TCP/IP

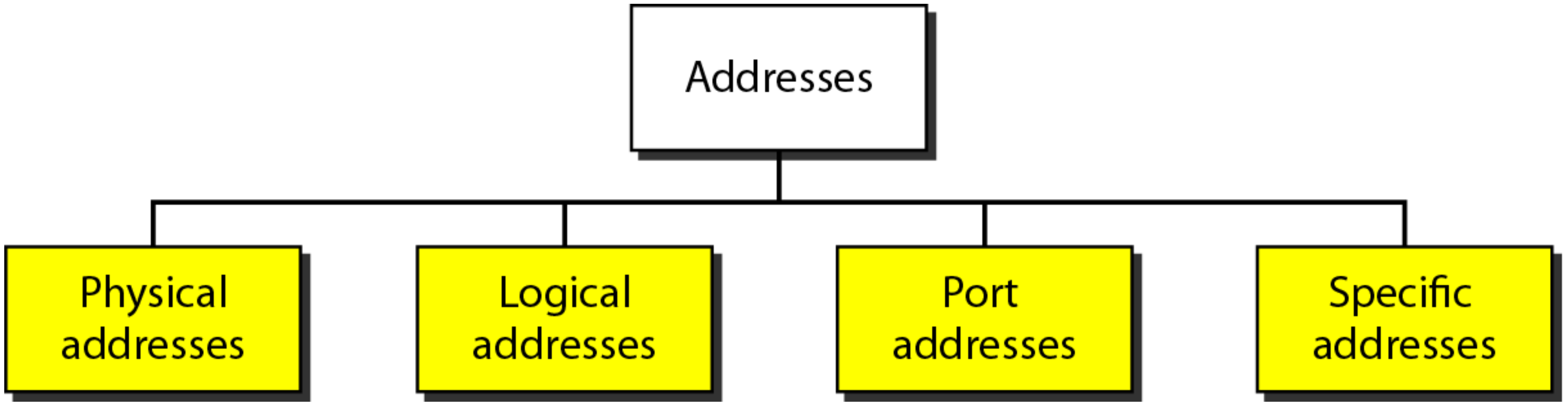
Dr. Lway Faisal

TCP/IP PROTOCOL SUITE

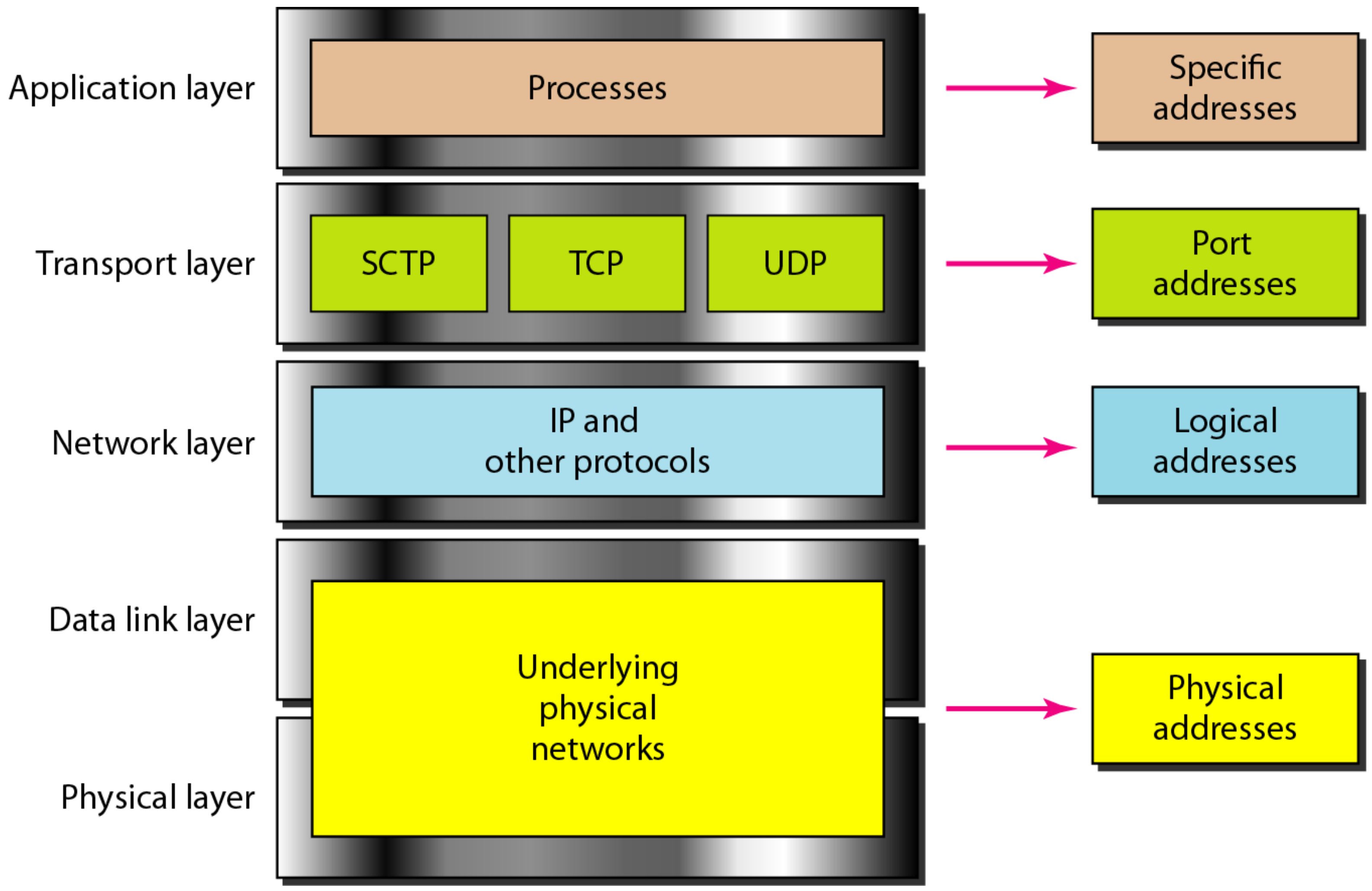
When TCP/IP is compared to OSI, we can say that the TCP/IP protocol suite is made of five layers: physical, data link, network, transport, and application.

ADDRESSING

Four levels of addresses are used in an internet employing the TCP/IP protocols: physical, logical, port, and specific.



Relationship of layers and addresses in TCP/IP



Physical Address

•Physical address or (hardware Address), or MAC address.

•Each node has a unique MAC Address: Globally identifier that burned into your RAM of your network interface card.

•MAC Address assigned by manufacturer , each factory has a block of address assigned by IEEE.

•No two networks in the world have the same Address.

•local-area networks use a 48-bit (6-byte) physical address written as 12 hexadecimal digits; every byte (2 hexadecimal digits) is separated by a colon, as shown below:

07:01:02:01:2C:4B

A 6-byte (12 hexadecimal digits) physical address.

Network Layer:
Logical Addressing

Dr. Lway Faisal

An IP address (Internet Protocol Address) or (logical Address) is a unique address that devices use it in order to communicate with each other.

IP addresses are managed and created by the Internet Assigned Numbers Authority (**IANA**).

IP have two versions: 1. IPv4 is 32bits

2. IPv6 is 128bits

The network layer is responsible for the delivery of individual packets from the source host to the destination host.

IPv4 ADDRESSES

An IPv4 address is 32 bits long, are unique and universal.

A protocol IPv4 has an address space. An **address space** is the total number of addresses used by the protocol. If a protocol uses *N* bits to define an address, the address space is 2^N.

IPv4 uses 32-bit addresses, which means that the address space is 2^{32} or 4,294,967,296 (more than 4 billion). This means that, theoretically, if there were no restrictions, *more than 4 billion devices could be connected to the Internet.*

Notations

There are two prevalent notations to show an IPv4 address: binary notation and dotted-decimal notation.

Binary Notation

In binary notation, the IPv4 address is displayed as 32 bits. Each octet is often referred to as a byte. So it is common to hear an IPv4 address referred to as a 32-bit address or a 4-byte address. Example:

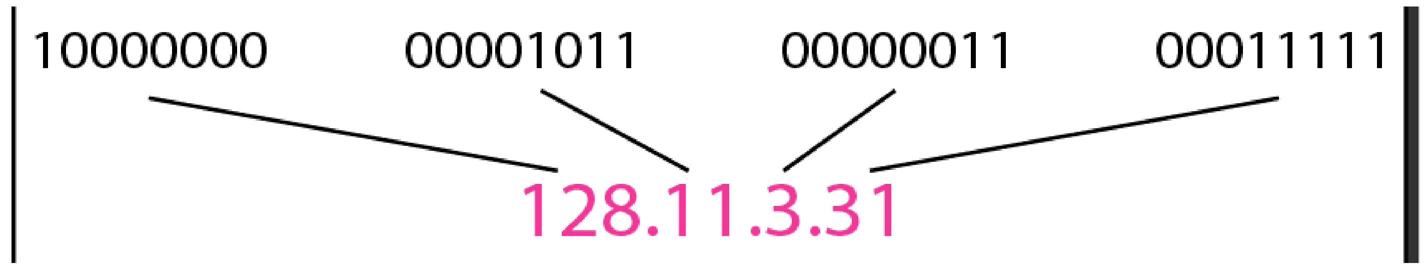
01110101 10010101 00011101 00000010

00000000.00000000.00000000.00000000

11111111.11111111.11111111.11111111

Dotted-Decimal Notation

To make the IPv4 address more compact and easier to read, Internet addresses are usually written in decimal form with a decimal point (dot) separating the bytes. Example:



Note that because each byte (octet) is 8 bits, each number in dotted-decimal notation is a value ranging from 0 to 255.

Example 2.1

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

a. 10000001 00001011 00001011 11101111

b. 11000001 10000011 00011011 11111111

Solution

We replace each group of 8 bits with its equivalent decimal number and add dots for separation.

a. 129.11.11.239

b. 193.131.27.255

Example 2.2

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

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 2.3

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).

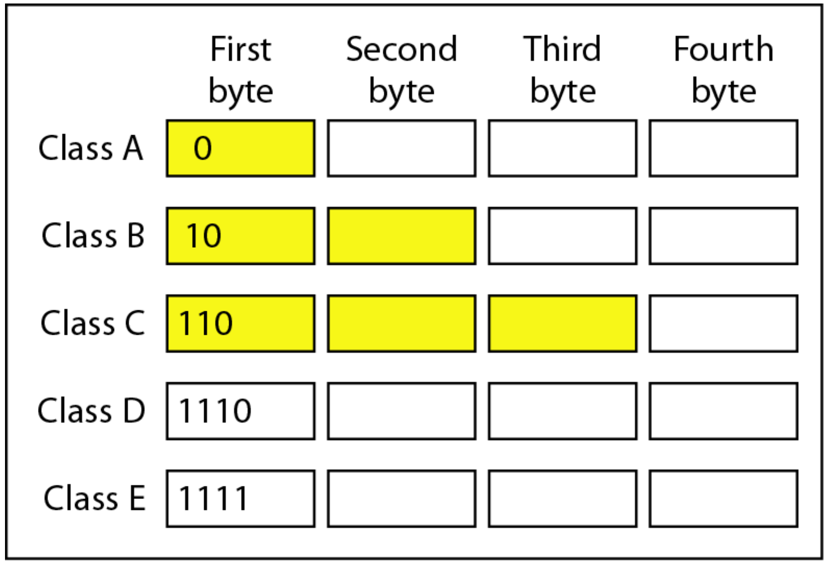
b. There can be no more than four numbers.

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.

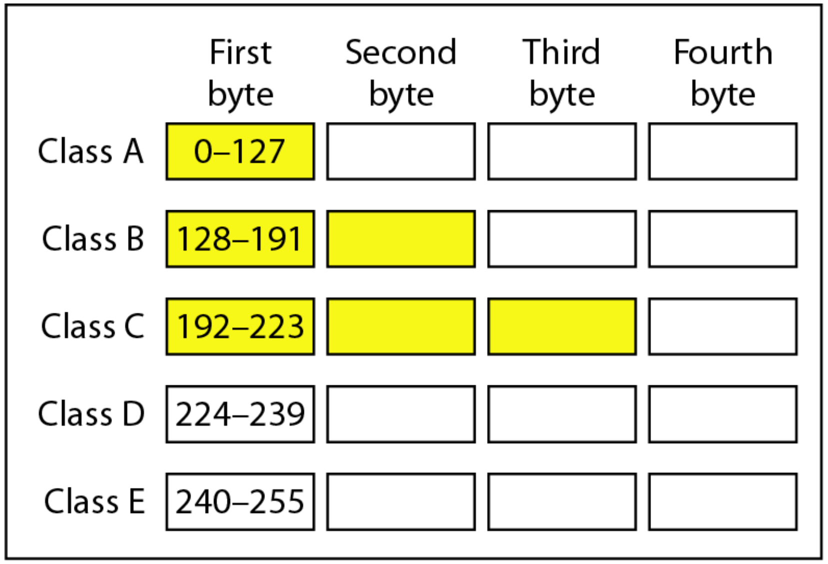
Classful Addressing

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



	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

Finding the classes in binary and dotted-decimal notation

Example 2.4

Find the class of each address?

a. <u>**0**</u>0000001 00001011 00001011 11101111

b. <u>**110**</u>00001 10000011 00011011 11111111

c. <u>**14**</u>.23.120.8

d. <u>**252**</u>.5.15.111

Solution

a. The first bit is 0. This is a class A address.

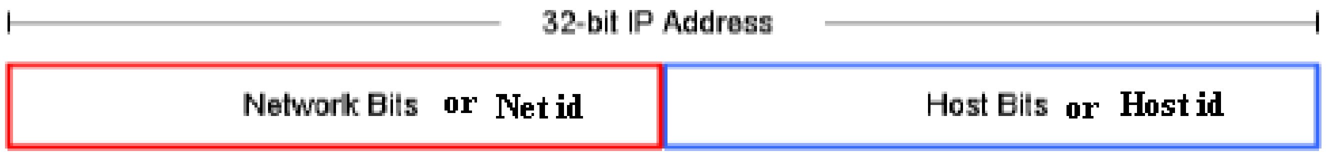
b. The first 2 bits are 1; the third bit is 0. This is a class C address.

c. The first byte is 14; the class is A.

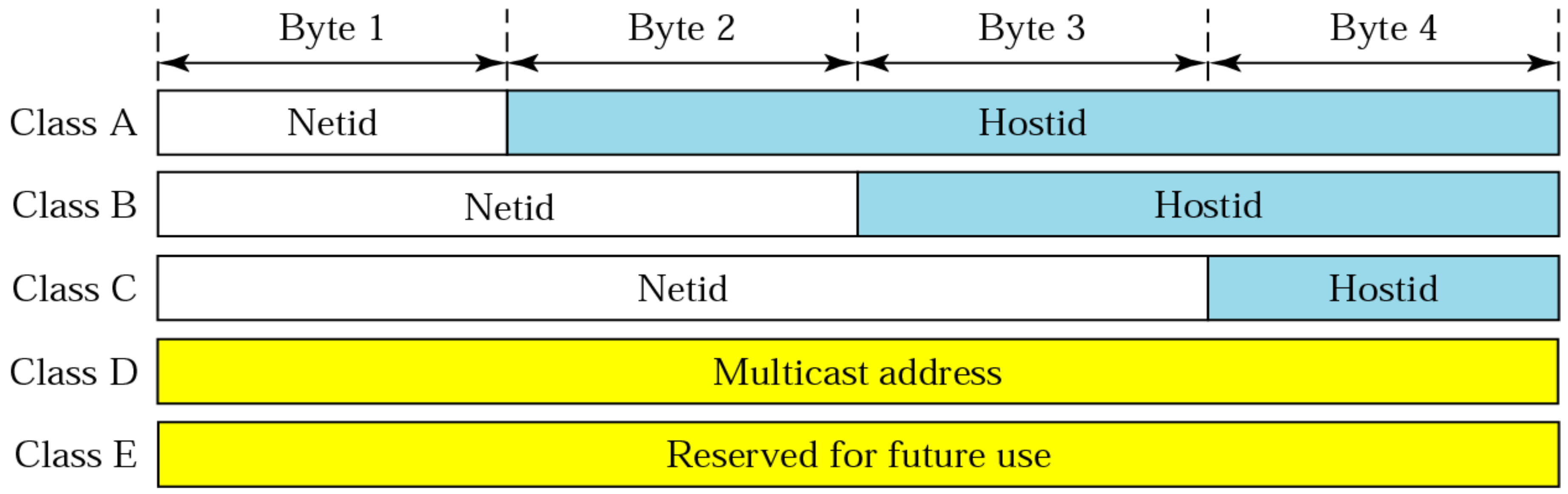
d. The first byte is 252; the class is E.

Anatomy of an IP Address

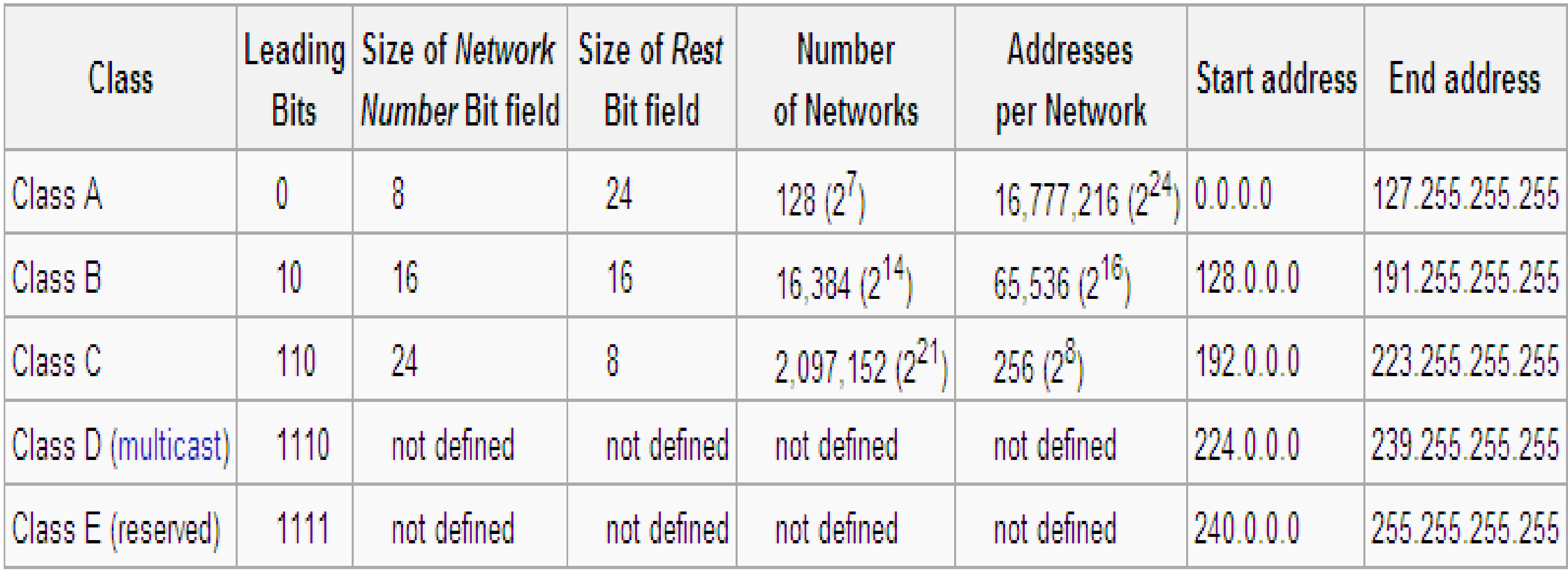
- The IP address consists of two components:
- First component is the network portion of the address, consisting of the network bits.
- Second component is the host portion of the address, consisting of the host bits. They consist of the remaining bits not included with the network bits. The part of an IP address that identifies a host.



IP Address Classes



IP Address Classes

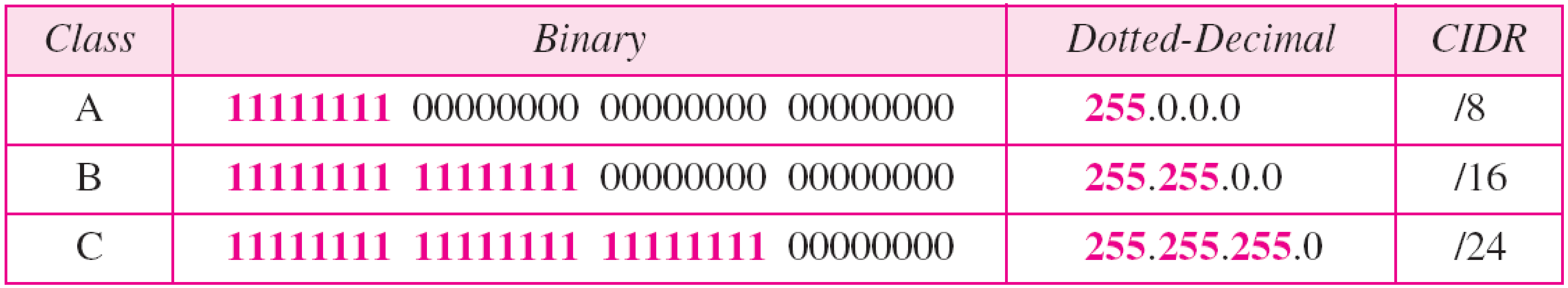


Class	Leading Bits	Size of *Network Number* Bit field	Size of *Rest* Bit field	Number of Networks	Addresses per Network	Start address	End address
Class A	0	8	24	128 (2^7)	16,777,216 (2^{24})	0.0.0.0	127.255.255.255
Class B	10	16	16	16,384 (2^{14})	65,536 (2^{16})	128.0.0.0	191.255.255.255
Class C	110	24	8	2,097,152 (2^{21})	256 (2^8)	192.0.0.0	223.255.255.255
Class D (multicast)	1110	not defined	not defined	not defined	not defined	224.0.0.0	239.255.255.255
Class E (reserved)	1111	not defined	not defined	not defined	not defined	240.0.0.0	255.255.255.255

Mask or Default Mask

The masks for classes A, B, and C are shown in Table below The concept does not apply to classes D and E.

The mask can help us to find the Net ID and the Host ID. For



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

Subnetting

If an organization was granted a large block in class A or B, it could **divide** the addresses into several contiguous groups and assign each group to smaller networks (called subnets) or, in rare cases, share part of the addresses with neighbors.

Supernetting:

A several networks are **combined** to create a super network.

Address Depletion

Yet the number of devices on the Internet is much less than the 2^{32} address space. We have run out of class A and B addresses, and a class C block is too small for most midsize organizations. *One solution that has alleviated the problem is the idea of classless addressing.*

Classful addressing: An IPv4 addressing mechanism in which the IP address space is divided into 5 classes: A, B, C, D, and E. Each class occupies some part of the whole address space.

Classless addressing: An addressing mechanism in which the IP address space is not divided into classes.

Classless Addressing

To overcome address depletion and give more organizations access to the Internet, classless addressing was designed and implemented. In this scheme.

Address Blocks

In classless addressing, when an entity, needs to be connected to the Internet, it is granted a block (range) of addresses. The size of the block (the number of addresses) varies based on the nature and size of the entity.

An ISP, as the Internet service provider, may be given thousands or hundreds of thousands based on the number of customers it may serve.

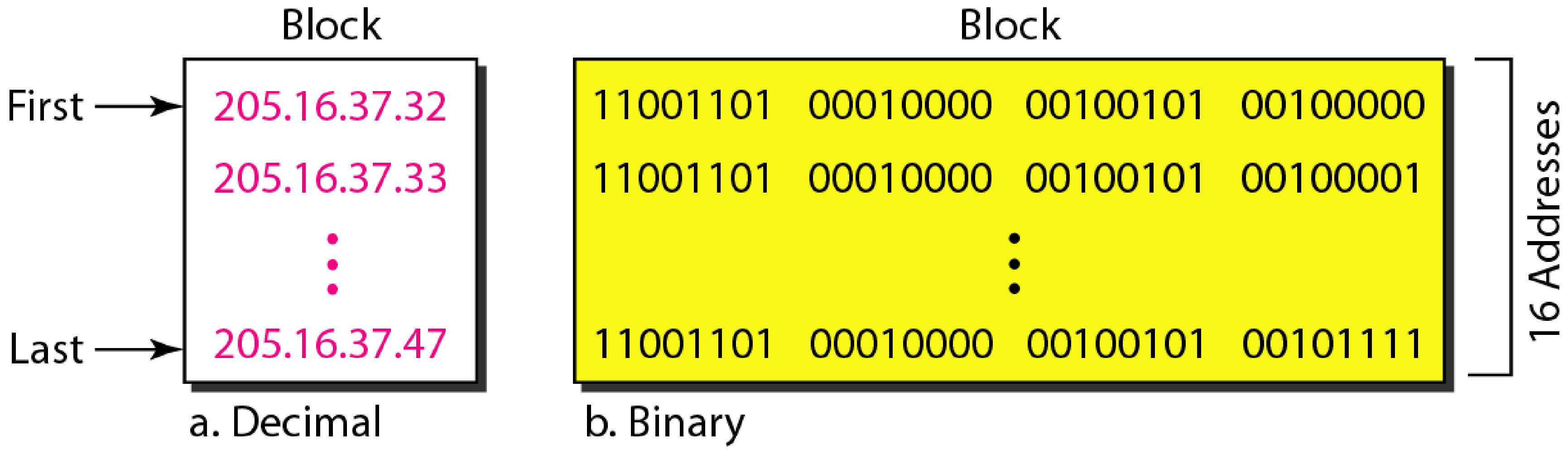
Restriction

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,.etc)
3. The first address must be evenly divisible by the number of addresses.

Example

Figure 2.3 shows a block of addresses, in both binary and dotted-decimal notation, granted to a small business that needs 16 addresses.



We can see 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 address and the */n* notation completely define the whole block (the first address, the last address, and the number of addresses).

First Address The first address in the block can be found by setting the 32 - *n* rightmost bits in the binary notation of the address to 0s.

Example

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 00100 111

If we set 32−28 rightmost bits to 0, we get

11001101 00010000 00100101 00100000

or

205.16.37.32/28

Last Address

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

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

Example

Find the last address for the block.

205.16.37.39/28

Solution

The binary representation of the given address is

11001101 00010000 00100101 00100111

If we set 32 − 28 rightmost bits to 1, we get

11001101 00010000 00100101 00101111

or

205.16.37.47

Number of Addresses

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

Example

Find the number of addresses in Example 6.6.

205.16.37.39/28

Solution

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

205.16.37.32 → 205.16.37.47

Example

Another way to find the first address, the last address, and the number of addresses is to represent the mask as a 32bit binary (or 8-digit hexadecimal) number. This is particularly useful when we are writing a program to find these pieces of information. Ex: The 205.16.37.39/28 , /28 can be represented as (Mask Definition)

11111111 11111111 11111111 11110000

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

Find

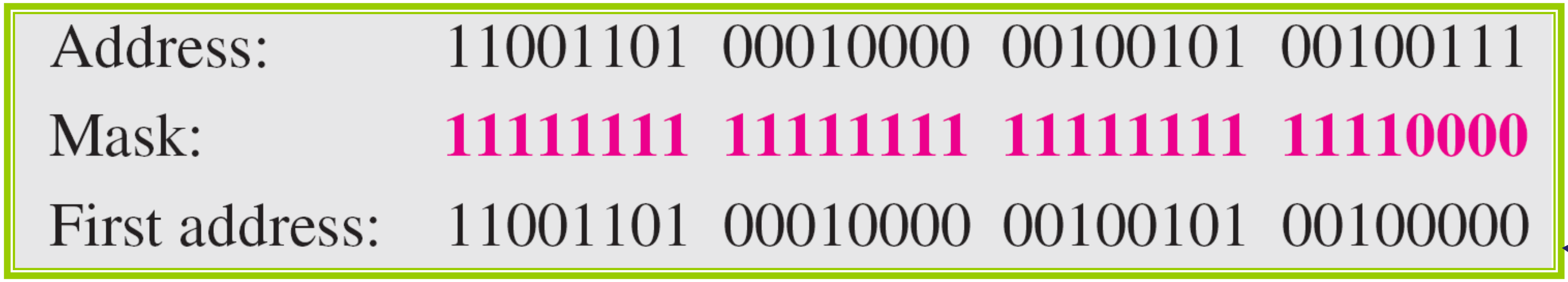
a. The first address

b. The last address

c. The number of addresses.

Solution

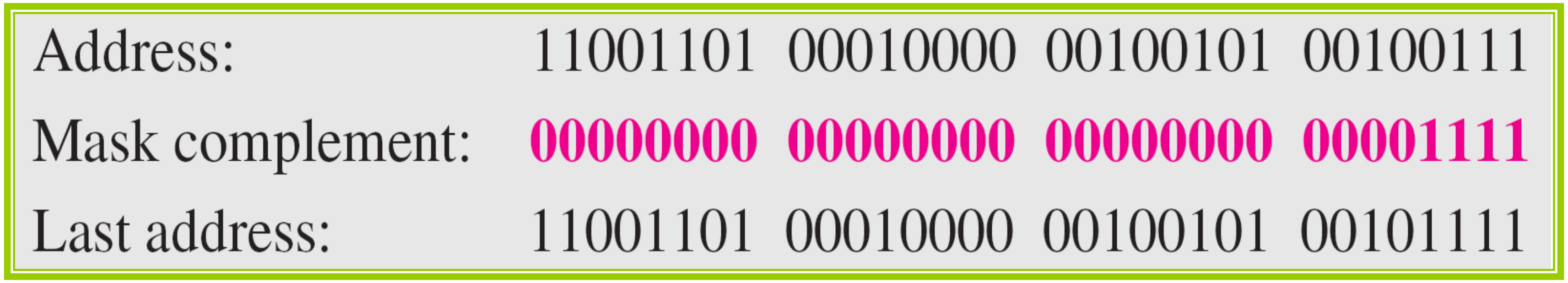
a. **The first address can be found by ANDing the given addresses with the mask. ANDing here is done bit by bit. The result of ANDing 2 bits is 1 if both bits are 1s; the result is 0 otherwise.**



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

205.16.37.32

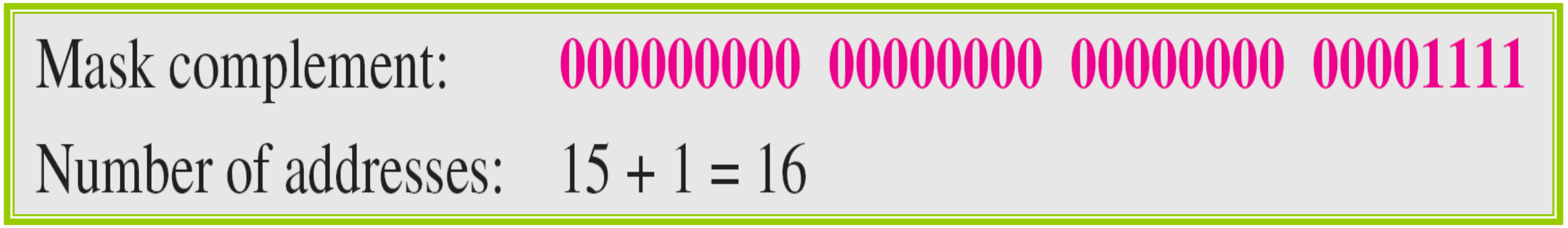
b. **The last address can be found by ORing the given addresses with the complement of the mask. ORing here is done bit by bit. The result of ORing 2 bits is 0 if both bits are 0s; the result is 1 otherwise. The complement of a number is found by changing each 1 to 0 and each 0 to 1.**



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

205.16.37.47

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



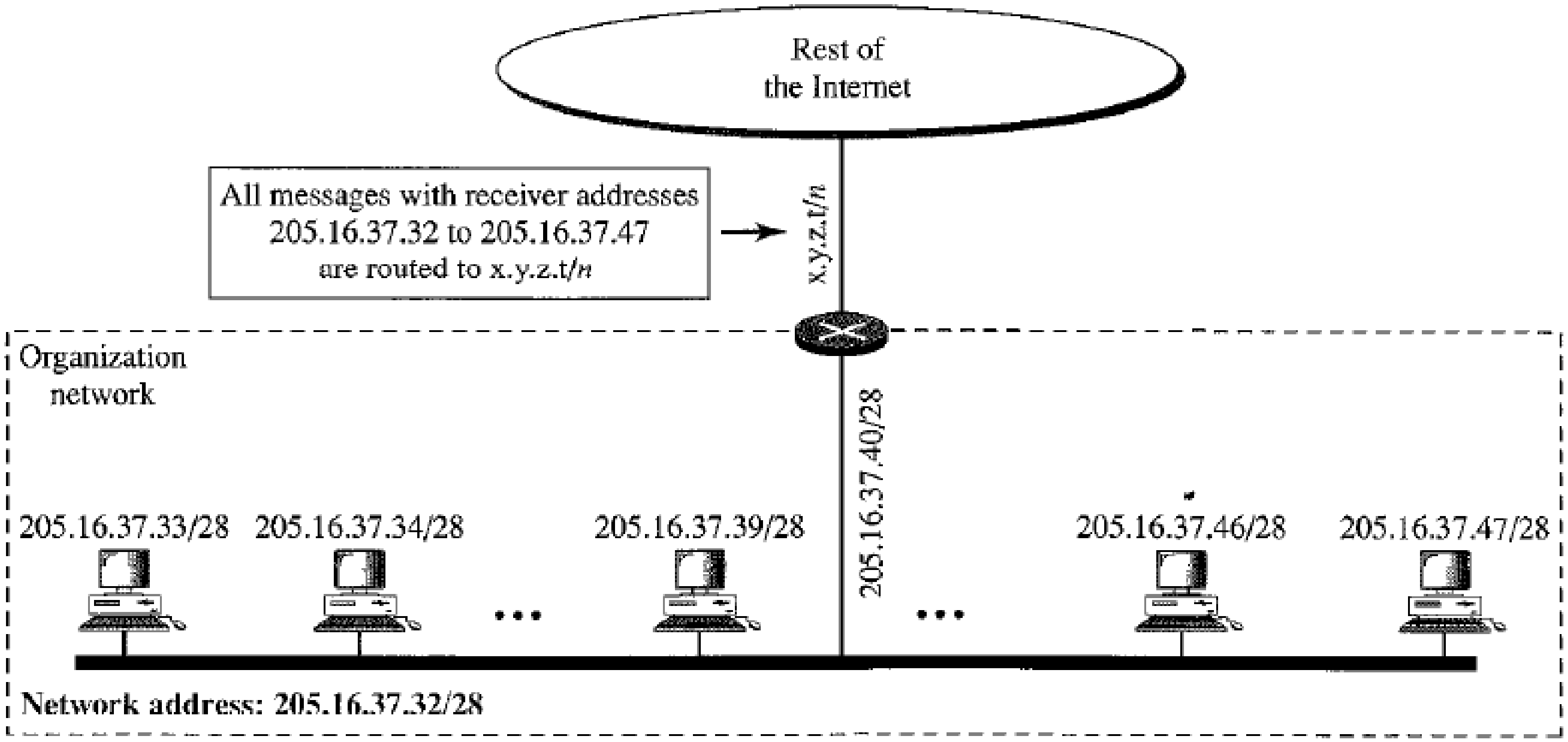
Mask complement:	000000000 00000000 00000000 00001111
Number of addresses:	15 + 1 = 16

Chapter 2 Part 2

Network Layer:
Logical Addressing- Subnetting

Dr. Lway Faisal

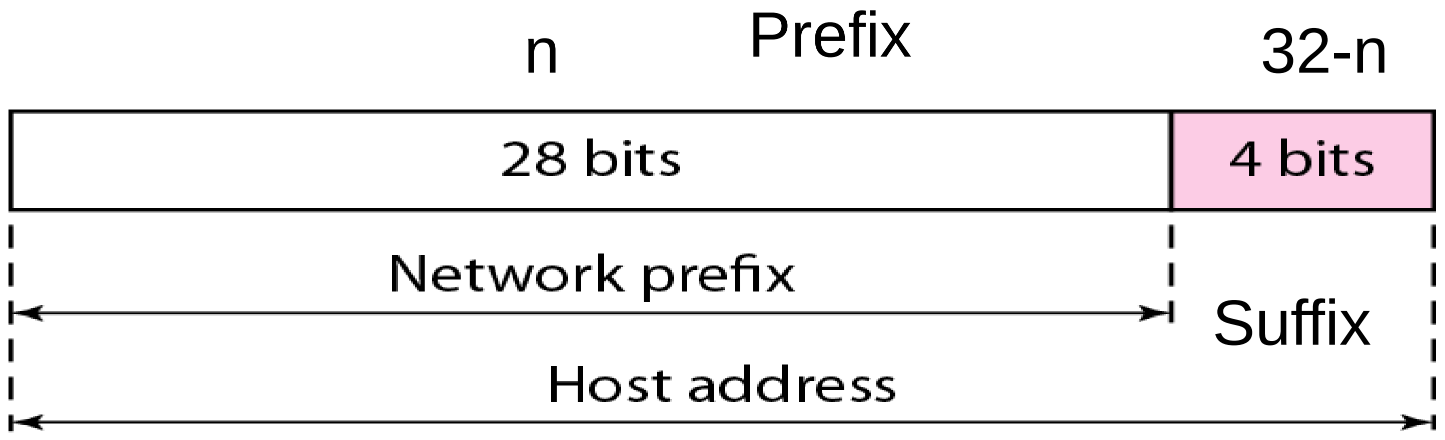
Network Addresses



The first address in a block is normally not assigned to any device; it is used as the network address that represents the organization to the rest of the world.

Two-Level Hierarchy: No Subnetting

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



Three-Levels of Hierarchy: Subnetting

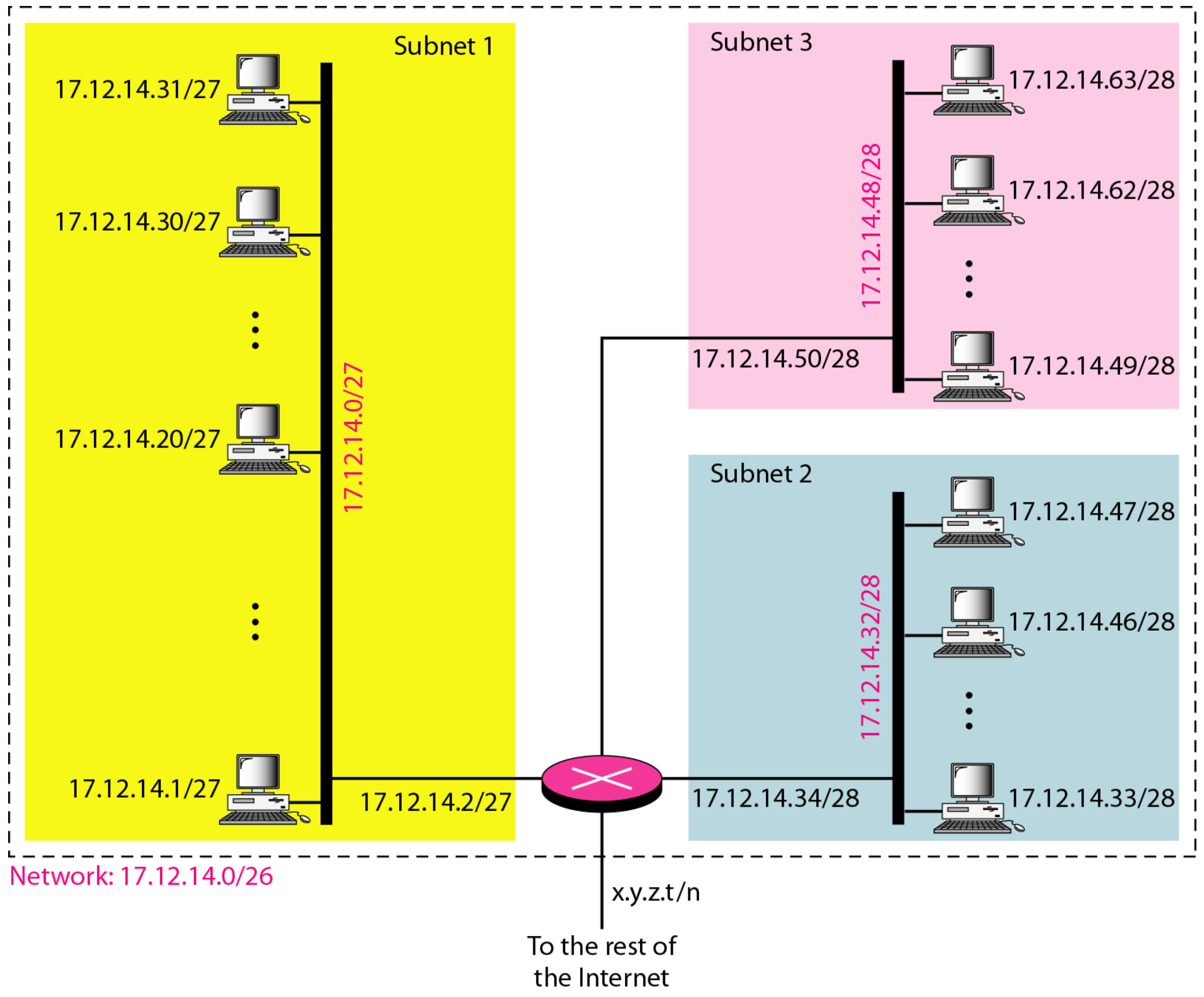
An organization that is granted a large block of addresses may want to create clusters of networks (called subnets) and divide the addresses between the different subnets.

The organization, however, needs to create small sub blocks of addresses, each assigned to specific subnets. The organization has its own mask; each subnet must also have its own.

As an example, suppose an organization is given the block 17.12.14.0/26, which contains 64 addresses. The organization has three offices and needs to divide the addresses into three sub blocks of 32, 16, and 16 addresses. We can find the new masks by using the following arguments:

1. Suppose the mask for the first subnet is n1, then 2^{32-n1} must be 32, which means that n1 =27.
2 . Suppose the mask for the second subnet is n2, then 2^{32-n2} must be 16, which means that n2 = 28.
3. Suppose the mask for the third subnet is n3, then 2^{32-n3} must be 16, which means that n3 =28.

Configuration and addresses in a subnetted network



Let us check to see if we can find the subnet addresses from one of the addresses in the subnet.

a. **In subnet 1**, the address 17.12.14.29/27 can give us the subnet address if we use the mask /27 because
Host: 00010001. 00001100. 00001110. 00011101
Mask:11111111 11111111 11111111 11100000 /27 (AND)
Subnet: 00010001 00001100 00001110 00000000 ....
(17.12.14.0)

b. **In subnet 2**, the address 17.12.14.45/28 can give us the subnet address if we use the mask /28 because
Host: 00010001 00001100 00001110 00101101
Mask:11111111 11111111 11111111 11110000 /28 (AND)
Subnet: 00010001 00001100 00001110 00100000 ....
(17.12.14.32)

c. **In subnet 3**, the address 17.12.14.50/28 can give us the subnet address if we use the mask /28 because
Host: 00010001 00001100 00001110 00110010
Mask:11111111 11111111 11111111 11110000 /28 (AND)
Subnet: 00010001 00001100 00001110 00110000 ....
(17.12.14.48)

We can say that through subnetting, we have three levels of hierarchy.

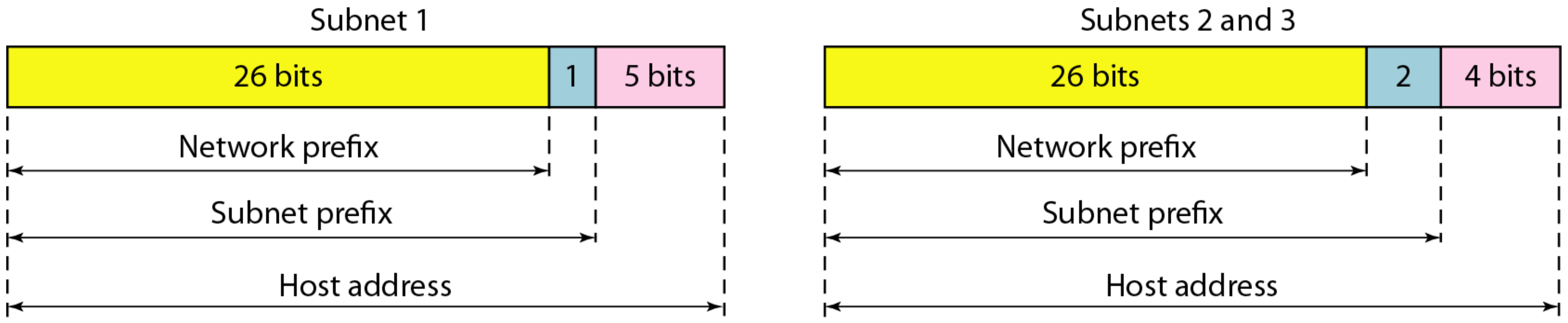
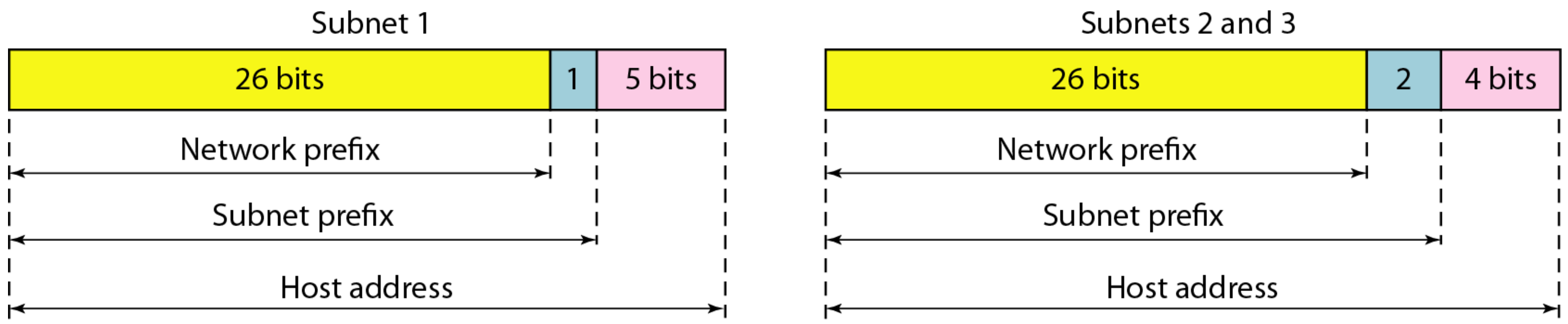


Figure 2.8 ***Three-level hierarchy in an IPv4 address***

Q/Analyze three – level of hierarchy as shown below for this network address 17.12.14.0/26 and draw the network diagram?



More Levels of Hierarchy

Large Block →Divide into →Small Blocks →Divide into→ Sub Blocks → Customers

National ISP → Regional ISP → Local ISP → Organization → Several Sub nets.

Address Allocation

How are the blocks allocated? The ultimate responsibility of address allocation is given to a global authority called the *Internet Corporation for Assigned Names and Numbers* (ICANN). However, ICANN does not normally allocate addresses to individual organizations. It assigns a large block of addresses to an ISP. Each ISP, in turn, divides its assigned block into smaller sub blocks and grants the sub blocks to its customers.

ICANN →National ISP → Regional ISP → Local ISP → Organization → Several Sub nets.

Example

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.

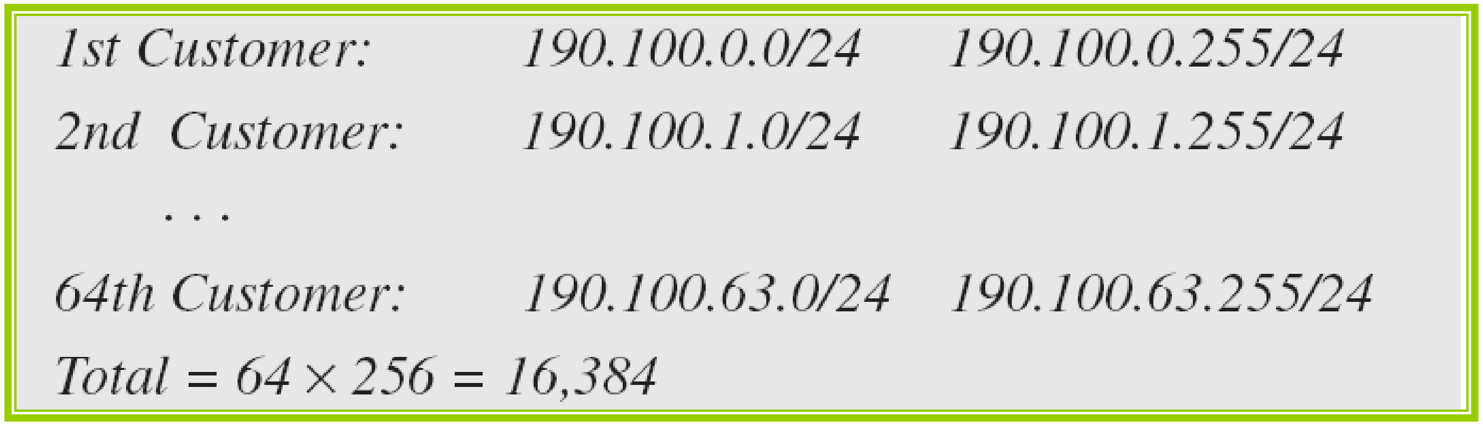
Design the sub blocks and find out how many addresses are still available after these allocations?

Solution

Figure below shows the situation.

Group 1

For this group, each customer needs 256 addresses. This means that 8 ($\log_2$ 256) bits are needed to define each host. The prefix length is then 32 − 8 = 24. The addresses are



1st Customer:	190.100.0.0/24	190.100.0.255/24
2nd Customer:	190.100.1.0/24	190.100.1.255/24
. . .		
64th Customer:	190.100.63.0/24	190.100.63.255/24

Total = 64 × 256 = 16,384

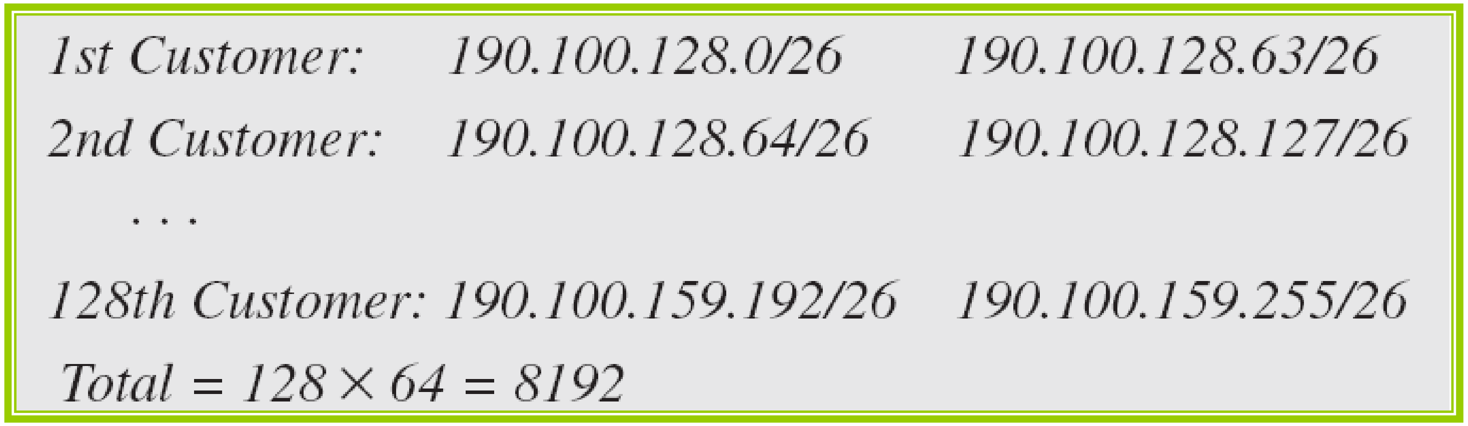
Group 2

For this group, each customer needs 128 addresses. This means that 7 ($\log_2$ 128) bits are needed to define each host. The prefix length is then 32 − 7 = 25. The addresses are

1st Customer:	*190.100.64.0/25*	*190.100.64.127/25*
2nd Customer:	*190.100.64.128/25*	*190.100.64.255/25*
. . .		
128th Customer:	*190.100.127.128/25*	*190.100.127.255/25*
Total = 128 × 128 = 16,384		

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

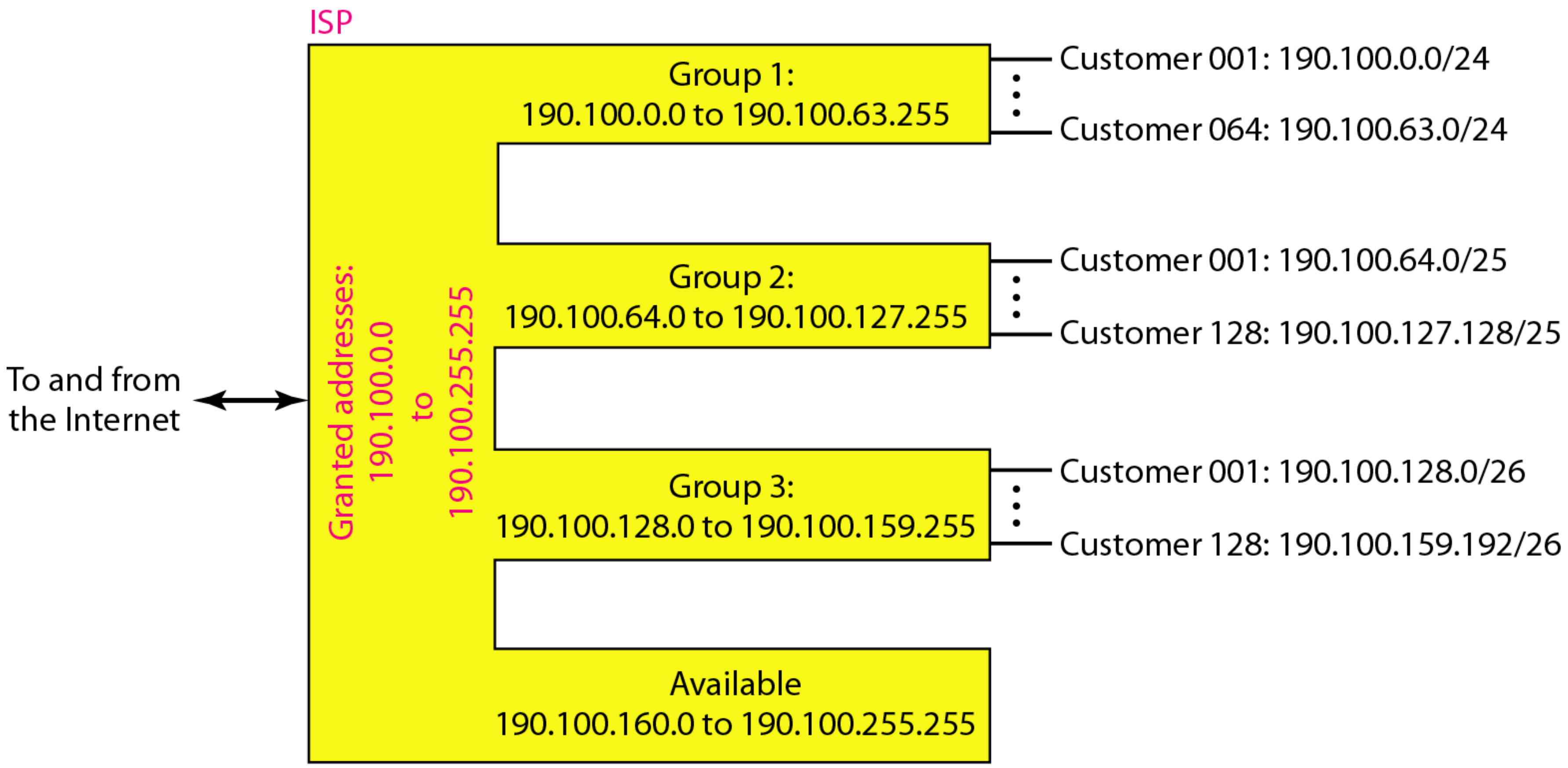


1st Customer:	*190.100.128.0/26*	*190.100.128.63/26*
2nd Customer:	*190.100.128.64/26*	*190.100.128.127/26*
. . .		
128th Customer:	*190.100.159.192/26*	*190.100.159.255/26*

Total = 128 × 64 = 8192

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

An example of address allocation and distribution by an ISP


ISP
To and from the Internet
Granted addresses: 190.100.0.0 to 190.100.255.255
Group 1: 190.100.0.0 to 190.100.63.255
Customer 001: 190.100.0.0/24
Customer 064: 190.100.63.0/24
Group 2: 190.100.64.0 to 190.100.127.255
Customer 001: 190.100.64.0/25
Customer 128: 190.100.127.128/25
Group 3: 190.100.128.0 to 190.100.159.255
Customer 001: 190.100.128.0/26
Customer 128: 190.100.159.192/26
Available 190.100.160.0 to 190.100.255.255

Network Address Translation (NAT)

A technology that allows a private network to use a set of private addresses for internal communication and a set of global Internet addresses for external communication.

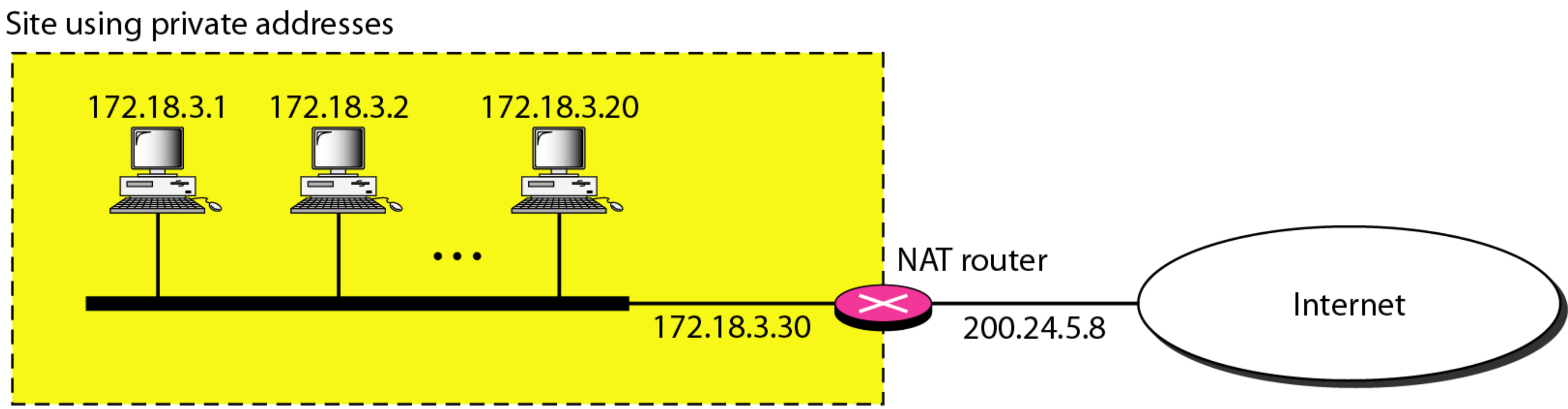
It provides a mapping between internal IP addresses and officially assigned external addresses.

The Internet authorities have reserved three sets of addresses as private addresses, shown below:

Range			*Total*
10.0.0.0	to	10.255.255.255	2^{24}
172.16.0.0	to	172.31.255.255	2^{20}
192.168.0.0	to	192.168.255.255	2^{16}

NAT enables a user to have a large set of addresses internally and one address, or a small set of addresses, externally. The traffic inside can use the large set; the traffic outside, the small set.

They are unique inside the organization, but they are not unique globally. No router will forward a packet that has one of these addresses as the destination address.

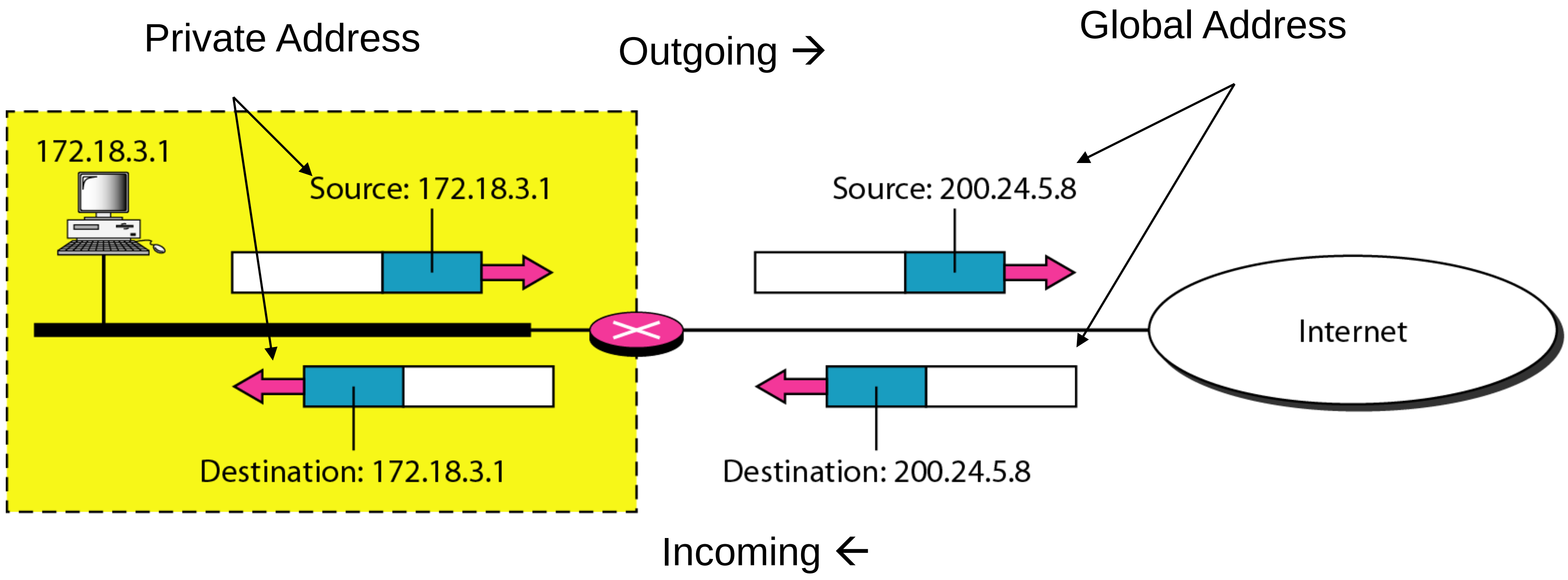


A NAT implementation

The router that connects the network to the global address uses one private address and one global address.

Address Translation

All the outgoing packets go through the NAT router, which replaces the *source address* in the packet with the global NAT address. All incoming packets also pass through the NAT router, which replaces the *destination address* in the packet with the appropriate private address.



Translation Table

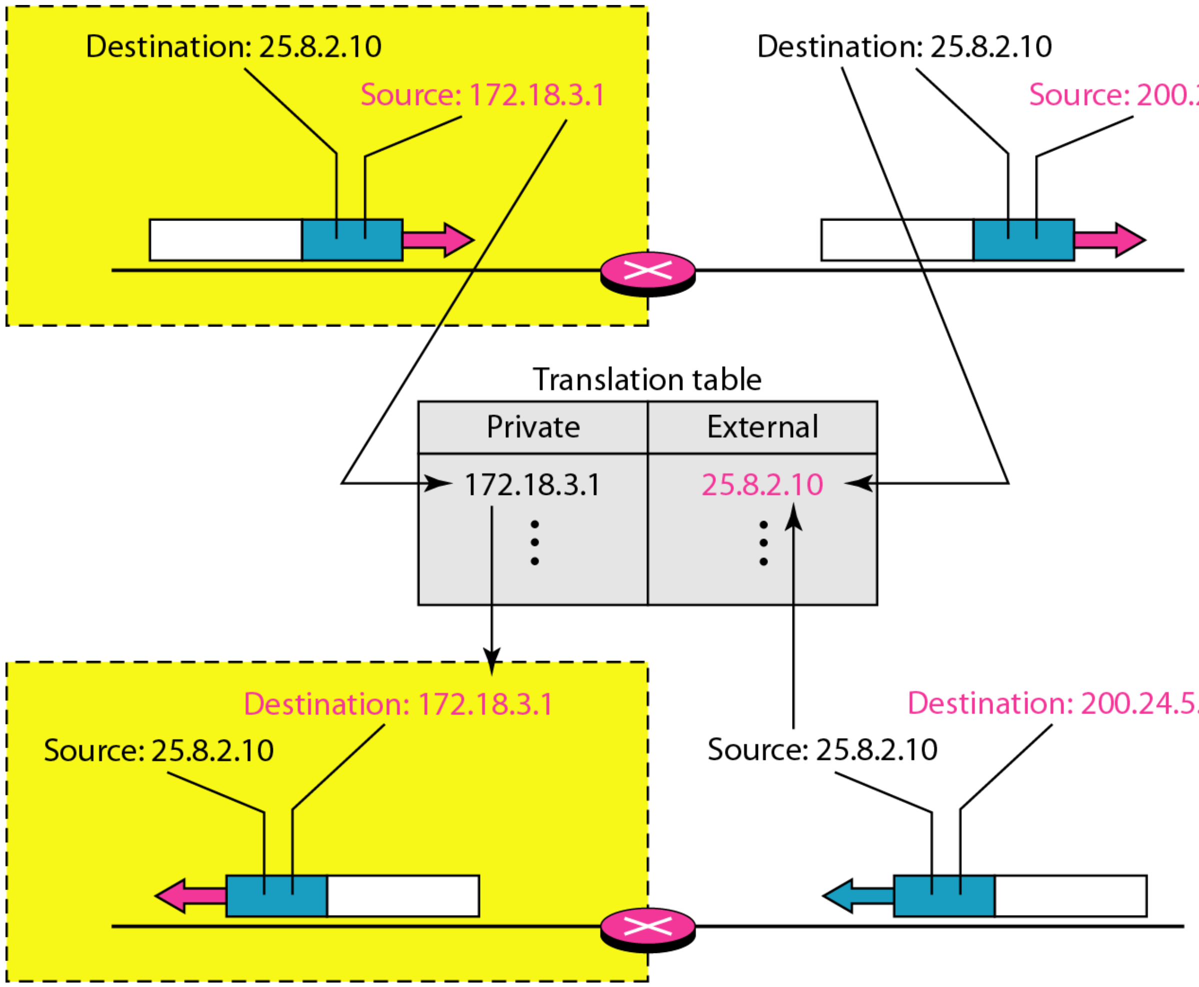
Translating the source addresses for outgoing packets is straightforward. But how does the NAT router know the destination address for a packet coming from the Internet? There may be tens or hundreds of private IP addresses, each belonging to one specific host. The problem is solved if the NAT router has a translation table.

The translation table configuration in NAT router has three types:

1. **Using One IP Address:**

A translation table has only two columns: the private address and the external address.

NAT Address Translation



In this strategy, communication must always be initiated by the private network. The NAT mechanism described requires that the private network start the communication.

1. *Using a Pool of IP Addresses*

Since the NAT router has only one global address, only one private network host can access the same external host.

To remove this restriction, the NAT router uses a pool of global addresses. For example, instead of using only one global address (200.24.5.8), the NAT router can use four addresses (200.24.5.8, 200.24.5.9, 200.24.5.10, and 200.24.5.11). In this case, four private network hosts can communicate with the same external host at the same time.

2. *Using Both IP Addresses and Port Numbers*

To allow a many-to-many relationship between private-network hosts and external server programs, we need more information in the translation table. For example, suppose two hosts with addresses 172.18.3.1 and 172.18.3.2 inside a private network need to access the HTTP server on external host 25.8.3.2.

If the translation table has five columns, instead of two, that include the source and destination port numbers of the transport layer protocol, the ambiguity is eliminated.

Table 2.4 ***Five-column translation table***

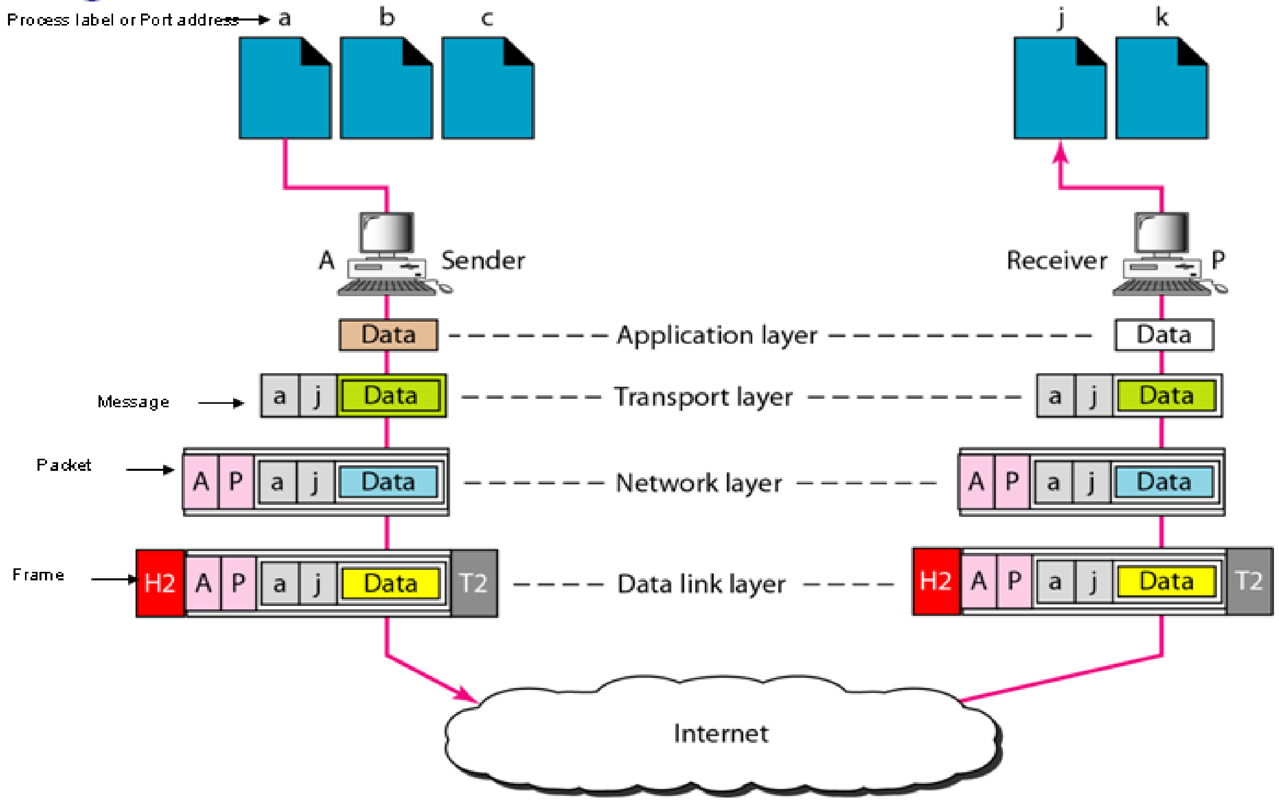
Private Address	*Private Port*	*External Address*	*External Port*	*Transport Protocol*
172.18.3.1	1400	25.8.3.2	80	TCP
172.18.3.2	1401	25.8.3.2	80	TCP
. . .	. . .	. . .	. . .	. . .

Transport Layer

The transport layer is responsible for the delivery of a message from one process to another.

Service-point addressing (Port Address): source-to-destination delivery means from a specific process (running program) on one computer to a specific process (running program) on the other. Such as http:// is Port 80, FTP:// is the port 21 and Telnet is the port 23

Figure 2.21 *Port addresses*



IPv6 ADDRESSES

Despite all short-term solutions, such as classless addressing but address depletion is still a long-term problem for the Internet.

This and other problems in the IP protocol itself such as lack of accommodation for real-time audio and video transmission, and encryption and authentication of data for some applications, have been the motivation for IPv6.

Structure

An IPv6 address consists of 16 bytes (octets); it is 128 bits long.

An IPv6 address is 128 bits long.

Hexadecimal Colon Notation

In this notation, 128 bits is divided into eight sections, each 2 bytes in length. Two bytes in hexadecimal notation requires four hexadecimal digits. Therefore, the address consists of 32 hexadecimal digits, with every four digits separated by a colon, as shown in Figure below.

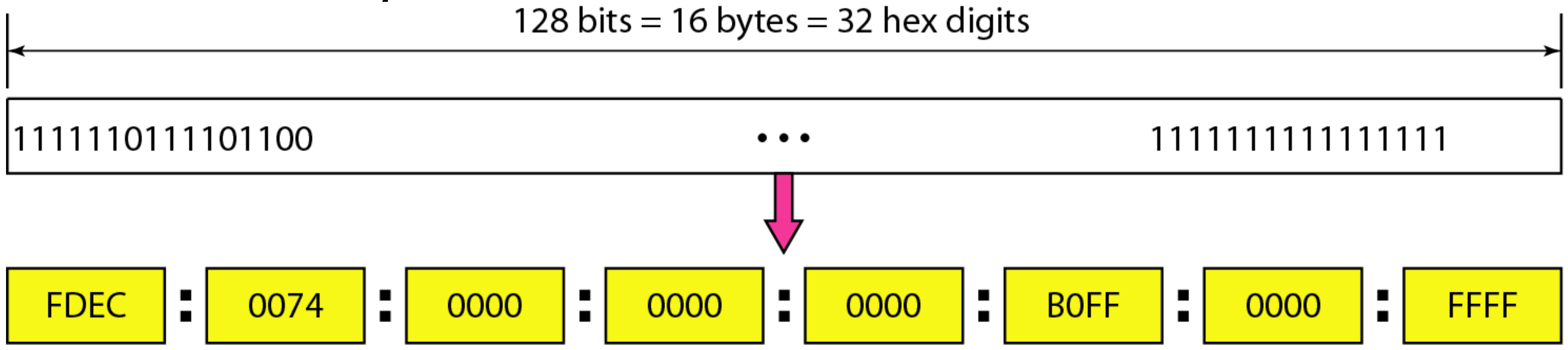

128 bits = 16 bytes = 32 hex digits
1111110111101100
• • •
1111111111111111
FDEC : 0074 : 0000 : 0000 : 0000 : B0FF : 0000 : FFFF

Figure 2.14 ***IPv6 address in binary and hexadecimal colon notation***

Abbreviation

Although the IP address, even in hexadecimal format, is very long, many of the digits are zeros. In this case, we can abbreviate the address. The leading zeros of a section (four digits between two colons) can be omitted. Only the leading zeros can be dropped, not the trailing zeros (see Figure 6.15).

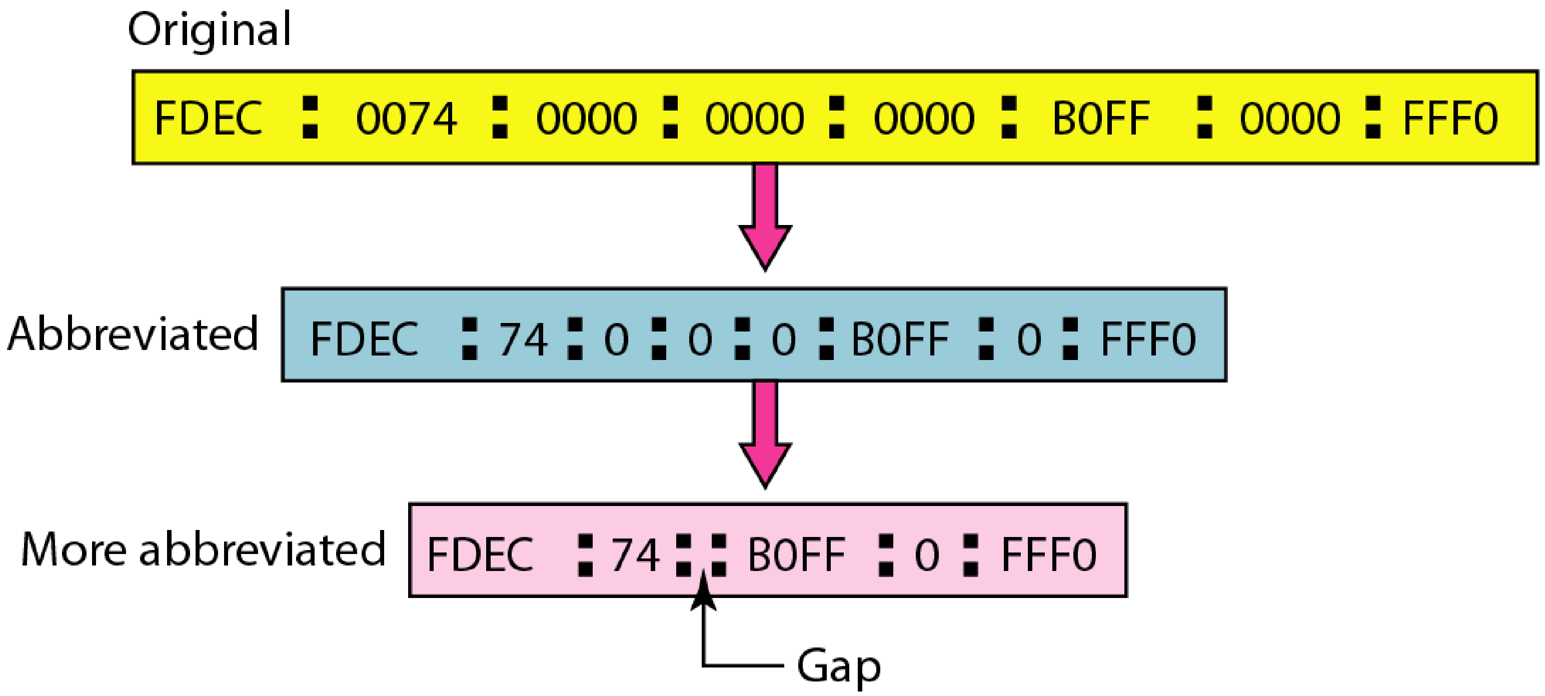

Original
FDEC : 0074 : 0000 : 0000 : 0000 : B0FF : 0000 : FFF0
Abbreviated
FDEC : 74 : 0 : 0 : 0 : B0FF : 0 : FFF0
More abbreviated
FDEC : 74 :: B0FF : 0 : FFF0
Gap

Figure 2.15 ***Abbreviated IPv6 addresses***

Using this form of abbreviation, 0074 can be written as 74, 000F as F, and 0000 as 0.Note that 3210 cannot be abbreviated.

We can remove the zeros altogether and replace them with a double colon. Note that this type of abbreviation is allowed only once per address.

Example

Expand the address 0:15::1:12:1213 to its original.

Solution

We first need to align the left side of the double colon to the left of the original pattern and the right side of the double colon to the right of the original pattern to find how many 0s we need to replace the double colon.

0:15::1:12:1213

```
xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx
   0:  15:               :   1:  12:1213
```

This means that the original address is.

```
0000:0015:0000:0000:0000:0001:0012:1213
```