Internet Protocol (IP)

Network Layer

Network layer is concerned with getting packets from the source **all the way** to the destination (**end-end to transmission)**

- May require many hops at intermediate routers (**multiple hops**),

- Its primary function is **routing**

- Network layer should provide either connection oriented or connectionless service

Network Layer Protocol
Internet Protocol (IP)

IP ADDRESSES

An IP address uniquely and universally defines the connection of a device (a computer or a router) to the Internet

Types of IP addresses in active use

- IP version 4 (IPv4)
- IP version 6 (IPv6)

Types of IP ADDRESSES

IP version 4 (IPv4)

- Initially deployed on Jan 1, 1983
- Still most commonly used version
- A 32 bit address in length
- Expressed as four 8 bit Octets separated by **a period symbol** (Dotted decimal notation)
  - 192.2.3.50
- Its **address space** is **2^{32} or 4,294,967,296**

Types of IP ADDRESSES

IP version 6 (IPv6)

- Its deployment started in 1999

- 128 bit numbers

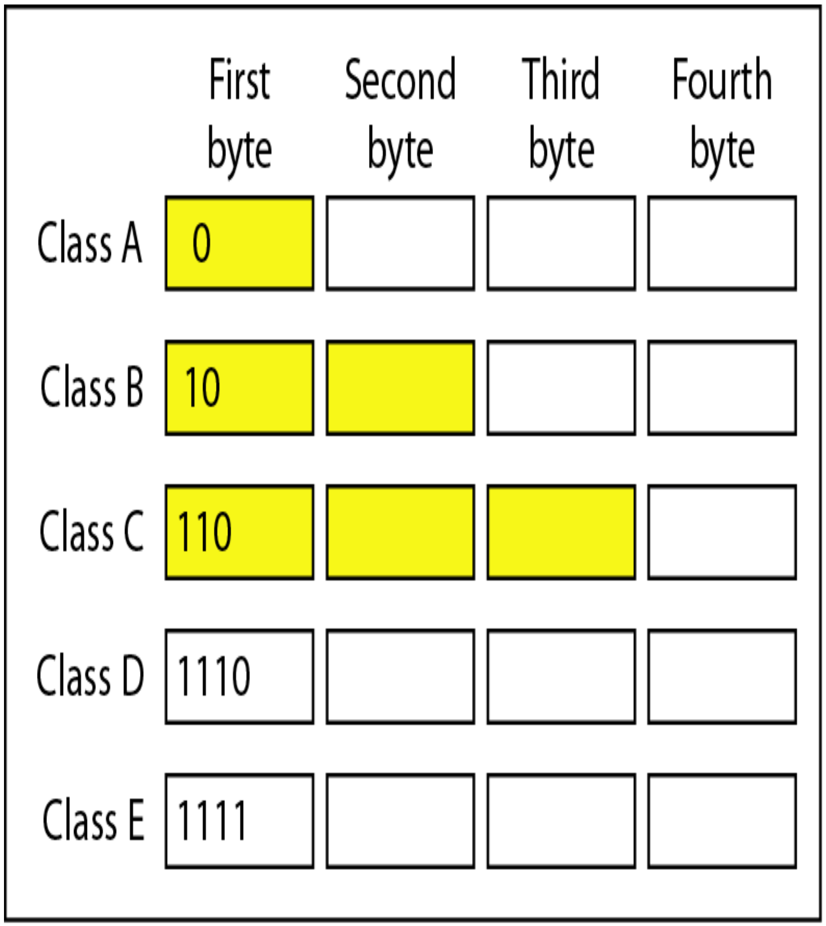
- conventionally expressed using hexadecimal strings
- *2001:0db8:582:ae33::29*

- *Apparently Address space is* $\mathbf{2^{128}}$

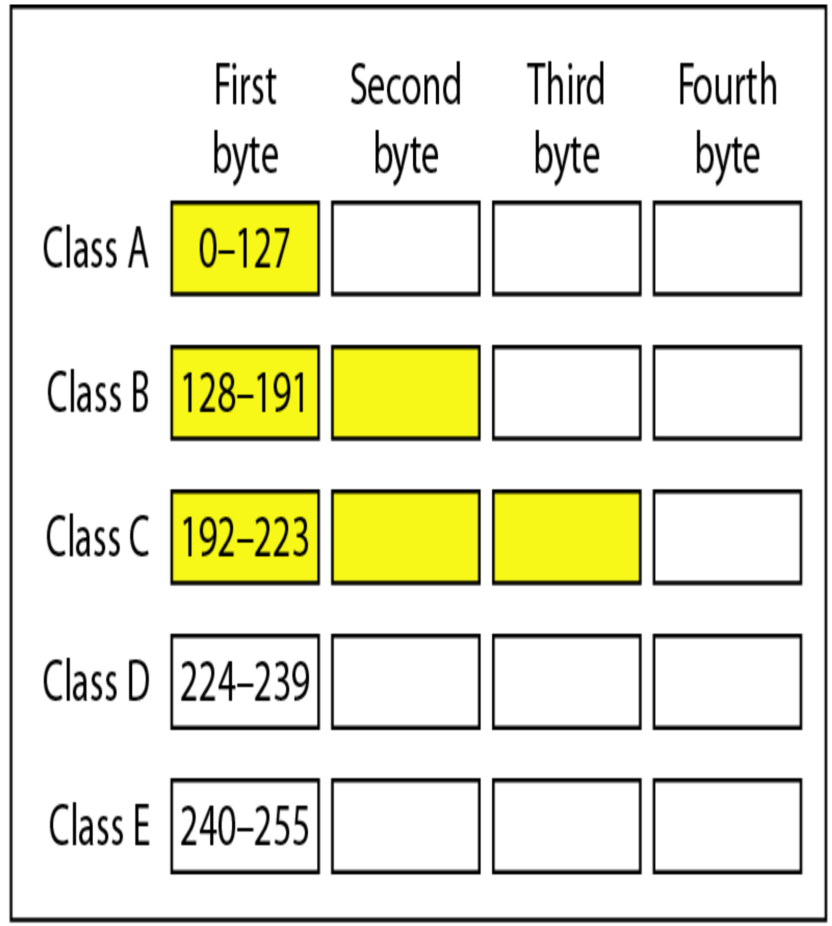
IPv4 Addresses (classful)

IPv4 classes in binary and dotted-decimal notation

The address space is divided into five classes:
A, B, C, D, and E

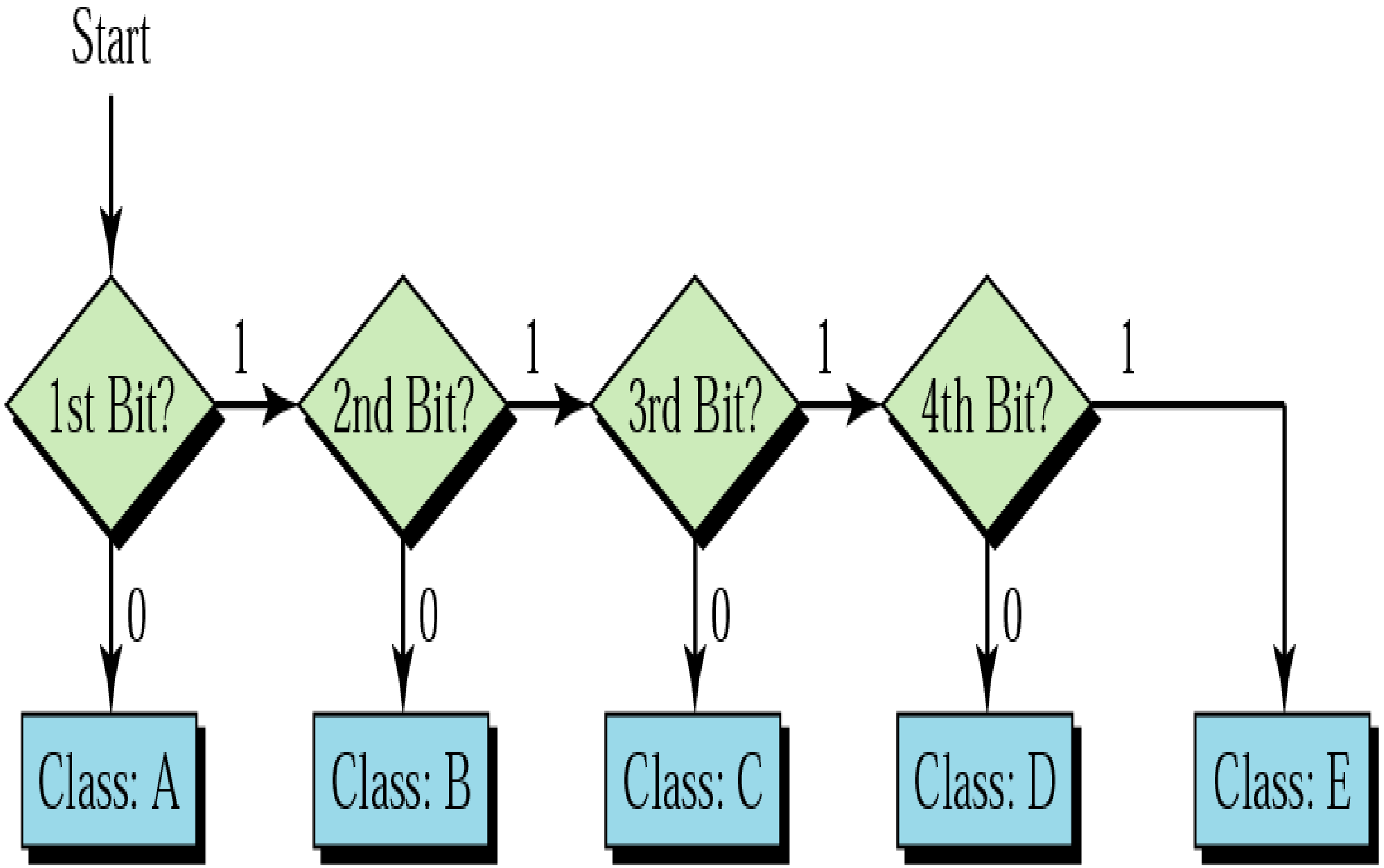


a. Binary notation



b. Dotted-decimal notation

Finding the Class



IPv4 ADDRESSES

IP address is divided into two parts

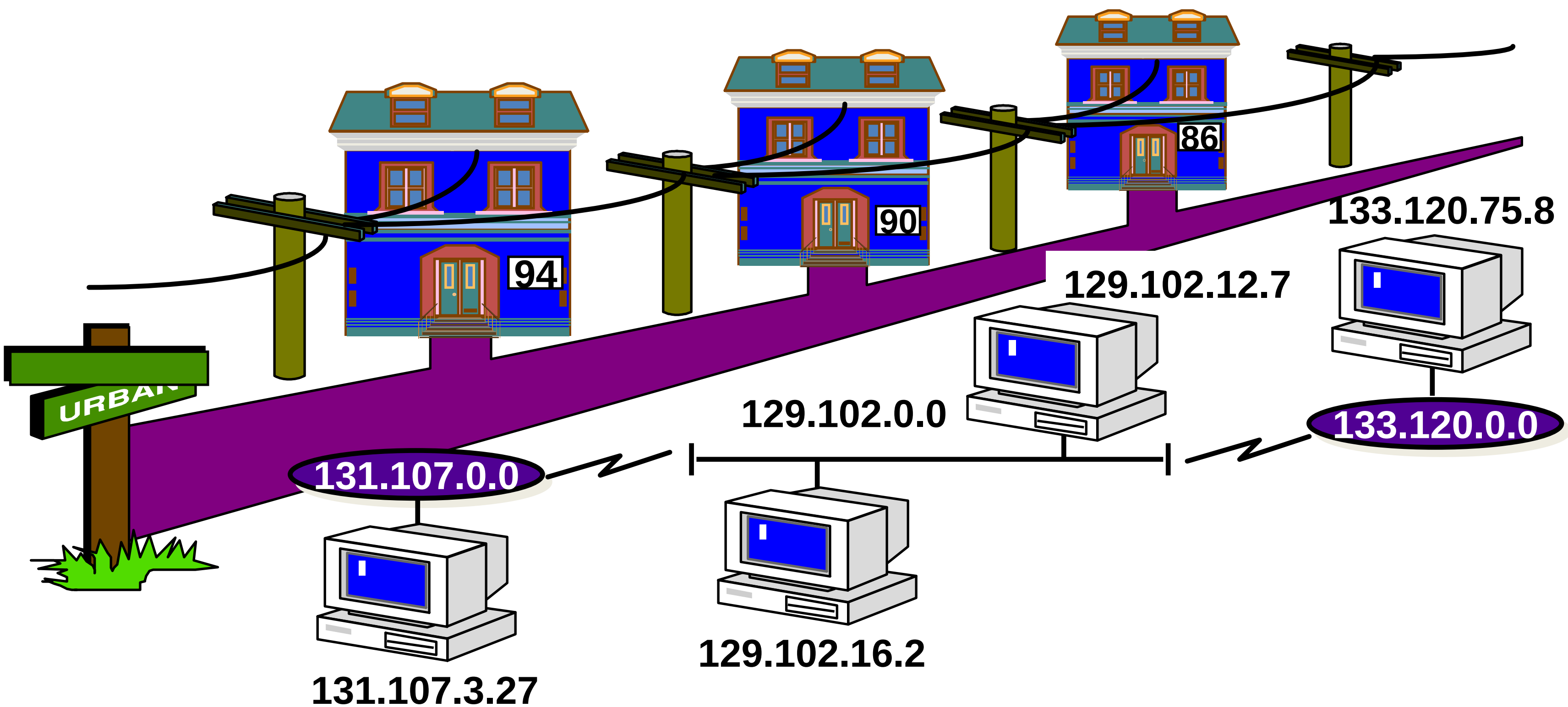
- **Prefix** defines the network (Network ID or **Net ID):**
  - like street address

- **Suffix (Host ID)** defines the node (connection of a device to the network):
  - Like house or building number

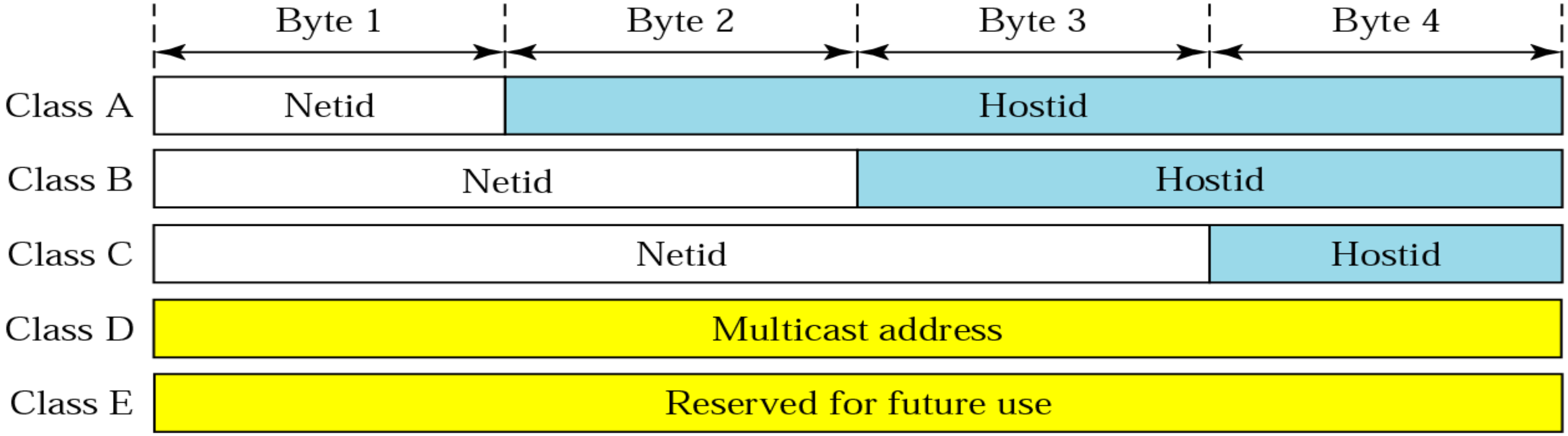
- A **host** is a device that has a network interface card (NIC) connected to a network

IPv4 ADDRESSES

- If a device has **two** network interfaces, it should be considered **two separate hosts**
- Each host that is attached to a TCP/IP network must have a **unique IP address**



Netid and Hostid



- **Hostid: cannot be all 0s**
  - If host portion is all 0s, represents a **network** address.
- **Hostid: cannot be all 1s**
  - If host portion is all 1s, represents **broadcast** address.

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

Class A Address

- First bit: always 0. Remaining bits can be either 0s or 1s.

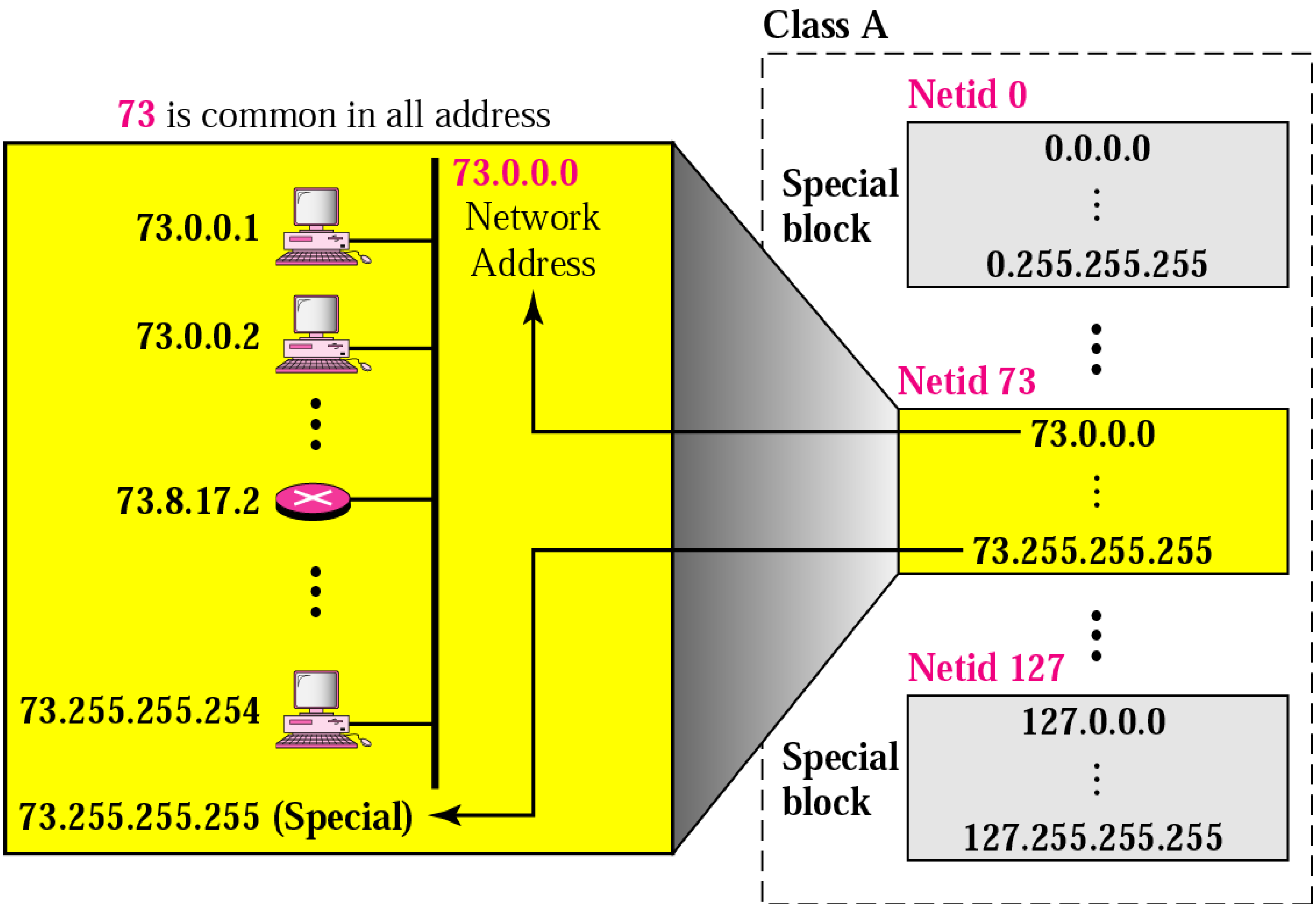
- Range of first octet is 00000000 to 01111111

  - Network addresses cannot be all 0s (**0.0.0.0** through **0.255.255.255**): nodes attempting to use them will be unable to communicate properly on the Internet

  - 127 is reserved for **loopback testing**

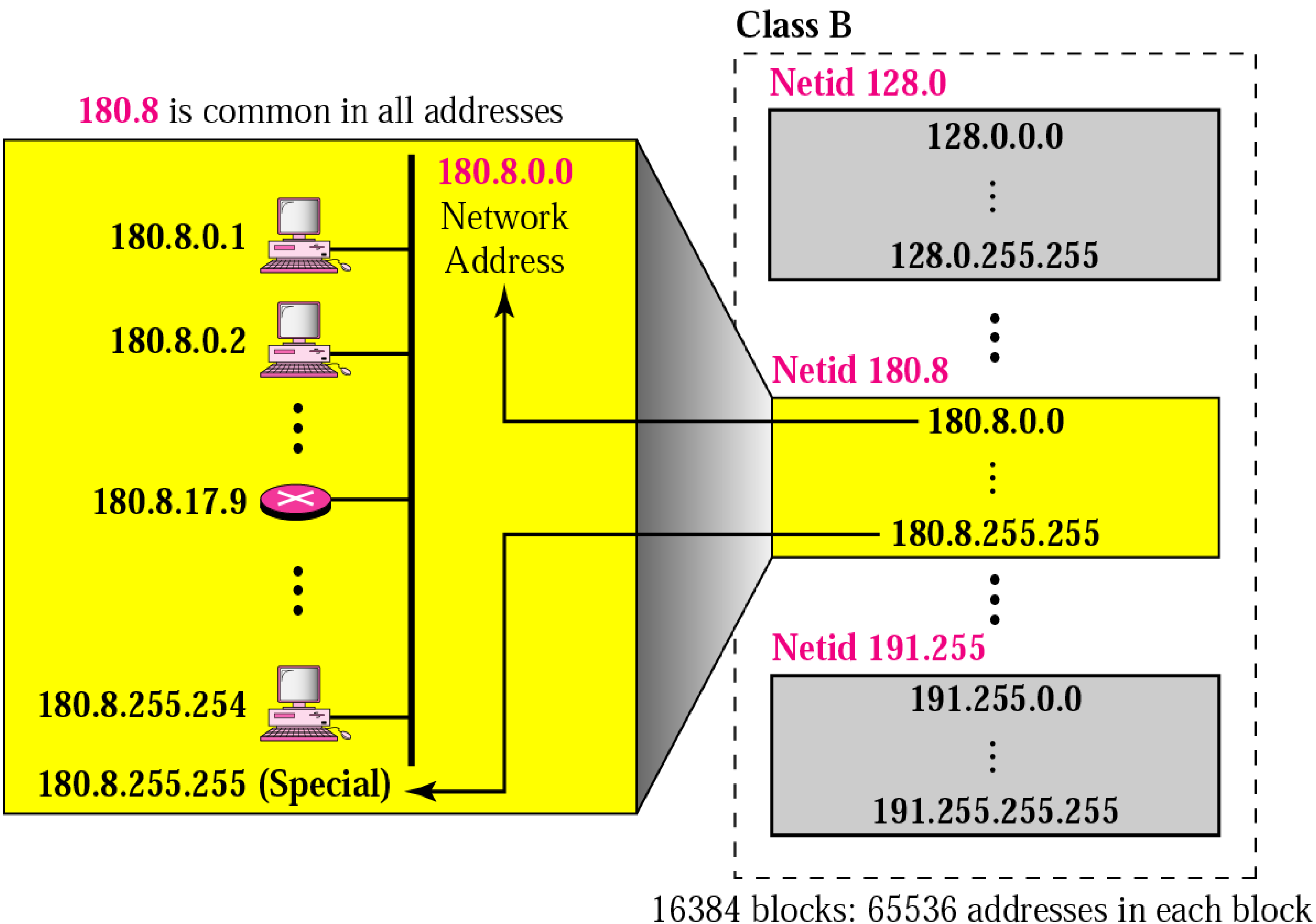
- **126 valid Class A network IDs**: 1.x.y.z to 126.x.y.z

Class A Address

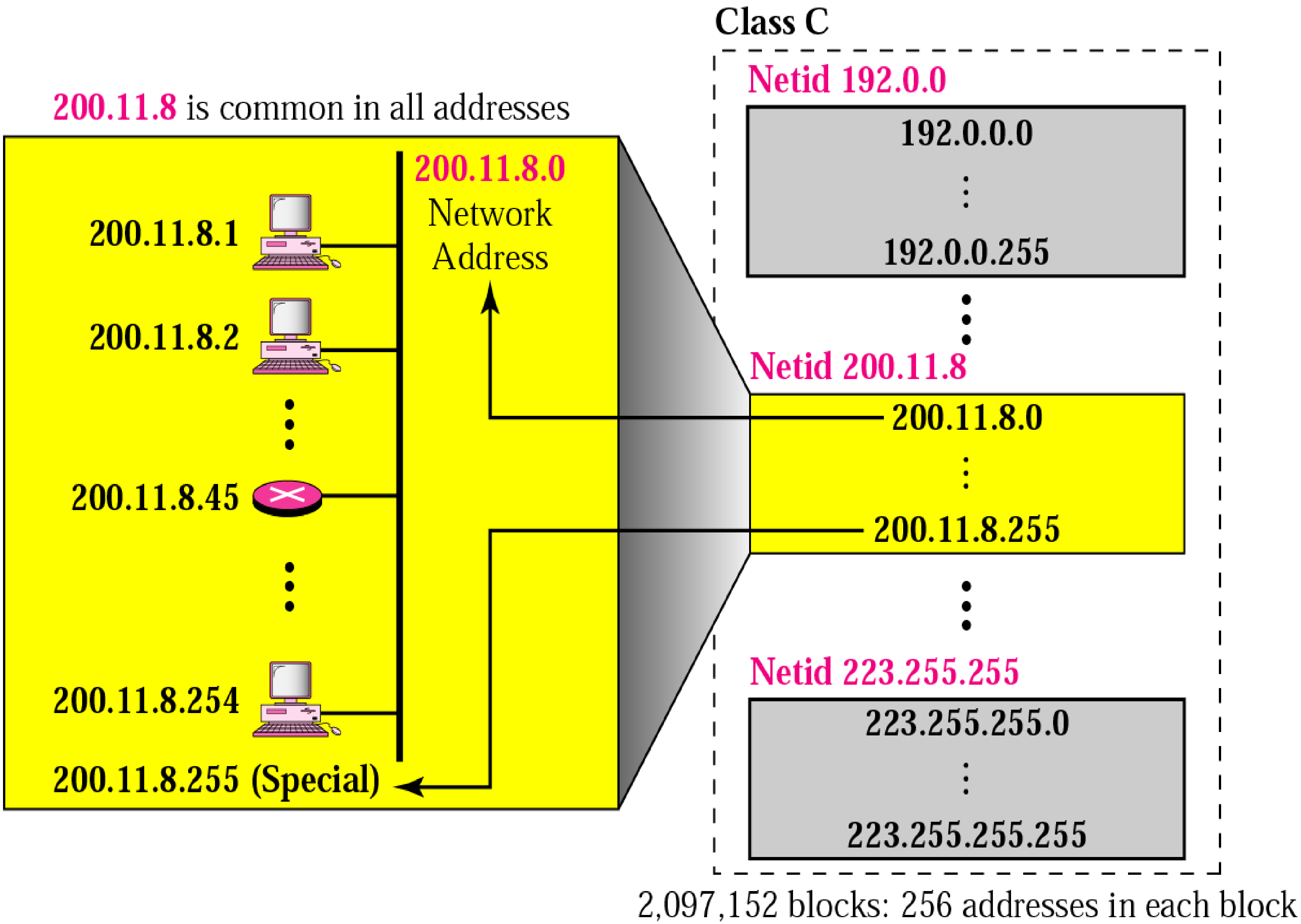


128 blocks: 16,777,216 addresses in each block

Blocks in class B



Blocks in Class C



- **Class D addresses** (from 224.x.y.z to 239.x.y.z) are used for multicasting
  - method of sending a single packet to multiple hosts

- **Class E addresses** (from 240.x.y.z to 255.x.y.z)
  - experimental address range
  - Not used in actual networks

Network Address

The **first address** is called the **network address**

- different from a **netid** and
- contains **both netid and hostid**
  - with **0s** for the **hostid**

➢It defines the **organization network** (to connect to the rest of world)

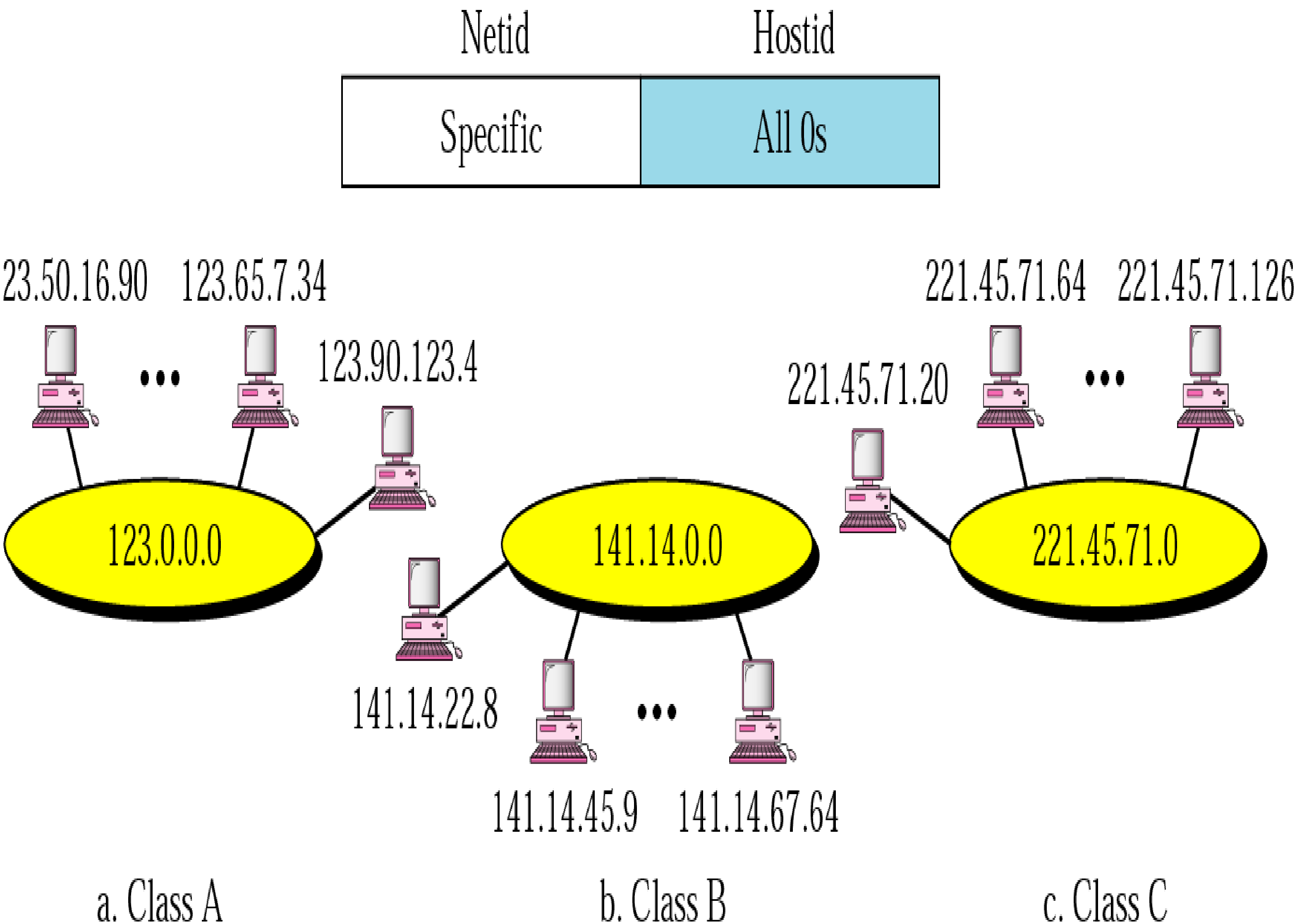
  ➢The **organization network** is **connected** to the **Internet** via a **router**.

➢The router has **two addresses**.

  ➢One belongs to the **granted block**;

  ➢the other belongs to the network that is **at the other side of the router**.

Network Address



Examples: Network Address

- **Question:** Given the address **23.56.7.91**, find the **network address**.
- **Answer:** Replacing the host-id bytes (56.7.91) with 0s.

So, the network address is **23.0.0.0**.

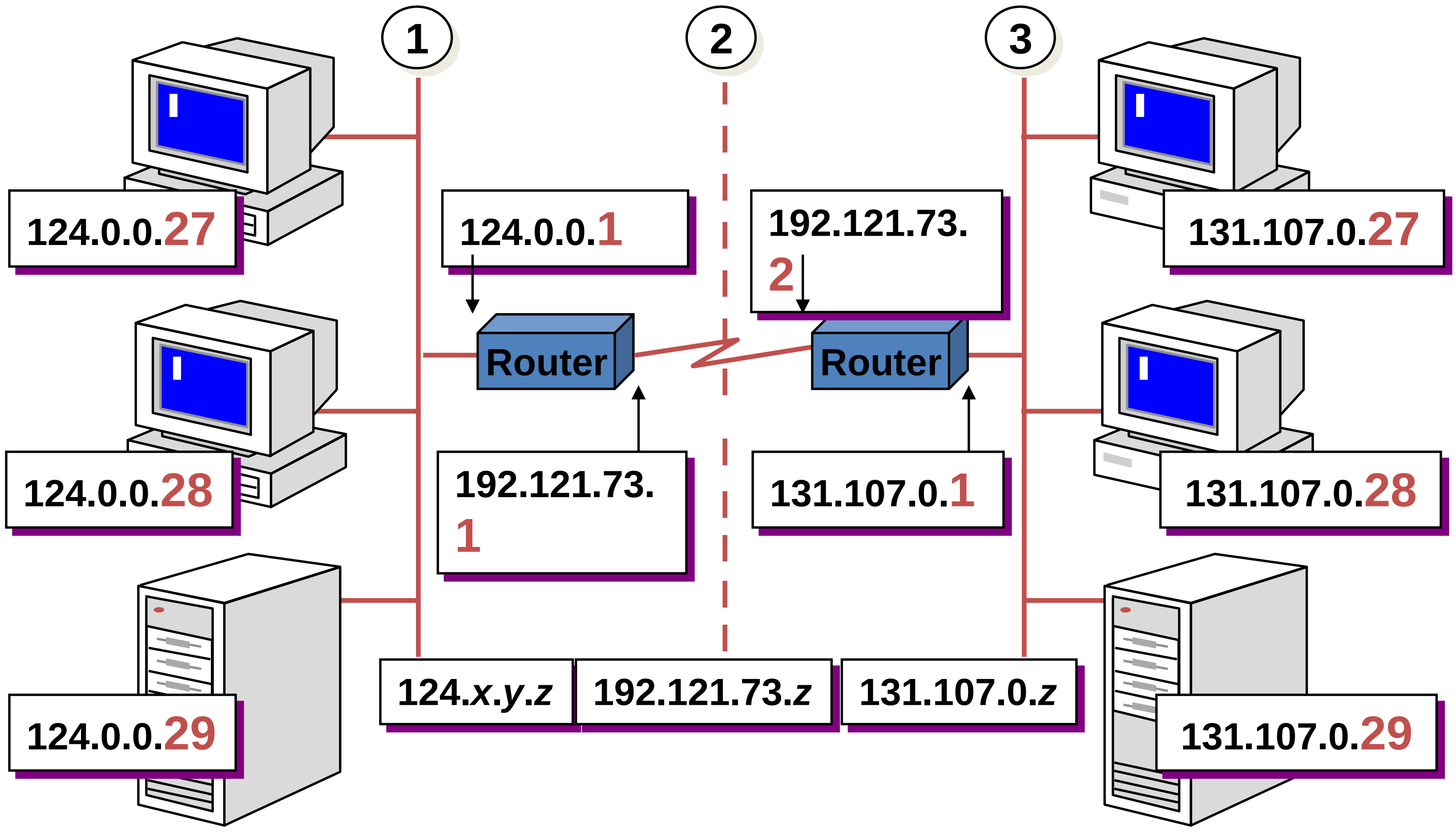
- **Question:** Given the address **132.6.17.85**, find the **network address**
- **Answer:** We can find the network address by replacing the hostid bytes (17.85) with 0s.

Thus, the network address is **132.6.0.0.**

Address Class Summary

	Number of Networks	Number of Hosts per Network	Range of Network IDs (First Octet)
Class A	126	16,777,214	1 – 126
Class B	16,384	65,534	128 – 191
Class C	2,097,152	254	192 – 223

Assigning Host IDs



Remember the following

Network address: If all the bits in the host part are "0", that represents the network id

First usable IPv4 address: If all the bits in the host part are "0" except the last bit

Last usable IPv4 address : If all the bits in the host part are "1" except the last bit

Broadcast address: If all the bits in the host part are "1"

Reserved, Private addresses

Private address block:
Class A: 10.0.0.0 to 10.255.255.255 [16777216] (Private use networks)
Class B: 172.16.0.0 to 172.31.255.255 [1048576]
Class C: 192.168.0.0 to 192.168.255.255 [65536]

Reserved:
Class A: 0.0.0.0 to 0.255.255.255 & 127.0.0.0 to 127.255.255.255 (LB)
Administered by RIRs: Examples
128.0.0.0 to 128.0.255.255 & 191.255.0.0 to 191.255.255.255
192.0.0.0 to 192.0.0.255 ---------196.-.-.-, 198.-.-.-

What is Subnet Mask?

IPv4 address has two components, the network part and the host part.

The purpose of subnet mask is to identify which part is the **network part** and which part is the **host part**.

Subnet mask: A 32 bit number where

- all the bits of the network part are shown as **1** and
- all the bits of the host part are represented as **0**.

For example, for a **Class C Network**, 192.168.10.0, the subnet mask is **255.255.255.0**

Subnet and Sub-netting

- A logical, visible subdivision of an IP network is called **subnet** or **subnetwork:**
  - It is created by dividing the host identifier

- **Sub-netting** is the practice of dividing a network into two or more networks

- **Sub-netting** is done by taking the bits from host part and adding it to the network part

Subnet and Sub-netting

In sub-netting, a class A or class B or class C block is divided into several subnets

- each subnet with larger prefix length than the original network).

➢**For example,** divide the **class A** into **four subnets**, (take two bits from host id part in order to obtain subnets).

   ➢then each subnet will have prefix length as 10

Class C – 1 bit sub-netting

Consider class C network 192.168.10.0 (subnet mask is 255.255.255.0)

- If we include one bit from the host part to the network part, the subnet mask changes into 255.255.255.128 (for 1 bit sub-netting)
- The single bit can have two values in last octet, either 0 or 1
- 11000000.10101000.00001010.**0** | 0000000
  11111111.11111111.11111111.**1** | 0000000
- So the network 192.168.10.0 is divided into two networks with a single bit sub-netting,
- each network has 128 total addresses of which 126 are usable
- two are used in each subnet to represent the network address and broadcast address.

Class C – 1 bit sub-netting

SN No.	Description	Binaries	Decimal
1	Network Address	11000000.10101000.00001010.00000000	192.168.10.0
	First usable address	11000000.10101000.00001010.00000001	192.168.10.1
	Last usable address	11000000.10101000.00001010.01111110	192.168.10.126
	Broadcast Address	11000000.10101000.00001010.01111111	192.168.10.127
2	Network Address	11000000.10101000.00001010.10000000	192.168.10.128
	First usable address	11000000.10101000.00001010.10000001	192.168.10.129
	Last usable address	11000000.10101000.00001010.11111110	192.168.10.254
	Broadcast Address	11000000.10101000.00001010.11111111	192.168.10.255

Class C – 2 bit sub-netting

SN No.	Description	Binaries	Decimal
1	NA	11000000.10101000.00001010.00000000	192.168.10.0
	1st	11000000.10101000.00001010.00000001	192.168.10.1
	Last	11000000.10101000.00001010.00111110	192.168.10.62
	BA	11000000.10101000.00001010.00111111	192.168.10.63
2	NA	11000000.10101000.00001010.01000000	192.168.10.64
	1st	11000000.10101000.00001010.01000001	192.168.10.65
	Last	11000000.10101000.00001010.01111110	192.168.10.126
	BA	11000000.10101000.00001010.01111111	192.168.10.127
3	NA	11000000.10101000.00001010.10000000	192.168.10.128
	1st	11000000.10101000.00001010.10000001	192.168.10.129
	Last	11000000.10101000.00001010.10111110	192.168.10.190
	BA	11000000.10101000.00001010.10111111	192.168.10.191
4	NA	11000000.10101000.00001010.11000000	192.168.10.192
	1st	11000000.10101000.00001010.11000001	192.168.10.193
	Last	11000000.10101000.00001010.11111110	192.168.10.254
	BA	11000000.10101000.00001010.11111111	192.168.10.255

Class C – 3 bit sub-netting

SN No.	Description	Binaries	Decimal
1	NA	11000000.10101000.00001010.**000**00000	192.168.10.0
	1st	11000000.10101000.00001010.00000001	192.168.10.1
	Last	11000000.10101000.00001010.00011110	192.168.10.30
	BA	11000000.10101000.00001010.**000**11111	192.168.10.31
2	NA	11000000.10101000.00001010.**001**00000	192.168.10.32
	1st	11000000.10101000.00001010.00100001	192.168.10.33
	Last	11000000.10101000.00001010.00111110	192.168.10.62
	BA	11000000.10101000.00001010.**001**11111	192.168.10.63
		....	
8	NA	11000000.10101000.00001010.**111**00000	192.168.10.224
	1st	11000000.10101000.00001010.11100001	192.168.10.225
	Last	11000000.10101000.00001010.11111110	192.168.10.254
	BA	11000000.10101000.00001010.**111**11111	192.168.10.255

Class B – 1 bit sub-netting

Consider class B network 172.16.0.0 (subnet mask is 255.255.0.0)

- ❖ If we include one bit from the host part to the network part, the **subnet mask** changes into 255.255.128.0 (with 1 bit sub-netting)
- ❖ The single bit can have two values in last octet, either 0 or 1
- ❖ 10101100.00010000.**0** | 0000000.00000000
  11111111.11111111.**1** | 0000000.00000000
- ❖ So the network 172.16.0.0 is divided into two networks with single bit sub-netting,
  - ❖ each network has 32768 total addresses of which 32766 are usable, two are used in each subnet to represent the network address and broadcast address.

Class B – 1 bit sub-netting

SN No.	Description	Binaries	Decimal
1	Network Address	10101100.00010000.00000000.00000000	172.16.0.0
	First address	10101100.00010000.00000000.00000001	172.16.0.1
	Last address	10101100.00010000.01111111.11111110	172.16.127.254
	Broadcast Address	10101100.00010000.01111111.11111111	172.16.127.255
2	Network Address	10101100.00010000.10000000.00000000	172.16.128.0
	First address	10101100.00010000.10000000.00000001	172.16.128.1
	Last address	10101100.00010000.11111111.11111110	172.16.255.254
	Broadcast Address	10101100.00010000.11111111.11111111	172.16.255.255

Class B – 2 bit sub-netting

SN No.	Description	Binaries	Decimal
1	NA	10101100.00010000.00000000.00000000	172.16.0.0
	1st	10101100.00010000.00000000.00000001	172.16.0.1
	Last	10101100.00010000.00111111.11111110	172.16.63.254
	BA	10101100.00010000.00111111.11111111	172.16.63.255
2	NA	10101100.00010000.01000000.00000000	172.16.64.0
	1st	10101100.00010000.01000000.00000001	172.16.64.1
	Last	10101100.00010000.01111111.11111110	172.16.127.254
	BA	10101100.00010000.01111111.11111111	172.16.127.255
3	NA	10101100.00010000.10000000.00000000	172.16.128.0
	1st	10101100.00010000.10000000.00000001	172.16.128.1
	Last	10101100.00010000.10111111.11111110	172.16.191.254
	BA	10101100.00010000.10111111.11111111	172.16.191.255
4	NA	10101100.00010000.11000000.00000000	172.16.192.0
	1st	10101100.00010000.11000000.00000001	172.16.192.1
	Last	10101100.00010000.11111111.11111110	172.16.192.254
	BA	10101100.00010000.11111111.11111111	172.16.192.255

Example: Route 193.205.102.36

193								205								102								36							
1	1	0	0	0	0	0	1	1	1	0	0	1	1	0	1	0	1	1	0	0	1	1	0	0	0	1	0	0	1	0	0

Class C address;
Outside private domain routed with mask 255.255.255.0

network																								host							
193								205								102								36							
1	1	0	0	0	0	0	1	1	1	0	0	1	1	0	1	0	1	1	0	0	1	1	0	0	0	1	0	0	1	0	0

Inside private domain, administrator has set netmask 255.255.255.248

255								255								255								248							
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0

Hence, route to subnet address and then to host id, computed as:

network																								subnet					host		
1	1	0	0	0	0	0	1	1	1	0	0	1	1	0	1	0	1	1	0	0	1	1	0	0	0	1	0	0	1	0	0
193.205.102.32 /29																													4		

Classless Addressing

- **Classless addressing**
  - uses a variable number of bits for the network and host portions of the address (**variable-length blocks)**
  - treats the IP address as a 32 bit stream of ones and zeroes,
  - where the boundary between network and host portions can fall anywhere between bit 0 and bit 31.

Classless Addressing

- **Classless addressing**
- How to find the prefix length if an address is given?
- As prefix length is not inherent in the address
  - Need to separately give the length of the prefix
- So, Prefix length is added to the address, separated by a slash

Format of classless address

The notation is informally referred to as slash notation and formally as ***classless inter-domain routing or CIDR***

x.y.z.t/n

Table Prefix lengths

/n	Mask	/n	Mask	/n	Mask	/n	Mask
/1	128.0.0.0	/9	255.128.0.0	/17	255.255.128.0	/25	255.255.255.128
/2	192.0.0.0	/10	255.192.0.0	/18	255.255.192.0	/26	255.255.255.192
/3	224.0.0.0	/11	255.224.0.0	/19	255.255.224.0	/27	255.255.255.224
/4	240.0.0.0	/12	255.240.0.0	/20	255.255.240.0	/28	255.255.255.240
/5	248.0.0.0	/13	255.248.0.0	/21	255.255.248.0	/29	255.255.255.248
/6	252.0.0.0	/14	255.252.0.0	/22	255.255.252.0	/30	255.255.255.252
/7	254.0.0.0	/15	255.254.0.0	/23	255.255.254.0	/31	255.255.255.254
/8	255.0.0.0	/16	255.255.0.0	/24	255.255.255.0	/32	255.255.255.255

Classful addressing is a special case of classless 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

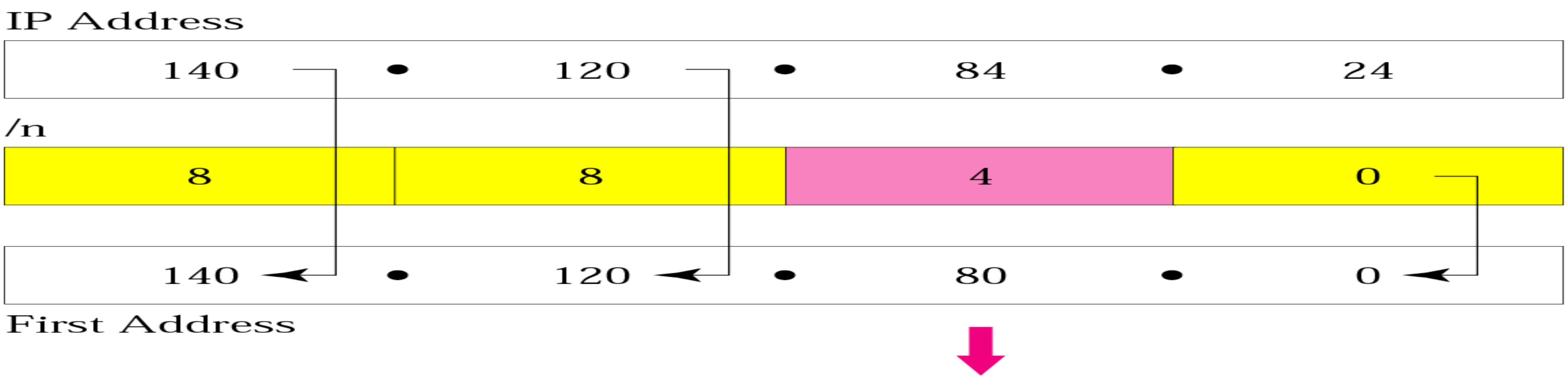
Example 1: What is the first address in the block if one of the addresses is ***167.199.170.82/27*****?**

Solution**:** The prefix length is 27, which means that we must keep the first 27 bits as is and change the remaining bits (5) to 0s. The following shows the process:

Address in binary:	10100111 11000111 10101010 01010010
Keep the left 27 bits:	10100111 11000111 10101010 01000000
Result in CIDR notation:	167.199.170.64/27

Example 2: What is the first address in the block if one of the addresses is 140.120.84.24/20? Number of addresses, first and last useable IPs in block? Broadcast address?

Answer: The first address is 140.120.80.0/20



84	0	1	0	1	0	1	0	0
Keep left 4 bits	0	1	0	1	0	0	0	0

Result in decimal: 80

	128	64	32	16	8	4	2	1
Write 84 as sum of:	0	64	0	16	0	4	0	0
Select only leftmost 4:	0	64	0	16				
Add to find the result:	80							

Example 4: find the first and last address in the block if one of the addresses is 140.120.84.24/20.

Answer: The **first** address is **140.120.80.0/20** (set all bits of host part to 0)

The **last** address is **140.120.95.255/20** (set all bits of host part to 1)

Another way to find the last address:

Add the **mask complement** to the **beginning address** to find the **last address**.

```
140 . 120 . 80 .   0
  0 .   0 . 15 . 255
--------------------------
140 . 120 . 95 . 255
```

Example 5: Find the block if one of the addresses is 190.87.140.202/29.

***Solution*:** To find the **first** address, we notice that the mask (/29) has five 1s in the last byte.

So write the last byte as powers of 2 and retain only the leftmost five:

202 → **128 + 64 + 0 + 0 + 8 + 0 + 2 + 0**

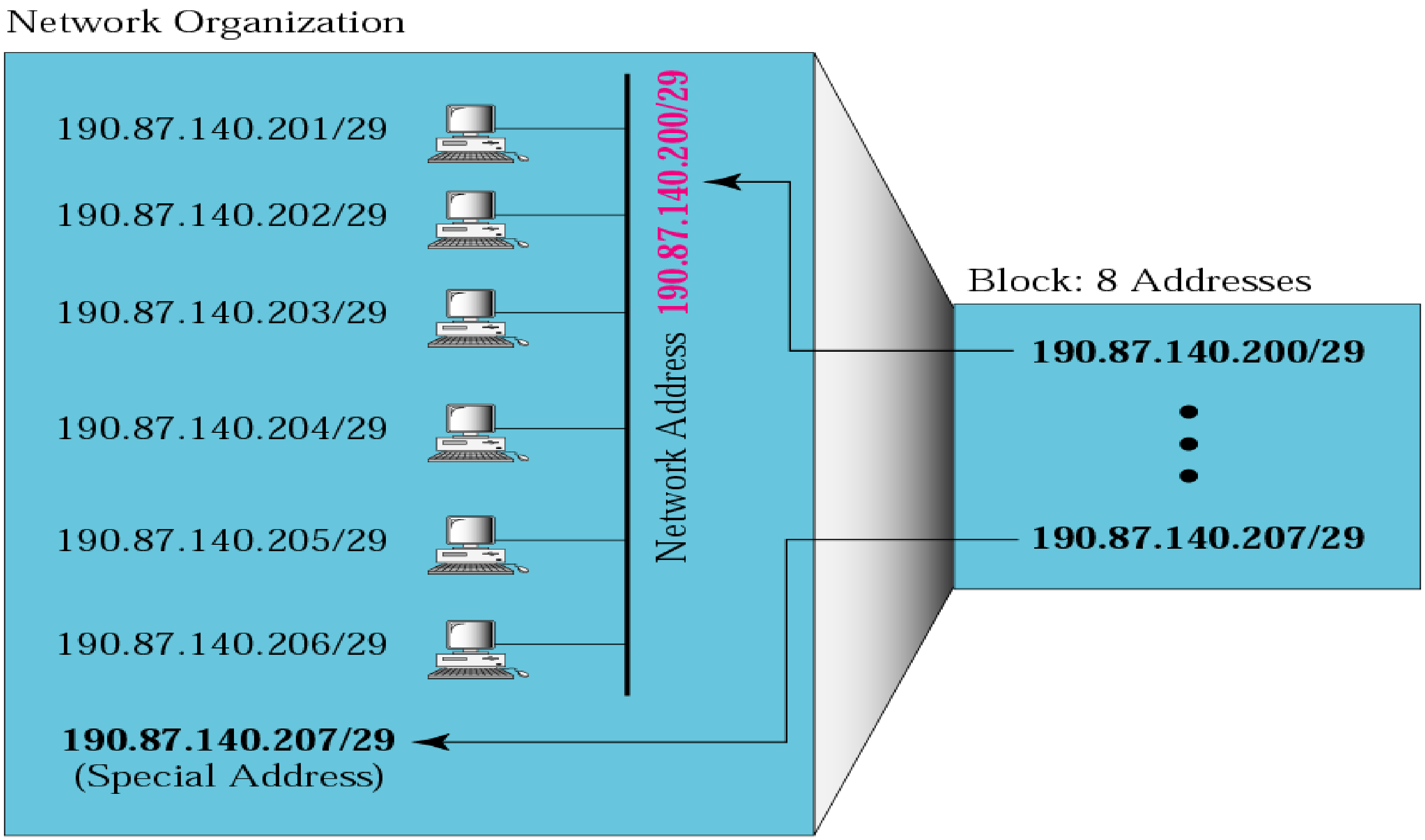
The leftmost 5 numbers are → **128 + 64 + 0 + 0 + 8**

The first address is 190.87.140.200/29

The number of addresses is 2^{32-29} or 8.

To find the last address, use the complement of the mask. The mask has twenty-nine 1s; the complement has three 1s. The complement is 0.0.0.7. Add this to the first address to get **190.87.140.207/29**. So, the first address is 190.87.140.200/29, the last address is 190.87.140.207/29. There are only 8 addresses in this block.

Previous Example: Network Configuration



In classless addressing, the last address in the block does not necessarily end in 255.

- In classless addressing, an address can belong to many blocks (depending on value of prefix associated with that block).

- For example, consider the address **230.8.24.56**

  - It can belong to many blocks

Prefix length	**Block**	
	From	To
16	230.8.0.0	230.8.255.255
20	230.8.16.0	230.8.31.255
26	230.8.24.0	230.8.24.63
27	230.8.24.32	230.8.24.63
29	230.8.24.56	230.8.24.63
31	230.8.24.56	230.8.24.57

Variable Length Subnet Mask (VLSM)

- VLSM is a way of further sub-netting a subnet.

- we can allocate IPv4 addresses to the subnets by the exact need by using VLSM

- VLSM allows us to use more than one subnet mask within the same network address space.

- In classful addressing, we can divide a network only into subnets with equal number of IPv4 addresses.

- VLSM allows to create subnets from a single network with unequal number of IPV4 addresses

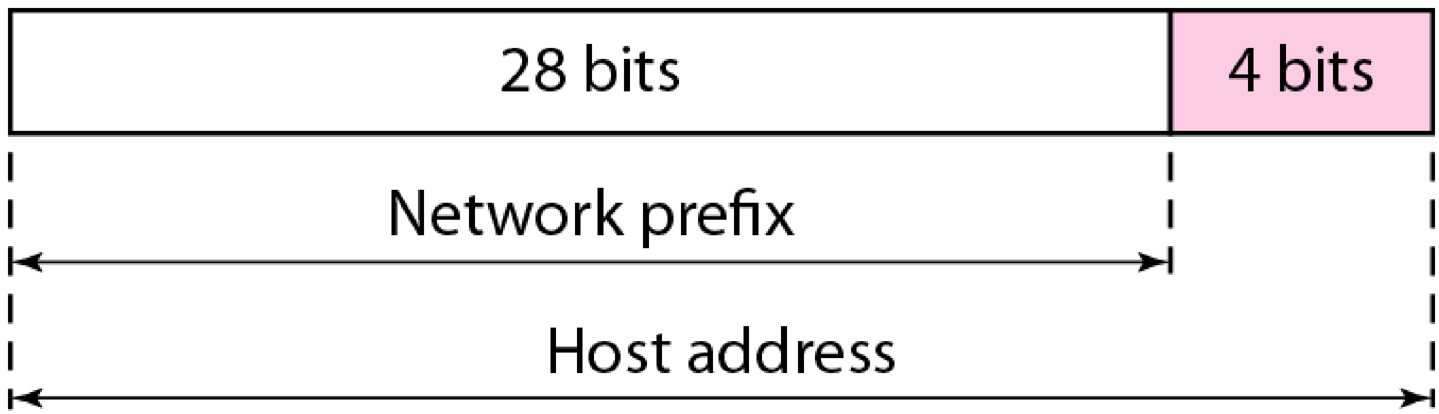
Variable Length Subnet Mask (VLSM)

- Suppose we want to divide 192.168.10.0 (a Class C network) into four networks each with unequal number of address as per the following requirements:

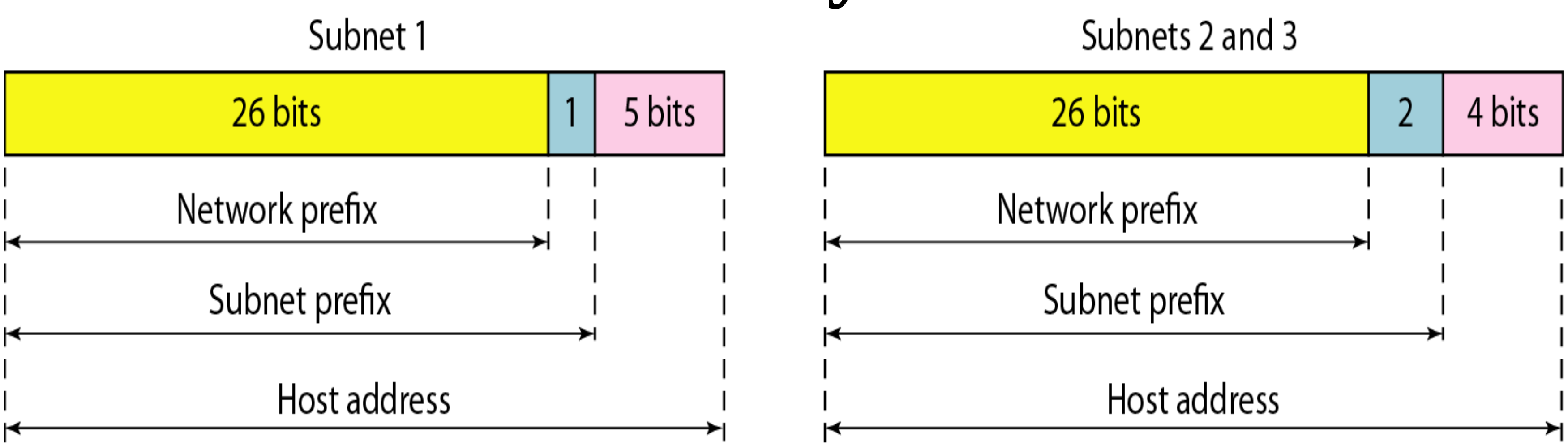
    Subnet A : 126 IPv4 Addresses.
    Subnet B : 62 IPv4 Addresses.
    Subnet C : 30 IPv4 Addresses.
    Subnet D : 30 IPv4 Addresses.

- Such division is not possible in classful addressing, since it divides the network equally,
- **but it is possible with VLSM.**

Two-level hierarchy in an IPv4 address



Three-level hierarchy in an IPv4 address



Variable Length Subnet Mask (VLSM)

Division of **192.168.10.0/24 (original network)** into four networks **with VLSM.**

FIRST DIVISION

- Divide into two networks equally with 128 addresses (126 usable) using subnet mask 255.255.255.128

1. 192.168.10.0/25 [255.255.255.128]
   - 11000000.10101000.00001010.00000000 [in binary]
     11111111.11111111.11111111.10000000 [subnet mask]
2. 192.168.10.128/25 [255.255.255.128]
   - 11000000.10101000.00001010.10000000
     11111111.11111111.11111111.10000000 [subnet mask]

Variable Length Subnet Mask (VLSM)

SECOND DIVISION

- Divide second subnet 192.168.10.128/25 (obtained from first division) again into two networks,
- Each with 64 addresses (62 usable) using subnet mask 255.255.255.192

1. 192.168.10.128/26 [255.255.255.192]
   - 11000000.10101000.00001010.10000000 [in binary]
     11111111.11111111.11111111.11000000 [subnet mask]
2. 192.168.10.192/26 [255.255.255.192]
   - 11000000.10101000.00001010.11000000
     11111111.11111111.11111111.11000000 [subnet mask]

Variable Length Subnet Mask (VLSM)

THIRD DIVISION

- Divide second subnet 192.168.10.192/26 (obtained from second division) again into two networks,
- each with 32 addresses (30 usable) using subnet mask 255.255.255.224
  1. 192.168.10.192/27 [255.255.255.224]
     - 11000000.10101000.00001010.11000000 [in binary]
       11111111.11111111.11111111.11100000 [subnet mask]
  2. 192.168.10.224/27 [255.255.255.224]
     - 11000000.10101000.00001010.11100000
       11111111.11111111.11111111.11100000 [subnet mask]

- So, splitting of 192.168.10.0/24 into four subnets using VLSM with unequal number of addresses is done

Example 2:

An organization is granted a block of addresses starting with 17.12.14.0/26 (64 addresses).

The organization needs to have three sub-blocks of addresses to use in its three subnets:

- one sub-block of 32 addresses, and
- two sub-blocks of 16 addresses each.

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

X.X.X.000xxxxx

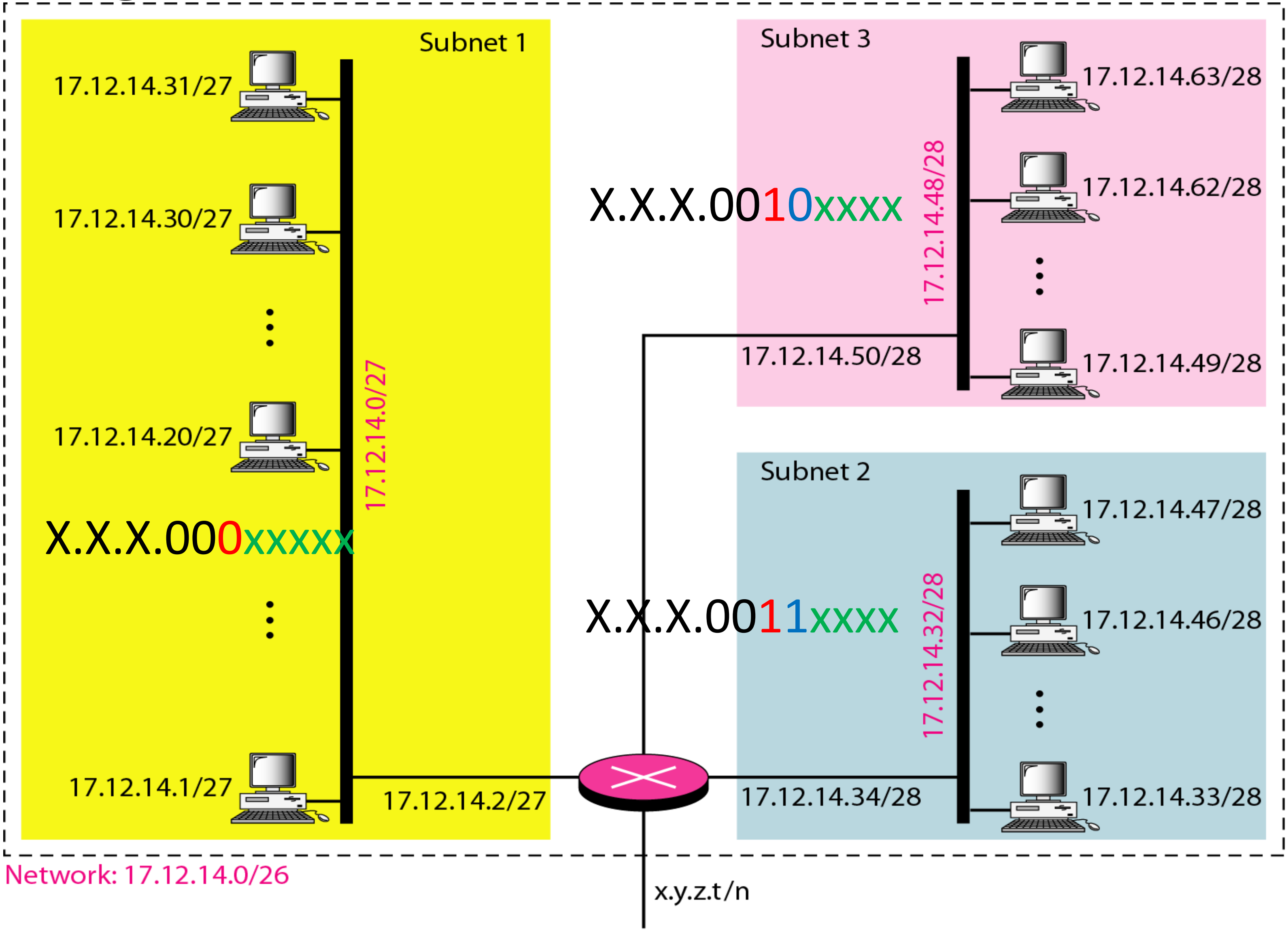
X.X.X.00xxxxxx

X.X.X.0010xxxx

X.X.X.001xxxxx

X.X.X.0011xxxx

Solution of Example 2:
Configuration and addresses in a sub-netted network



Example 3:

An organization is granted a block of addresses starting with 14.24.74.0/24.

The organization needs to have three sub-blocks of addresses to use in its three subnets:

- one sub-block of 10 addresses,
- one sub-block of 60 addresses, and
- one sub-block of 120 addresses.

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

Solution to Example 3:

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

Assign addresses to sub-blocks starting with the largest and ending with the smallest one.

- Mask n1 for the **first** (largest) subnet 2^{32-n1} must be 128 (a number with power of 2 nearest to 120). **So n1 = 25.** We allocate 128 addresses instead of 120 to this subnet

  - The first address in this subnet is 14.24.74.0/25 and last address is 14.24.74.127/25

Solution (previous example 3 continue)

- Mask for the **second subnet** 2^{32-n2} must be 64 (a number with power of 2 nearest to 60). **So n2 = 26**
  - The first address in this subnet is 14.24.74.128/26 and last address is 14.24.74.191/26

- Mask for the **third subnet** 2^{32-n3} must be 16 (a number with power of 2 nearest to 10). So **n3 = 28**
  - The first address in this subnet is 14.24.74.192/28 and last address is 14.24.74.207/28

- So we have 128+64+16 = 208 addresses in all three sub-blocks.
  - Therefore, 48 address are still left in reserve.

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.

Example (continued)

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 $32 - 8 = 24$.

Example (continued)

X.X.00000000.x
X.X.00000001.x
…
X.X.00111111.x
…

Solution

Group 1: 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

Example (continued)

X.X.01000000.0--------
X.X.01000000.1--------
...
X.X.01111111. 0--------
X.X.01111111. 1--------
...

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 = $128 \times 128 = 16{,}384$

Example (continued)

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

Example (continued)

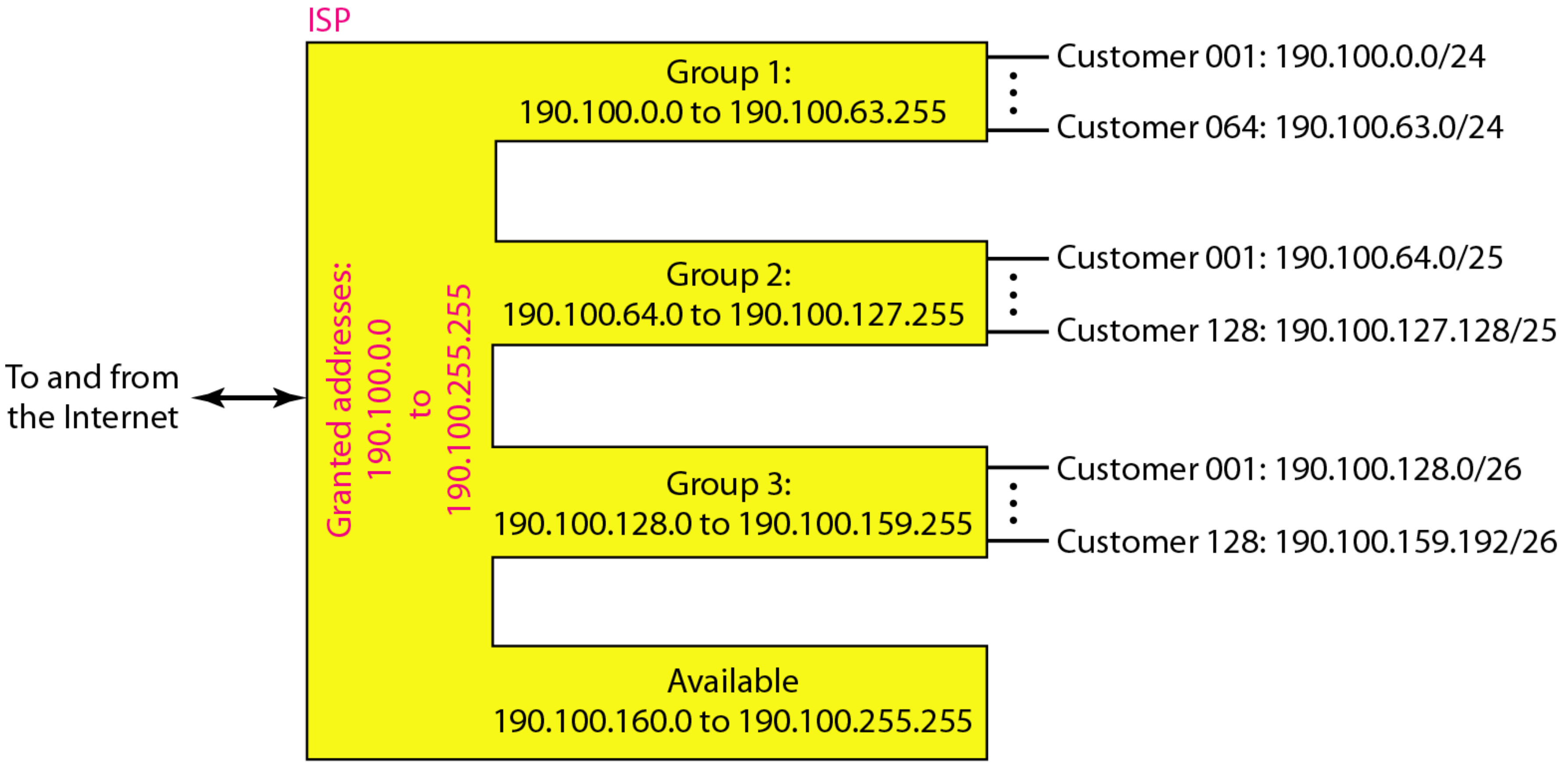
X.X.10000000.00------
X.X.10000000.01------
X.X.10000000.10------
X.X.10000000.11------
...
X.X.10011111. 10-------
X.X. 10011111. 11-------
...

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

$Total = 128 \times 64 = 8192$

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

Figure: An example of address allocation and distribution by an ISP



Address depletion and IPv6

Classless addressing is a short term solution to solve the address depletion problem

- Uses the same address space but change the distribution of addresses to provide a fair share to each organization.

- Still uses the IPv4 addresses (class privilege was removed from the distribution)

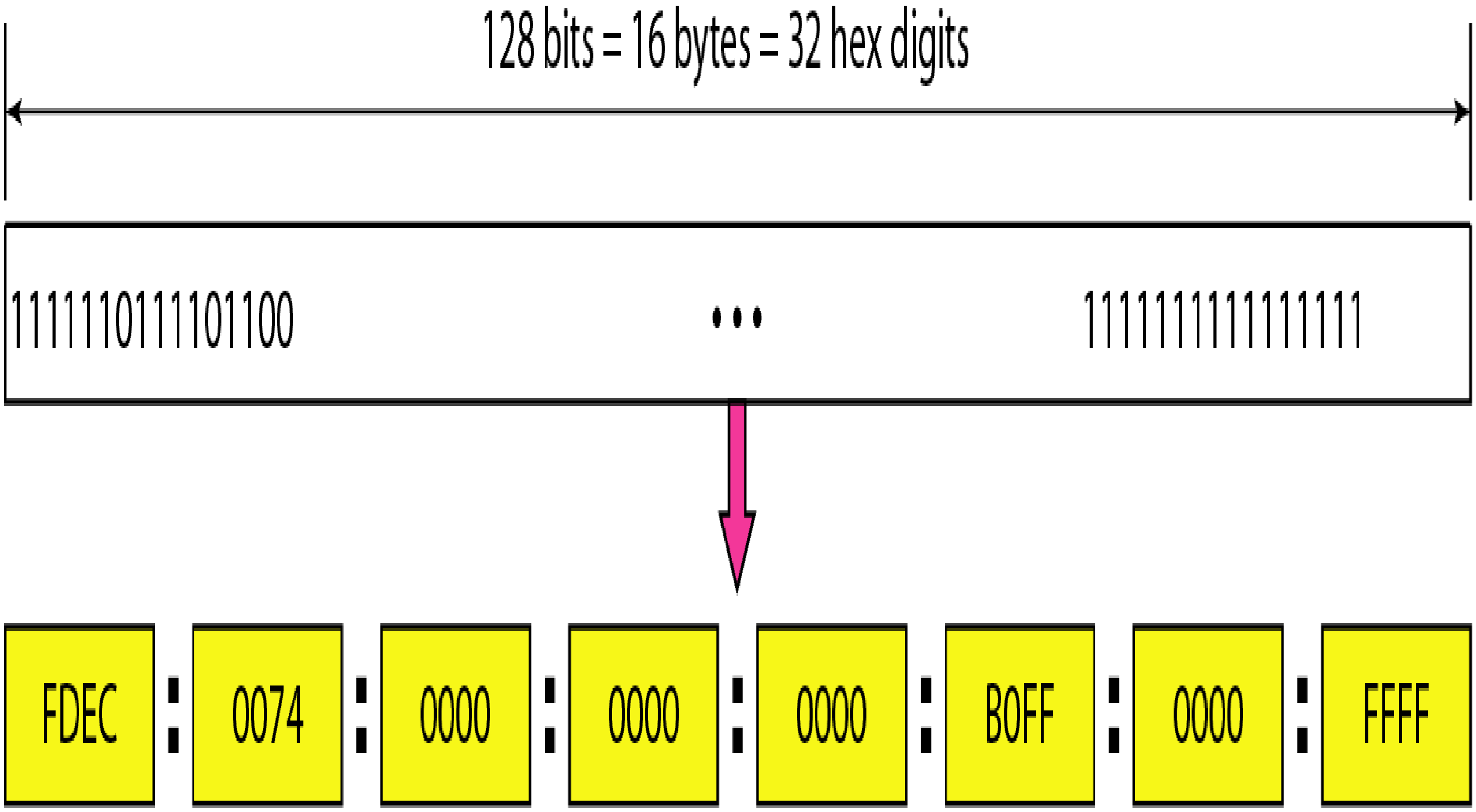
Address depletion and IPv6

- Despite all short-term solutions, address depletion is still a long-term problem for the Internet.

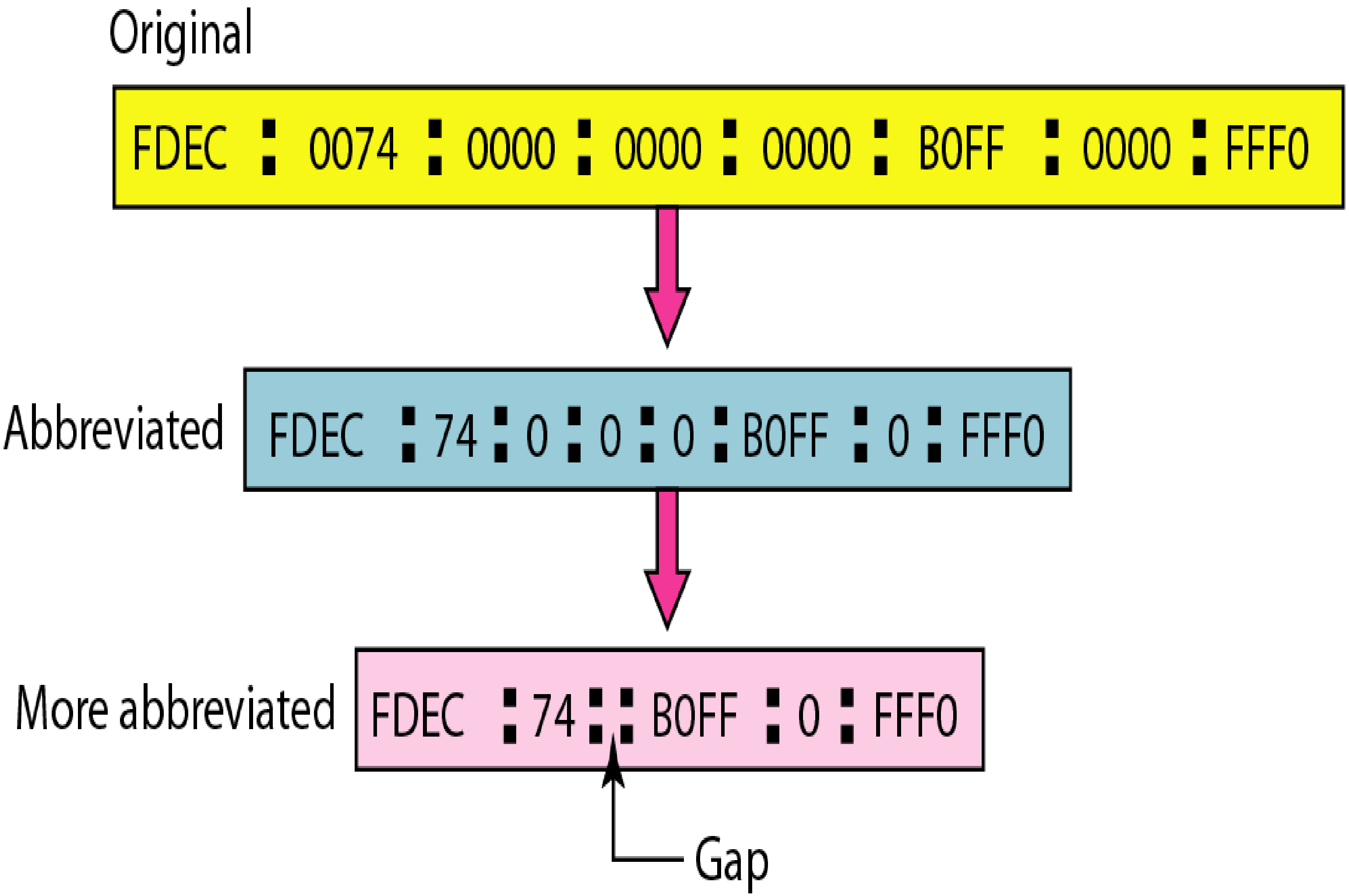
- This and other problems in the IP protocol itself have been the motivation for already devised **IPv6**.
  - **The long-range** solution
  - The larger address space is obtained by **increasing the length of IP addresses** (128 bits)

- It means that **format of IP** packets need to be changed

IPv6 ADDRESSES



Abbreviated IPv6 addresses



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

in order 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
```