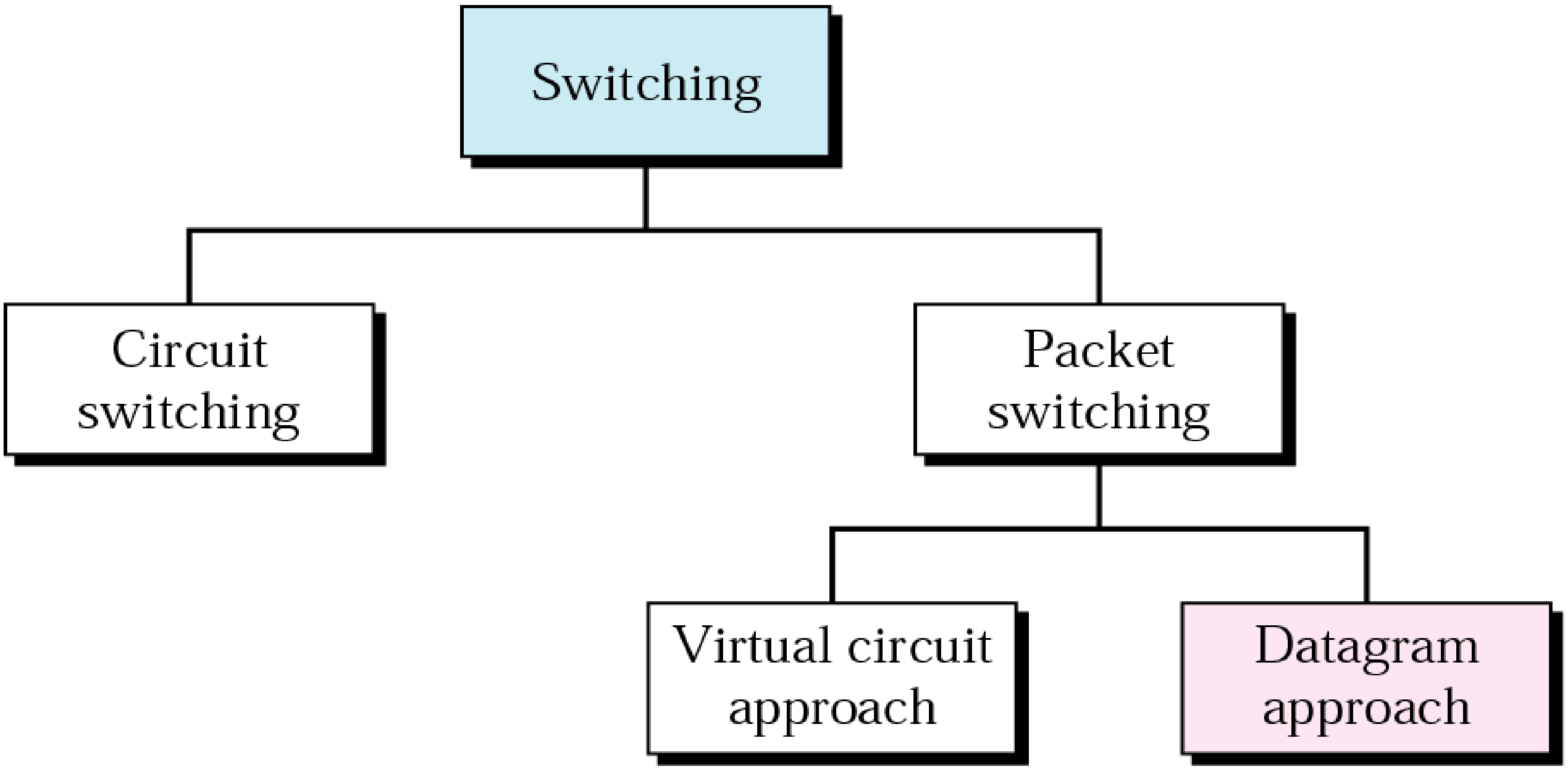
# Data Communication & Computer Networks

7. Network Layer :
Logical Addressing

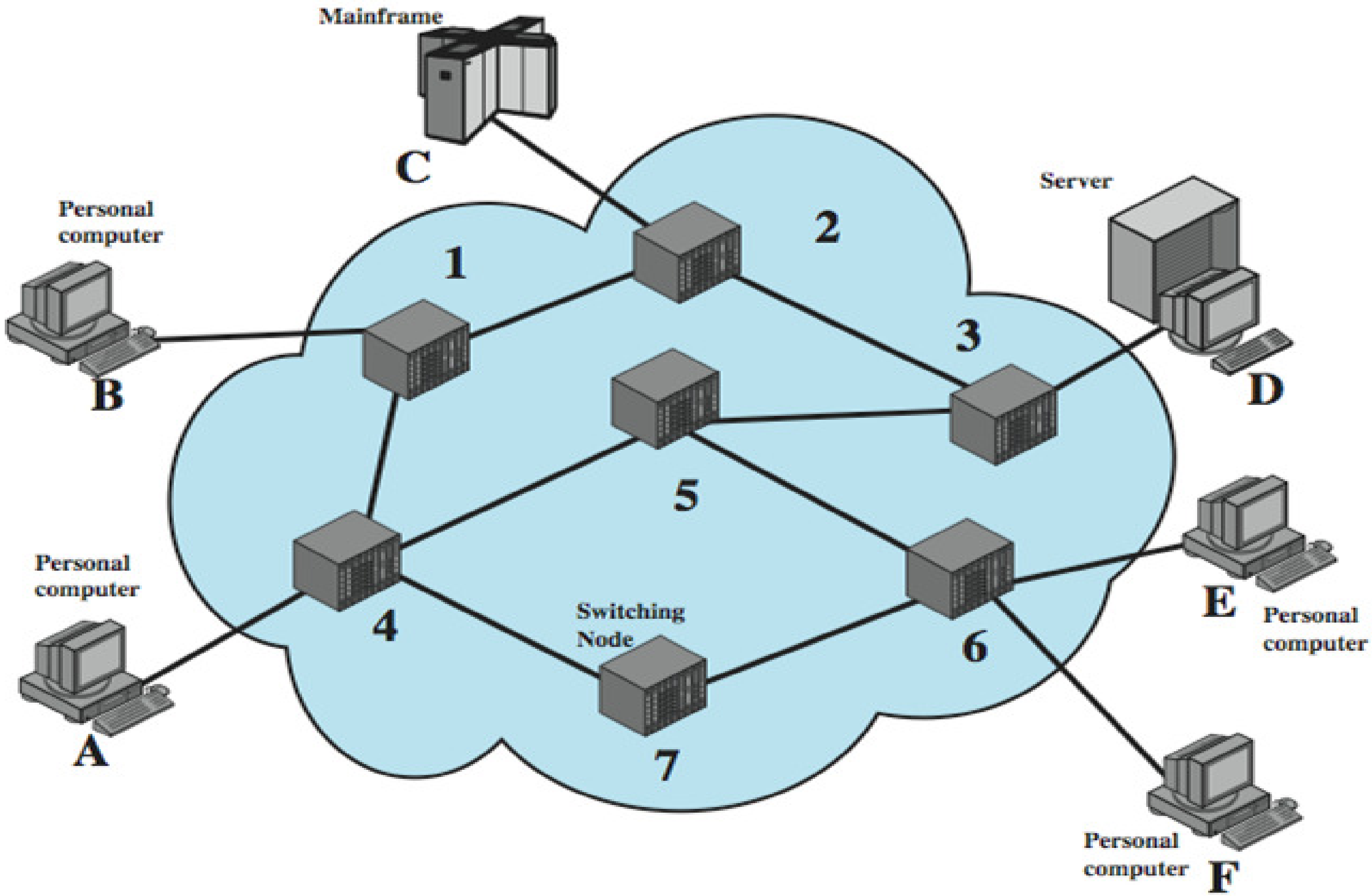
# Background

# Switching



Switching
Circuit switching
Packet switching
Virtual circuit approach
Datagram approach

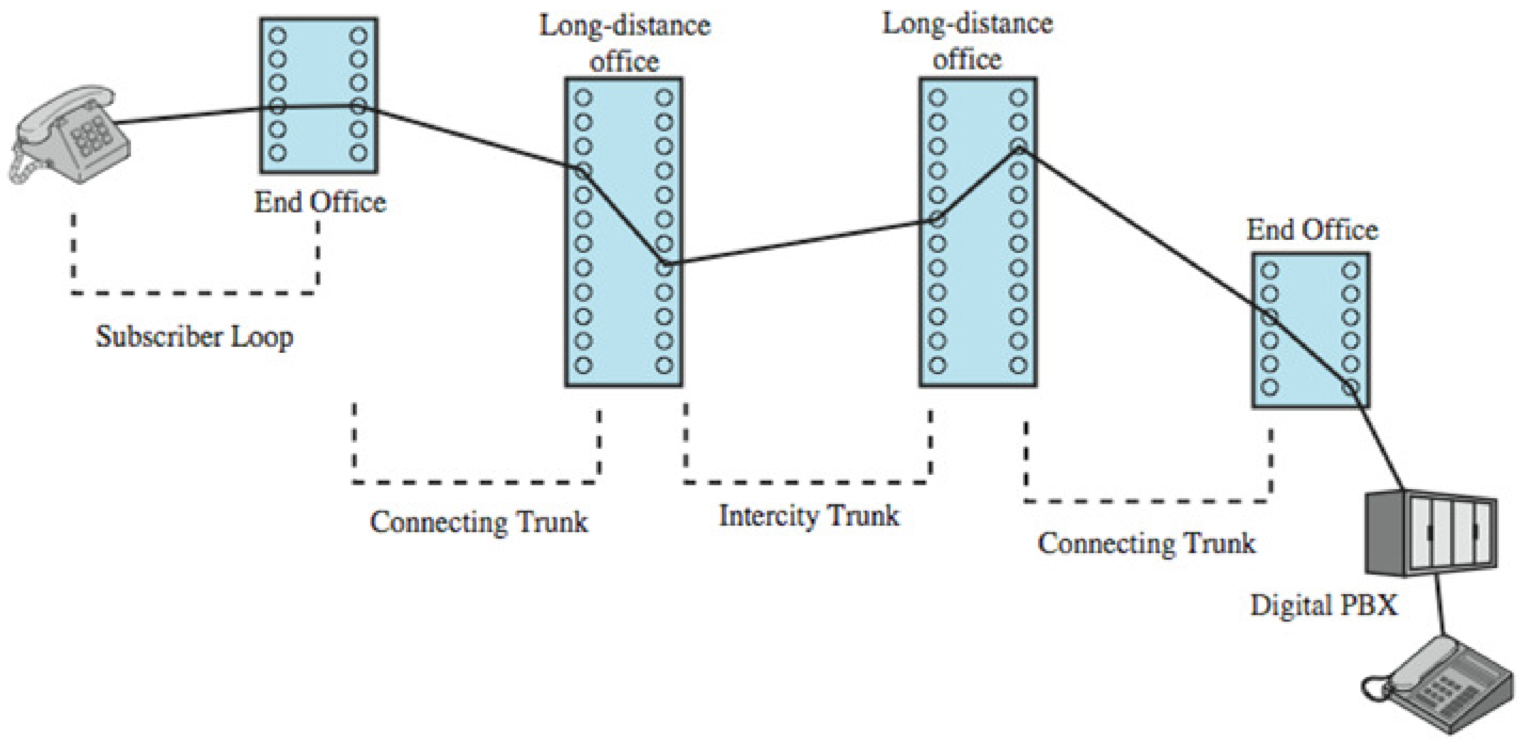
# Switched Network



# Circuit Switching

- uses a dedicated path between two stations
- once connected, transfer is transparent
- set up (connection) takes time
- inefficient
  - channel capacity dedicated for duration of connection
  - if no data, capacity wasted

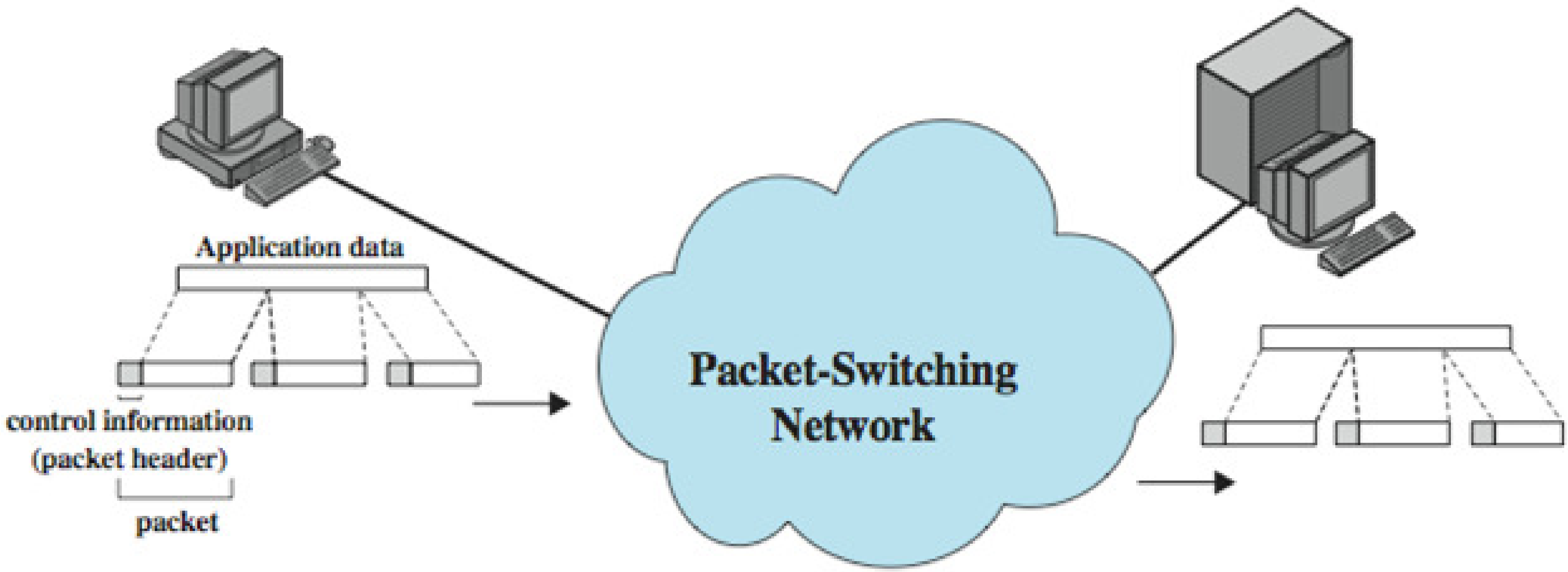
# Public Circuit Switched Network



# Packet Switching

- circuit switching was designed for voice
- packet switching was designed for data transmitted in small packets
- packets contains user data and control info
  - user data may be part of a larger message
  - control info includes routing (addressing) info
- Advantage: line efficiency
  - single link shared by many packets over time

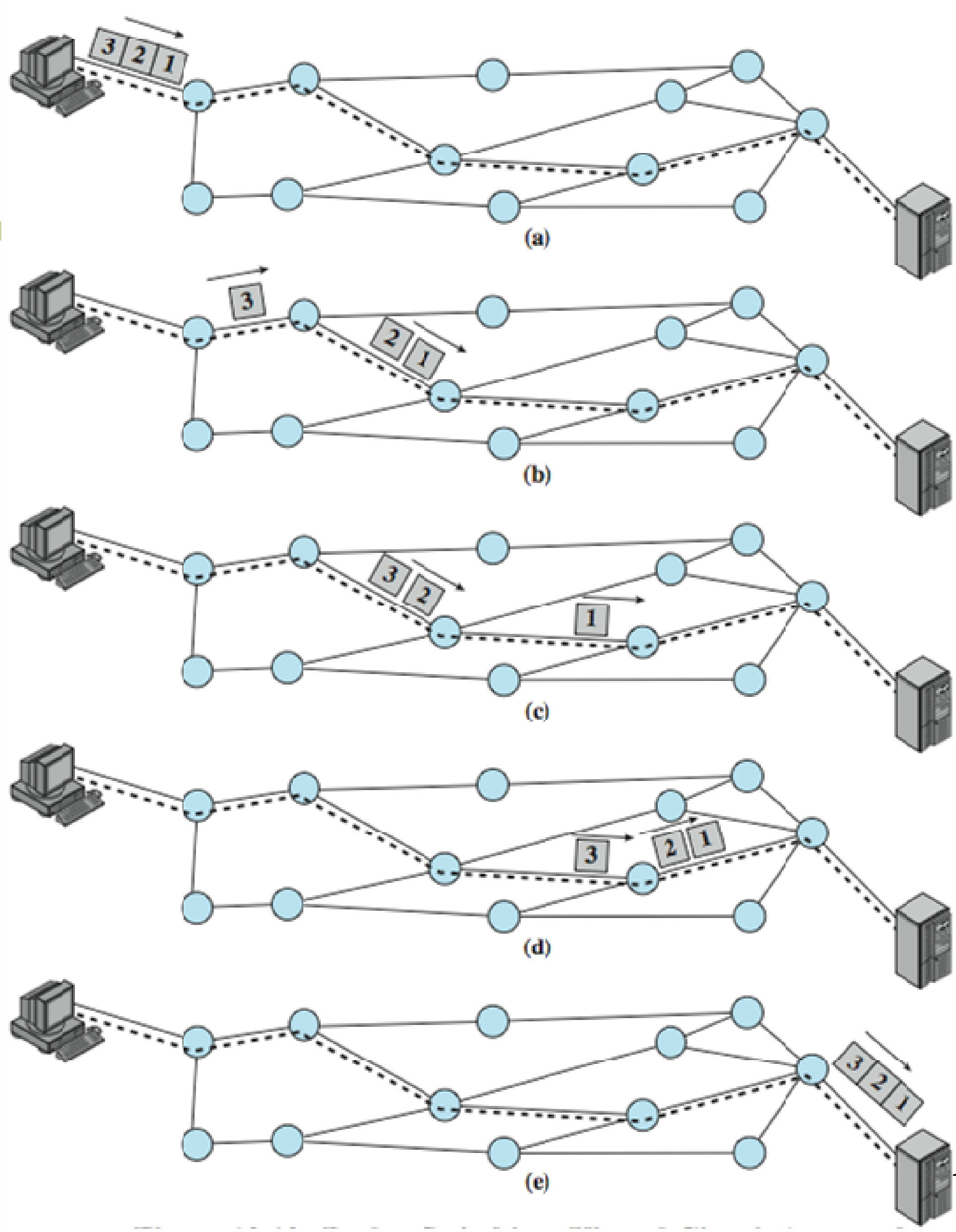
# Packet Switching



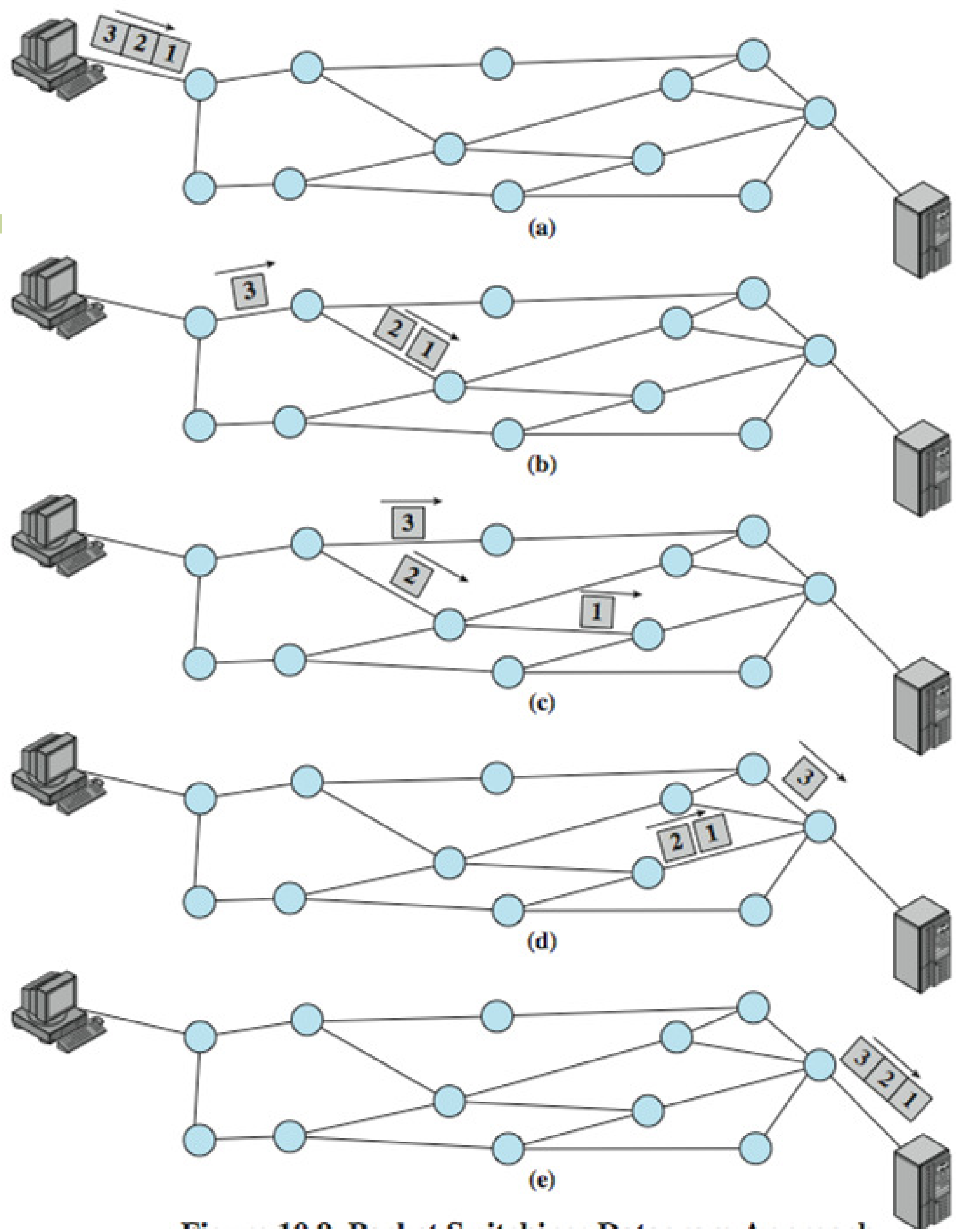
# Switching Techniques

- station breaks long message into packets
- packets sent one at a time to the network
- packets can be handled in two ways
  - virtual circuit
  - datagram

# Virtual Circuit Approach



# Datagram Approach

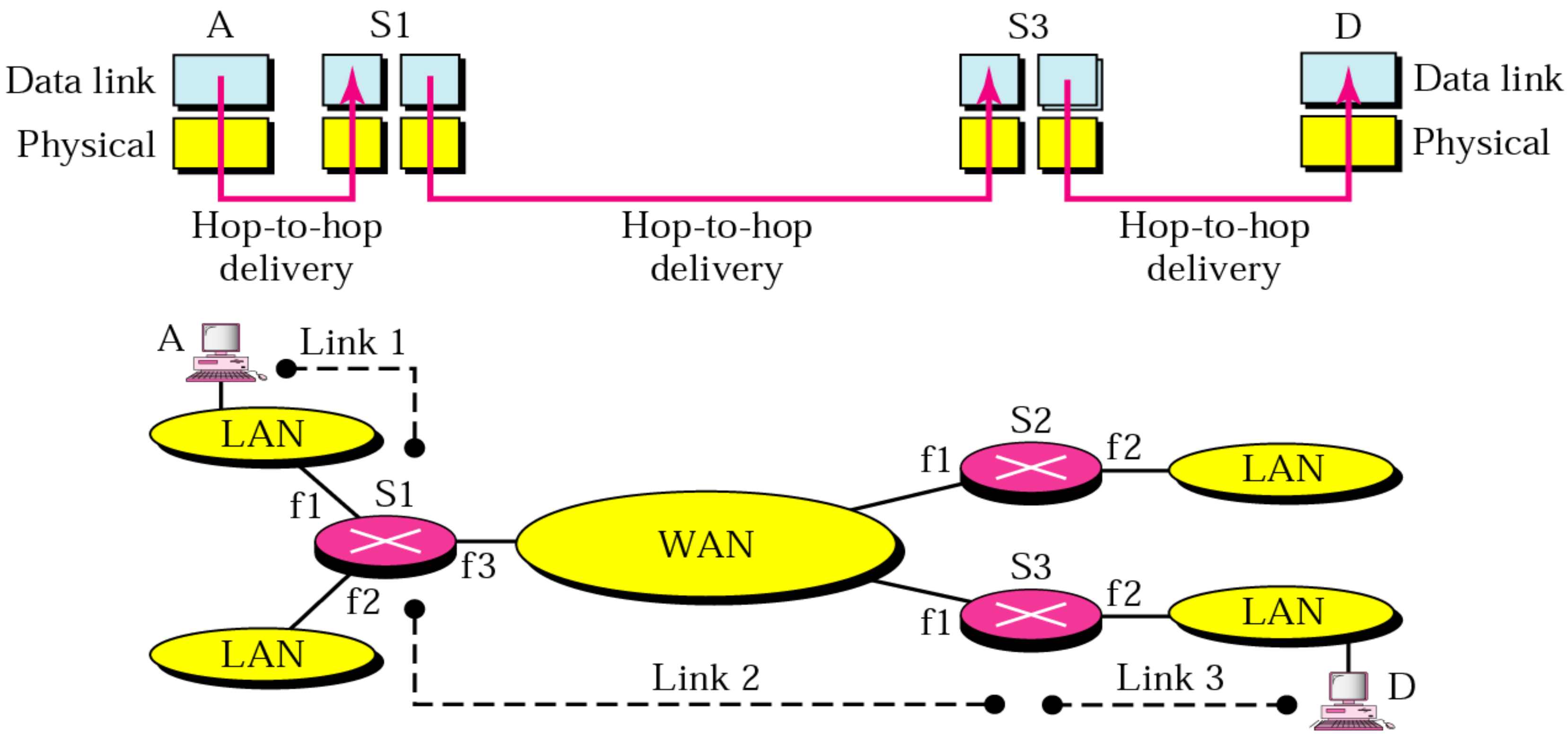


# Virtual Circuits vs. Datagram

- virtual circuits
  - network can provide sequencing and error control
  - packets are forwarded more quickly
  - less reliable
- datagram
  - no call setup phase
  - more flexible
  - more reliable

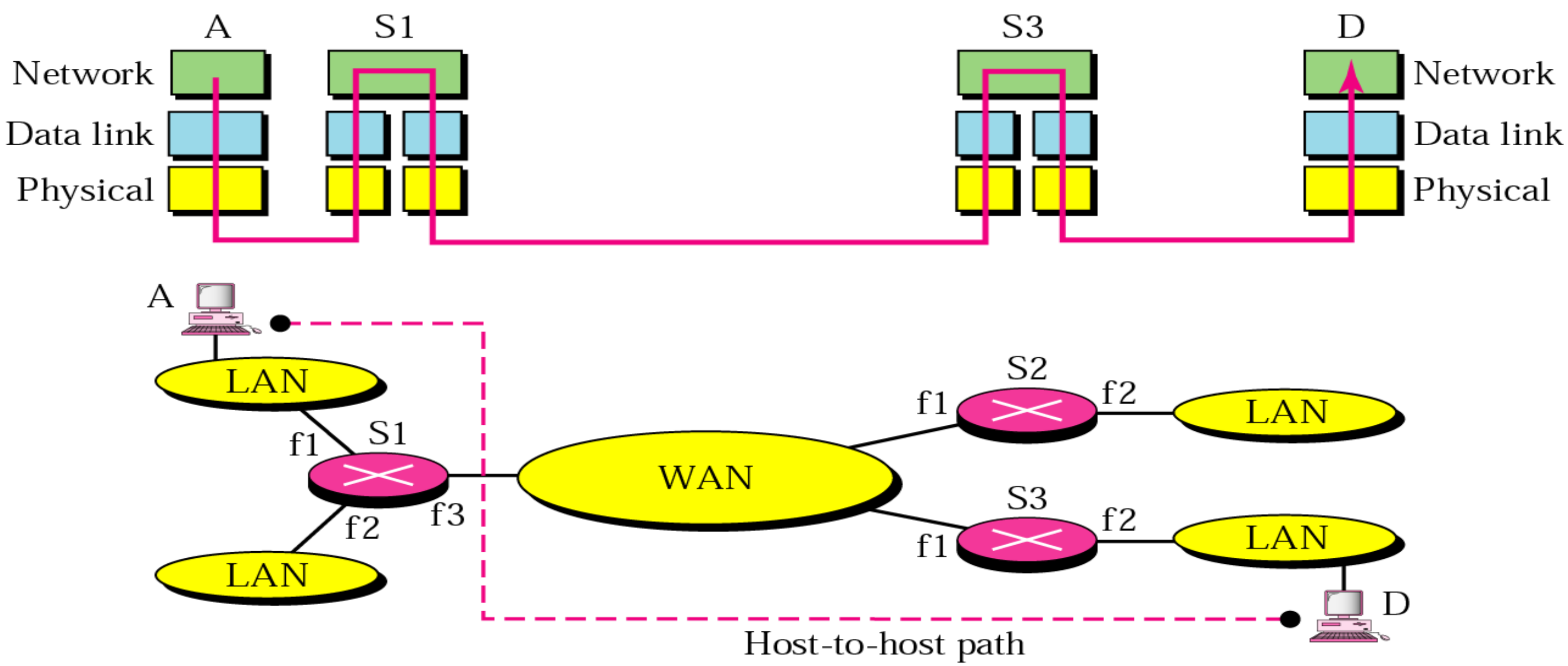
# Links

- The physical and data link layers deliver data within a network.

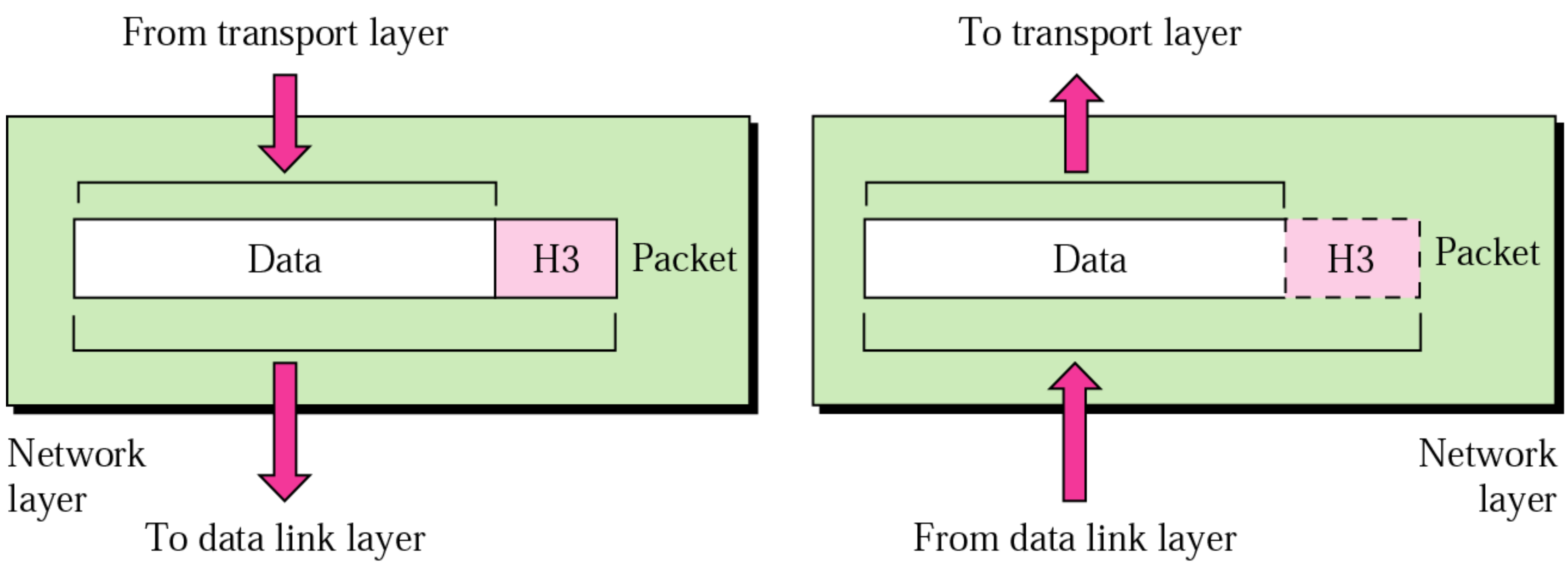


# Network Layer

- Network layer is responsible for host-to-host delivery of data across networks



# Network Layer



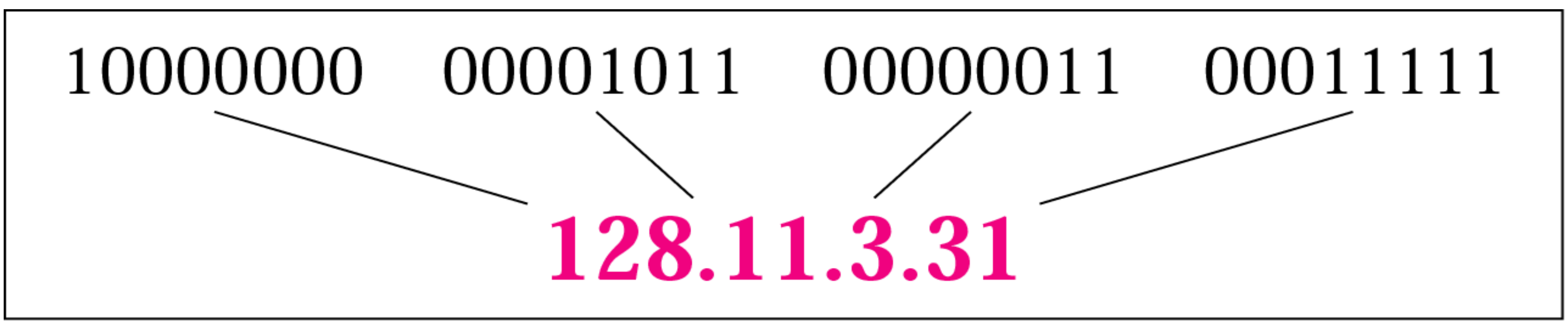
# IPv4 Addresses

# Network layer addressing

- An *Internet address* or *IP address* uniquely identifies a device connected to the Internet at the network layer
- IP = Internet Protocol
- IP is the base protocol at network layer in the Internet model or the TCP/IP protocol suit

# IPv4 address

- An IPv4 address is a 32-bit address
- The address space of IPv4 is $2^{32}$ or 4,294,967,296.
- Represented in 4 blocks of 1 byte each



# 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

# Classful Addressing

In classful addressing, the address space is divided into five classes:

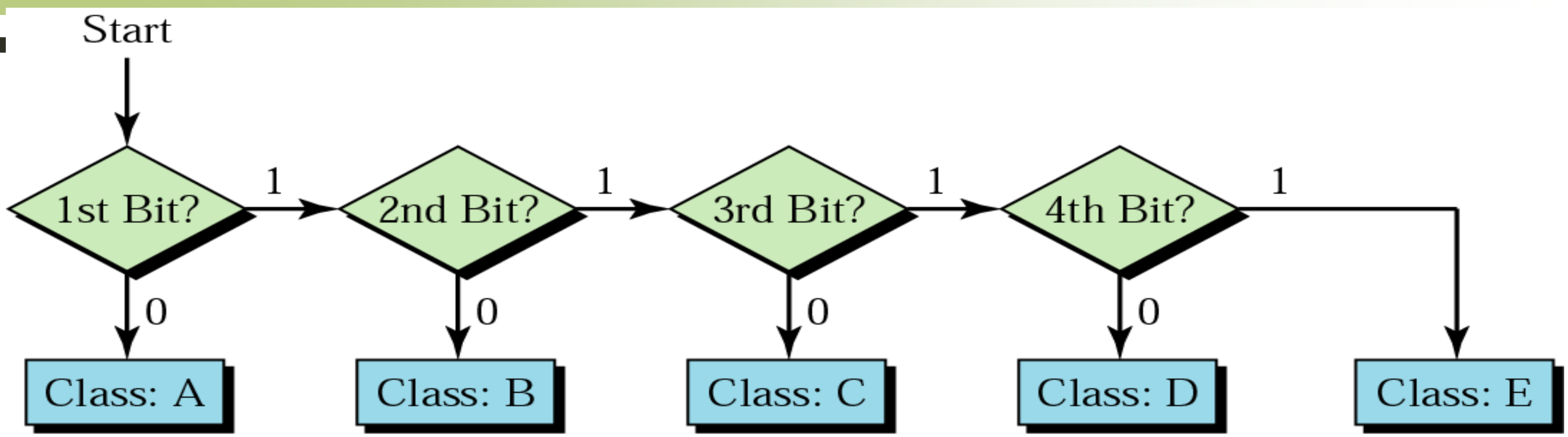
- A
- B
- C
- D
- E

Unicast communication

Multicast communication

Reserved for special use

# Finding the class in binary notation




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

# Finding the class in dotted-decimal notation

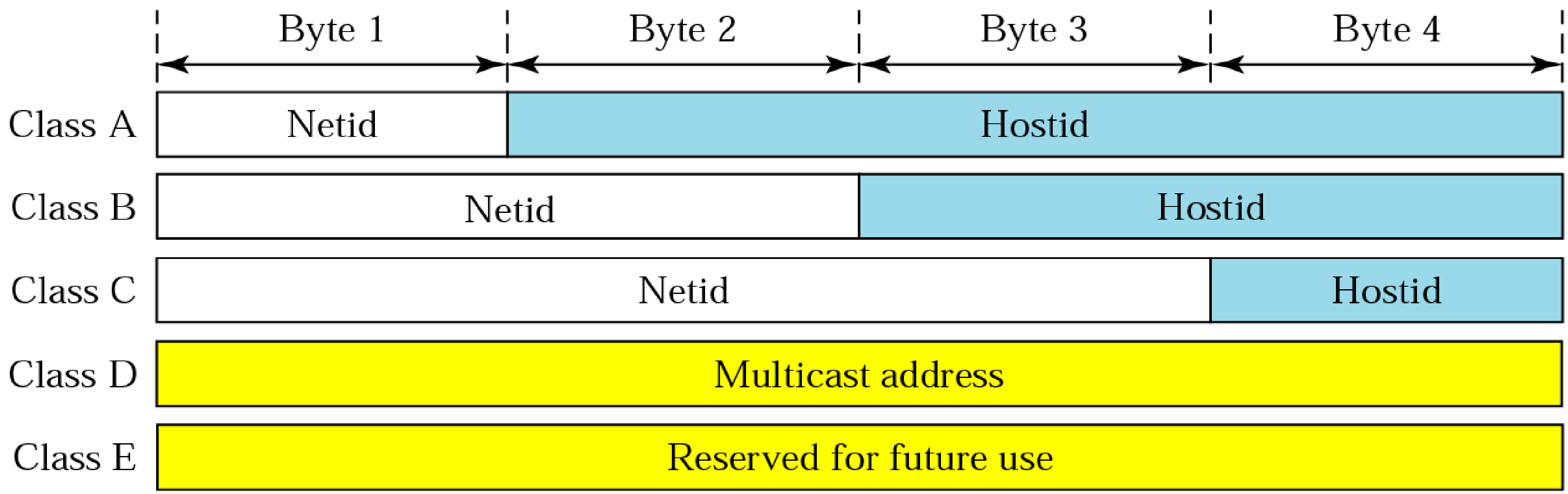
| | First byte | Second byte | Third byte | Fourth byte |
|---|---|---|---|---|
| Class A | 0 | | | |
| Class B | 10 | | | |
| Class C | 110 | | | |
| Class D | 1110 | | | |
| Class E | 1111 | | | |

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

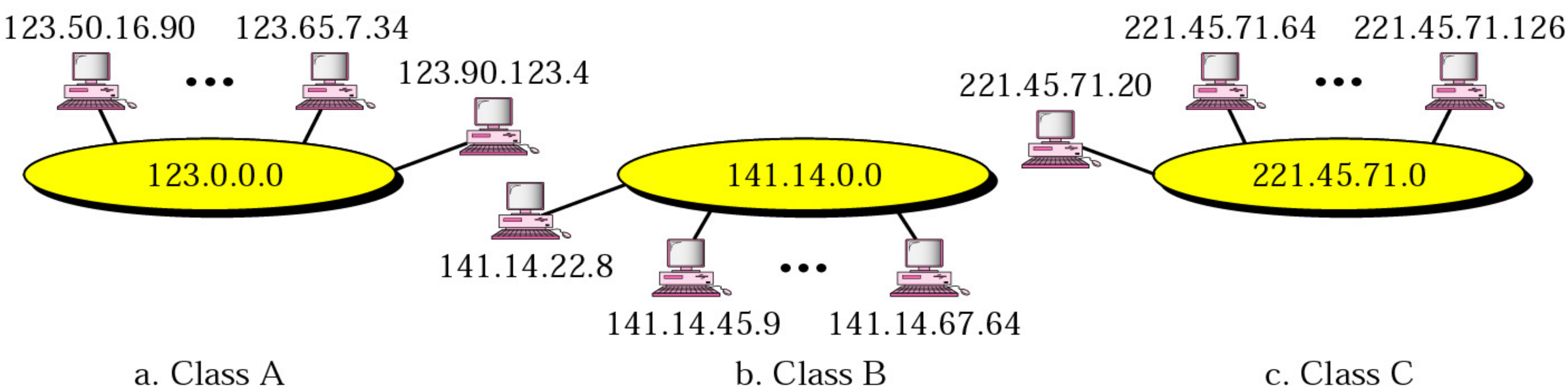
# Network ID and Host ID



# Network Address

- Network address is an address that defines the network itself; it cannot be assigned to a host.
- A network address is different from a netid. A network address has both netid and hostid, with 0s for the hostid.

| Netid | Hostid |
| --- | --- |
| Specific | All 0s |

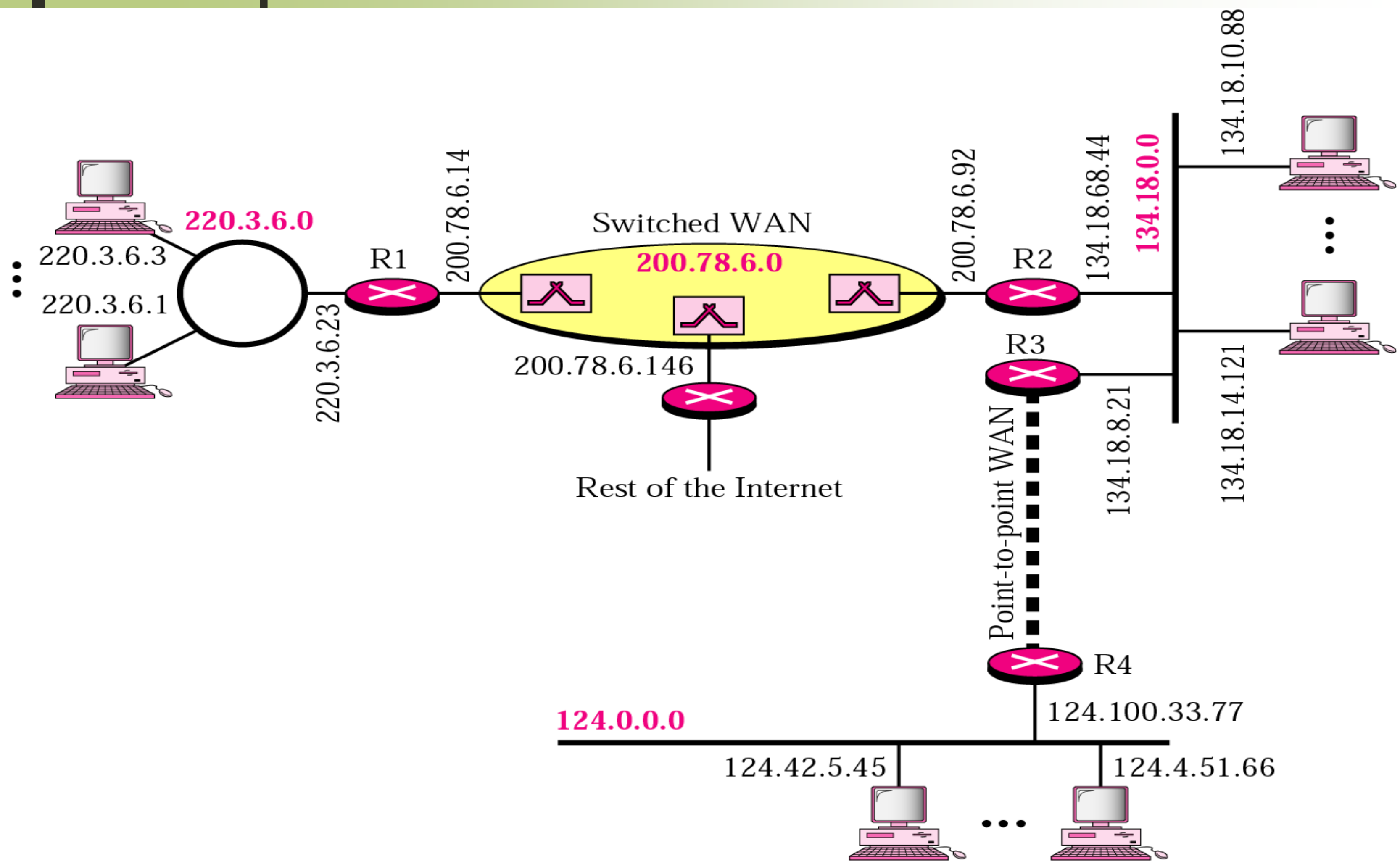


# Mask

- Although the length of the netid and hostid (in bits) is predetermined in classful addressing, we can also use a mask
- The mask can help us to find the netid and 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 |

# Sample Internet



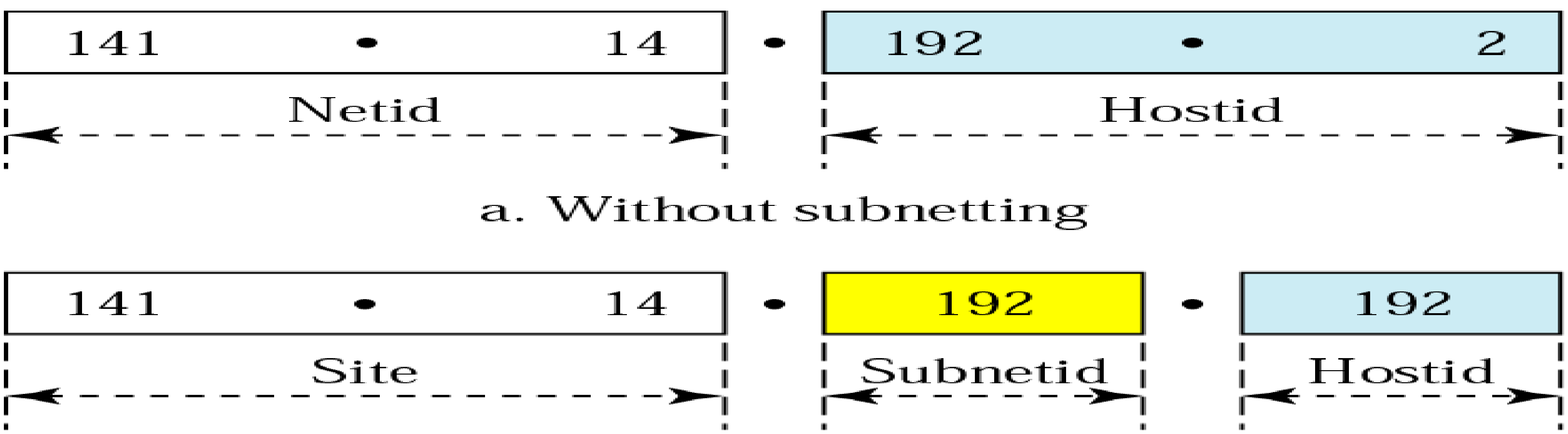
# Subnetwork

- Q. What if an organisation wants to
  - arrange its hosts into groups?
  - break its large network into smaller networks?

- A. Divide its network into smaller *subnetworks* or *subnets*.

# Subnets and Subnet Masks

- host portion of address partitioned into subnet number and host number
- local routers route within subnetted network
- subnet mask indicates which bits are subnet number and which are host number

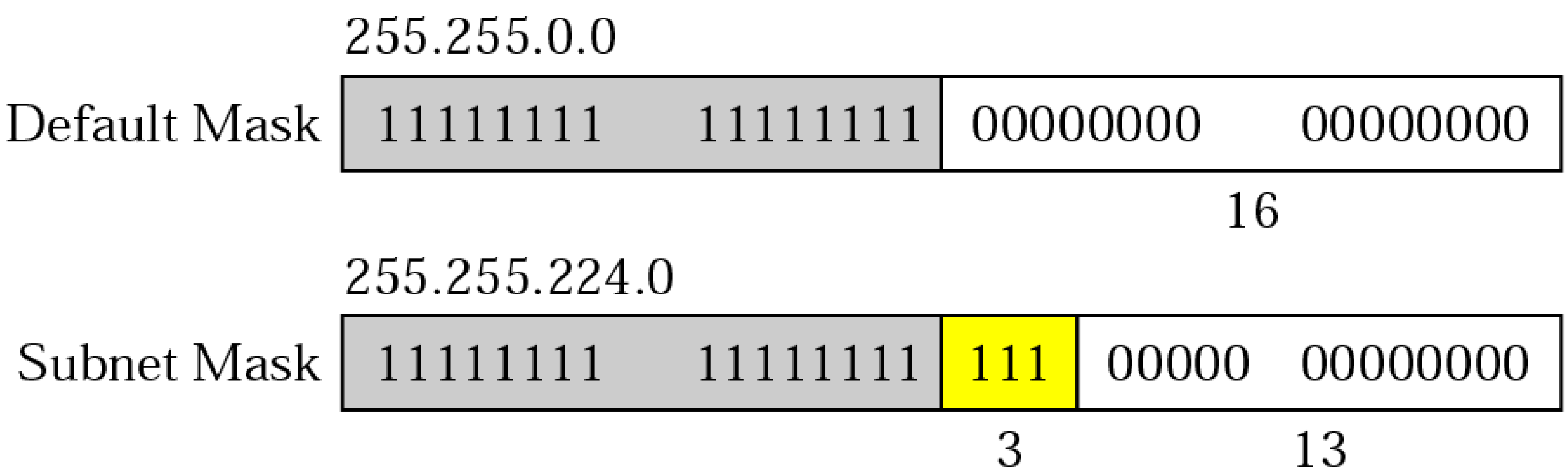



a. Without subnetting

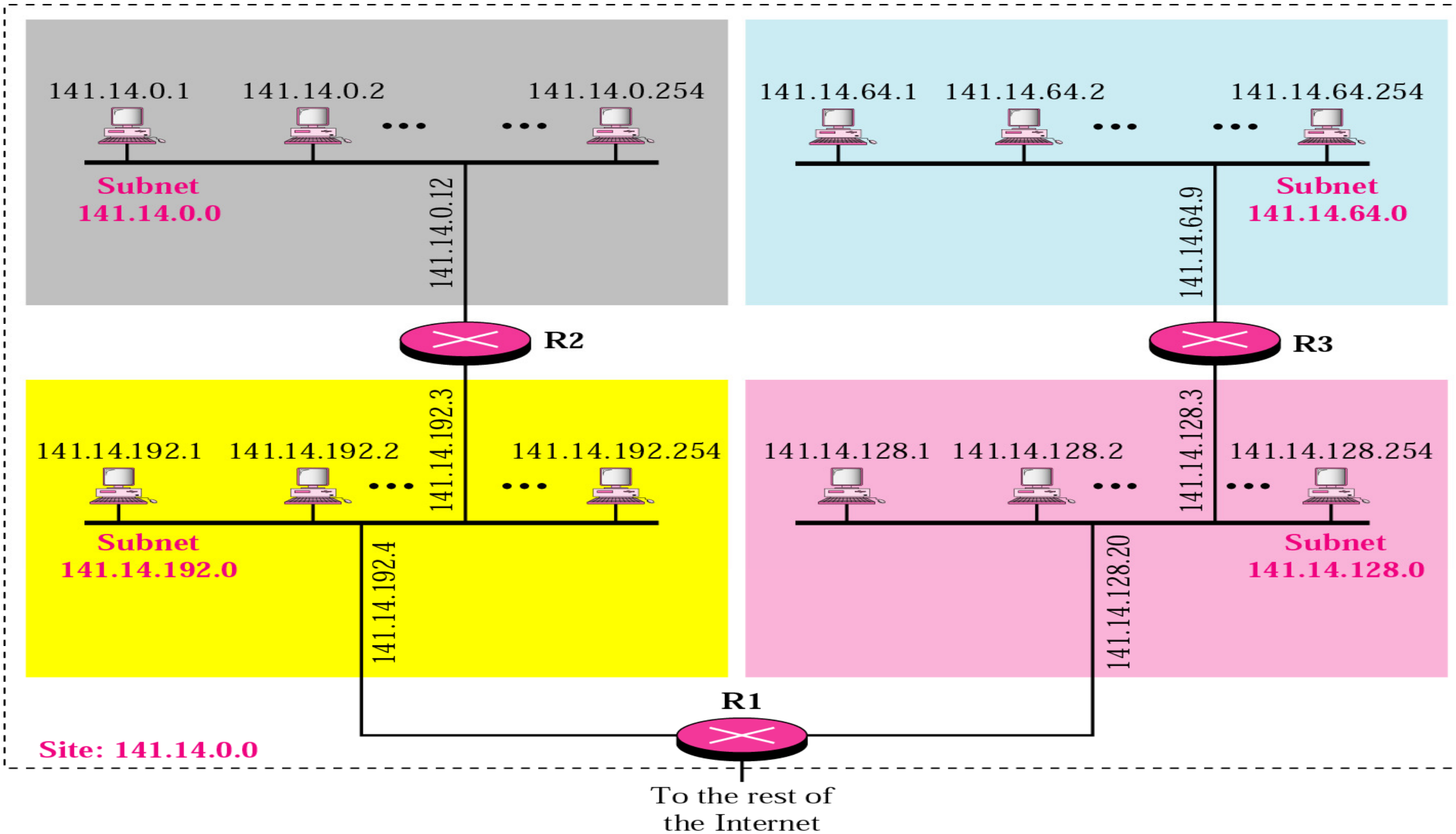
b. With subnetting

# Subnet Masks

- Routers outside a subnetted organisation use default masks
- Routers inside the organisation use subnet masks



# Subnetworks



# Supernetting

- In supernetting several blocks of addresses (networks) are combined to create a supernetwork of large address range
  - e.g. an organisation with 1000 computers can be allocates four class C network addresses to satisfy its needs
    - 254 x 4 >1000

# Classless Addressing

- Problem with classful addresses
  - wastage due fix blocks of addresses
- Solution: Classless addressing
  - Variable length address blocks
  - No classes
  - CIDR = Classless Inter-Domain Routing
- Classful addressing is almost obsolete and replaced with classless addressing.

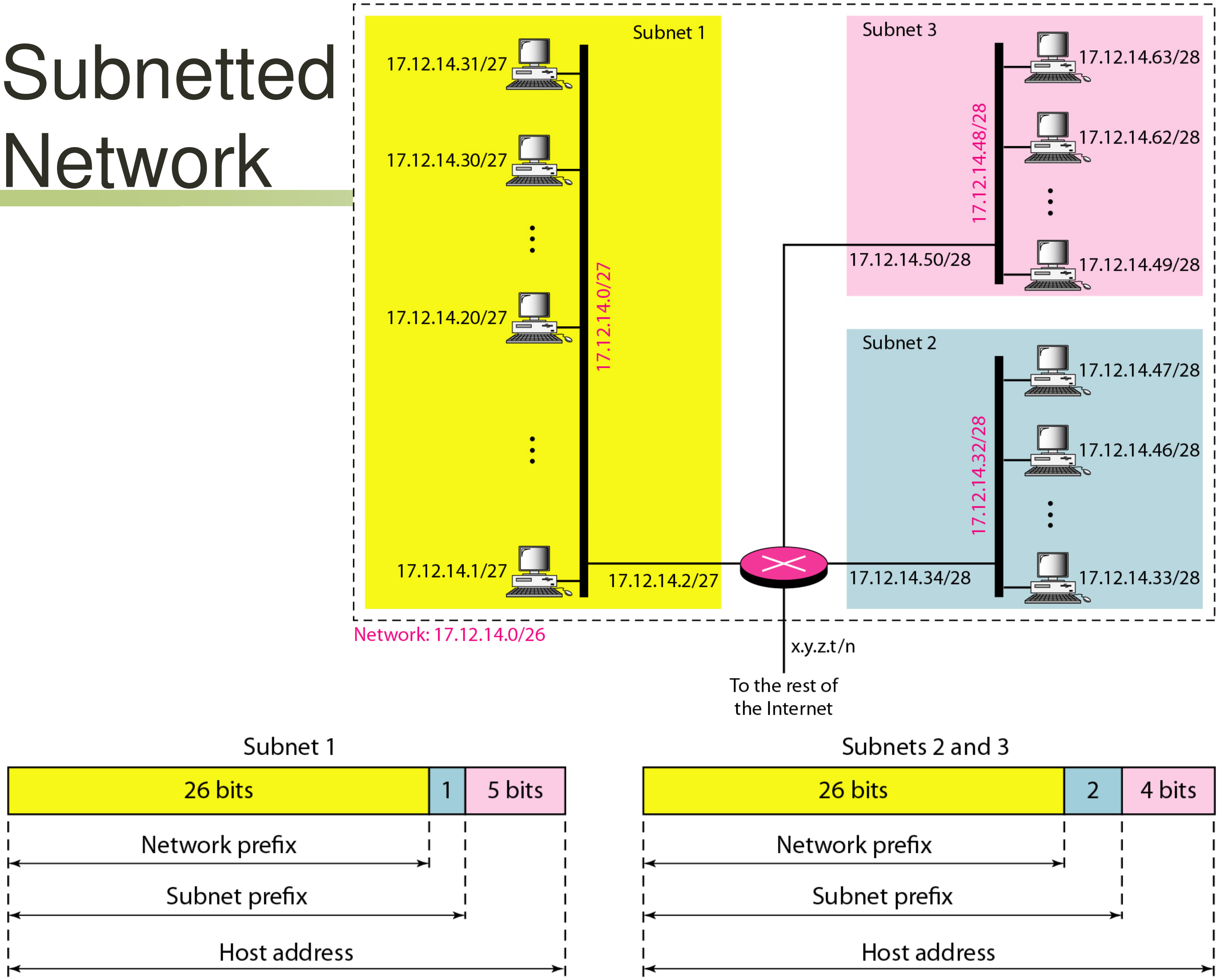
# Classless address blocks

- An entity 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
- Restrictions:
  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.

# Mask in Classless Addressing

- 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 rightmost *32-n* bits to 0s.

# Subnetted Network



# Example 1

*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 2

- An ISP is granted a block of addresses starting with 190.100.0.0/16 The ISP needs to distribute these addresses to three groups of customers as follows:

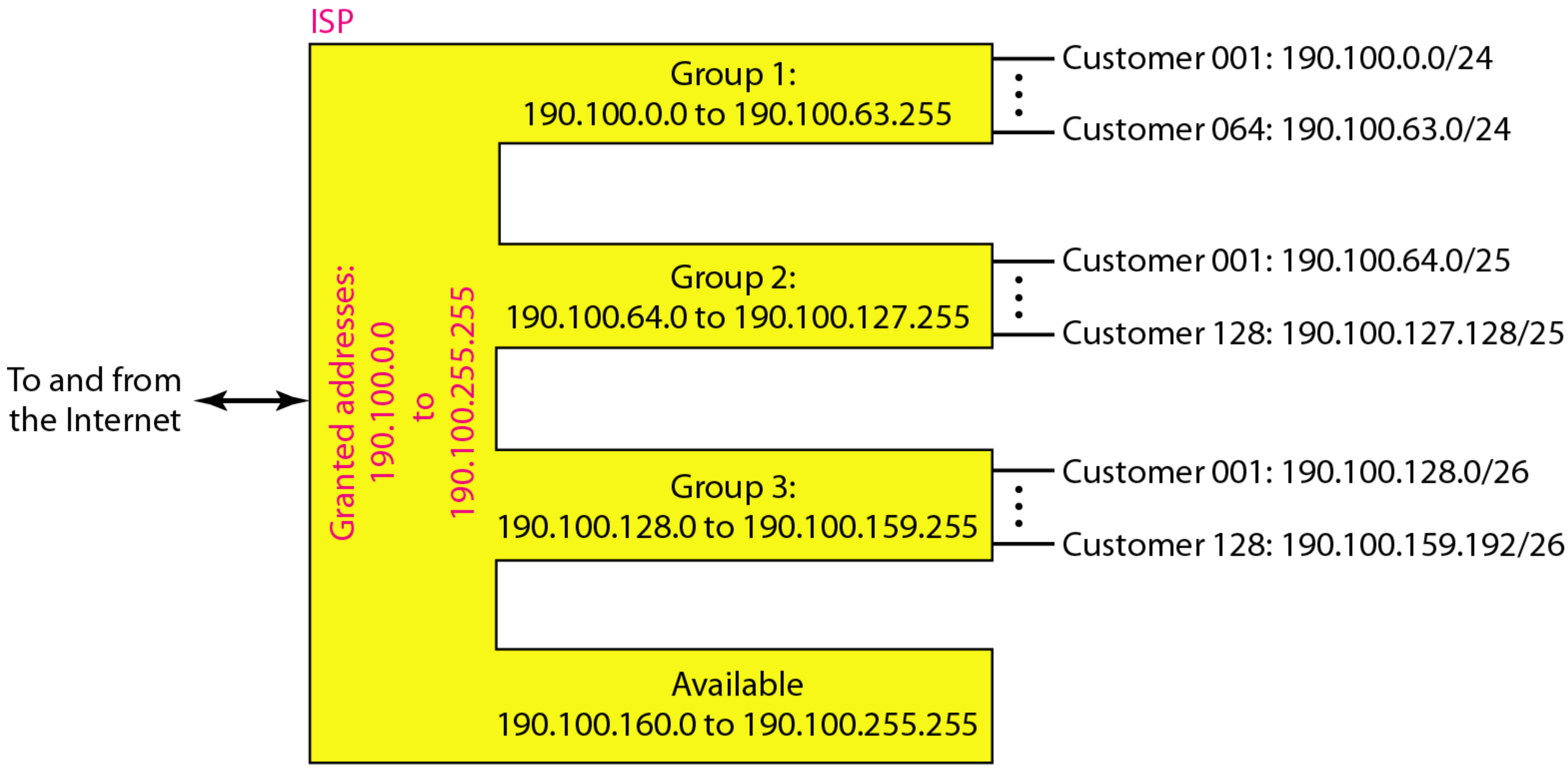
a) 1st group has 64 customers; each needs 256 addresses.

b) 2nd group has 128 customers; each needs 128 addrs.

c) 3rd group has 128 customers; each needs 64 addresses.

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

# Example 2


ISP
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
Granted addresses:
190.100.0.0
to
190.100.255.255
To and from
the Internet

# Network Address Translation

- IP addresses are scarce.
- The long-term solution is for the whole Internet to migrate to IPv6
  - 128-bit addresses
- The quick fix is NAT (Network Address Translation)
  - described in RFC 3022

# NAT – basic idea

- Assign each company a single IP address (or at most, a small number of them) for Internet traffic.
- Within the company, every computer gets a unique IP address (private)
- When a packet exits the company and goes to the Internet, an address translation takes place.
  - (private -> global)

# Private addresses

| Class | RFC 1918 Internal Address Range | CIDR Prefix |
|---|---|---|
| A | 10.0.0.0 - 10.255.255.255 | 10.0.0.0/8 |
| B | 172.16.0.0 - 172.31.255.255 | 172.16.0.0/12 |
| C | 192.168.0.0 - 192.168.255.255 | 192.168.0.0/16 |

- 172.16.0.0 – 172.31.255.255: 172.16.0.0/12
  - Where does the /12 come from?

**12 bits in common**

10101100 . 00010000 . 00000000 . 00000000 – 172.16.0.0

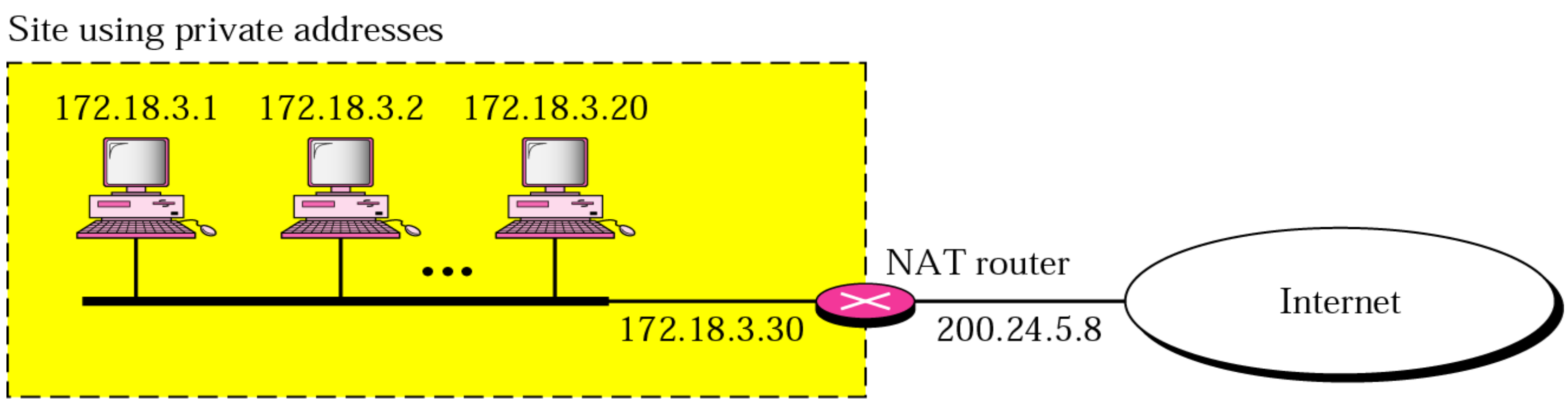
10101100 . 00011111 . 11111111 . 11111111 – 172.31.255.255

------------------------------------------------------------

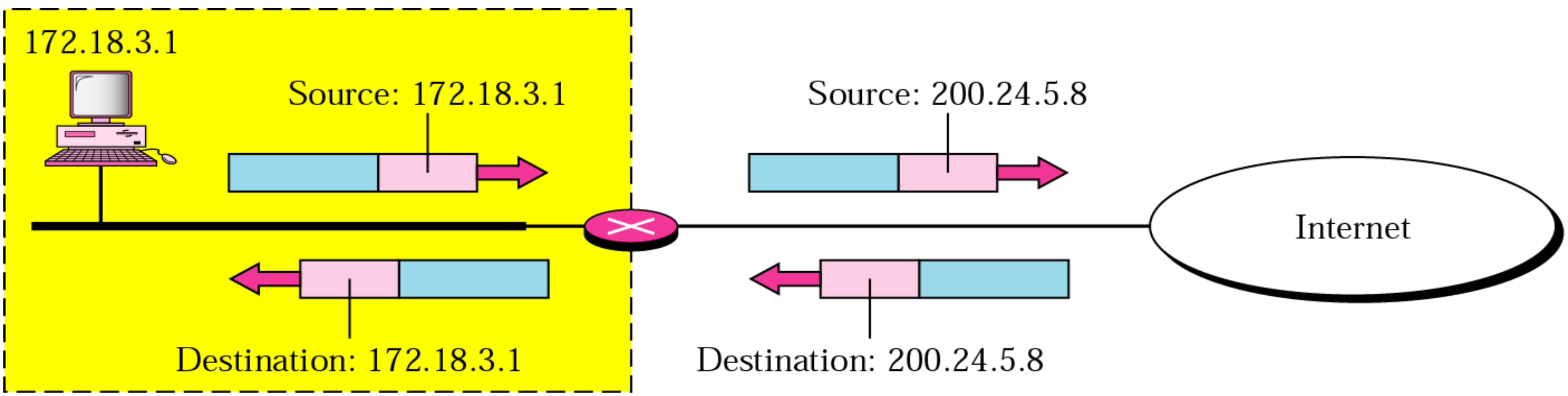
10101100 . 00010000 . 00000000 . 00000000 – 172.16.0.0/12

# NAT

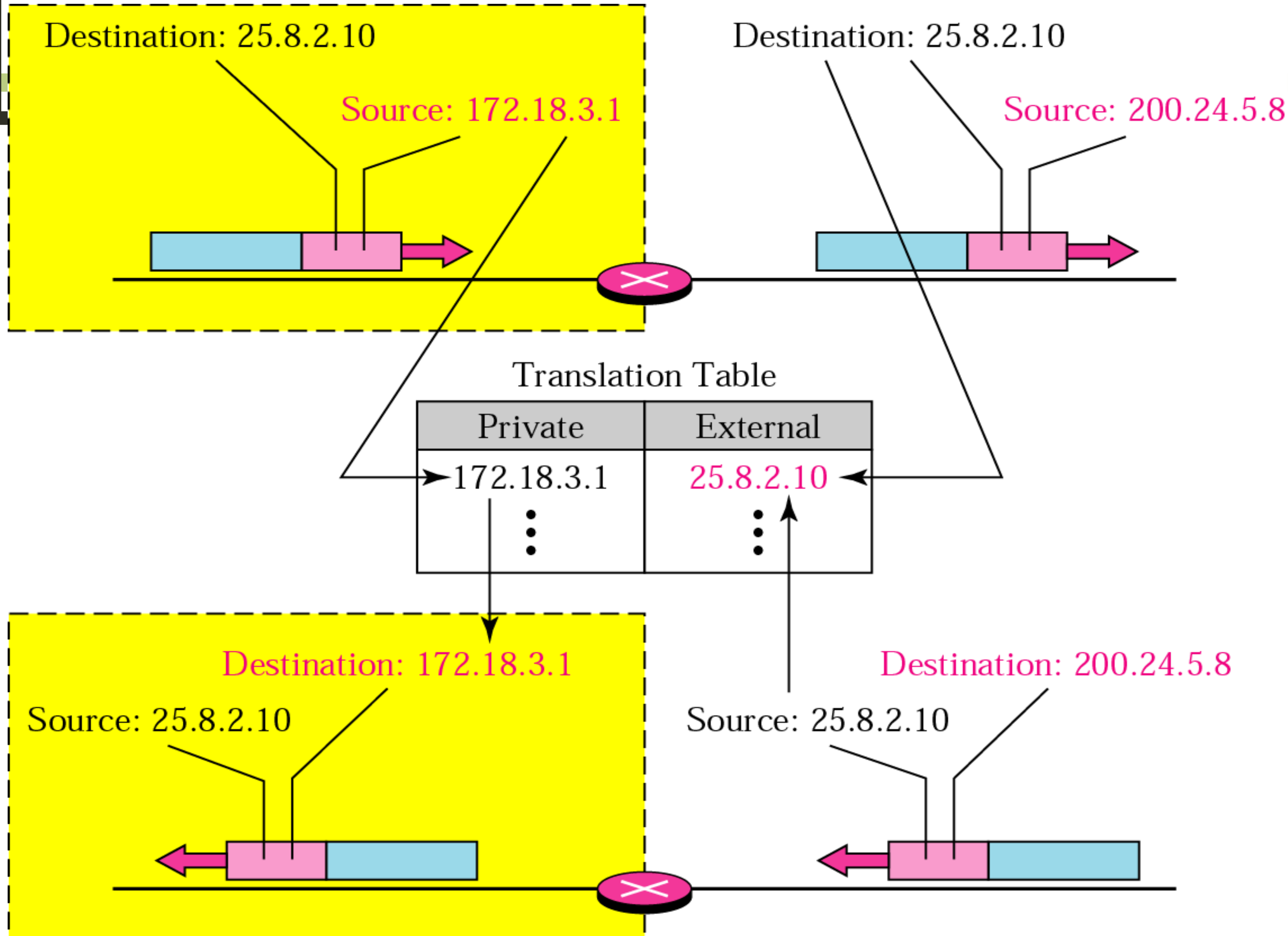
- Private network uses private addresses
- NAT router translates private addresses to global address
- The private network is transparent to the rest of the Internet



# Address Translation



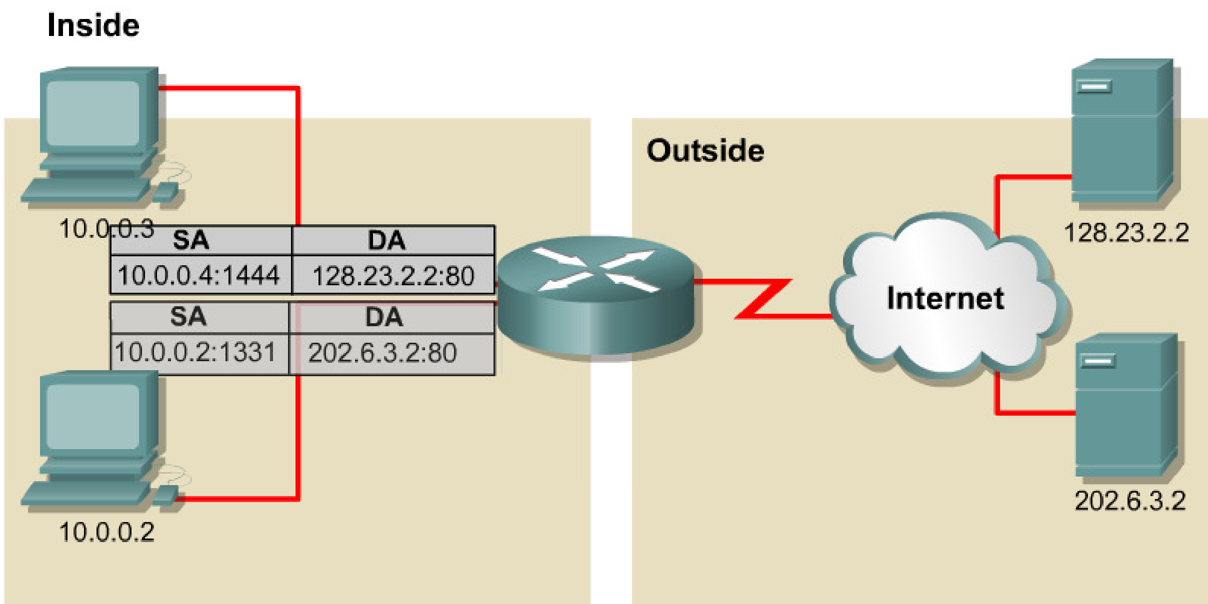
# Translation table



# NAT

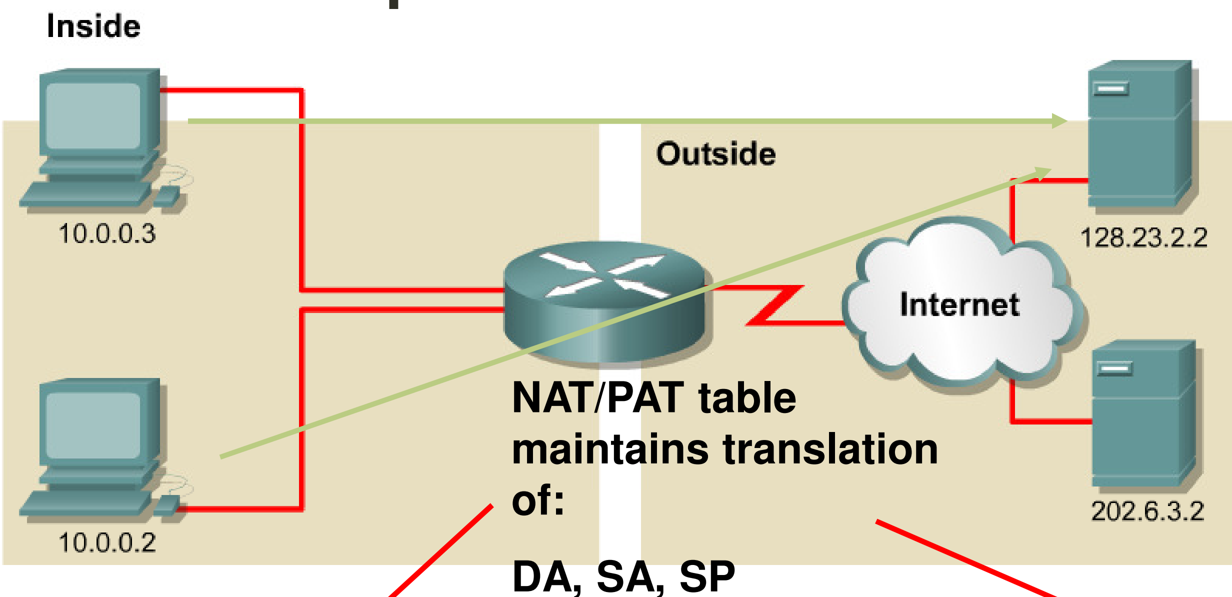
- Using one IP address
  - 1 private host can access an external host at a time
- Using a pool of IP addresses
  - N global addresses = N private hosts can access same external host at a time
- Using both IP addresses and port numbers
  - many private hosts can access many external host at a time

# PAT – Port Address Translation




- PAT (Port Address Translation) allows you to use a single Public IP address and assign it up to 65,536 inside hosts (4,000 is more realistic).
- PAT translates and records the TCP/UDP source port address to track inside Host addresses.
- Also known as Network Address and Port Translation (NAPT)

# PAT Example



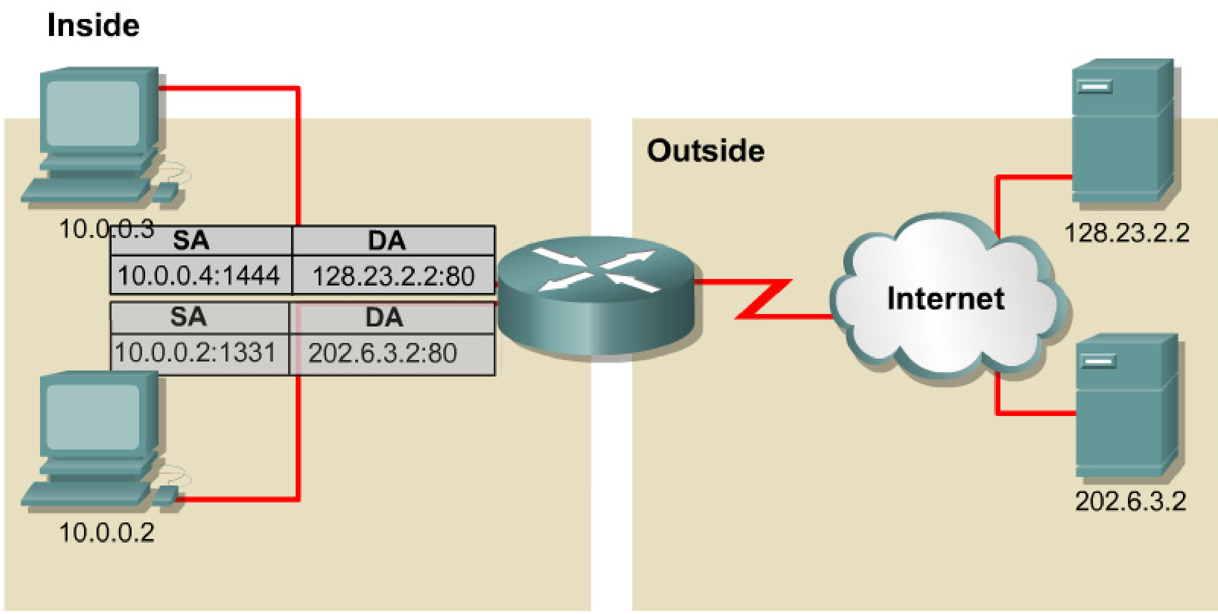

| | DA | SA | DP | SP | |
|---|---|---|---|---|---|
| 1 | 128.23.2.2 | 10.0.0.3 | 80 | 1331 | Data |
| | | IP Header | TCP/UDP Header | | |

| | DA | SA | DP | SP | |
|---|---|---|---|---|---|
| 2 | 128.23.2.2 | 179.9.8.80 | 80 | 3333 | Data |
| | | IP Header | TCP/UDP Header | | |

| | DA | SA | DP | SP | |
|---|---|---|---|---|---|
| 1 | 128.23.2.2 | 10.0.0.2 | 80 | 1555 | Data |
| | | IP Header | TCP/UDP Header | | |

| | DA | SA | DP | SP | |
|---|---|---|---|---|---|
| 2 | 128.23.2.2 | 179.9.8.80 | 80 | 2222 | Data |
| | | IP Header | TCP/UDP Header | | |

# PAT – Port Address Translation




- With PAT multiple private IP addresses can be translated by a single public address (many-to-one translation).
- This solves the limitation of NAT which is one-to-one translation.

# IPv6 Addresses

# IPv6 Addresses

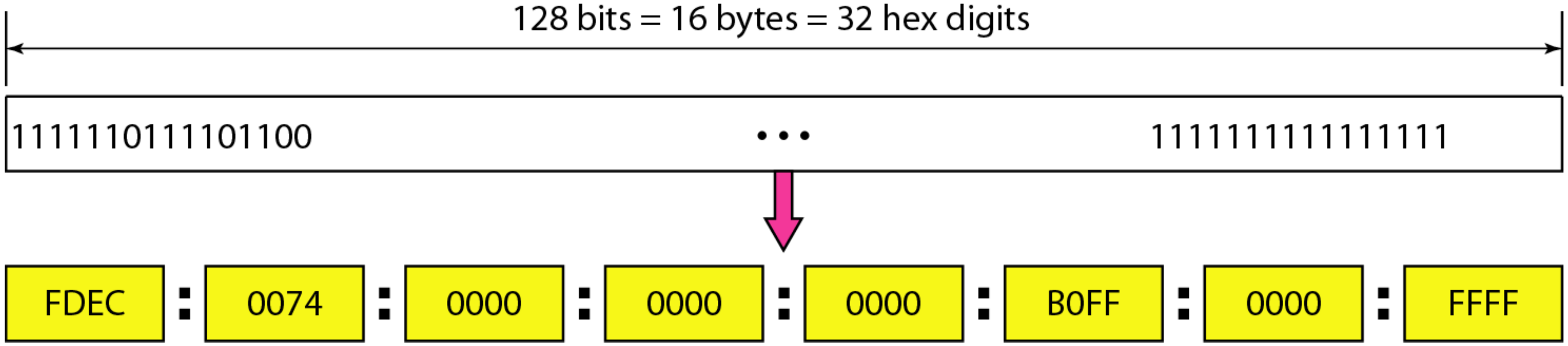
- Despite all short-term solutions (e.g. NAT/PAT), address depletion is still a problem for the Internet.
- This and other problems in the IP protocol itself have been the motivation for designing IPv6.

# IPv6 Addresses

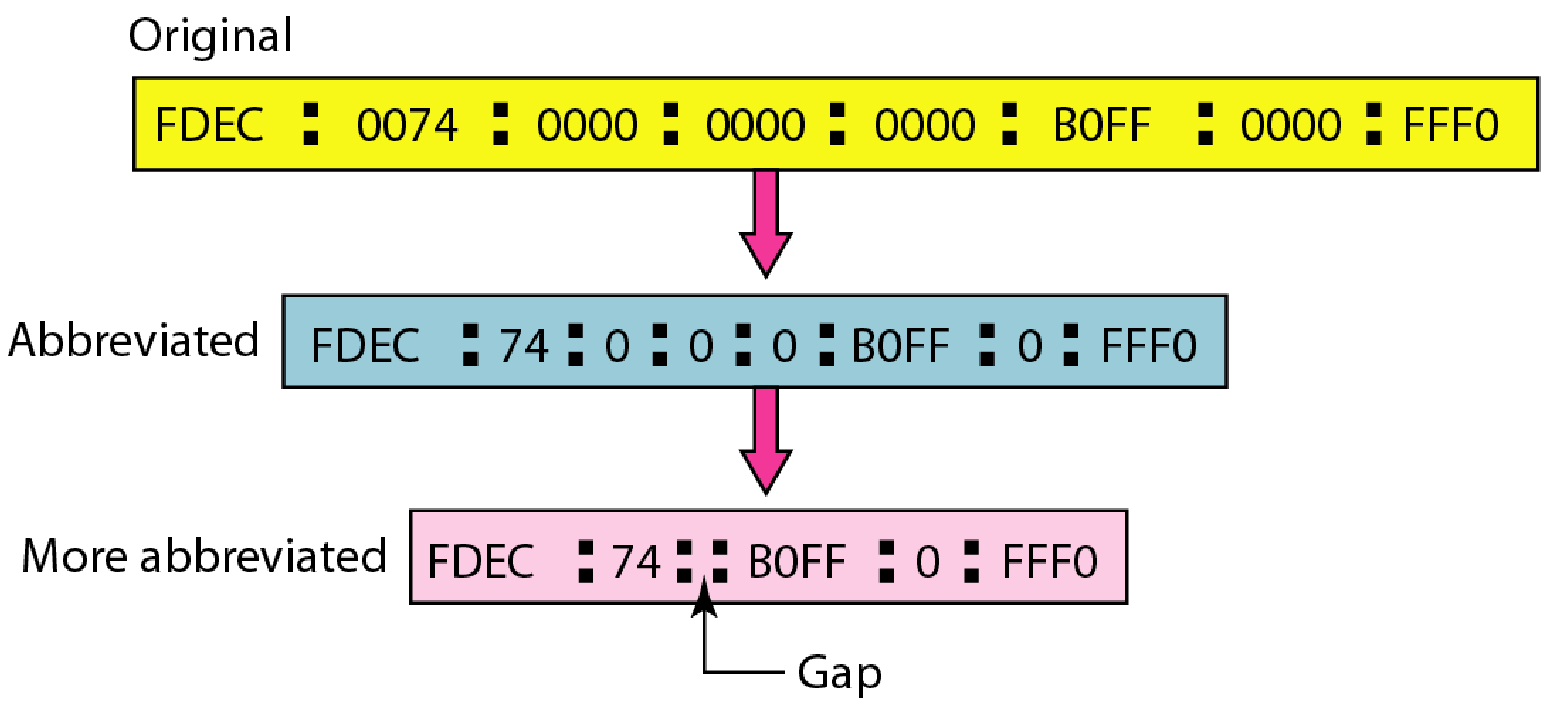
- 128 bits long
- assigned to interface
- single interface may have multiple unicast addresses
- three types of addresses:
  - unicast - single interface address
  - anycast - one of a set of interface addresses
  - multicast - all of a set of interfaces

# IPv6 address structure

- An IPv6 address is 128 bits long
- It is usually presented in hexadecimal colon notation



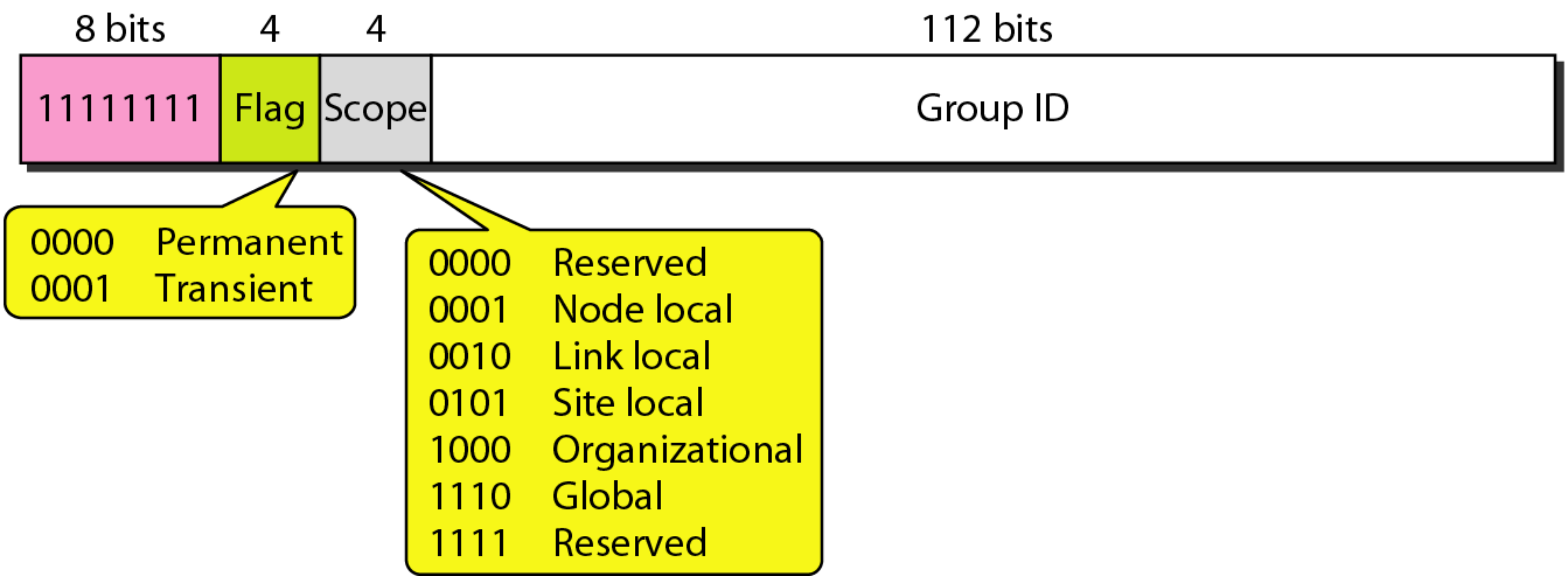
# Abbreviated IPv6 addresses



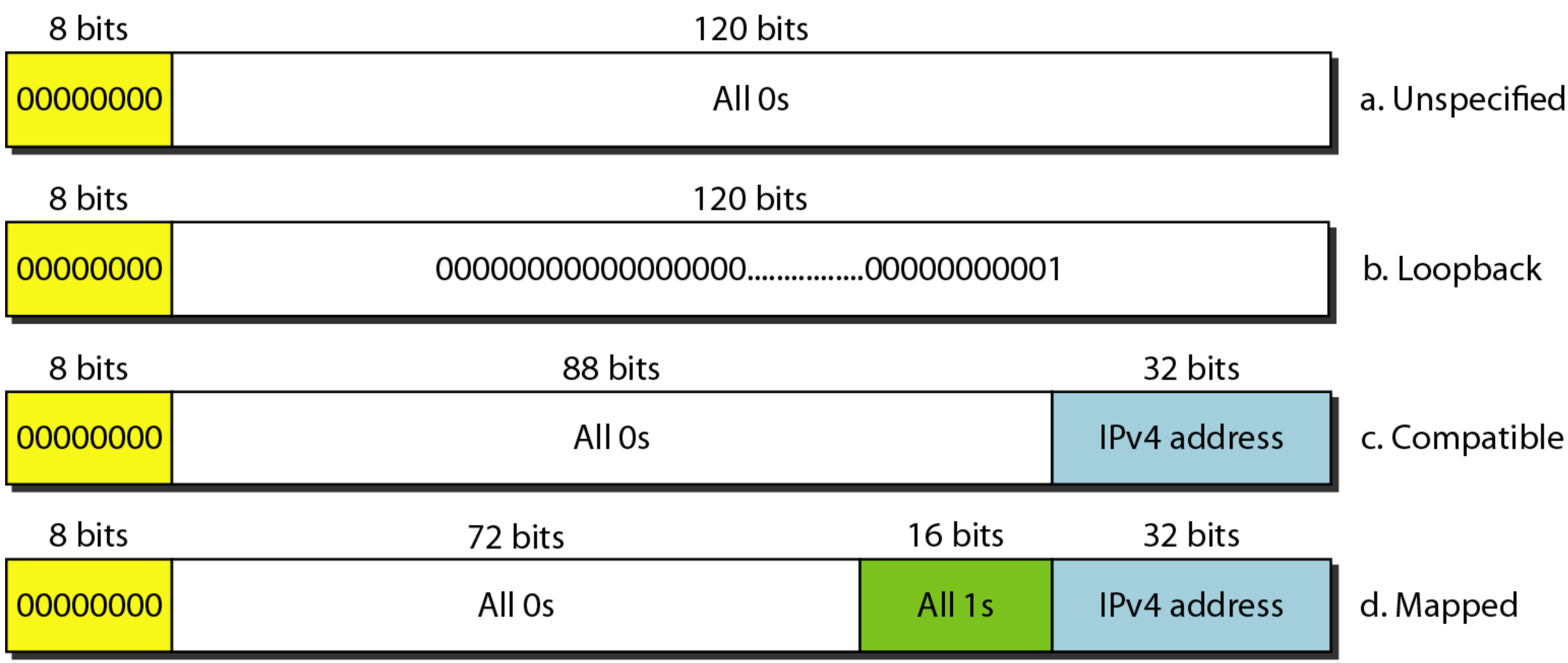
# Type prefixes for IPv6 addresses

| *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 |

# Multicast address in IPv6



# Reserved addresses in IPv6



# Local addresses in IPv6