block having a fixed size as shown in below table.

Class A addresses were designed for large organizations with a large number of attached hosts or routers.

Class B addresses were designed for midsize organizations with ten of thousands of attached hosts or routers.

Class C addresses were designed for small organizations with a small number of attached hosts or routers.

Table 1: *Number of blocks and block size in classful IPv4 addressing*

Class	*Number of Blocks*	*Block Size*	*Application*
A	128	16,777,216	Unicast
B	16,384	65,536	Unicast
C	2,097,152	256	Unicast
D	1	268,435,456	Multicast
E	1	268,435,456	Reserved

Netid and Hostid

•An IP address in class A, B, or C is divided into netid and hostid.
•These parts are of varying lengths, depending on the class of the address.
•The netid is shown in colour and hostid is in white in the table below.

Table 2 *Default masks for classful addressing*

Class	*Binary*	*Dotted-Decimal*	*CIDR*
A	**11111111** 00000000 00000000 00000000	**255**.0.0.0	/8
B	**11111111 11111111** 00000000 00000000	**255.255**.0.0	/16
C	**11111111 11111111 11111111** 00000000	**255.255.255**.0	/24

In classful addressing, a large part of the available addresses were wasted.
Classful addressing, which is almost obsolete, is replaced with classless addressing.

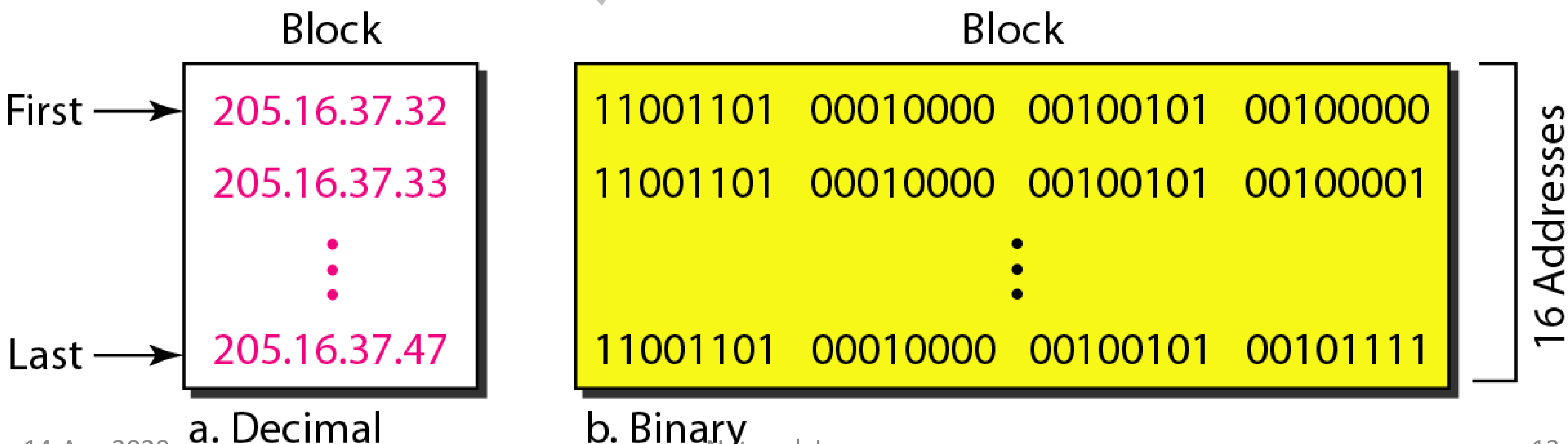
Classless Addressing

In classless addressing, the size of the block varies based on the nature and size of the entity.

Restrictions: To simplify the handling of addresses, the Internet authorities impose Three restrictions on classless address blocks:

1. The addresses in a block must be contiguous, one after another
2. The number of addresses in a block must be a power of 2 (1,2,4,8,….)
3. The first address must be evenly divisible by the number of addresses.

Example: *A block of 16 addresses granted to a small organization as shown below*



Contd.

Previous figure shows a block of addresses, in both binary and dotted-decimal notation, granted to a small business that needs 16 addresses.

It is seen that the restrictions are applied to this block. The addresses are contiguous. The number of addresses is a power of 2 ($16 = 2^4$), and the first address is divisible by 16. The first address, when converted to a decimal number, is 3,440,387,360, which when divided by 16 results in 215,024,210.

Mask

A mask is a 32 bit number in which the n leftmost bits are 1s and 32-n rightmost bits are 0s.
The mask value may vary from 0 to 32.

In IPv4 addressing, a block of addresses can be defined as
x.y.z.t /*n*
in which x.y.z.t defines one of the addresses and the /*n* defines the mask.

Mask

The first address in the block can be found by setting the rightmost
$32 - n$ bits to 0s.

The last address in the block can be found by setting the rightmost
$32 - n$ bits to 1s.

The number of addresses in the block can be found by using the formula
2^{32-n}.

Example 5

A block of addresses is granted to a small organization. We know that one of the addresses is 205.16.37.39/28. What is the first address in the block?

Solution

The binary representation of the given address is

11001101 00010000 00100101 00100111

If we set 32−28 rightmost bits to 0 (red color), we get

11001101 00010000 00100101 00100000

or

205.16.37.32.

Example 6

Find the (i) last address and (ii)number of addresses for the block in Example 5.

Solution -i

The binary representation of the given address is

11001101 00010000 00100101 00100111

If we set 32 − 28 rightmost bits to 1 (red color), we get

11001101 00010000 00100101 00101111

or

205.16.37.47

Solution -ii

The value of n is 28, which means that number of addresses is 2^{32-28} *or 16.*

Example 7

Another way to find the first address, the last address, and the number of addresses is to represent the mask as a 32-bit binary (or 8-digit hexadecimal) number. This is particularly useful when we are writing a program to find these pieces of information. In Example 5 the /28 can be represented as

11111111 11111111 11111111 11110000

(twenty-eight 1s and four 0s).

Now find

a. The first address

b. The last address

c. The number of addresses.

Example 7 (Contd.)

Solution

a. *The first address can be found by ANDing the given addresses with the mask. ANDing here is done bit by bit. The result of ANDing 2 bits is 1 if both bits are 1s; the result is 0 otherwise.*

Address:	11001101	00010000	00100101	00100111
Mask:	**11111111**	**11111111**	**11111111**	**11110000**
First address:	11001101	00010000	00100101	00100000

Example 7 (Contd.)

b. *The last address can be found by ORing the given addresses with the complement of the mask. ORing here is done bit by bit. The result of ORing 2 bits is 0 if both bits are 0s; the result is 1 otherwise. The complement of a number is found by changing each 1 to 0 and each 0 to 1.*

Address:	11001101	00010000	00100101	00100111
Mask complement:	**00000000**	**00000000**	**00000000**	**00001111**
Last address:	11001101	00010000	00100101	00101111

Example 7 (Contd.)

c. *The number of addresses can be found by complementing the mask, interpreting it as a decimal number, and adding 1 to it.*

Mask complement:	000000000 00000000 00000000 00001111
Number of addresses:	15 + 1 = 16

Network Address

- A very important concept in IP addressing is the **network address**.
- When an organization is given a block of addresses, the organization is free to allocate the addresses to the devices that need to be connected to the Internet.
- The **first address** is called the **network address** and defines the organization network.

Contd.

The figure below shows network configuration for the block 205.16.37.32/28 and granted a 16-address block.

The 16 address block are ranges from block 205.16.37.32/28 to block 205.16.37.47/28

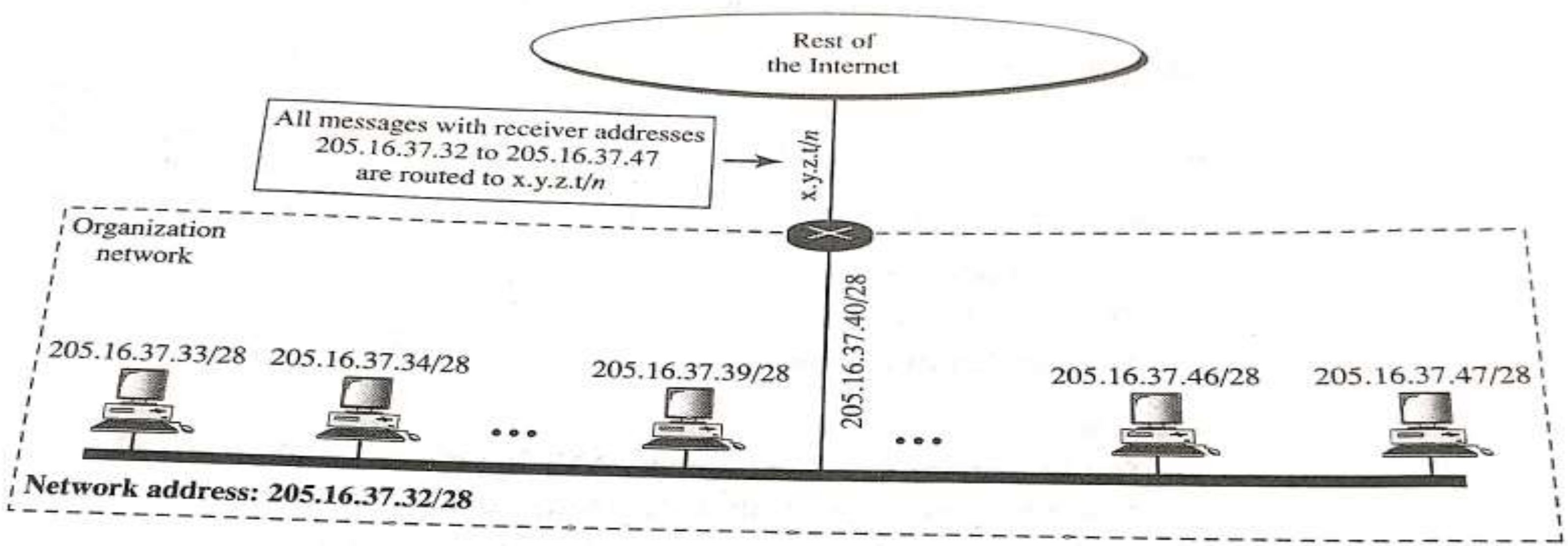


Figure 1: *A network configuration for the block 205.16.37.32/28*

Two-Level Hierarchy: No Subnetting

IP address can define only two levels of hierarchy when not subnetted. The n left-most bits of the address x.y.z.t/n define the network and 32-n rightmost bits define the particular host to the network.
The part of the address that defines the network is called ***prefix*** and the part that defines the host is called the ***suffix***.

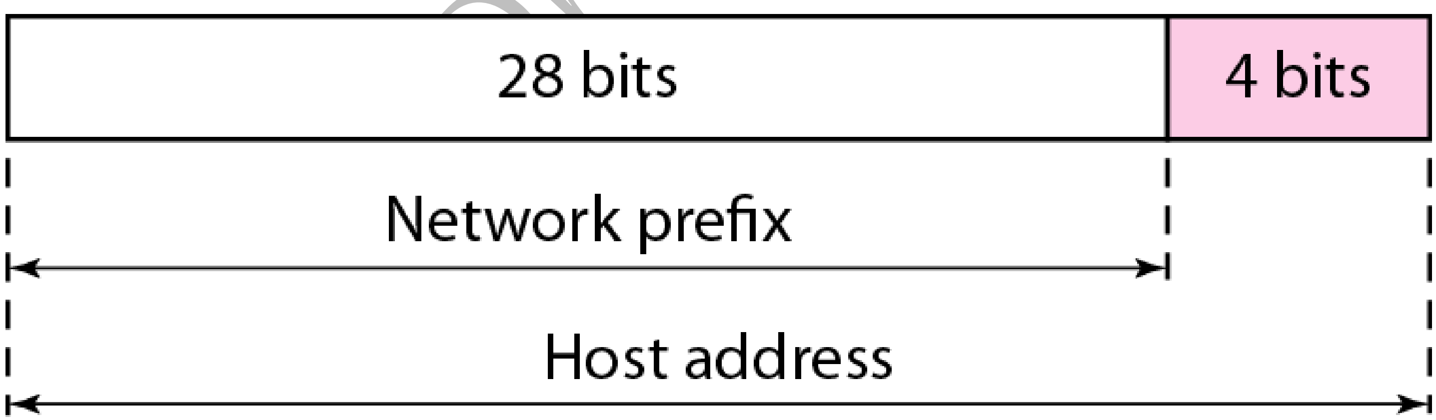


Figure 2:*Two levels of hierarchy in an IPv4 address*

Three-Level Hierarchy: Subnetting

An organization that is granted a large block of addresses may create clusters of networks, called ***subnets and divide the addresses between the different subnets.***

All messages are sent to the router address that connects the organization to the rest of the Internet.

The router delivers the messages to the appropriate subnets.

Example: **Suppose an organization is given the block 17.12.40.0/26, which contains 64 addresses. Organization has three (3) offices and divide the addresses into three subnets of 32, 16, 16 addresses.**

Find out the new masks for the three offices having 32, 16, 16 addresses respectively.

Solution: 1. Let the mask for the first subnet is n1, then 2^{32-n1}=32 [required address]

therefore, $2^{32-n1}= 2^5$

=> n1= 27

2. Similarly, for the second subnet mask is n2, then 2^{32-n2}= 16 [required address]

therefore, $2^{32-n1}= 2^4$

=> n2= 28

3. Similarly, for third subnet mask is n3, and then 2^{32-n3}= 16 [required address]

=> n3= 28

So, 17.12.14.0/27 is the subnet address of subnet 1, and 17.12.14.0/27 to 17.12.14.31/27 a total of 32 address is given to subnet1.

Similarly, for subnet2 and subnet 3 the subnet addresses are 17.12.14.32/28 and 17.12.14.48/28 Respectively.

Both subnet2 and subnet3 have 16 addresses.

The configuration and addresses are explained in the figure below.

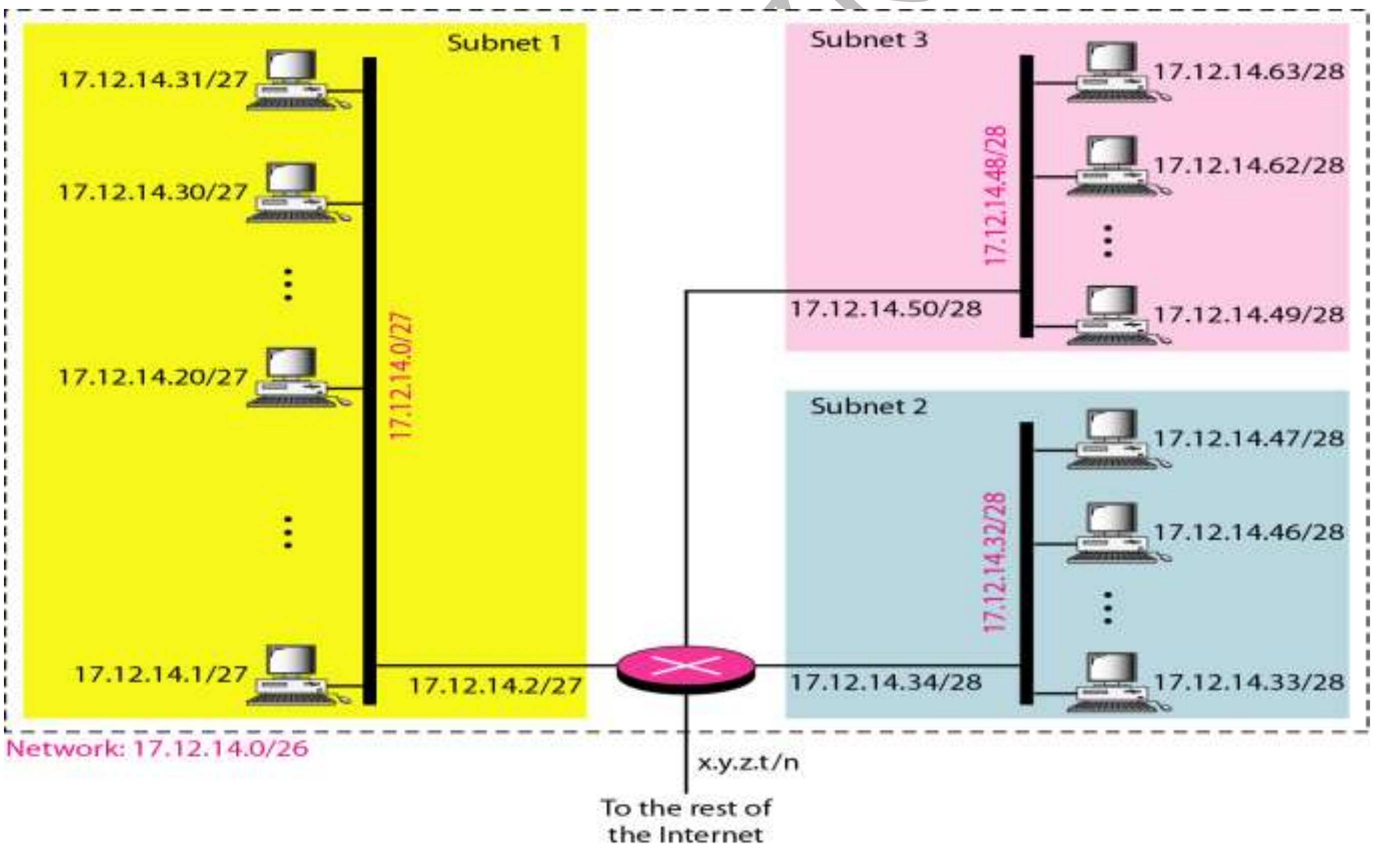


Figure 3: *Configuration and addresses in a subnetted network*

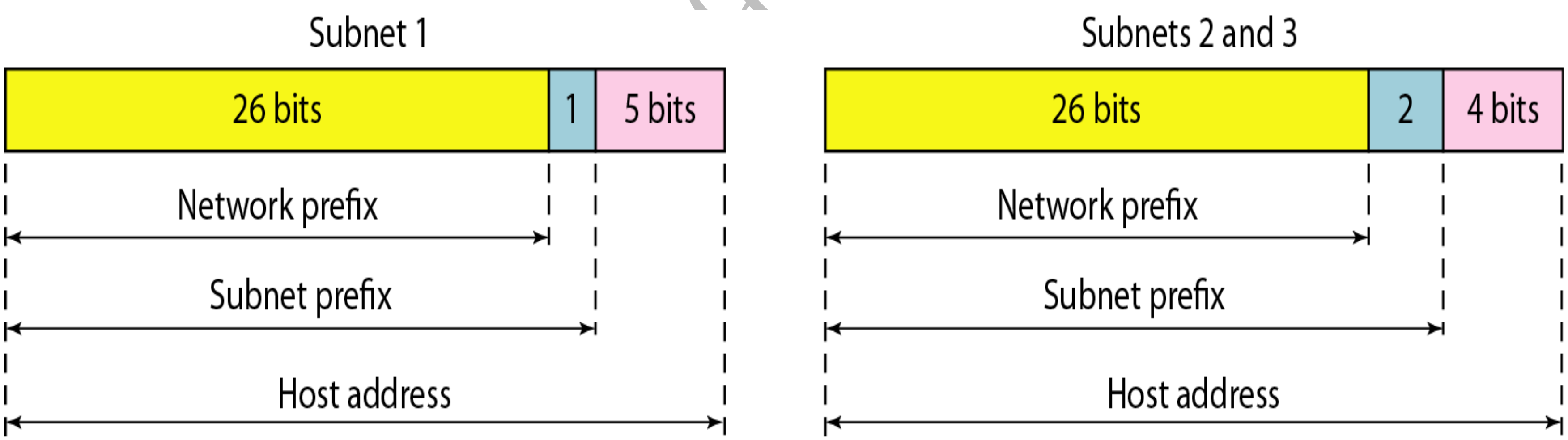


Figure 4: *Three-level hierarchy in an IPv4 address*

Example-8

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:

a. The first group has 64 customers; each needs 256 addresses.
b. The second group has 128 customers; each needs 128 addresses.
c. The third group has 128 customers; each needs 64 addresses.

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

Example-8 (CONTD.)

Solution

Group 1 (Total customer 64, each need 256, so, 256x64=16,384 addresses)

For this group, each customer needs 256 addresses. This means that 8 (log2 256) bits are needed to define each host. The prefix length is then 32 − 8 = 24.

The addresses are as follows:

1^{st} customer: 190.100.0.0./24 to 190.100.0.255/24

[the fourth octet is used for host address and the third octet is used for customer id]

2^{nd} customer: 190.100.1.0/24 to 190.100.1.255/24

3^{rd} customer : 190.100.2.0/24 to 190.100.2.255/24

……..

64^{th} customer: 190.100.63.0/24 to 190.100.63.255/24

Example-8 (CONTD.)

Group 2 Total customer 128, each need 128, so, 128x128=16,384 addresses)

For this group, each customer needs 128 addresses. This means that 7 (log2 128) bits are needed to define each host. The prefix length is then 32 − 7 = 25. The addresses are:

Note:Gr-1, last address ends at 190.100.63.255, so here starts from 190.100.64.0

1st customer: 190.100.64.0./25 to 190.100.64.127/25

[the fourth octet is used for host address and the third octet is used for customer id]

2nd customer: 190.100.64.128/25 to 190.100.1.255/25

3rd customer : 190.100.65.0/25 to 190.100.65.127/25

… …..

128th customer: 190.100.127.0/24 to 190.100.127.255/25

Example-8 (CONTD.)

Group 3

For this group, each customer needs 64 addresses. This means that 6 ($\log_2 64$) bits are needed to each host. The prefix length is then 32 − 6 = 26. The addresses are

Note:Gr-2,last address ends at 190.100.127.255, so here starts from 190.100.128.0

1st customer: 190.100.128.0./26 to 190.100.128.63/26

[the fourth octet is used for host address and the third octet is used for customer id]

2nd customer: 190.100.128.64./26 to 190.100.128.127/26

3rd customer: 190.100.128.128./26 to 190.100.128.191/26

… ……

128th customer: 190.100.159.192/26 to 190.100.159.255/26

Total=128x64=8192

Example-8 (CONTD.)

Number of granted addresses to the ISP: 65,536
Number of allocated addresses by the ISP: 40,960
Number of available addresses: 24,576

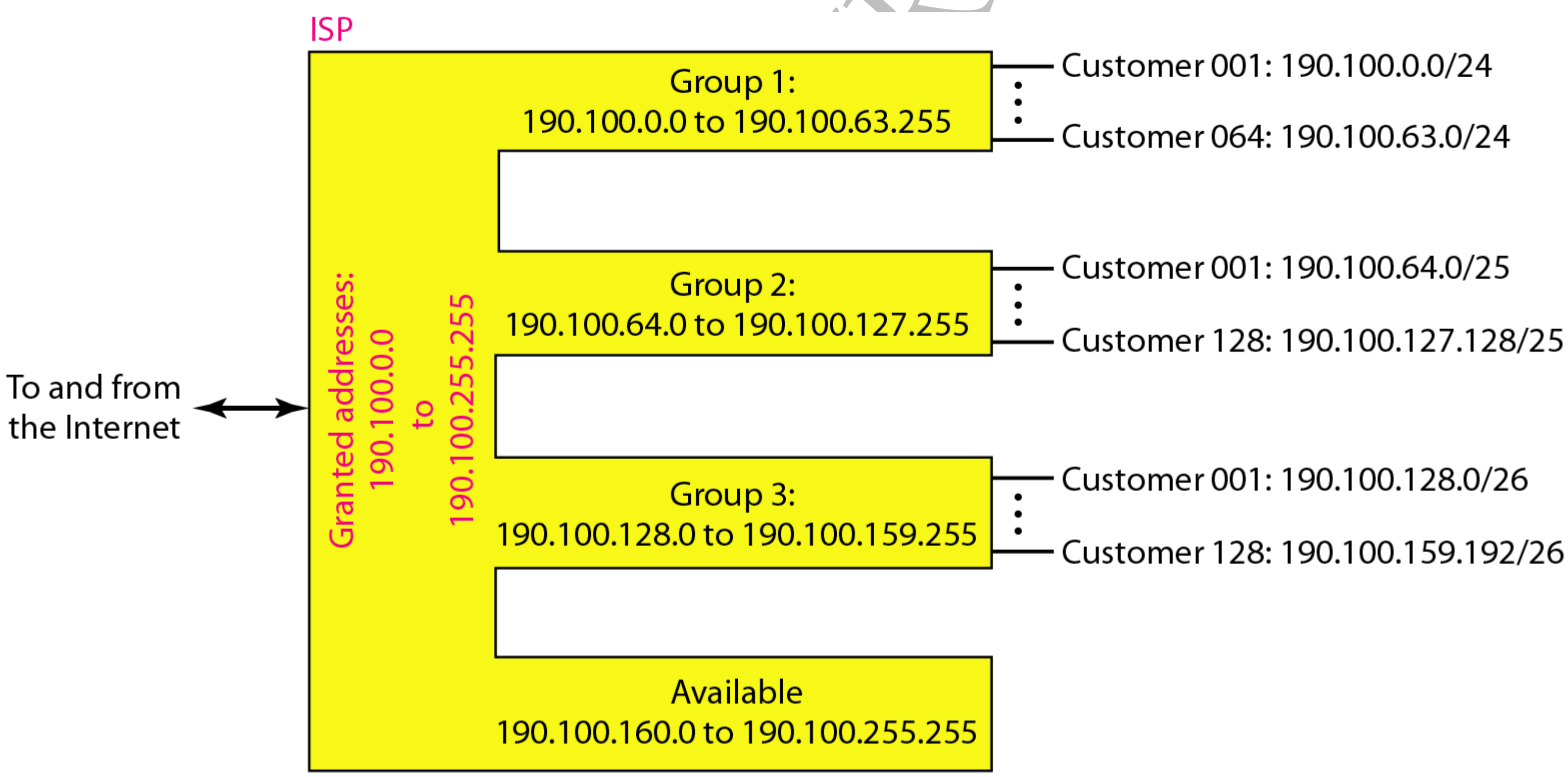


Figure 19.9 *An example of address allocation and distribution by an ISP*

Reference Book: Data Communications and Networking (4th Edition) by B A Forouzan

For any queries/doubts

WhatsApp me : +91 9732870280

Or

Email: zeenatrehena@yahoo.co.in