COMPUTER NETWORK (BCA301)

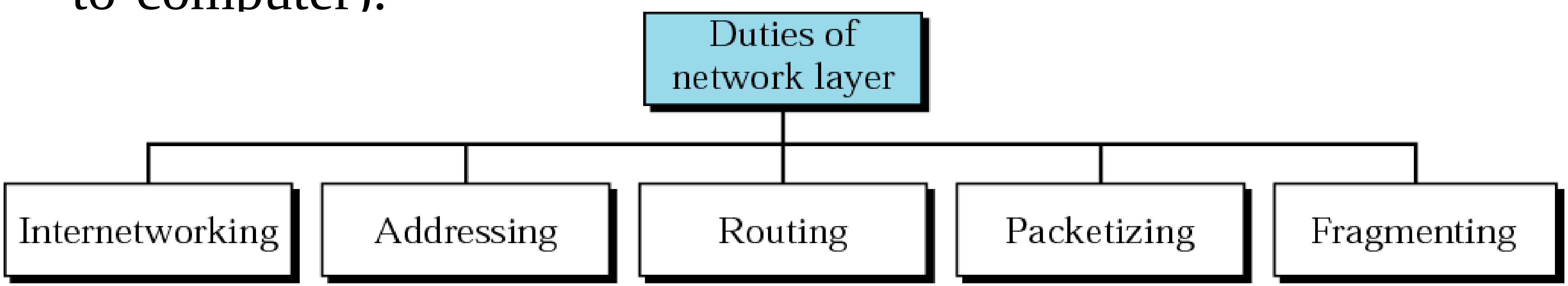
DEPARTMENT OF COMPUTER SCIENCE

PROGRAMME: BCA

CENTRAL UNIVERSITY OF ORISSA
KORAPUT

NETWORK LAYER

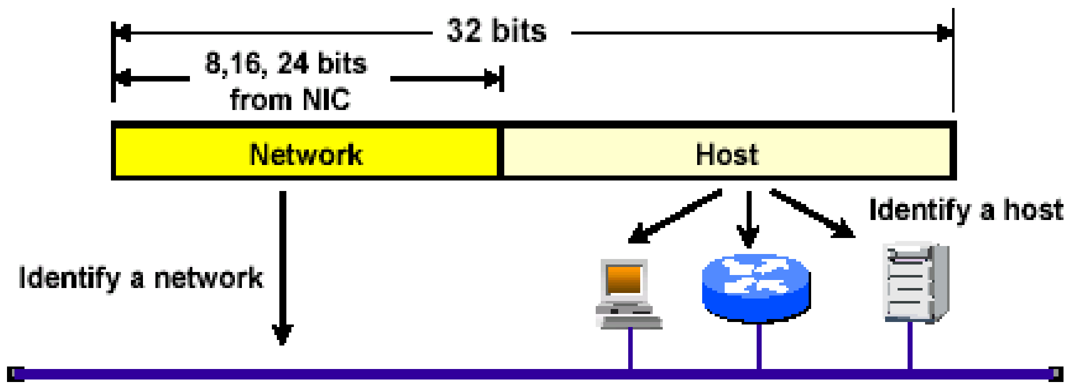
- The network layer is responsible for the source-to-destination delivery of a packet, possibly across multiple networks (links). Whereas the data link layer oversees the delivery of the packet between two systems on the same network (links).
- The network layer adds a header that includes the logical addresses of the sender and receiver to the packet corning from the upper layer.
- Communication at the network layer is host-to-host (computer-to-computer).



- How to connect multiple devices or to form a large network?
  - Mesh topology
  - Star topology
- Both are <u>not practical and costly</u> when applied to very large networks because of:
    - Number of links
    - Length of the links
  - Many links will stay idle for most of the time
- Solution: Switched networks

IPv4 ADDRESSES

- The address in the network layer of the TCP/IP model is called Internet Address or IP address
- 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.
- On the other hand, if a device operating at the network layer has m connections to the Internet, it needs to have m addresses.



- Network + Host: Complete IP address
- Network Address: Host part set to 0
- Network ID: identifies the network to which the host is connected
- Host ID: identifies the interface of the network connection to the host not the host itself

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 because each bit can have two different values (0 or 1) and N bits can have 2^N values.
- IPv4 uses 32-bit addresses, which means that the address space is 2^{32} or 4,294,967,296 (more than 4 billion).

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.

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.

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

EXAMPLE

- 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 and add dots for separation.

  a. 129.11.11.239

  b. 193.131.27.255

EXAMPLE

- 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.

a. 01101111 00111000 00101101 01001110

b. 11011101 00100010 00000111 01010010

EXAMPLE

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

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).

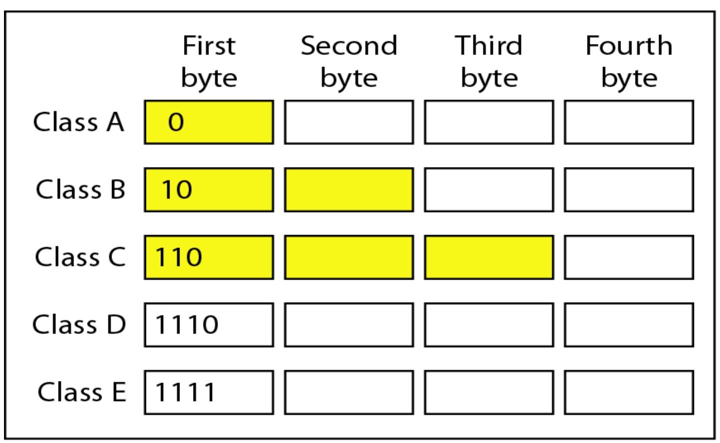
b. There can be no more than four numbers.

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

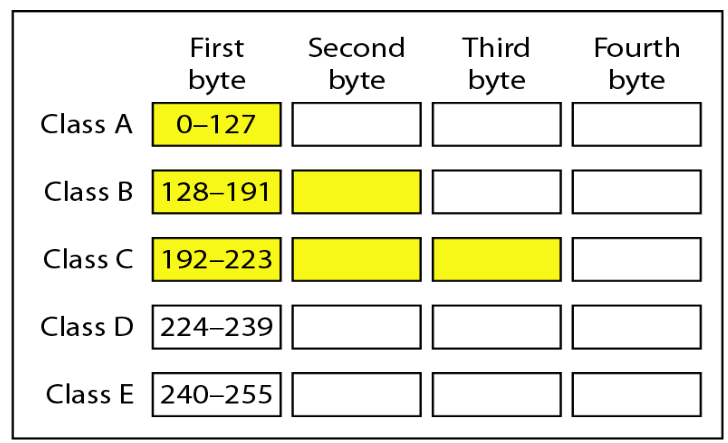
d. A mixture of binary notation and dotted-decimal notation is not allowed.

CLASSFUL ADDRESSING

- IPv4 addressing, at its inception, used the concept of classes.
- This architecture is called classful addressing. Although this scheme is becoming obsolete.
- In classful addressing, the address space is divided into five classes: A, B, C, D, and E.
- If the address is given in binary notation, the first few bits can immediately tell us the class of the address. If the address is given in decimal-dotted notation, the first byte defines the class.

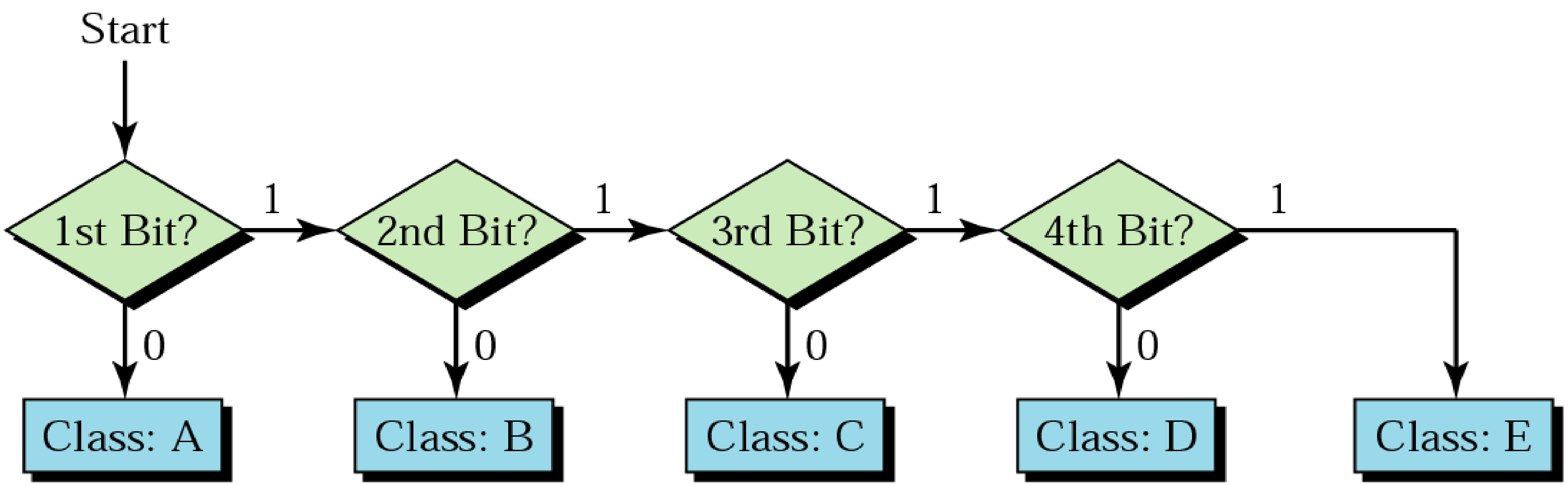


a. Binary notation



b. Dotted-decimal notation

Finding the address class



Classful Addressing

- Class A
  - Start with binary 0
  - All 0 reserved (default route) or any network
  - 01111111 (127) reserved for **loopback**
  - 2^{31} or 2,147,483,648 **class A complete IP addresses**
  - 2^7 =128 blocks (network addresses)
  - Number of complete IP addresses in **each block** is 2^{24}=16777216 – (all zeros host **- *network address***, and all ones – ***broadcast address***)
  - Valid Range 1.x.x.x to 126.x.x.x (126 valid blocks)
  - All allocated

- Class B
  - Start with binary 10
  - Range 128.x.x.x to 191.x.x.x
  - 2^{30} class B **complete IP addresses**
  - 2^{14}=16384 blocks (network addresses)
  - Number of addresses in each block is 2^{16}=65536 – (all zeros host, and all ones)
  - All allocated

Classful Addressing

- Class C
  - Start with binary 110
  - Range 192.x.x.x to 223.x.x.x
  - 2^{29} Class C complete IP addresses
  - 2^{21}=2097152 blocks (network addresses)
  - Number of addresses in each block is 256 – (all zeros host, and all ones) class
  - Nearly all allocated
- Class D
  - Multicast addresses
  - No network/host hierarchy

Private addresses

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}

EXAMPLE

- 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

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.

CLASSES AND BLOCKS

- Each class is divided into a fixed number of blocks with each block having a fixed size.
- The classful addressing wastes a large part of the address space.
  - Class A:
  - Class B:
  - Class C:
  - Class D:

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

- Previously, when an organization requested a block of addresses, it was granted one in class A, B, or C.
- 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 tens of thousands of attached hosts or routers.
- Class C addresses were designed for small organizations with a small number of attached hosts or routers.
- A block in class A address is too large for almost any organization.
- This means most of the addresses in class A were wasted and were not used.
- A block in class B is also very large, probably too large for many of the organizations that received a class B block.
- A block in class C is probably too small for many organizations.
- Class D addresses were designed for multicasting.
- The class E addresses were reserved for future use; only a few were used, resulting in another waste of addresses.
- NOTE: In c1assfUl addressing, a large part of the available addresses were wasted.

Netid and hostid

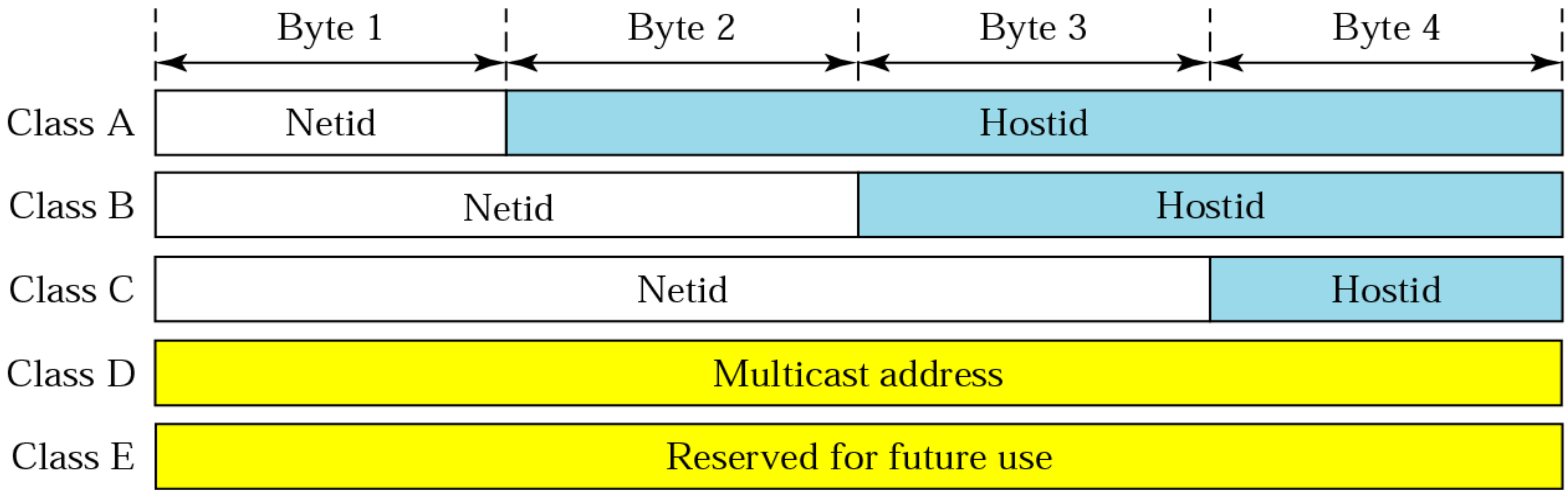
- In classful addressing, an IP address in class A, B, or C is divided into netid and hostid.
- These parts are of varying lengths, depending on the class of the address.
- The concept does not apply to classes D and E.
- In class A, one byte defines the netid and three bytes define the hostid.
- In class B, two bytes define the netid and two bytes define the hostid.
- In class C, three bytes define the netid and one byte defines the hostid.

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

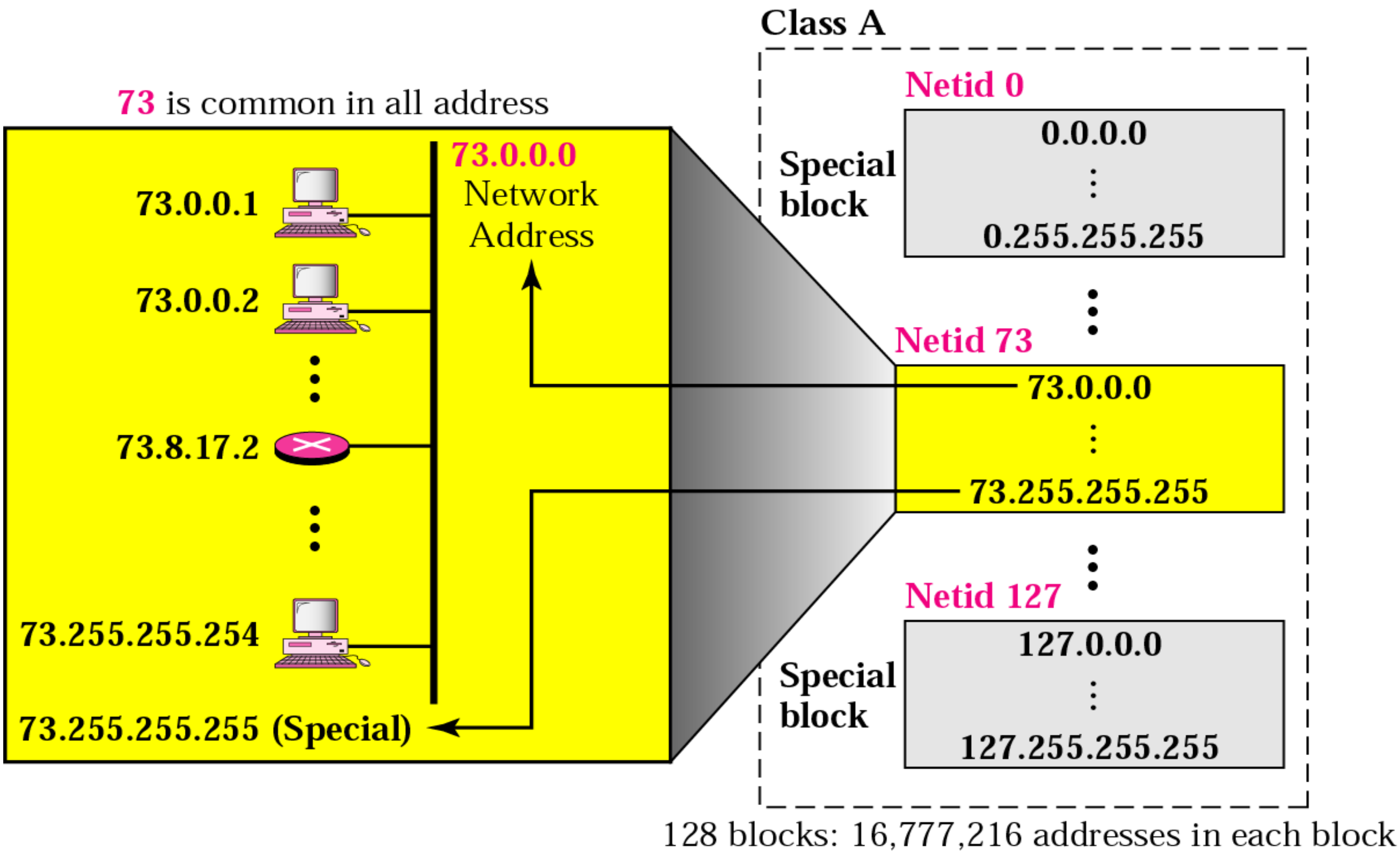
Netid and hostid

A network address is different from a netid. A network address has both netid and hostid, with 0s for the hostid.

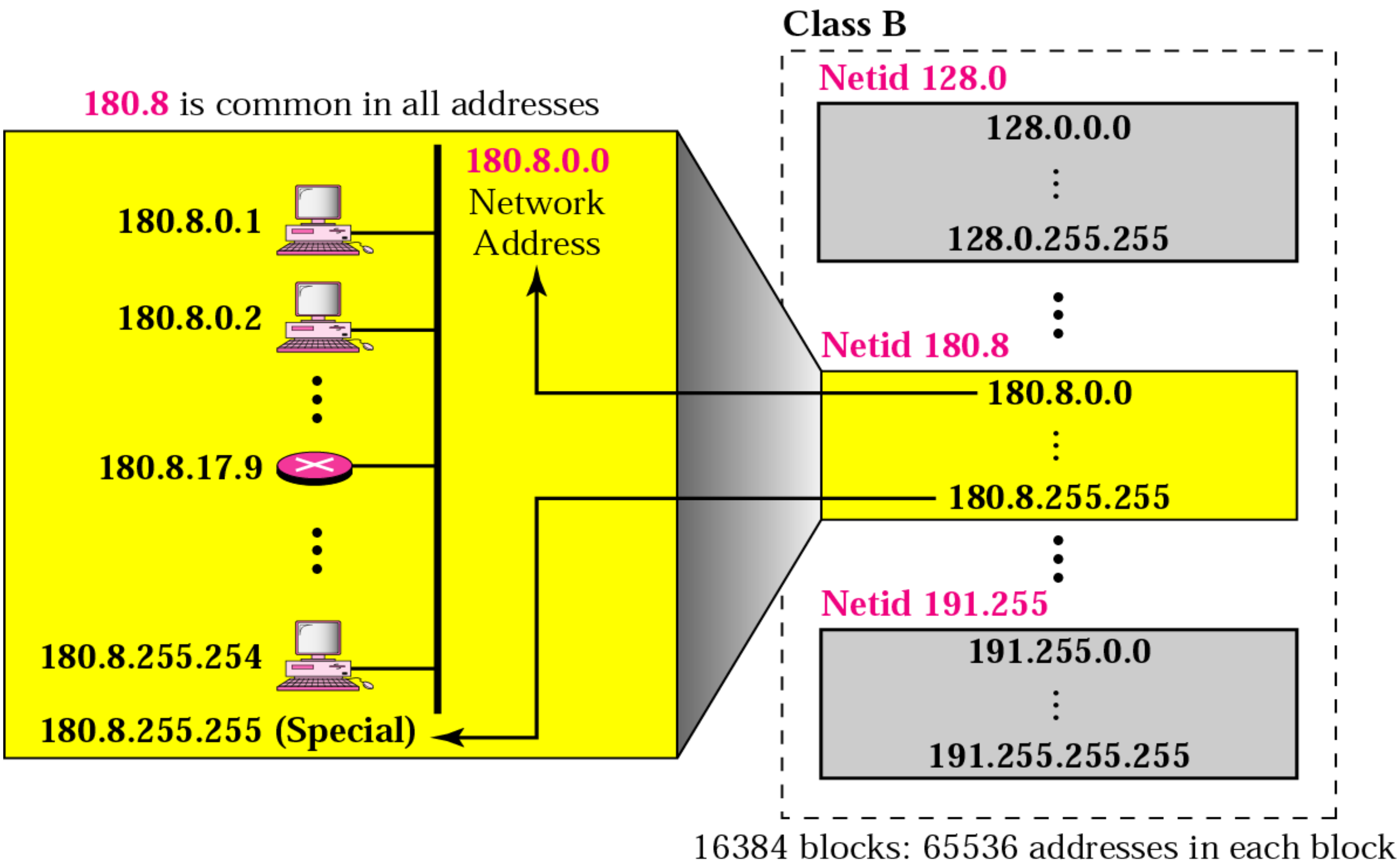
In classful addressing, the network address is the one that is assigned to the organization.



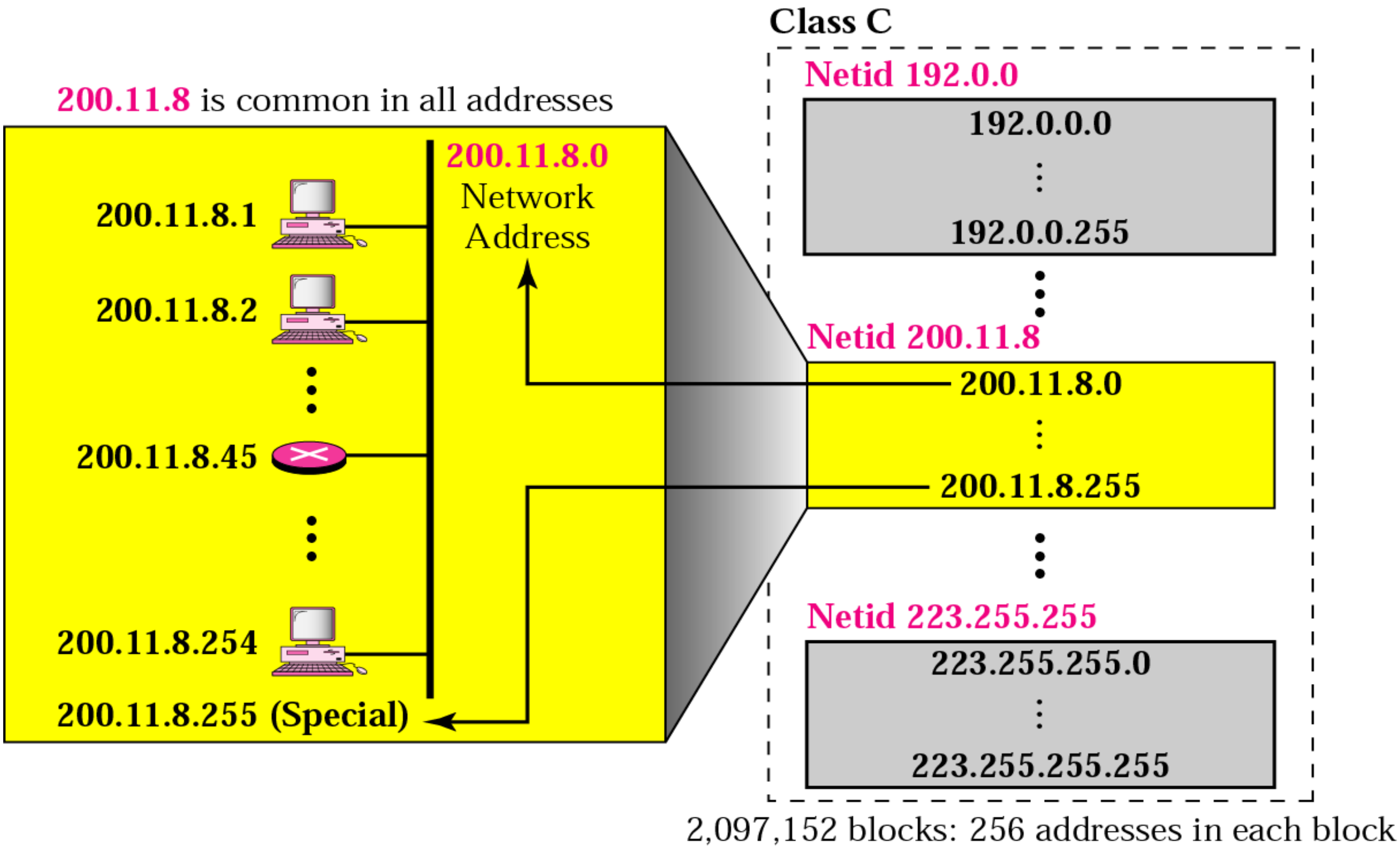
Blocks in class A



Blocks in class B



Blocks in class C



Mask

- Mask
  - 32-bit number of contiguous 1's followed by contiguous 0's.
  - To help to find the net ID and the host ID.

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

This notation is also called slash notation or Classless Interdomain Routing (CIDR) notation.

The notation is used in classless addressing.

EXAMPLE

Q. Given the address 23.56.7.91, find the network address?

Solution:

The class is A. Only the first byte defines the netid. We can find the network address by replacing the hostid bytes (56.7.91) with 0s. Therefore, the network address is 23.0.0.0.

Q. Given the address 132.6.17.85, find the network address.

Solution:

The class is B. The first 2 bytes defines the netid. We can find the network address by replacing the hostid bytes (17.85) with 0s. Therefore, the network address is 132.6.0.0.

Q. Given the network address 17.0.0.0, find the class.

Solution:

The class is A because the netid is only 1 byte.

SUBNETTING VS SUPERNETTING

- **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)

Subnetting increases the number of 1s in the mask.

- **Supernetting:**

Exhausted class A and B address space

Huge demand for class B address space

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 or a supernet.

An organization can apply for a set of class C blocks instead of just one.

Supernetting decreases the number of 1s in the mask

To combine several contiguous address spaces into a larger single address space

Subnetting

- Dividing the network into several smaller groups (subnets) with each group having its own **subnet IP address**
- Site looks to rest of internet like **single network** and routers outside the organization route the packet based on the main Network address
- Local routers route within subnetted network using subnet address
- Host portion of address partitioned into subnet number (most significant part) and host number (least significant part)
- In this case, IP address will have **3 levels** (Main network, subnet, host)
- **Subnet mask** is a 32-bit consists of zeros and ones that indicates which bits of the IP address are subnet number and which are host number
- Subnet mask when ANDed with the IP address it gives the subnetwork address

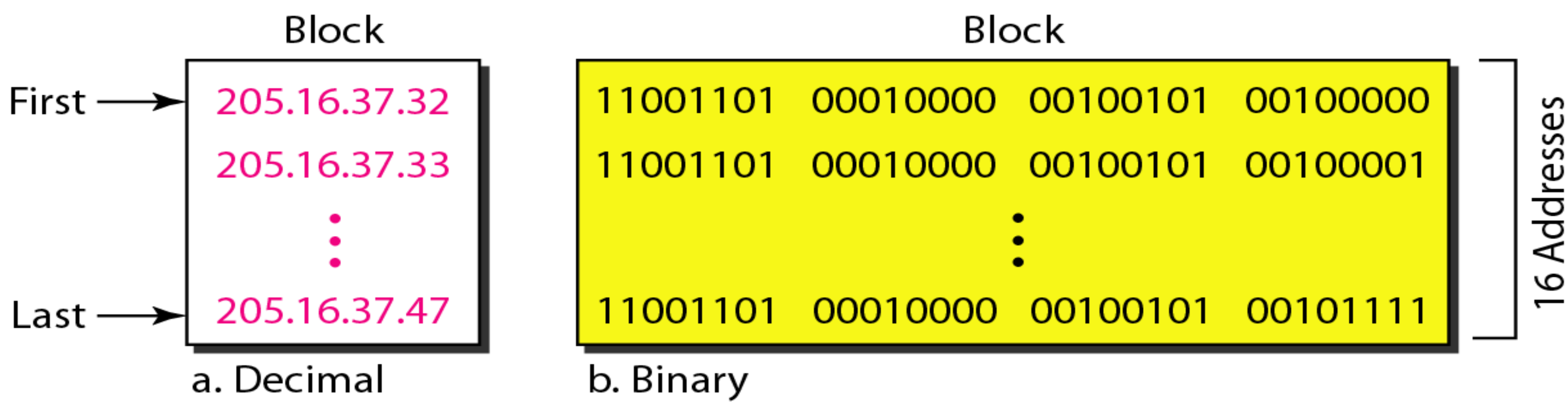
CLASSLESS ADDRESSING

- To overcome address depletion and give more organizations access to the Internet, classless addressing was designed and implemented.
- There are no classes, but the addresses are still granted in blocks.
- **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, ...).
3. The first address must be evenly divisible by the number of addresses.

A block of 16 addresses granted to a small organization

- 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 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 the 32 - *n* rightmost bits are 0s.
- However, in classless addressing the mask for a block can take any value from 0 to 32.
- It is very convenient to give just the value of *n* preceded by a slash (**CIDR notation**).
- 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.
- The **address** and the ***/n*** notation completely define the whole block (the first address, the last address, and the number of addresses).
- 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.
- 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 **number of addresses** in the block is the difference between the last and first address. It can easily be found using the formula 2^{32-n}.

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 00100111

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

11001101 00010000 00100101 0010000

or

205.16.37.32.

Example

Q. Find the last address for the block in *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

Q. Find the number of addresses in 205.16.37.39/28.

Solution:

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

Example

- 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 *205.16.37.39/28* can be represented as

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

- 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

- 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 ADDRESSES

- The first address in the class, however, is normally (not always) treated as a special address.
- The first address is called the network address and defines the organization network.
- It defines the organization itself to the rest of the world.
- Broadcast address: The last address in a block is used for broadcasting to all devices under the network.
-

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 Addresses* (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 subblocks and grants the subblocks to its customers.
- In other words, an ISP receives one large block to be distributed to its Internet users.
- This is called address aggregation: many blocks of addresses are aggregated in one block and granted to one ISP.

HIERARCHY

- **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 (organization network); the 32 – n rightmost bits define the particular host (computer or router) 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.

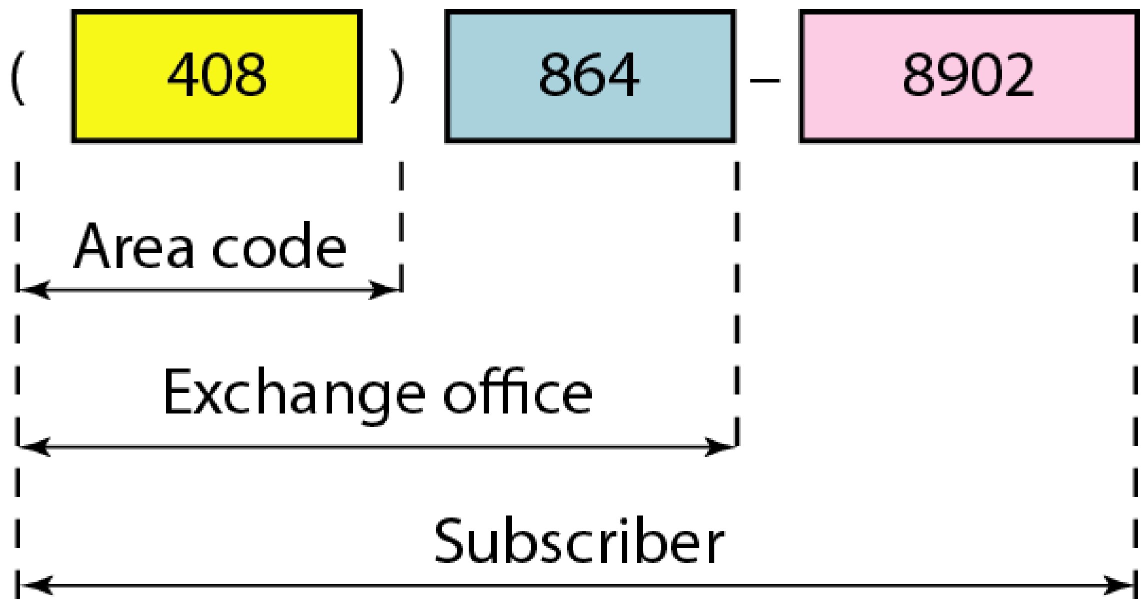
The prefix is common to all addresses in the network; the suffix changes from one device to another.

Three-Levels of Hierarchy: Subnetting

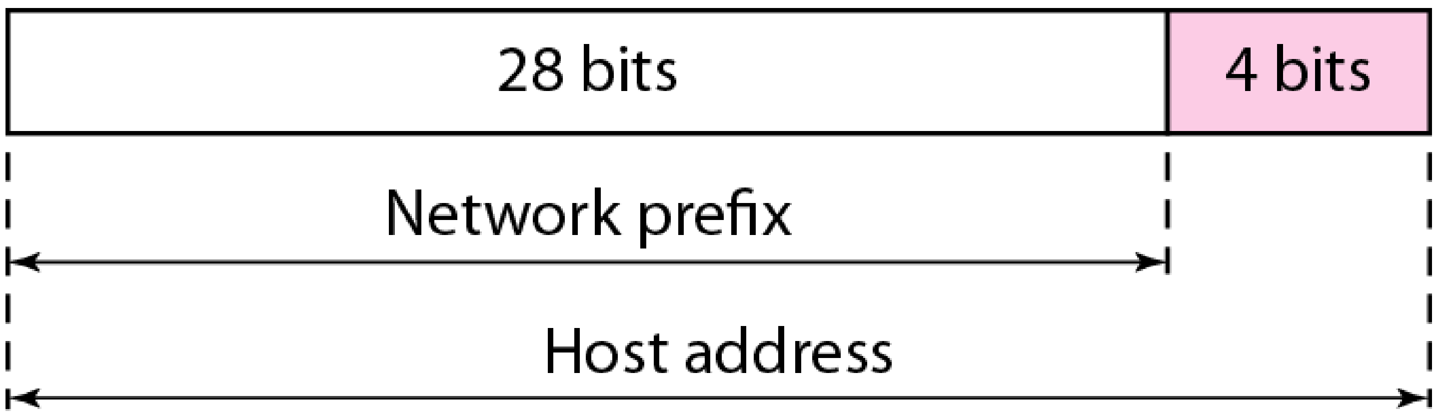
An organization that is granted a large block of addresses may want to create clusters of networks (called subnets) and divide the addresses between the different subnets.

The rest of the world still sees the organization as one entity; however, internally there are several subnets.

Two levels of hierarchy in an IPv4 address



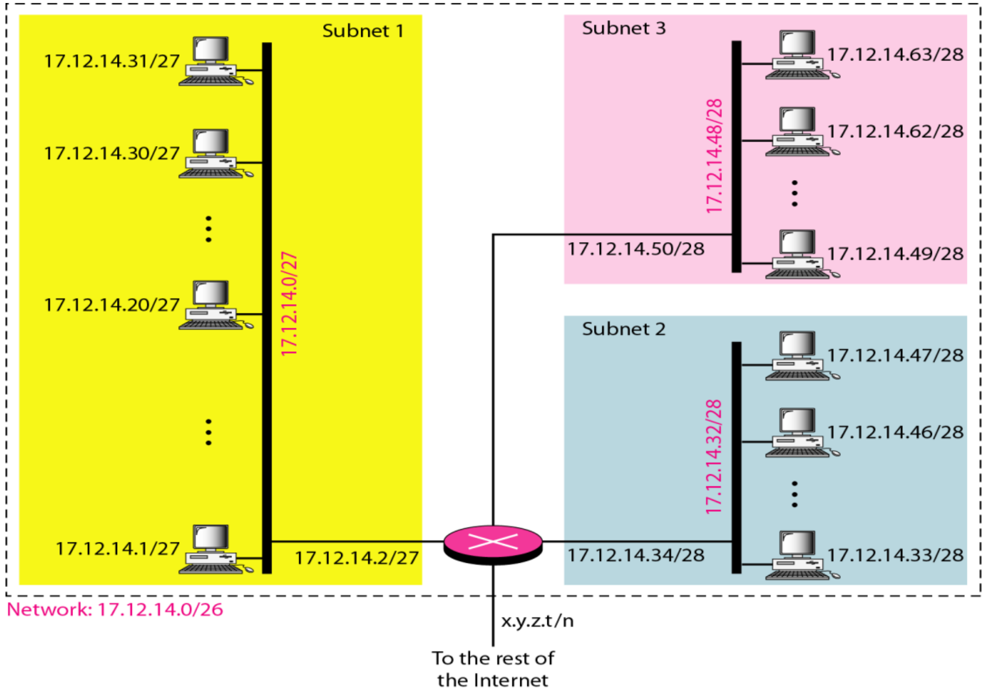
Two Level of Hierarchy

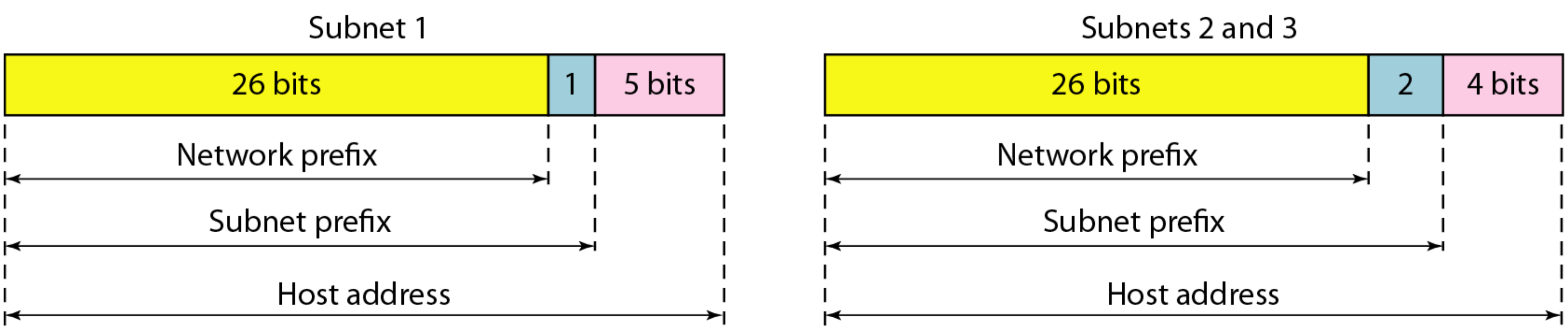


More Levels of Hierarchy

- The structure of classless addressing does not restrict the number of hierarchical levels.
- An organization can divide the granted block of addresses into subblocks.
- Each subblock can in turn be divided into smaller subblocks. And so on.

Three Levels of Hierarchy





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.

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

Solution:

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		

Solution:

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

Network Address Translation (NAT)

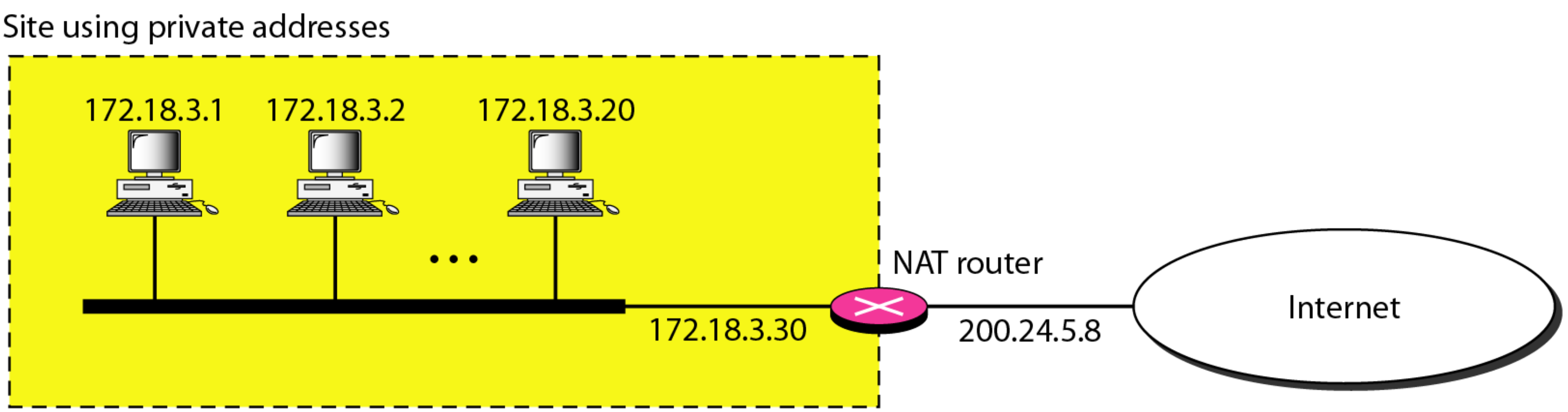
- NAT enables a user to have a large set of addresses internally and one address, or a small set of addresses, externally.
- Use of a single IP address among many devices in a network.
- Use of a dynamic IP address for home user for sharing.
- The traffic inside can use the large set; the traffic outside, the small set.
- **Private Addresses**

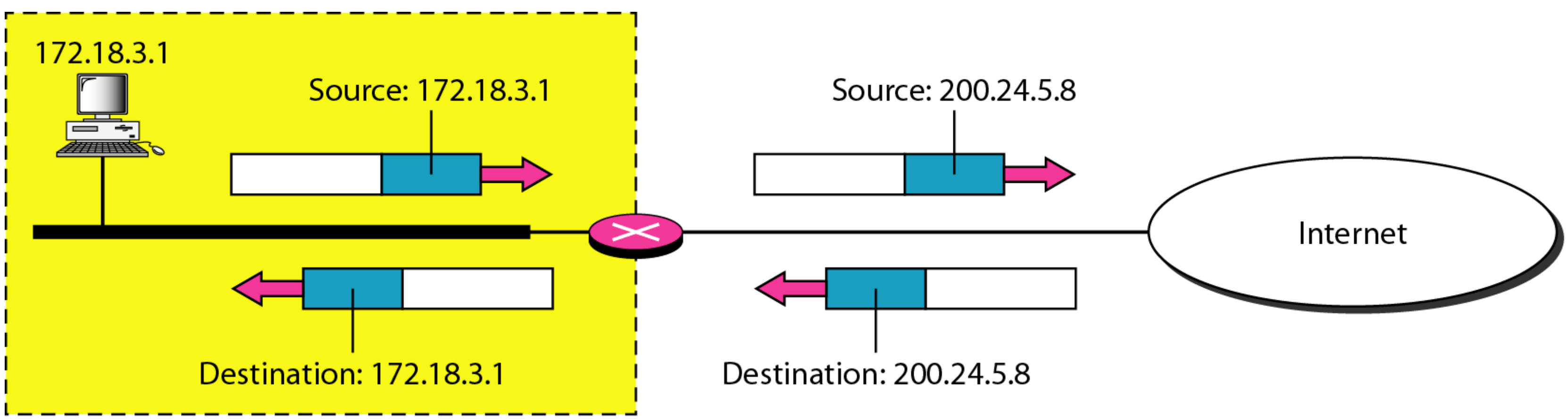
Range			*Total*
10.0.0.0	to	10.255.255.255	2^{24}
172.16.0.0	to	172.31.255.255	2^{20}
192.168.0.0	to	192.168.255.255	2^{16}

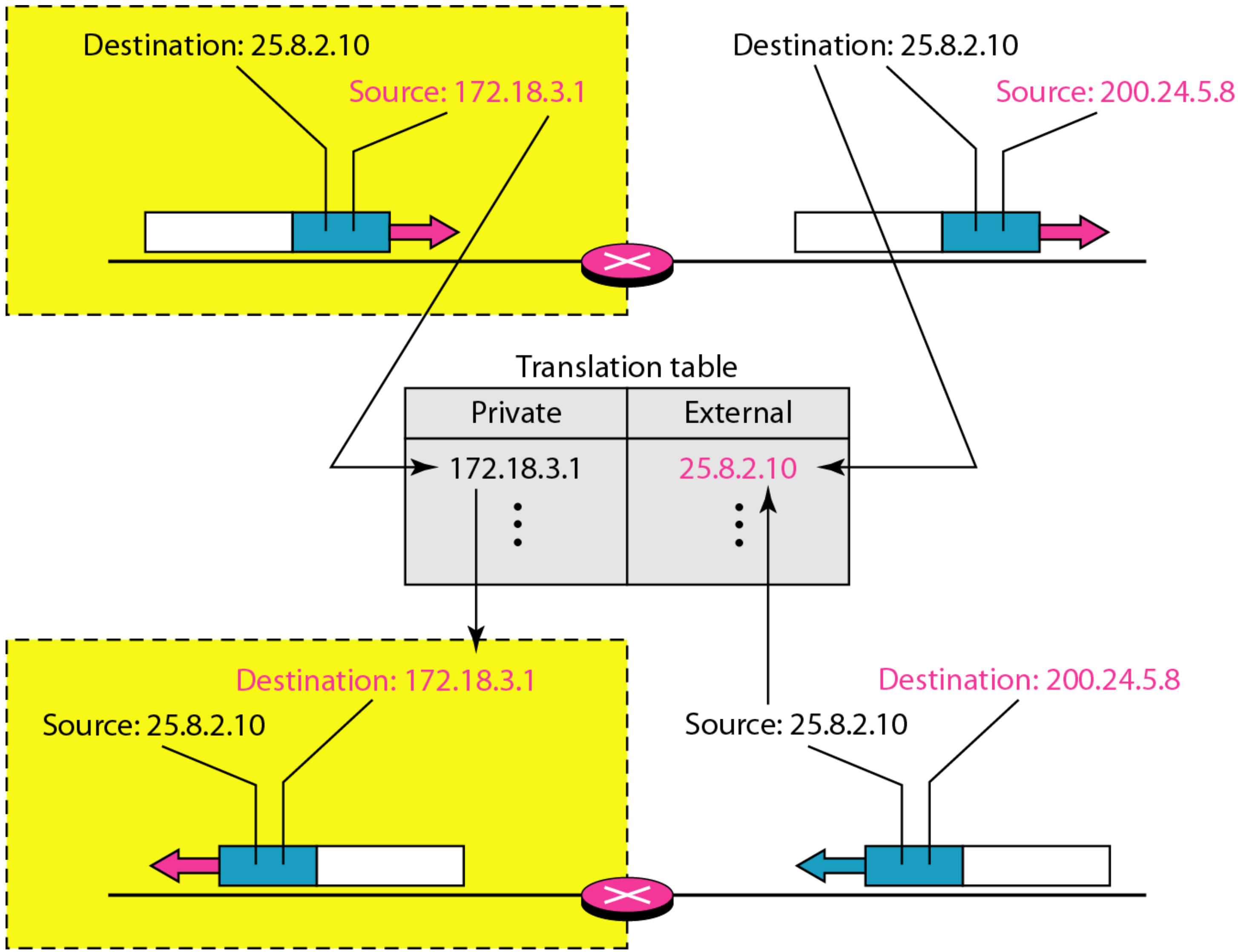
Address Translation

- 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 (the NAT router global address) with the appropriate private address.
- **Translation Table :**
- 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.
-

A NAT implementation






Destination: 25.8.2.10
Source: 172.18.3.1
Destination: 25.8.2.10
Source: 200.24.5.8
Translation table
Private
External
172.18.3.1
25.8.2.10
Destination: 172.18.3.1
Source: 25.8.2.10
Destination: 200.24.5.8
Source: 25.8.2.10

Translation Table

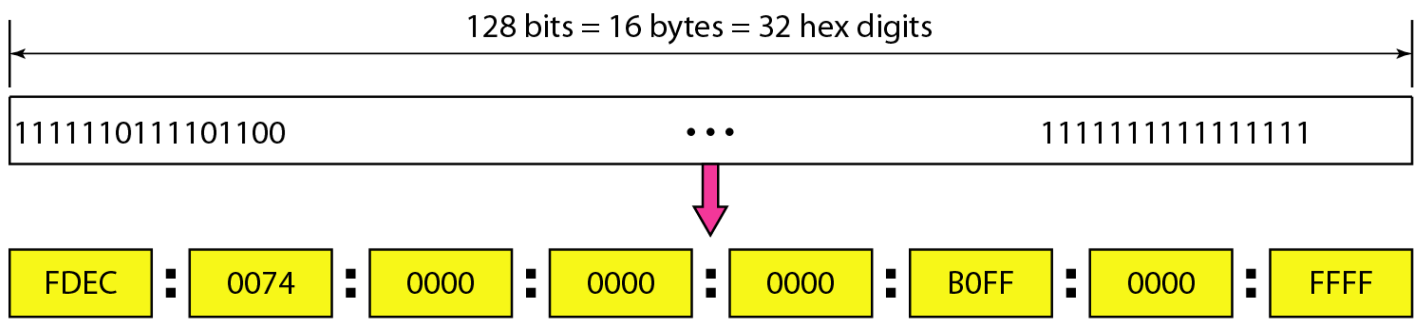
- **Using One IP Address** a translation table has only two columns: **the private' address** and **the external address** (destination address of the packet).
- When the router translates the source address of the outgoing packet, it also makes note of the destination address-where the packet is going.
- When the response comes back from the destination, the router uses the source address of the packet (as the external address) to find the private address of the packet.
- **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.
- No more than required connections can be made to the same destination.

- **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.
- 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
.	.	.	.	.
.	.	.	.	.

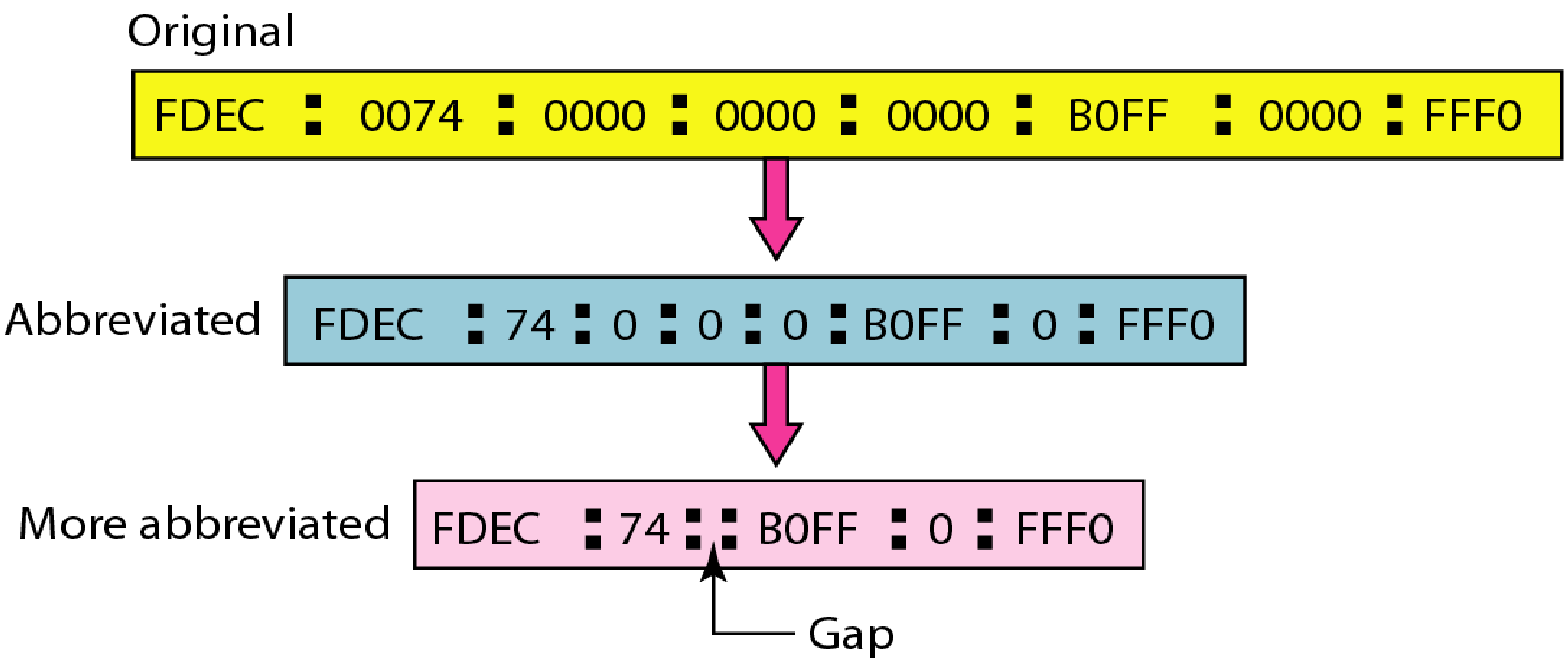
IPV6 ADDRESS

- An IPv6 address is 128 bits long (16-byte).
- 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



Abbreviation

- Although the IP address, even in hexadecimal format, is very long, many of the digits are 0's.
- 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.



- 0074 can be written as 74, OOOF as F, and 0000 as O.
- Note that 3210 cannot be abbreviated.
- We can remove the zeros altogether and replace them with a double semicolon.
- Note that this type of abbreviation is allowed only once per address.
- If there are two runs of zero sections, only one of them can be abbreviated.
- Reexpansion of the abbreviated address is very simple: Align the unabbreviated portions and insert zeros to get the original expanded address.

Q. 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.

```
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
```

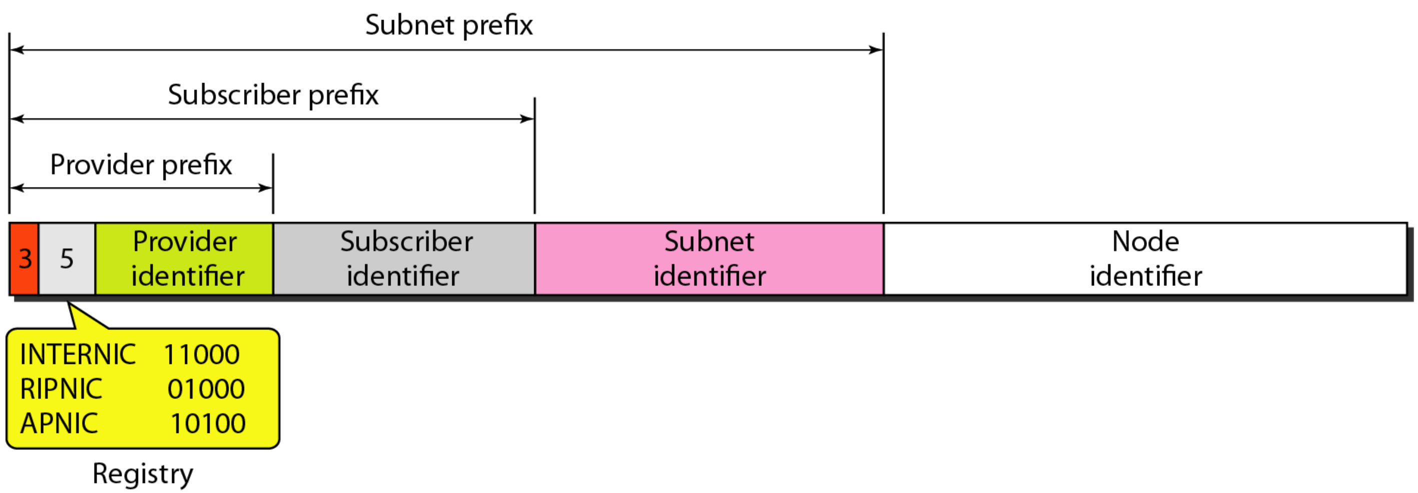
Address Space

- IPv6 has a much larger address space; $\mathbf{2^{128}}$ addresses are available.
- The designers of IPv6 divided the address into several categories.
- A few leftmost bits, called the type prefix, in each address define its category.
- The type prefix is variable in length, but it is designed such that no code is identical to the first part of any other code.
- In this way, there is no ambiguity; when an address is given, the type prefix can easily be determined.
- The third column shows the fraction of each type of address relative to the whole address space.

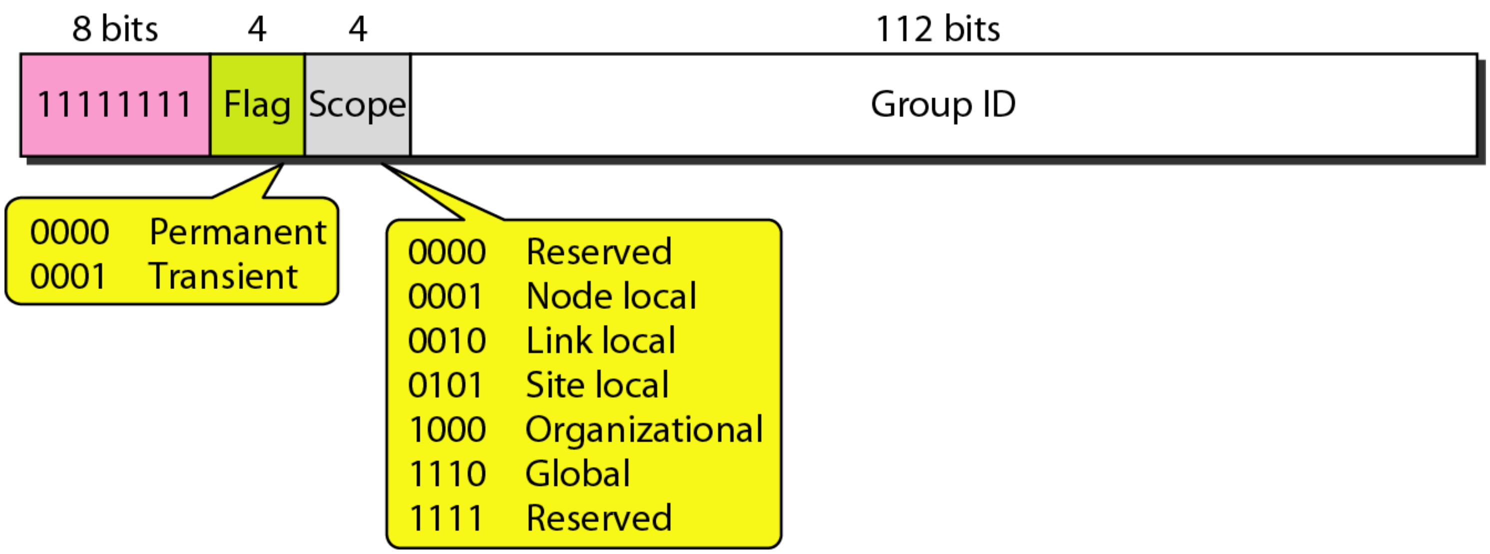
Type Prefix	*Type*	*Fraction*
0000 0000	Reserved	1/256
0000 0001	Unassigned	1/256
0000 001	ISO network addresses	1/128
0000 010	IPX (Novell) network addresses	1/128
0000 011	Unassigned	1/128
0000 1	Unassigned	1/32
0001	Reserved	1/16
001	Reserved	1/8
010	**Provider-based unicast addresses**	**1/8**

Type Prefix	*Type*	*Fraction*
011	Unassigned	1/8
100	Geographic-based unicast addresses	1/8
101	Unassigned	1/8
110	Unassigned	1/8
1110	Unassigned	1/16
1111 0	Unassigned	1/32
1111 10	Unassigned	1/64
1111 110	Unassigned	1/128
1111 1110 0	Unassigned	1/512
1111 1110 10	Link local addresses	1/1024
1111 1110 11	Site local addresses	1/1024
1111 1111	Multicast addresses	1/256

UNICAST ADDRESS

- A **unicast address** defines a single computer. The packet sent to a unicast address must be delivered to that specific computer.
- IPv6 defines two types of unicast addresses: geographically based and provider-based.
- Fields : **Type ID (3-bit), Registry ID (5-bit), Provider ID (16-bit), Subscriber ID (24-bit), Subnet ID (32-bit), Node ID (48-bit)**
- 

Multicast addresses

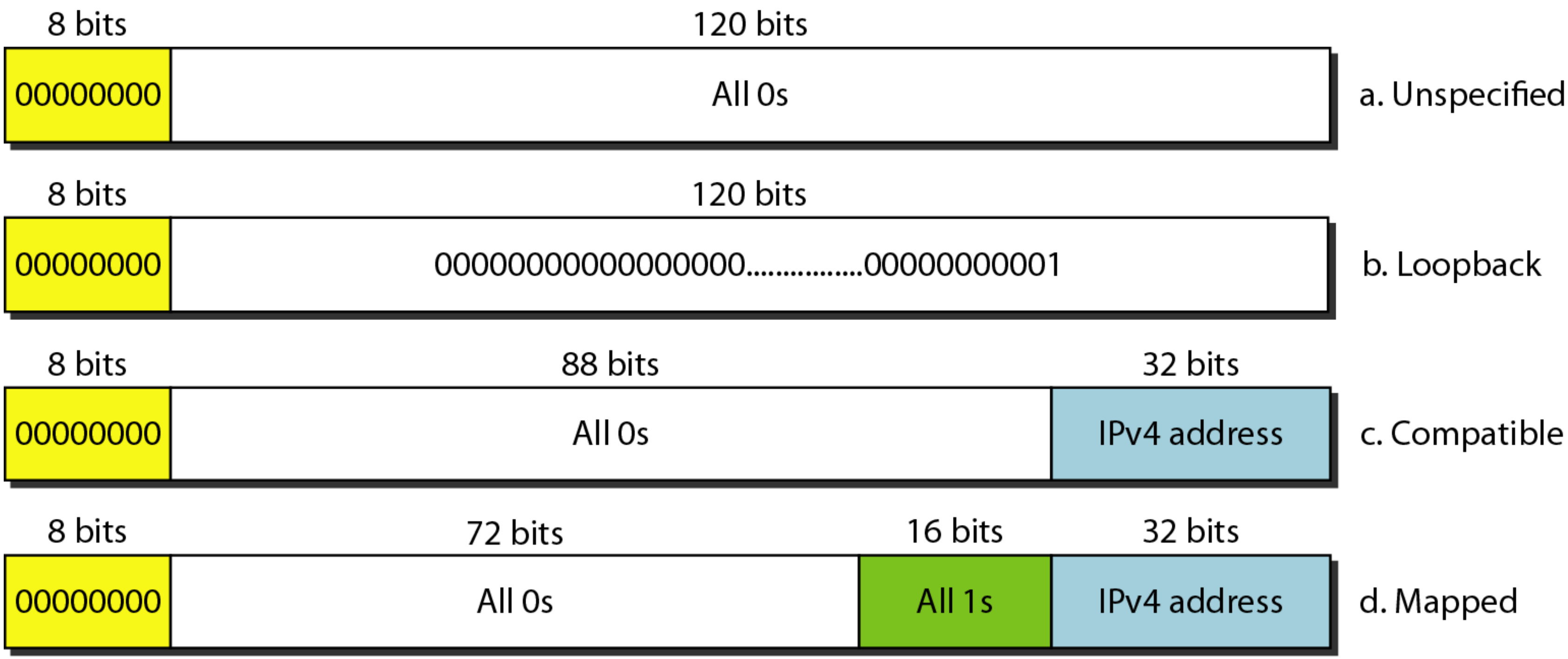
- Multicast addresses are used to define a group of hosts instead of just one.
- A packet sent to a multicast address must be delivered to each member of the group.
- 

ANYCAST ADDRESS

- An anycast address, like a multicast address, also defines a group of nodes.
- However, a packet destined for an anycast address is delivered to only one of the members of the anycast group, the nearest one (the one with the shortest route).

Reserved Addresses

- These addresses start with eight Os (type prefix is 00000000).



Local Addresses

- These addresses are used when an organization wants to use IPv6 protocol without being connected to the global Internet.
- In other words, they provide addressing for private networks.
- Nobody outside the organization can send a message to the nodes using these addresses.
- Two types of addresses are defined for this purpose