

Lecture 1 Overview and Network Models

Textbook: Ch.1 and Ch.2

Main Topics

Chapter 1

- ❖ Computer Networking
- ❖ Connection and Transmission Mode
- ❖ Topology
- ❖ Categories of Networks and Internetworks

Chapter 2

- ❖ Protocols, Standards and Standard Organizations
- ❖ OSI Model: Open System Interconnection by ISO
- ❖ TCP/IP Protocol Suite

1. Computer Networking

- ❖ Computer Networking facilitates data communication among computing devices
 - ❖ **Data communications** :The exchange of data between two devices via some form of transmission medium.
- ❖ Communication effectiveness depends on
 - ❖ Delivery (to the correct destination)
 - ❖ Accuracy
 - ❖ Timeliness
 - ❖ Jitter

Networks

- ❖ A network is a set of devices (called *nodes*) connected by media *links* (called communication channels).
- ❖ What is a *good* network?

Performance

- ❖ A number of measurements, e.g.
 - ❖ Propagation/Transmission time
 - ❖ Response time
- ❖ Performance is often evaluated by two networking metrics:
 - ❖ Throughput
 - ❖ Delay

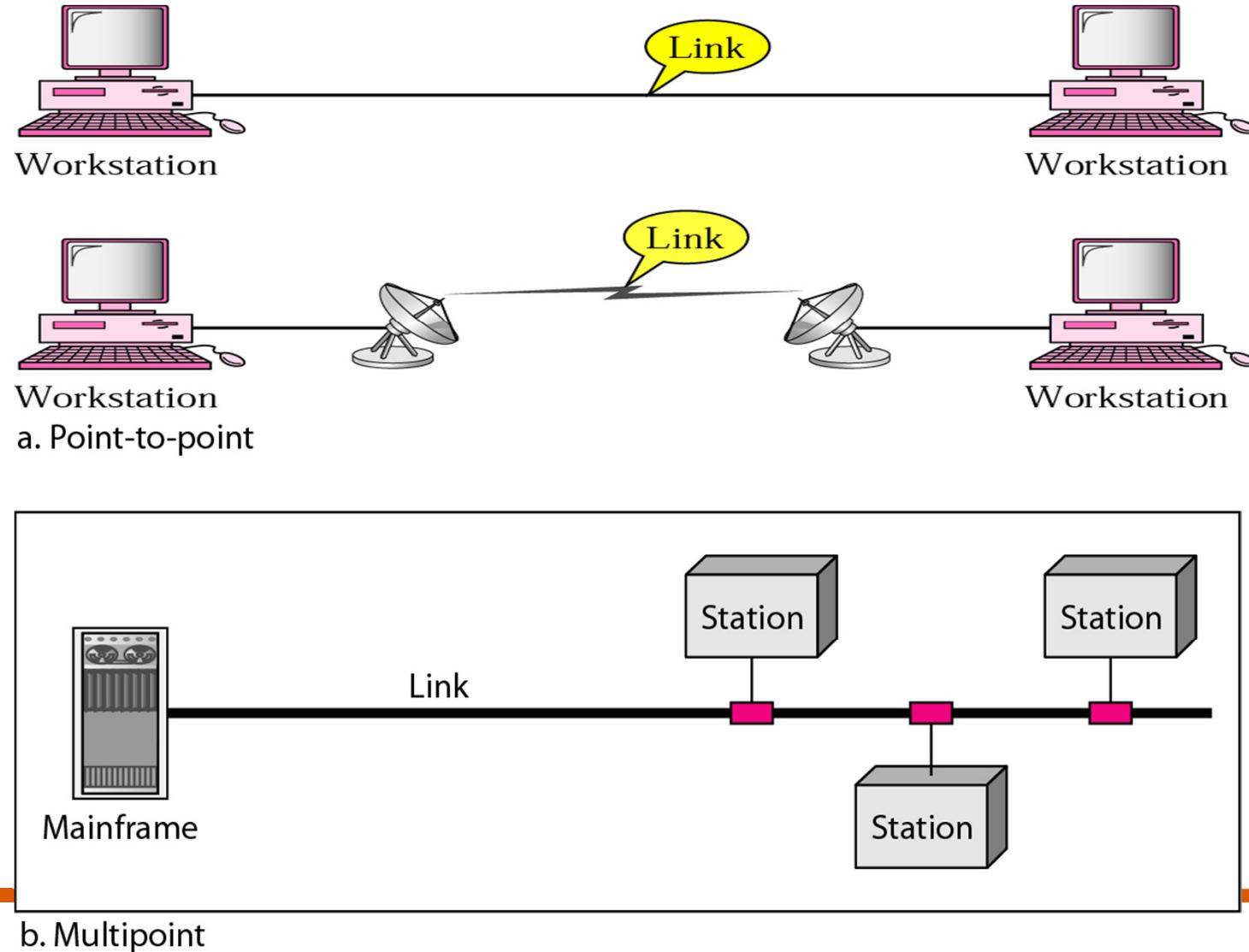
Reliability

- ❖ Frequency of failure
- ❖ Recovery time of a network after a failure
 - ❖ Catastrophe
 - ❖ Fire , earthquake, etc.
 - ❖ Backup system
 - ❖ Contingency plan
- ❖ Resistant to:
 - ❖ Unauthorized access
 - ❖ Data damage
 - ❖ Viruses

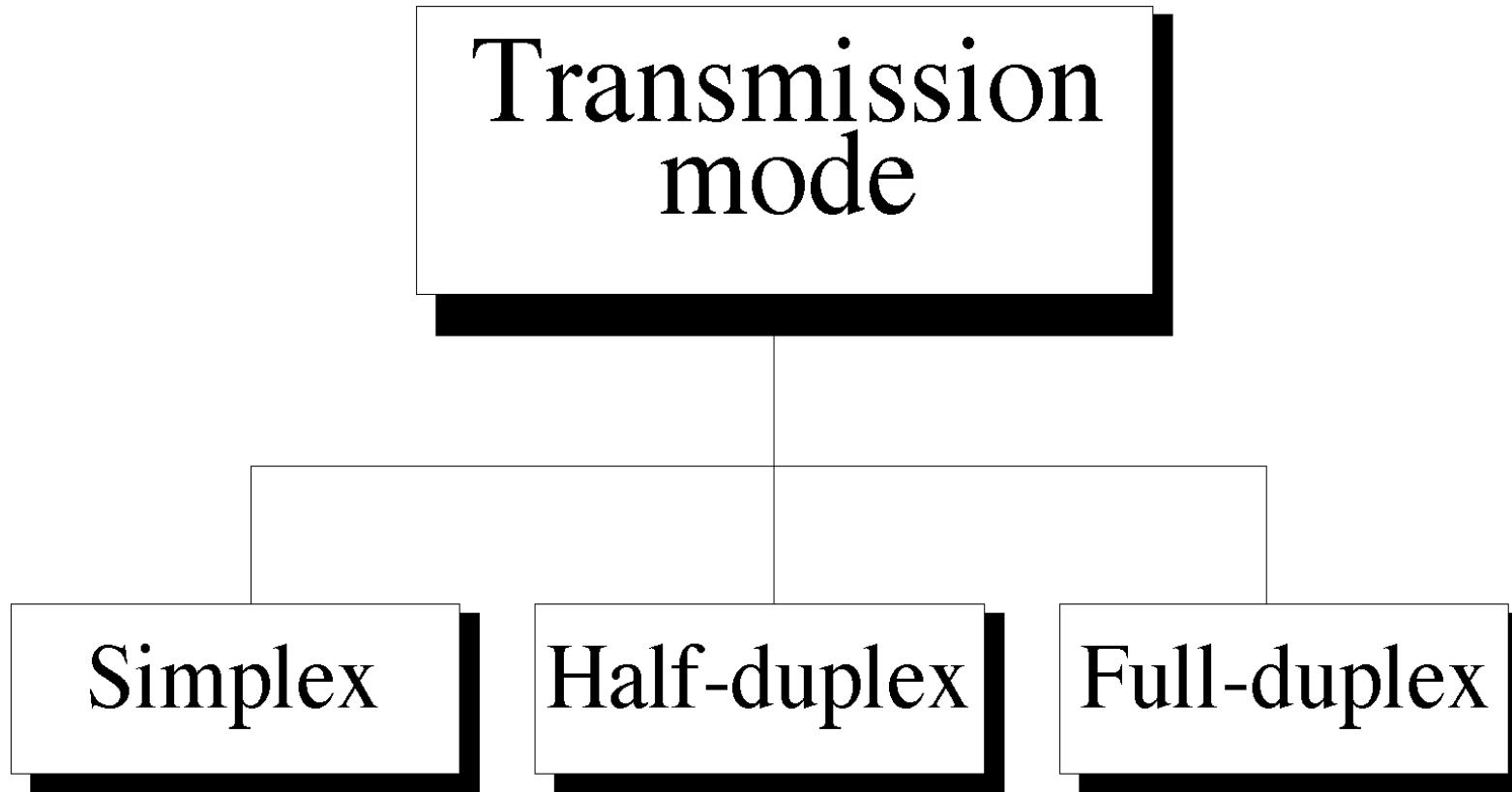
2. Types of Connections

- ❖ Defines the attachment of communication devices to a link
- ❖ Two Types:
 - ❖ **Point-to-point**: a dedicated link between two devices
 - ❖ **Multipoint** (multidrop): more than two devices share a single link

Figure 1.3 *Types of connections: point-to-point and multipoint*

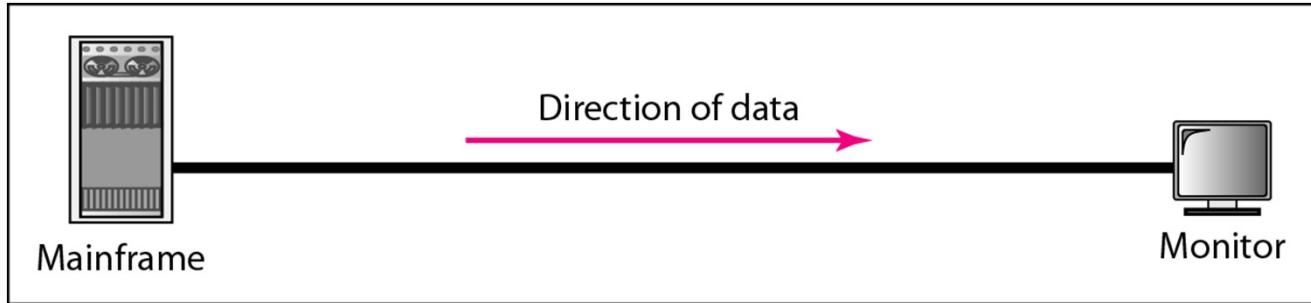


3. Transmission Mode

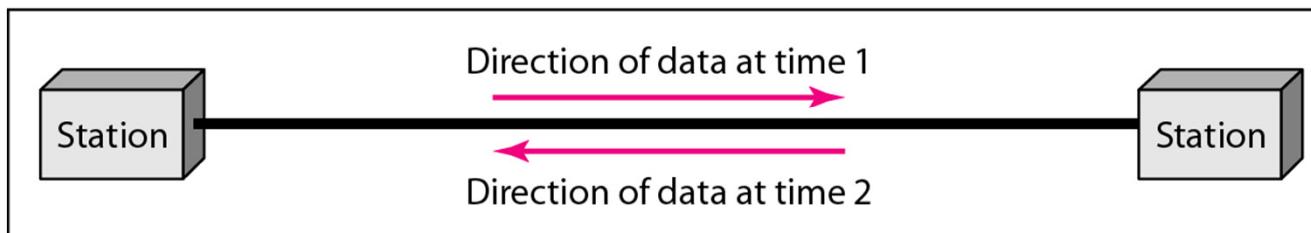


Refers to the direction of information flow

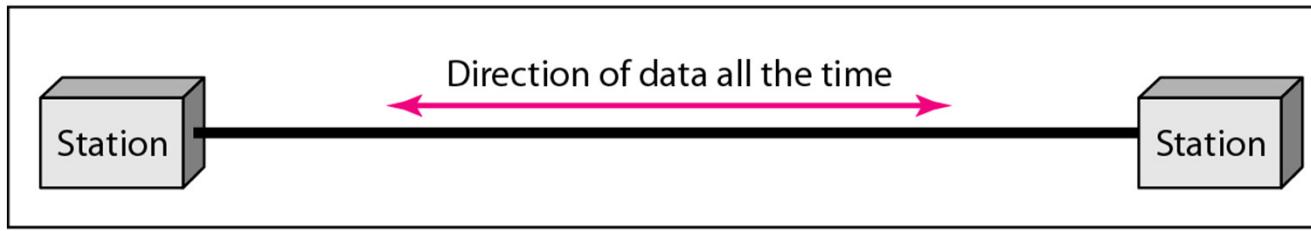
Data flow (simplex, half-duplex, and full-duplex)



a. Simplex - Communication is unidirectional.



b. Half-duplex - Each station can both transmit and receive, but not at the same time.



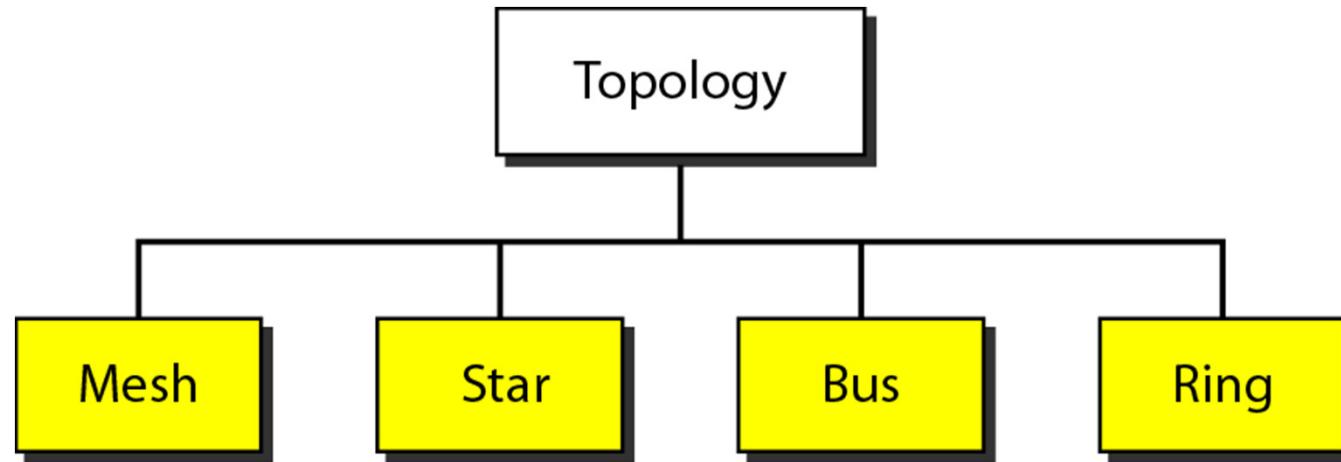
c. Full-duplex - Both stations can transmit and receive simultaneously.

Figure 1.2 *Data flow (simplex, half-duplex, and full-duplex)*

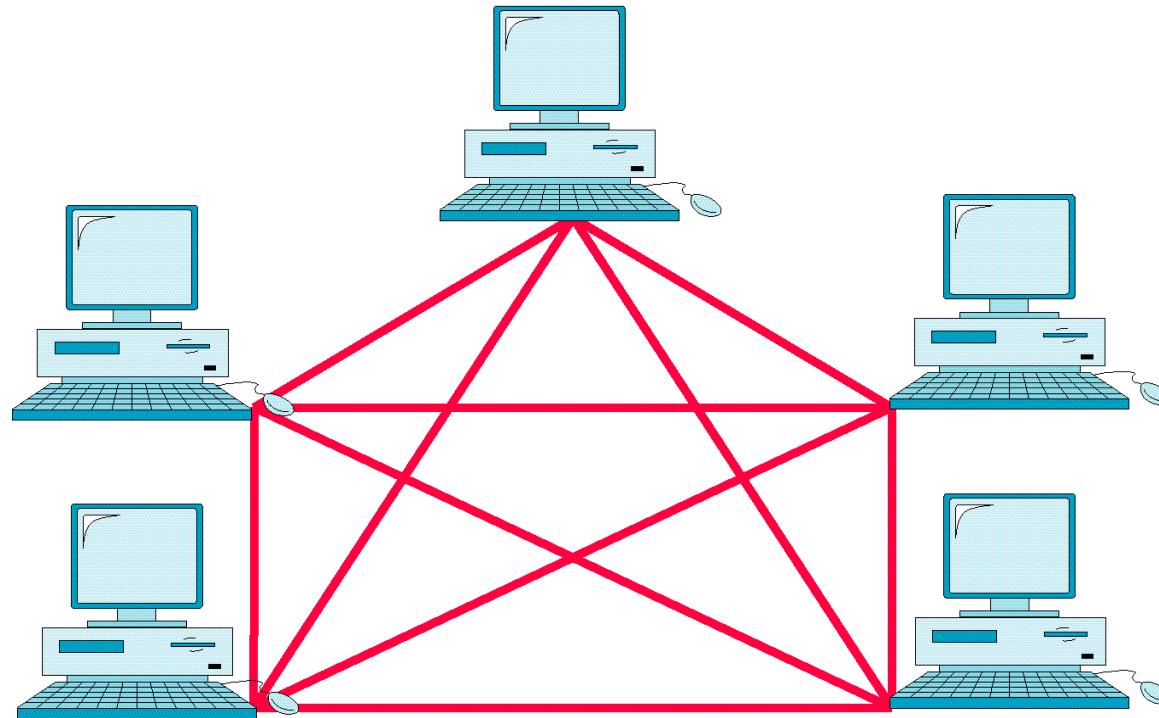
4. Topology

- ❖ Defines the physical or logical ***arrangement of links*** in a network
- ❖ It is the ***geometric representation*** of the relationship of all the links and nodes to each other (simply speaking, the ***shape*** of the network)
- ❖ A consideration when choosing a topology is the relative status of the devices to be linked
- ❖ Relationships: Peer-to peer or Primary-secondary

Figure 1.4 *Categories of topology*



Mesh Topology



- ❖ A *fully connected mesh topology (five devices)*
- ❖ *How many links are needed?*

Mesh Topology

- ❖ Every device has a dedicated point-to-point link to every other device
- ❖ A fully connected mesh network has $n(n-1)/2$ physical channels to link n devices
- ❖ Convenient for peer-to-peer transmission
- ❖ What are the advantages of Mesh topology?
What are the costs?

Mesh Topology

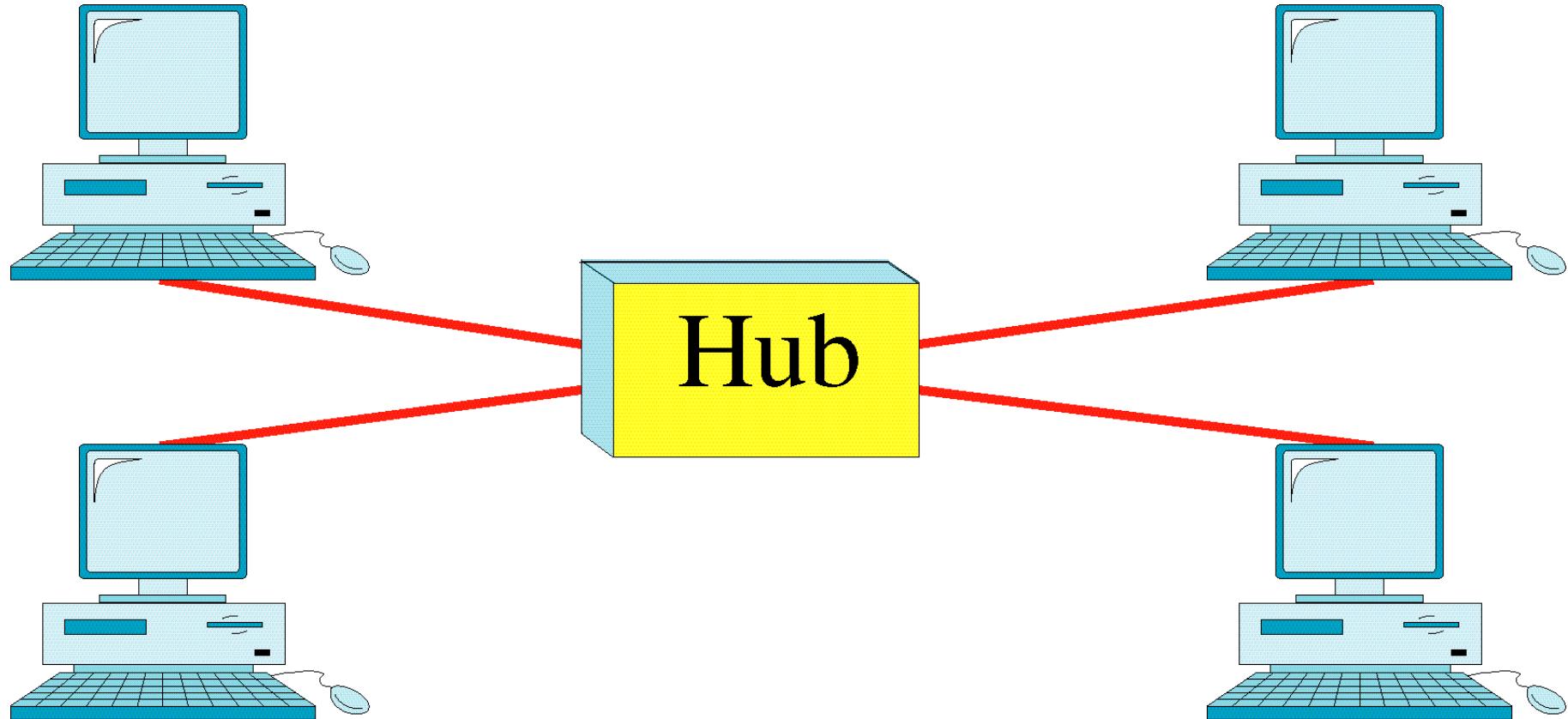
❖ Advantages

- ❖ Dedicated links eliminate the traffic problem
- ❖ Robust: failure of one link does not affect the whole network
- ❖ Privacy/Security provided by dedicated links
- ❖ Easy fault identification and isolation

❖ Disadvantages

- ❖ Expensive
 - ❖ cost of cabling and
 - ❖ the I/O ports

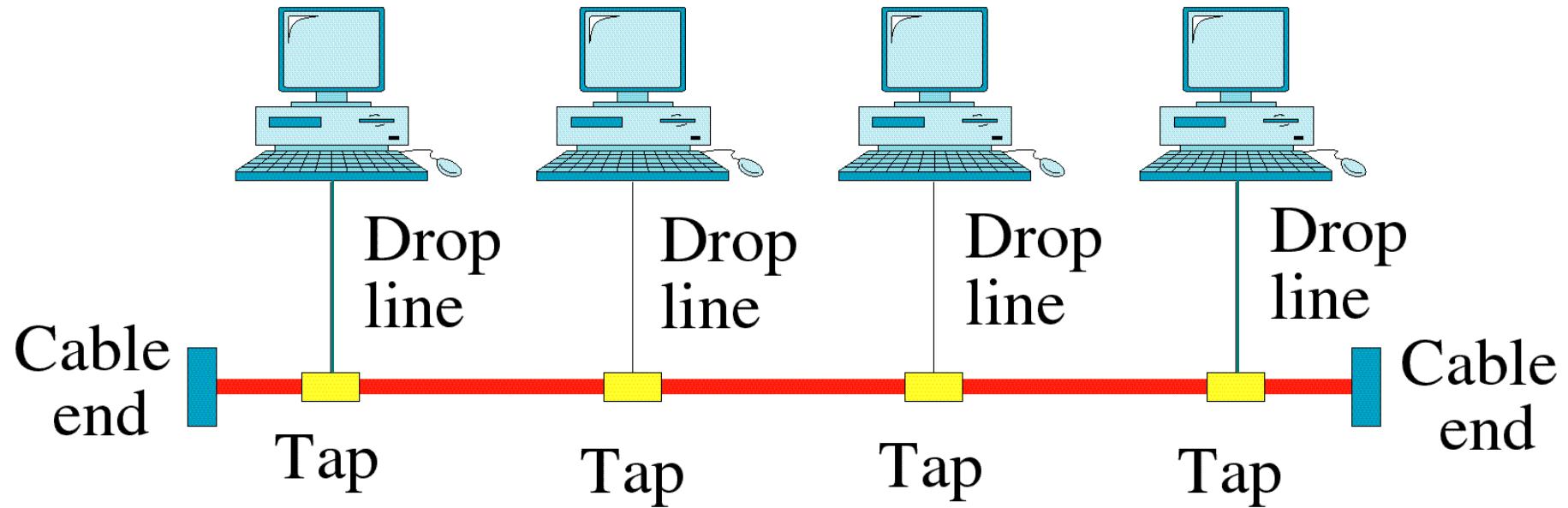
Star Topology



Star Topology

- ❖ Each device has a dedicated link only to a ***central controller*** (called a hub) which acts as an exchange
- ❖ ***No direct traffic*** between devices
- ❖ Advantages:
 - ❖ Less expensive for cabling and I/O ports
 - ❖ Robustness, easy fault identification and isolation
- ❖ Disadvantage:
 - ❖ **Single point of failure** (what if the hub goes down?)

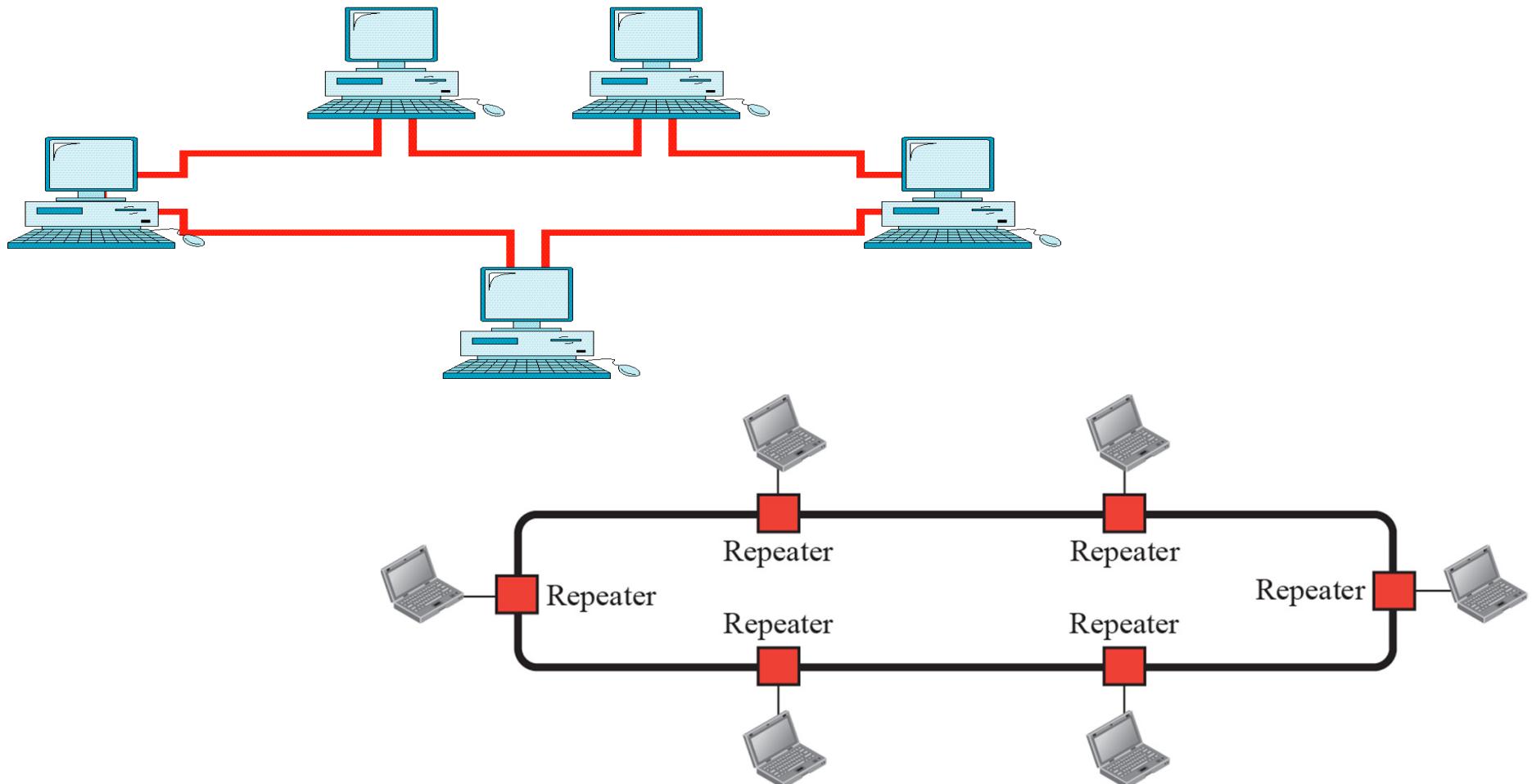
Bus Topology



Bus Topology

- ❖ One long **cable** acts as a backbone to link all the devices
- ❖ A **broadcast** channel
- ❖ Easy installation, least cabling
- ❖ Due to power loss; no. of taps and distance between taps are limited
- ❖ Difficult reconfiguration and fault isolation

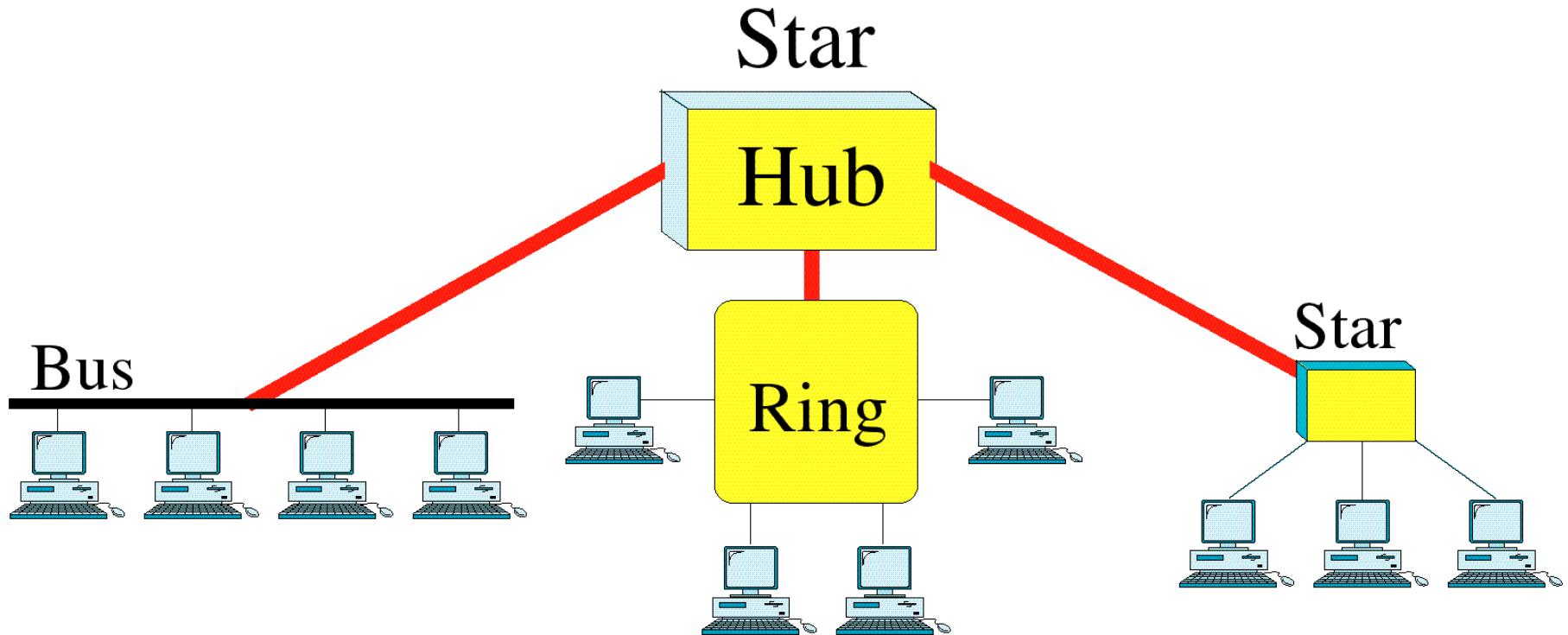
Ring Topology



Ring Topology

- ❖ Each device has a dedicated link **only with the two neighbor devices**
- ❖ A signal is passed along the ring in **one direction** from device to device (which has a repeater)
- ❖ Relatively easy to install and reconfigure
- ❖ Constraints on maximum ring length & no. of devices
- ❖ Unidirectional traffic: a break in the ring can disable the entire network

Hybrid Topology



Hybrid Topology

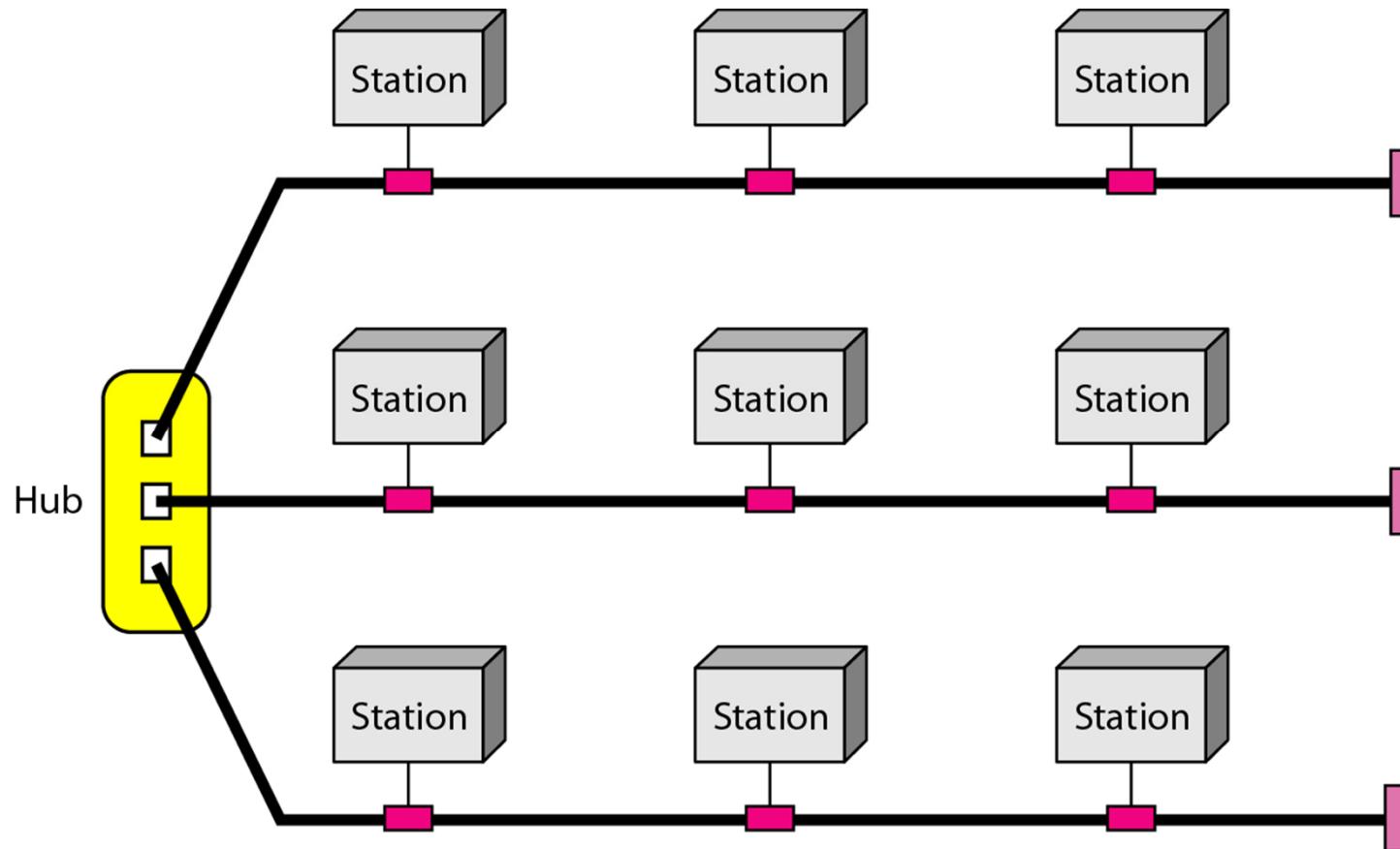
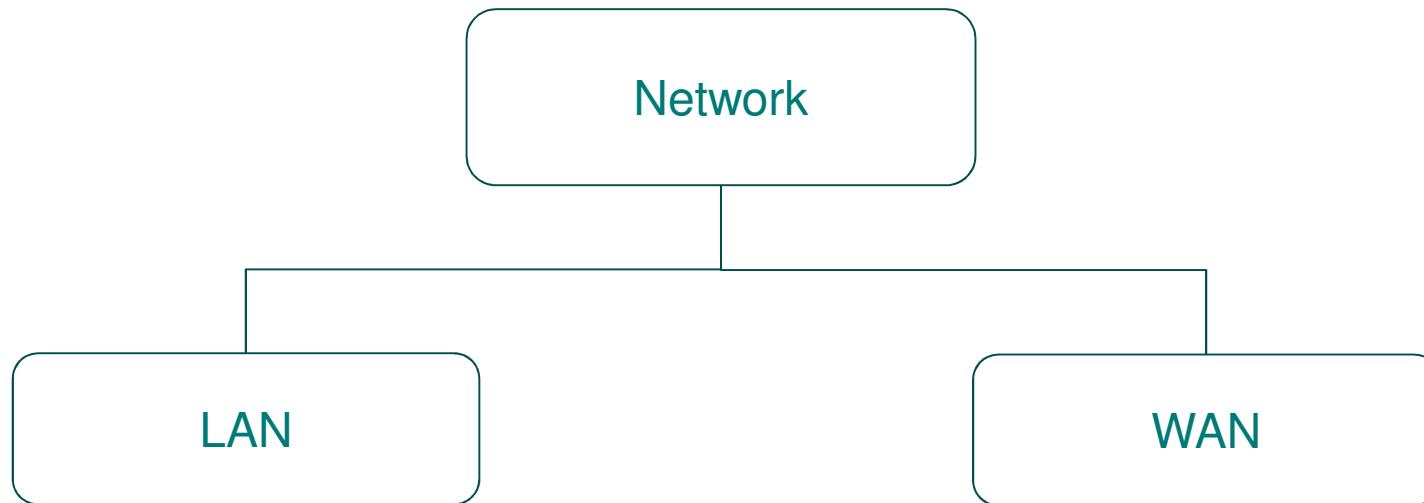


Figure 1.9 A hybrid topology: a star backbone with three bus networks

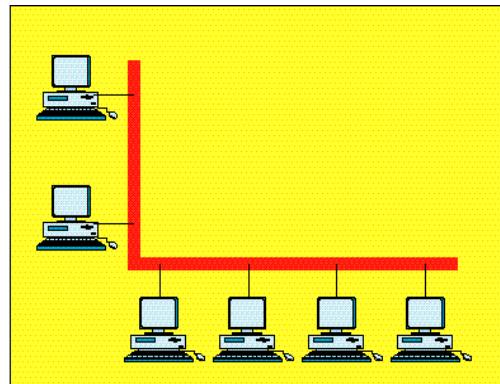
5. Categories of Networks

Classify by its size, ownership, covering distance and physical architecture

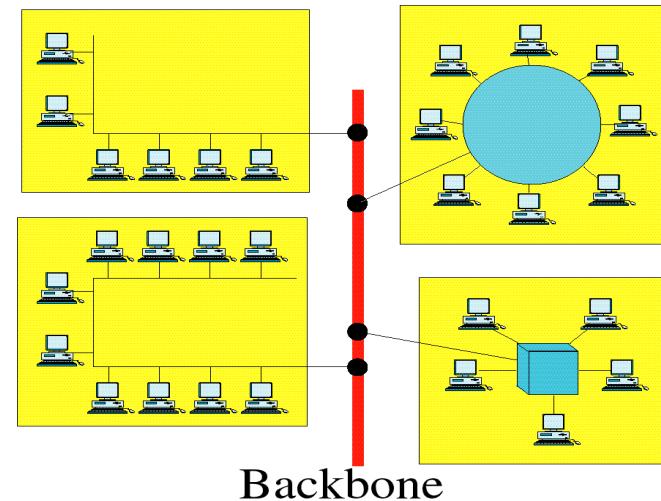


Local Area Network

- ❖ LAN is usually privately owned
- ❖ Connecting hosts in a single office, building, or campus.



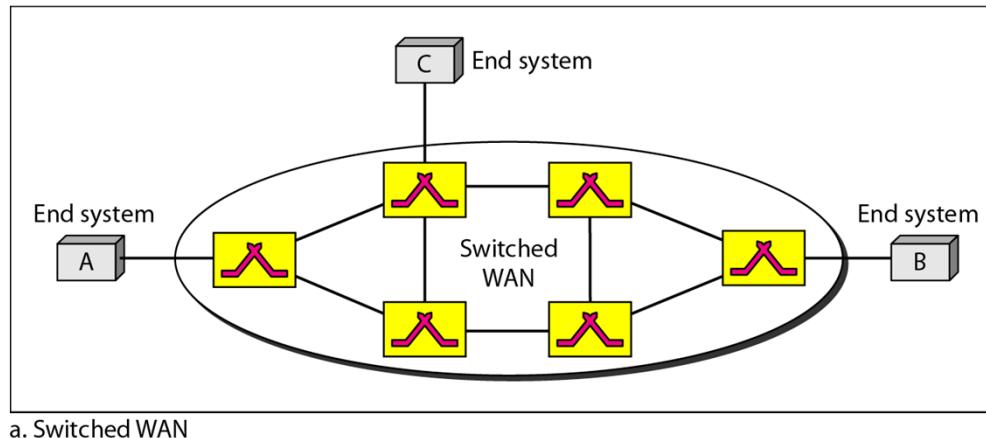
Single building LAN



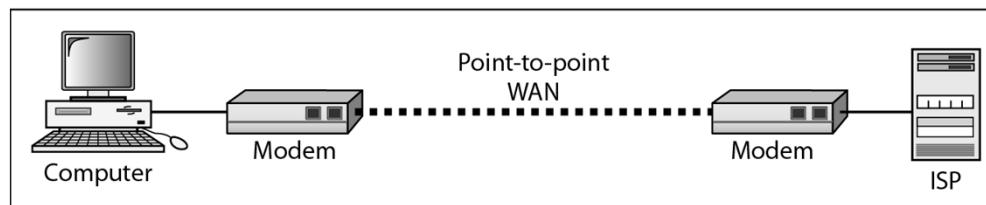
Multiple building LAN

Wide Area Network

- ❖ Connecting devices in a wider geographical area, e.g. town, country, or even the world.



a. Switched WAN



b. Point-to-point WAN

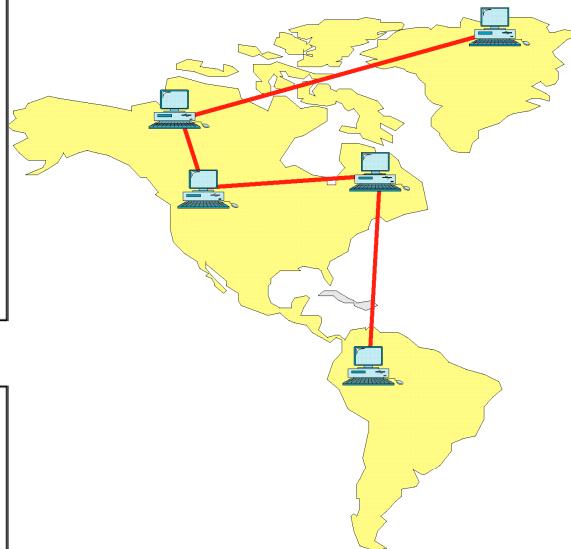
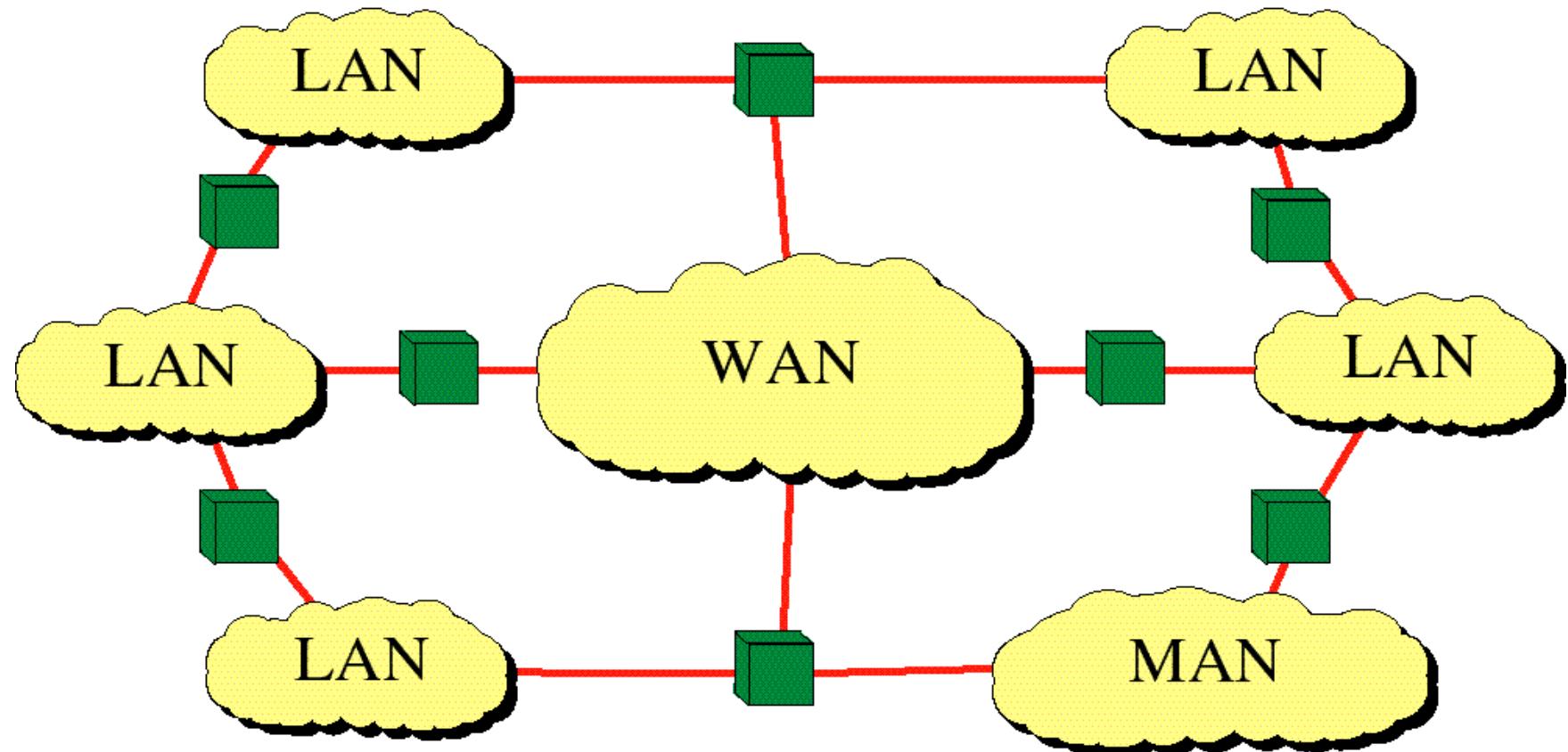


Figure 1.11 *WANs: a switched WAN and a point-to-point WAN*

Internet (Internet) is a network of networks



Example of LAN and WAN

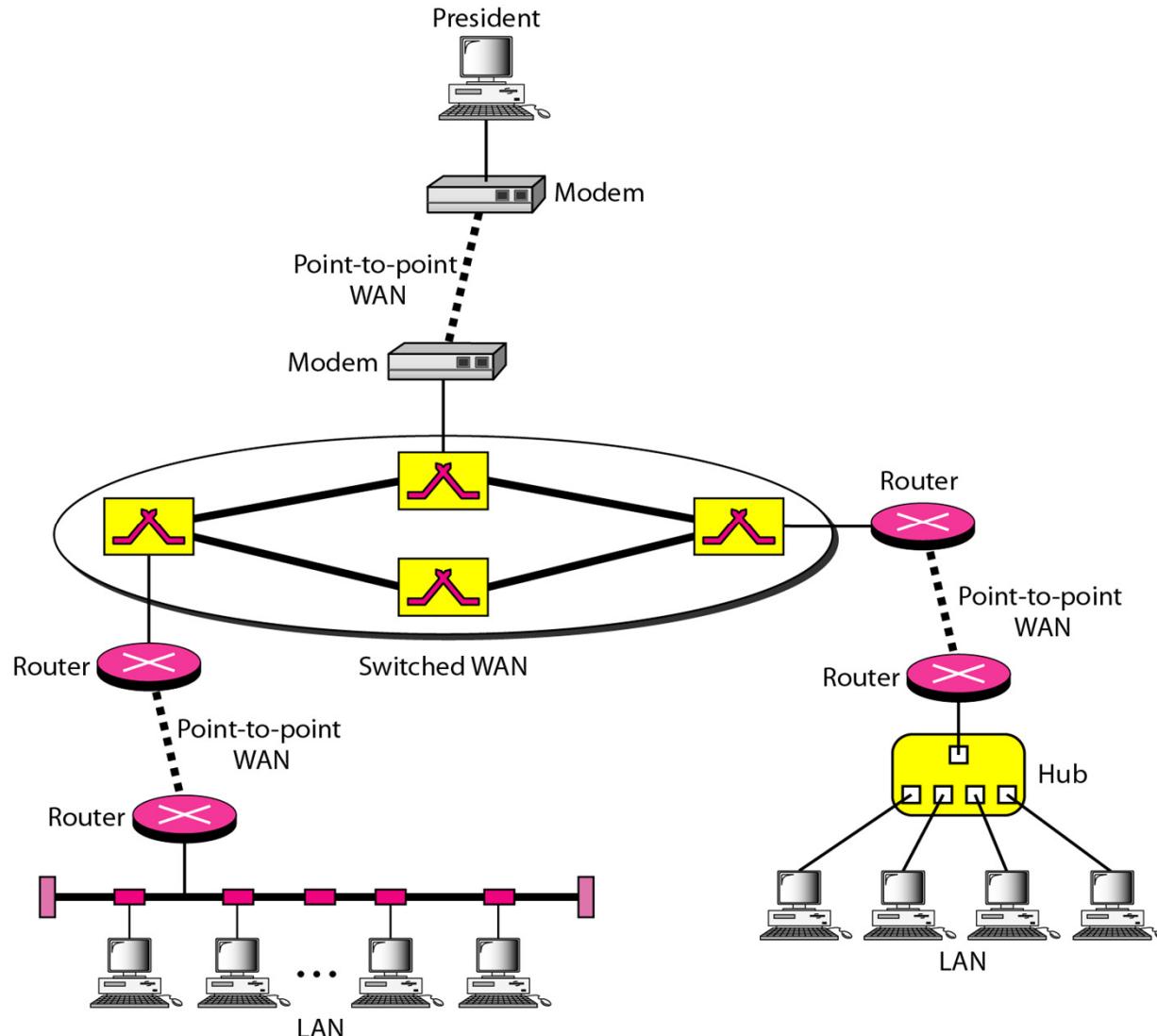


Figure 1.12 A heterogeneous network made of four WANs and two LANs

6. Protocols

- ❖ A ***set of rules*** (conventions) that govern all aspects of information exchange.
- ❖ The key elements:
 - ❖ *Syntax* : Structure or format of the data
 - ❖ *Semantics* : Meaning of different part
 - ❖ *Timing* : When to send and how fast

Standards

- ❖ Provides a model for development that makes it possible for a product to work regardless of the individual manufacturer.
- ❖ Ensures that products from different manufacturers can work together
- ❖ ISO – International Standards Organization
- ❖ ANSI – American National Standards Institute
- ❖ IEEE – Institute of Electrical and Electronics Engineers

Layering in Network Models

- ❖ Data communication systems consists of a lot of rules and procedures for different functions
- ❖ Divide the complex tasks into layers for simpler implementation and maintenance
 - ❖ Each layer only focuses on its own task
 - ❖ Protocols are designed for specific layers

Consider the scenario

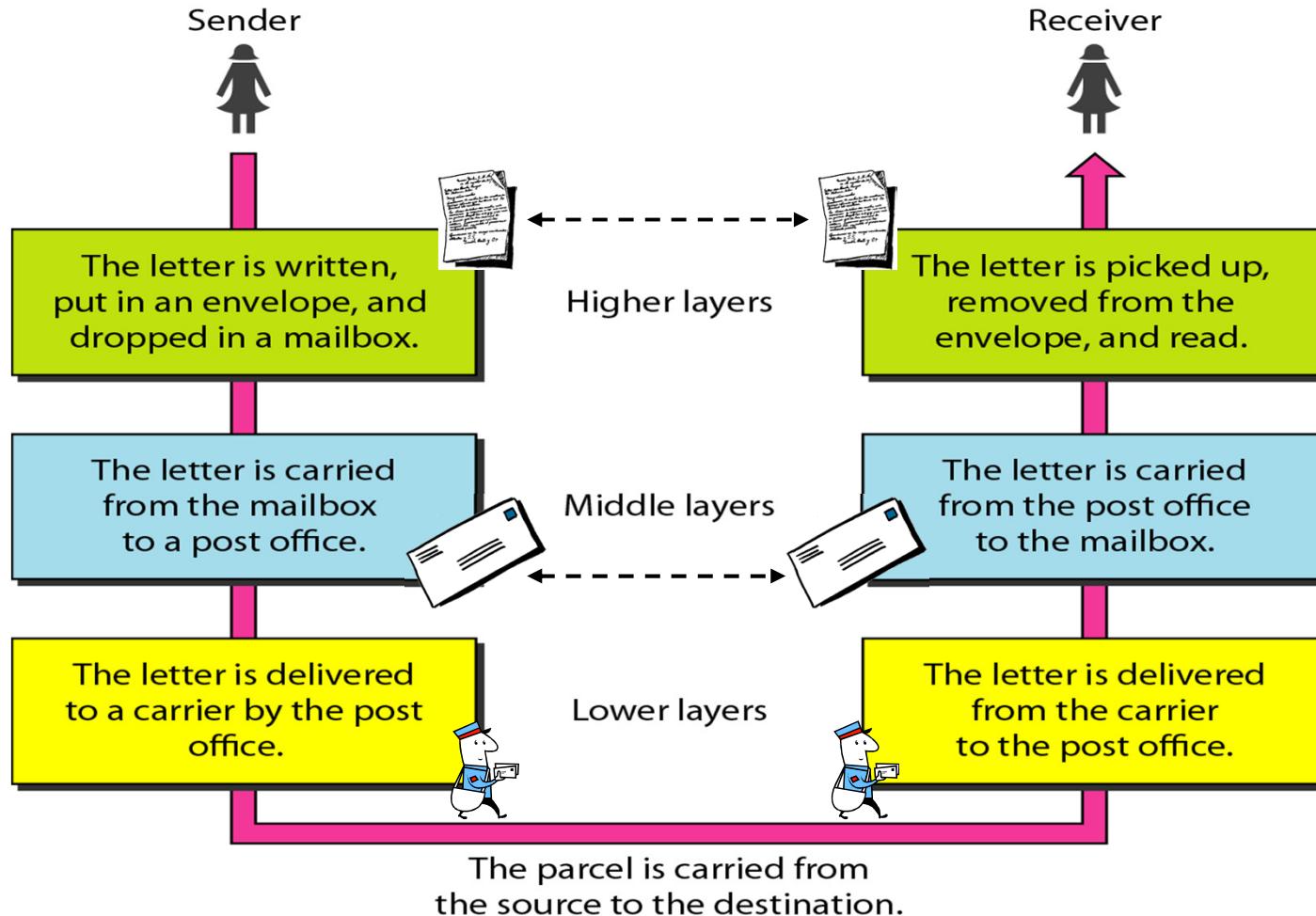
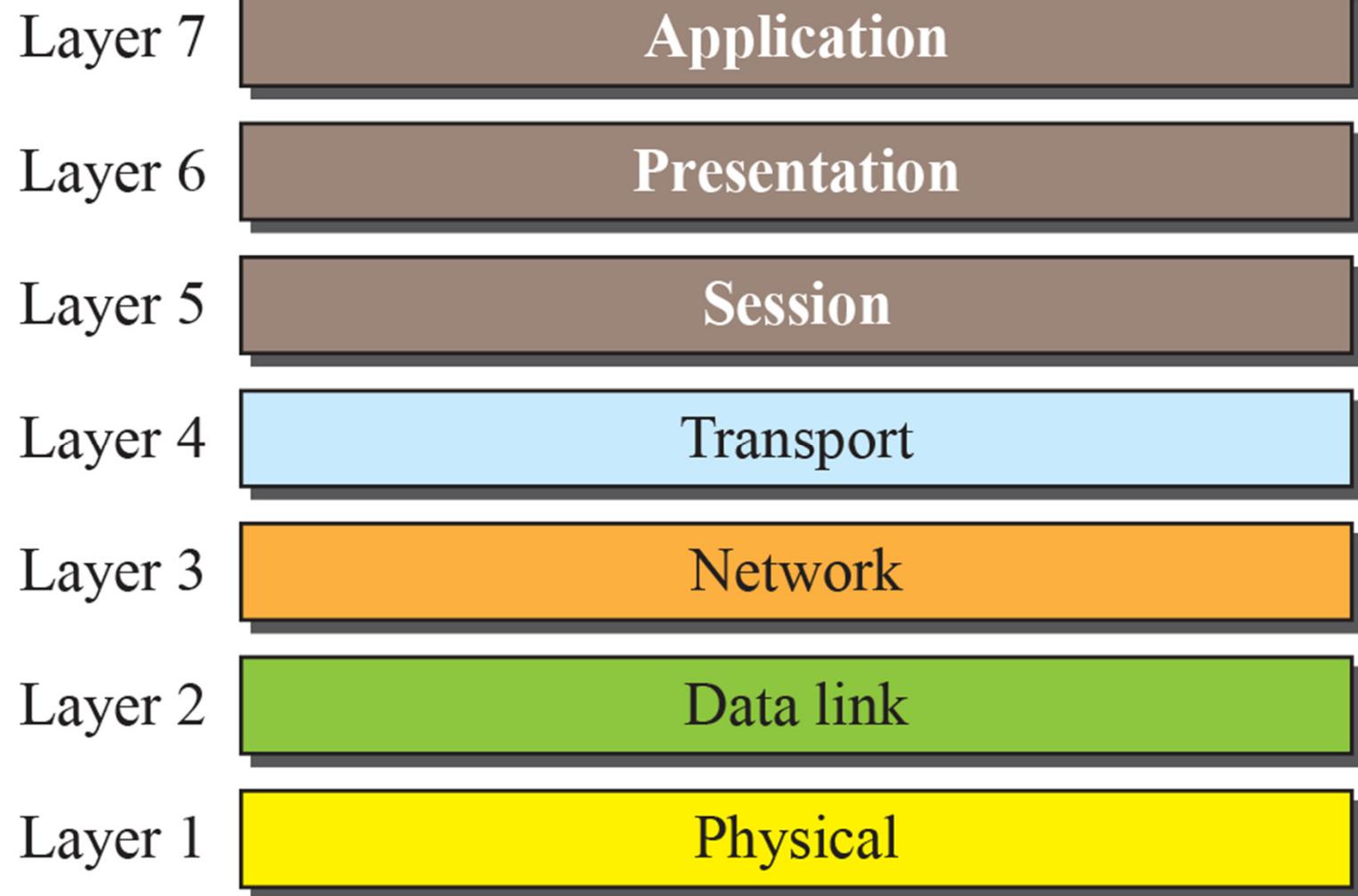


Figure 2.1 Tasks involved in sending a letter

OSI Model

- ❖ 7- layered architecture
- ❖ Provides guidelines for the development of universally compatible architecture, hardware and software
- ❖ Each layer
 - ❖ *provides **services** to the layer **above***
 - ❖ *utilizing the **services** of the layer **below***
- ❖ Communications between computers is a peer-to-peer process using the protocols appropriate to a given layer

OSI Model



Functions of layers

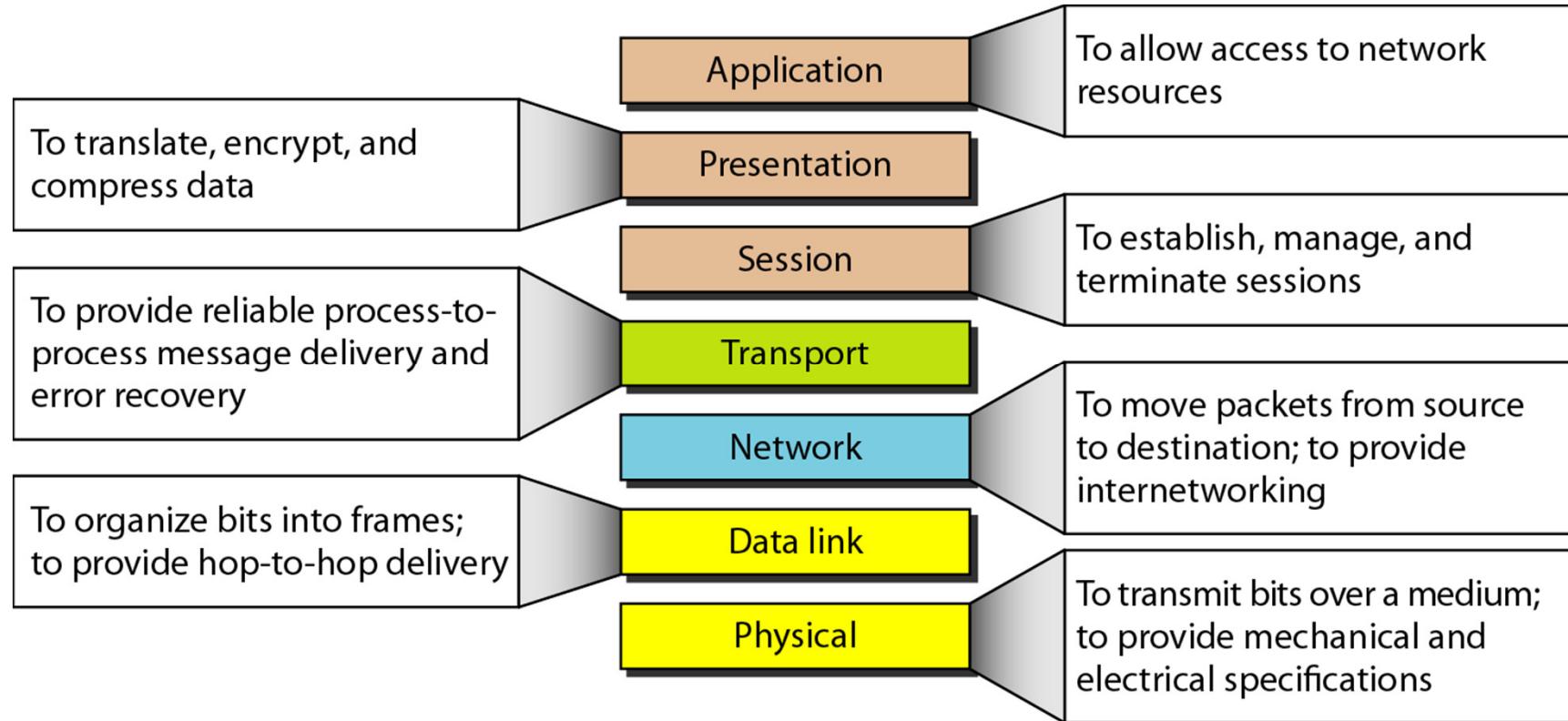
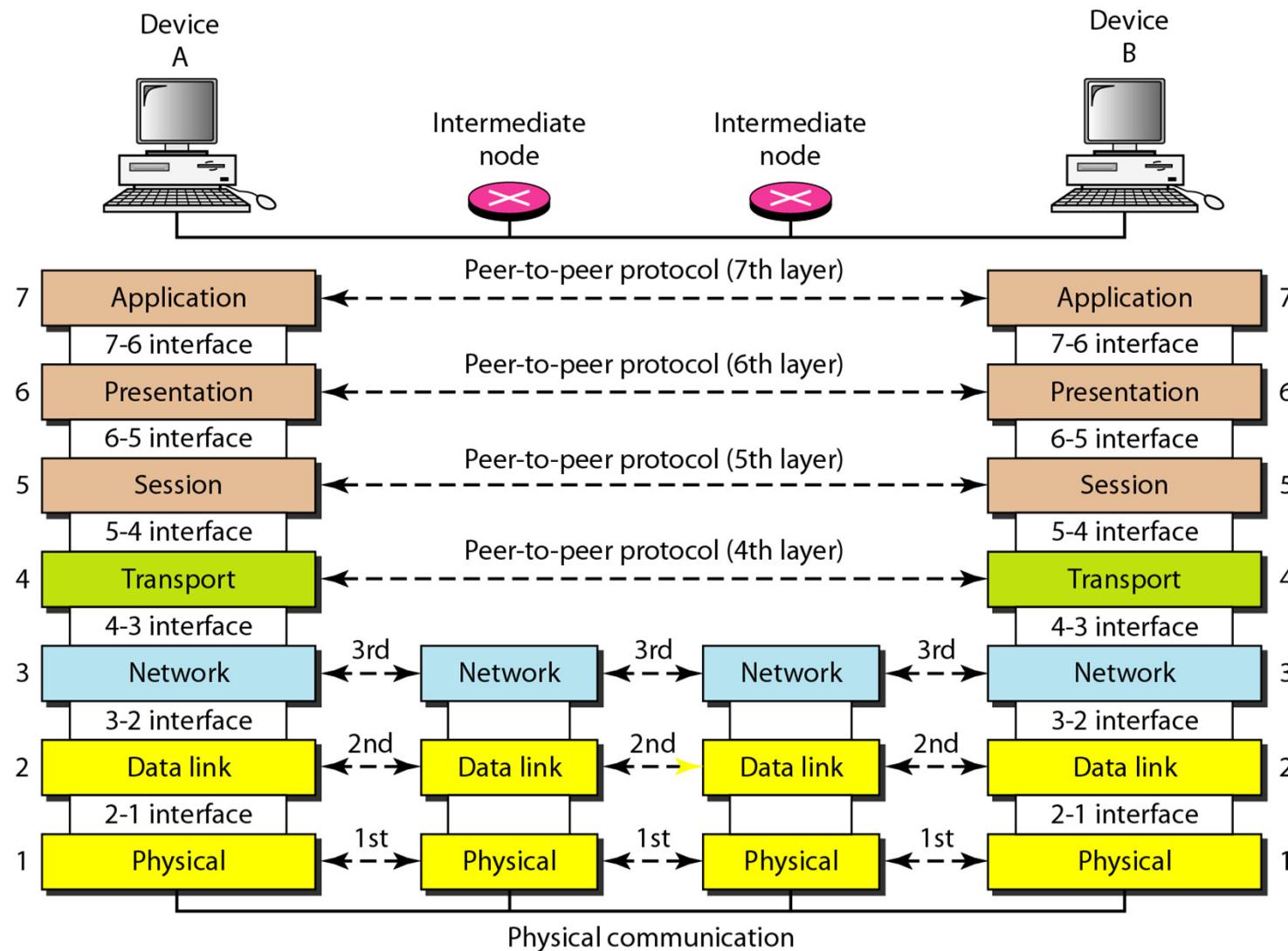


Figure 2.15

Figure 2.3 *The interaction between layers in the OSI model*



TCP/IP Protocol Suite

- ❖ 5-layered architecture
- ❖ Being used by current Internet

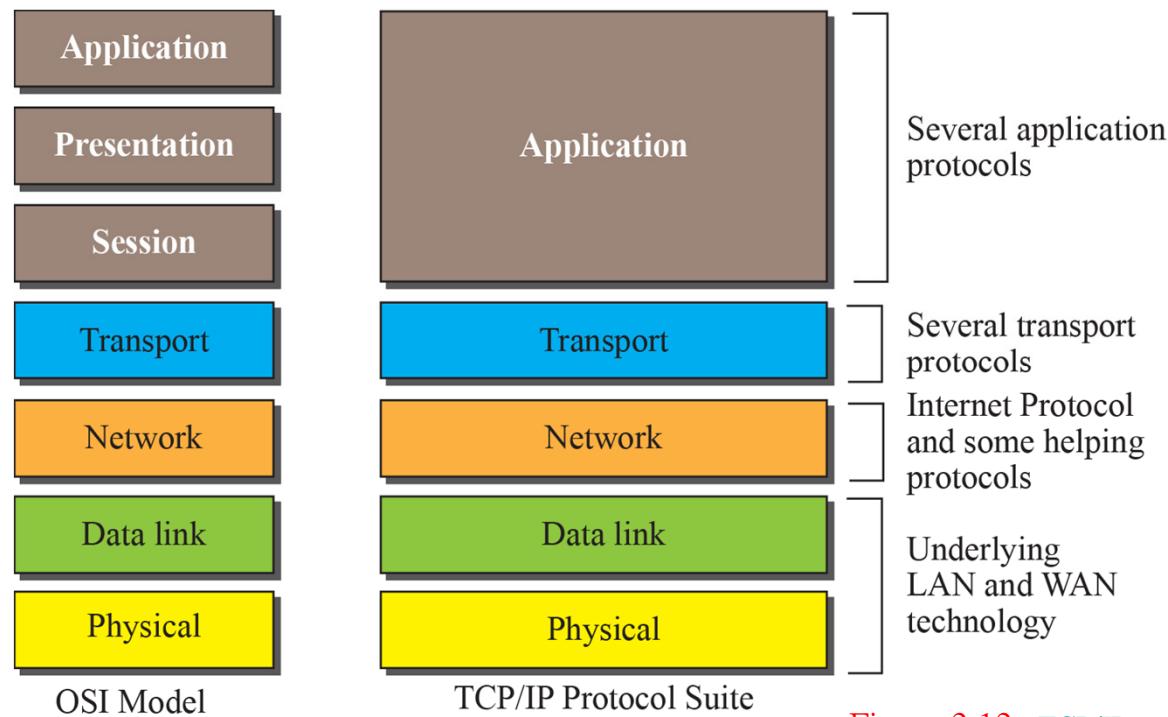
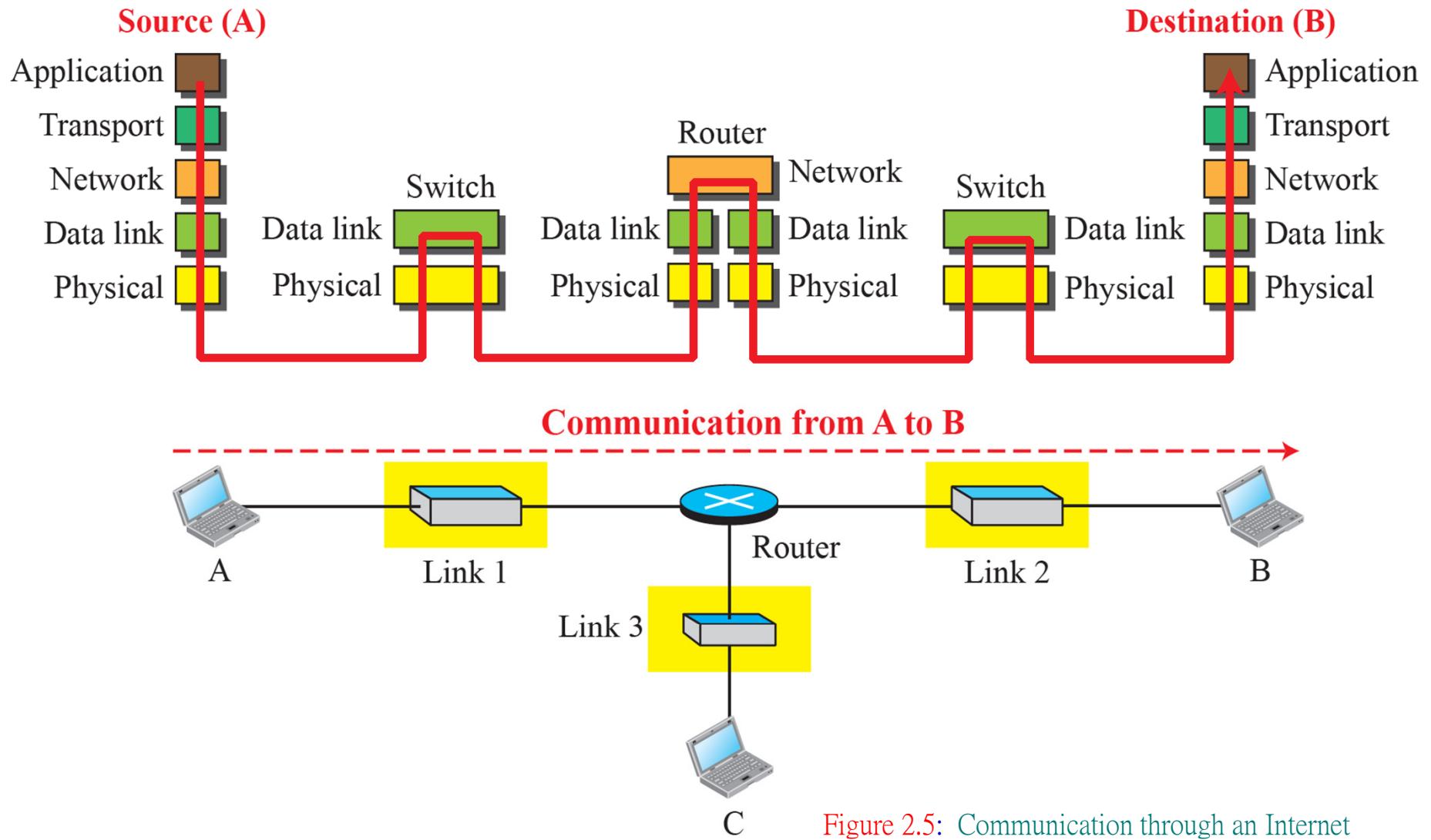


Figure 2.12: TCP/IP and OSI model

Communication through the Internet



How data is transmitted

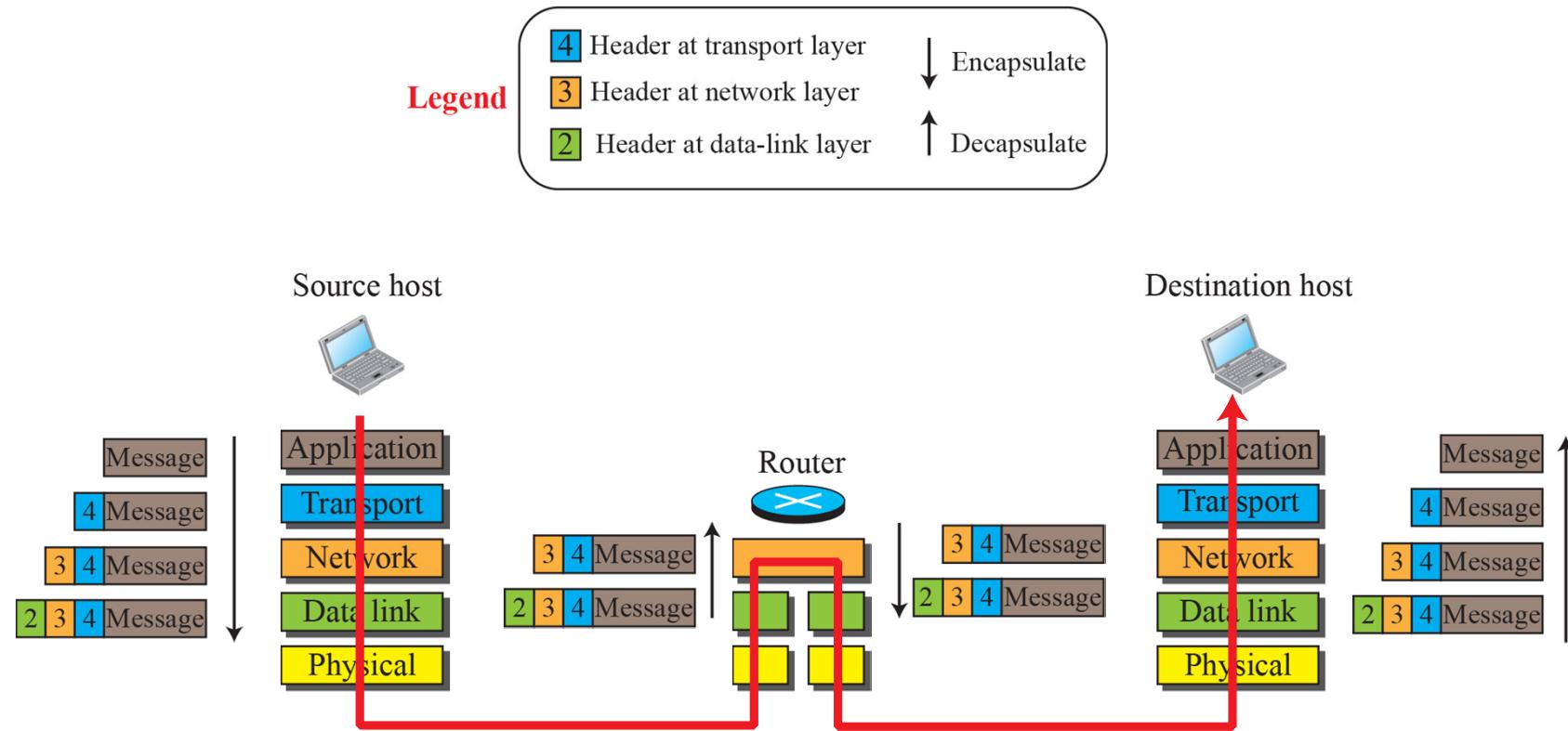


Figure 2.8: Encapsulation / Decapsulation

Examples of Protocols

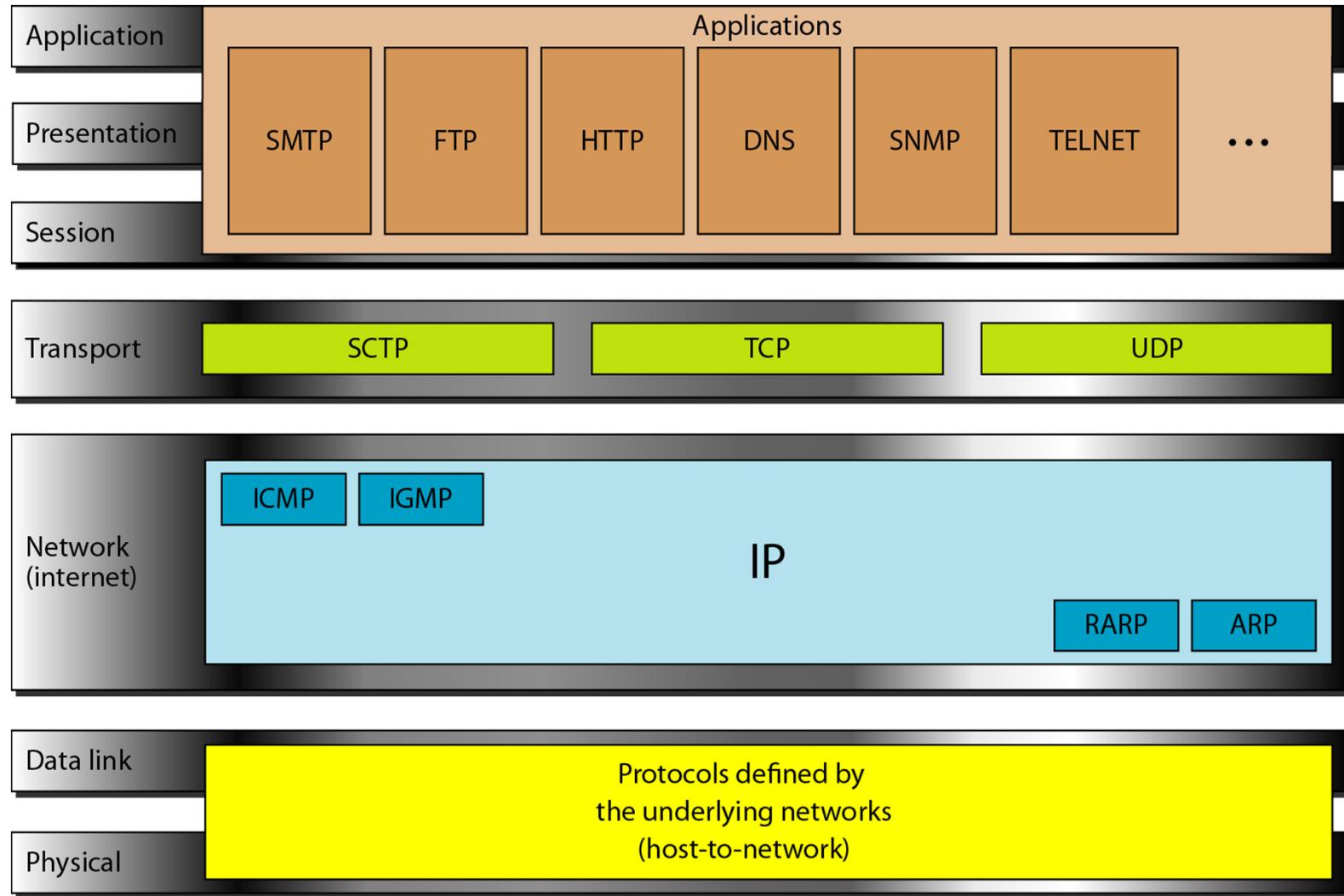


Figure 2.16

Summary

- 1. Types of connection: Point-to-point and Multipoint**
- 2. Transmission Mode: Simplex, ...**
- 3. Topology: Mesh, bus, ring, star, hybrid**
- 4. OSI Model, TCP/IP protocol suite**

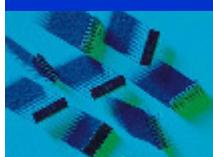
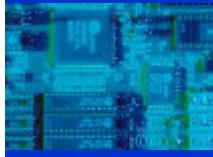
Revision Quiz:

❑ Chapter 1

- http://highered.mheducation.com/sites/0073376221/student_view0/chapter1/quizzes.html

❑ Chapter 2

- http://highered.mheducation.com/sites/0073376221/student_view0/chapter2/quizzes.html



Lecture 2 Basic Communication Principles

Textbook: Ch.3 and Ch.4

Main Topics

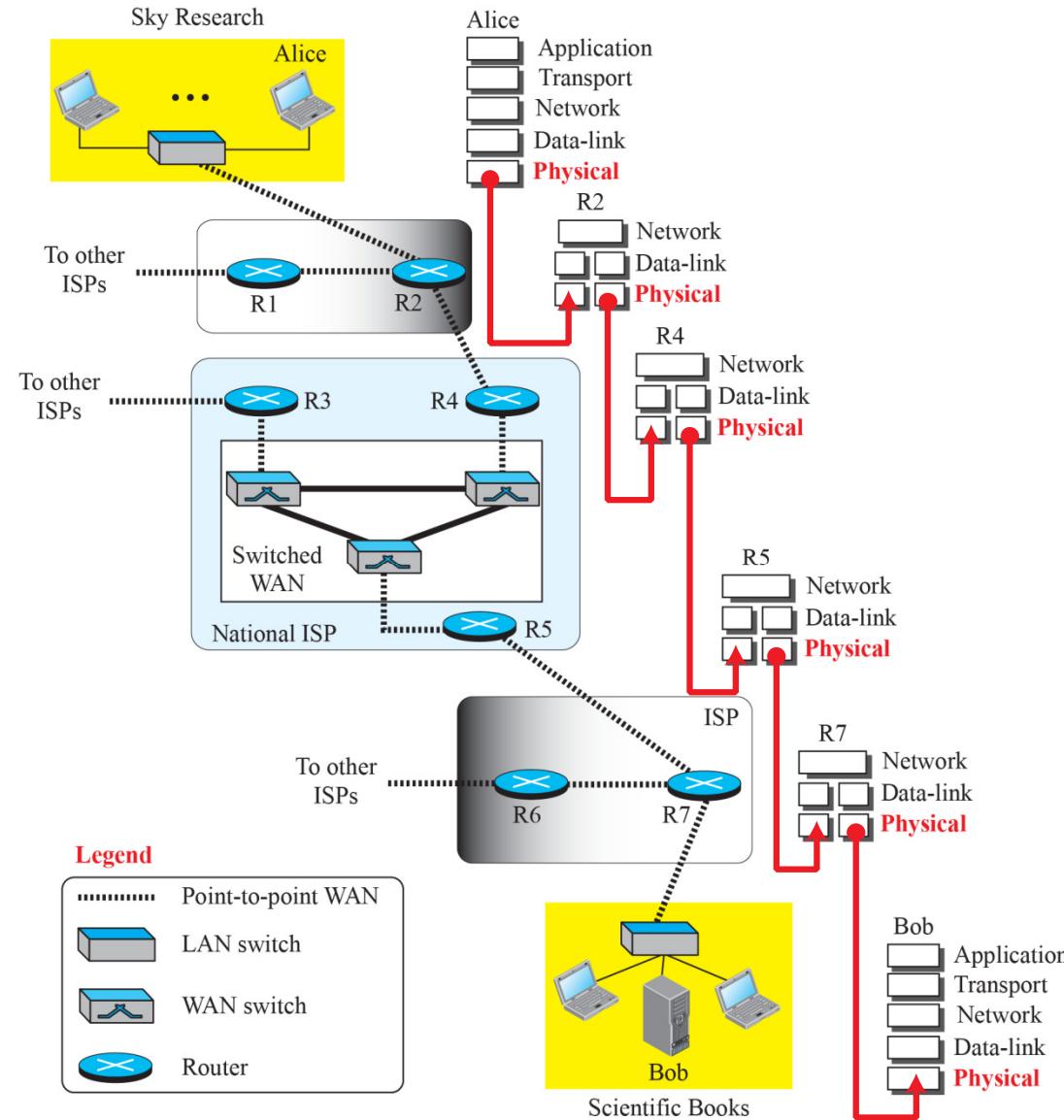
Ch 3. Physical Layer

- ❑ **3.1 Data and Signal**
- ❑ **3.2 Periodic Analogue Signals**
- ❑ **3.3 Digital Signals**
- ❑ **3.4 Transmission Impairment**
- ❑ **3.5 Data rate limit**
- ❑ **3.6 Performance**

Ch 4. Digital Transmission

- ❑ **Analog-to-digital Conversion (PCM)**

Physical layer



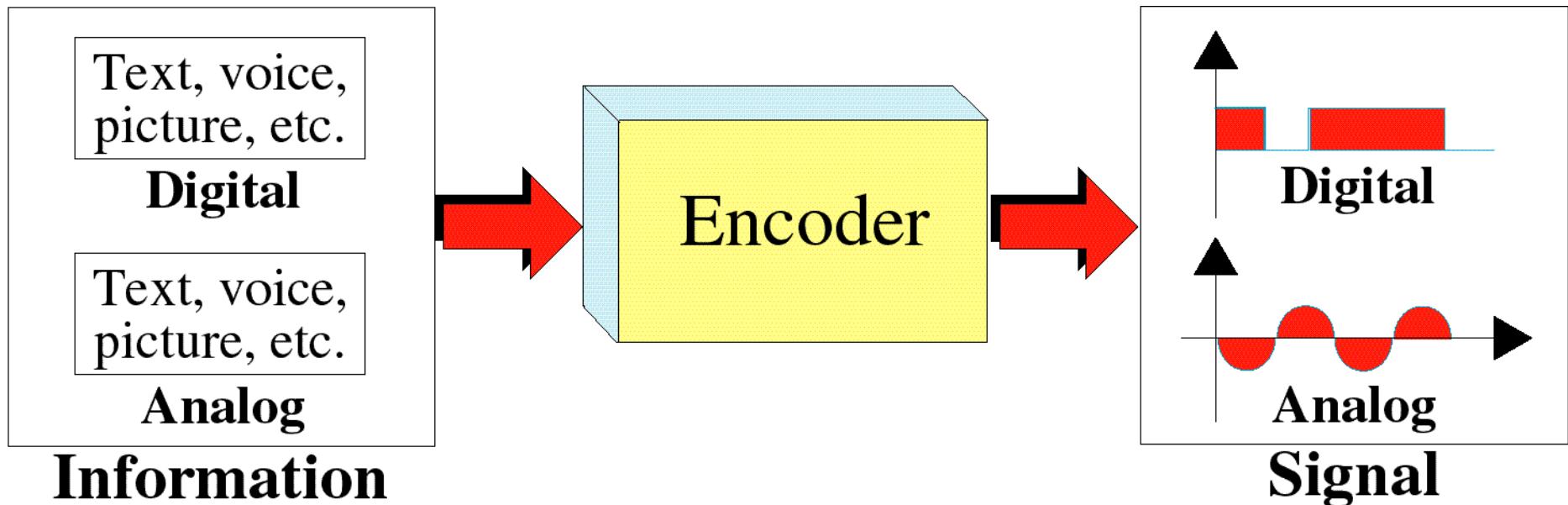
1.3

Figure 3.1: Communication at the physical layer

3.1 Data and Signal

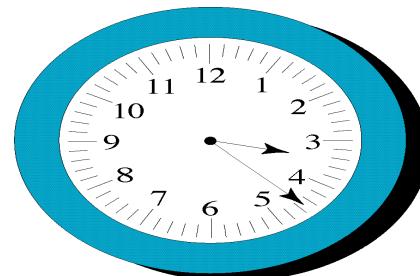
Transformation of Information to Signals

- ❖ To be transmitted, data must be transformed to electromagnetic signals



Analog and Digital

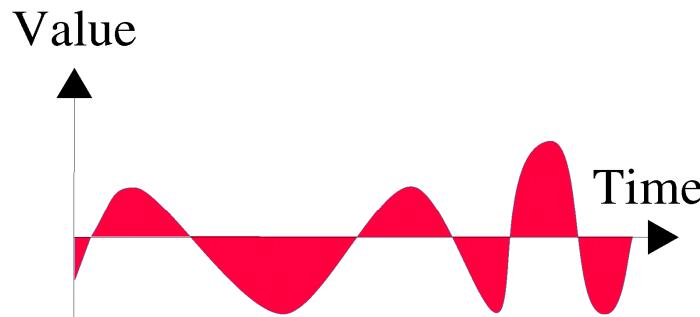
- ❖ **Analog signals** can have any value in a range (continuous values)
- ❖ **Digital signals** can have only a limited number of values (discrete values)



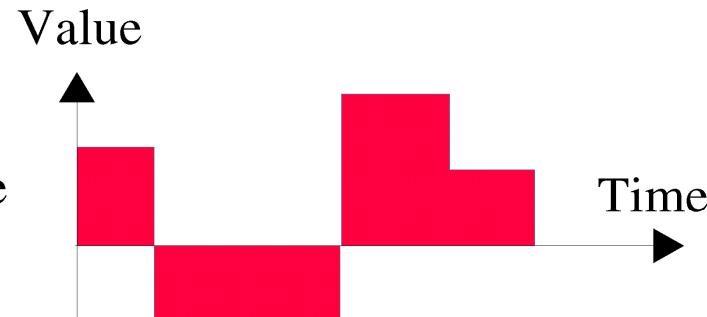
a. Analog



b. Digital



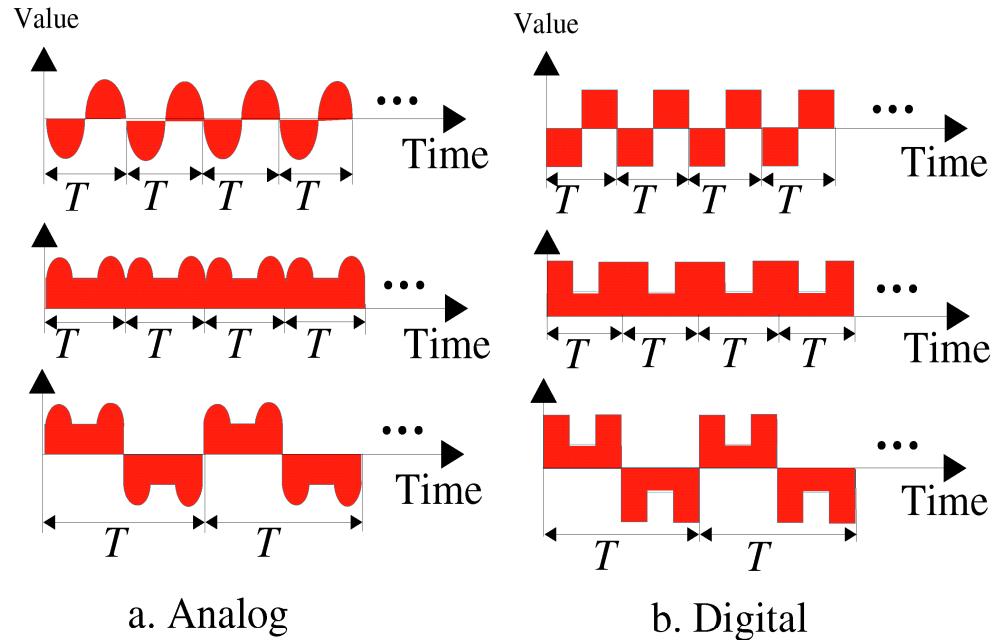
a. Analog signal



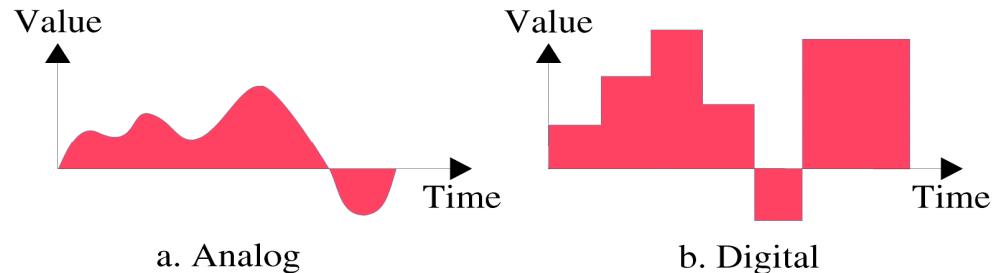
b. Digital signal

3.2 Periodic Analogue Signals

Periodic Signals
consists of a
continuously
repeated pattern

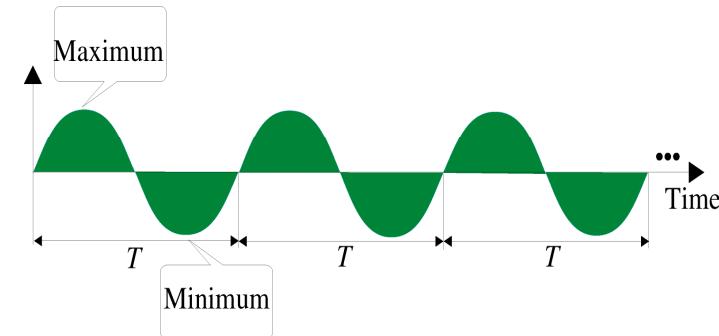


Aperiodic Signals
has no repetitive
pattern



Periodic Analog Signals

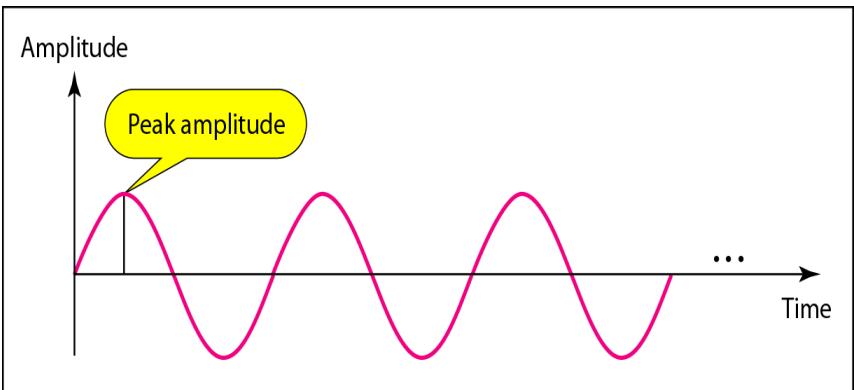
- ❖ The **sine wave** is the most fundamental form of a periodic signal
- ❖ A periodic signal can be decomposed into a set of sine waves



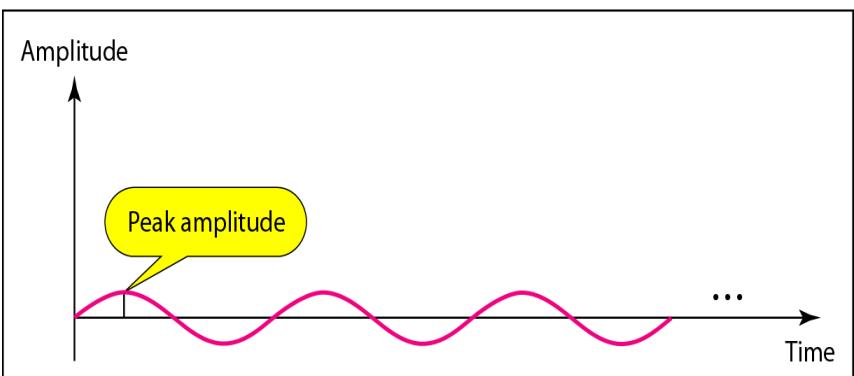
- ❖ Characteristics of a sine wave:
 - ❖ Amplitude – the instantaneous height
 - ❖ Frequency – the no. of cycles per second (Hz)
 - ❖ Frequency and period are the inverse of each other
 - ❖ Phase – the shift of the wave along the time axis (relative to time zero) measured in degrees or radians

$$f = \frac{1}{T} \quad \text{and} \quad T = \frac{1}{f}$$

Different Amplitudes

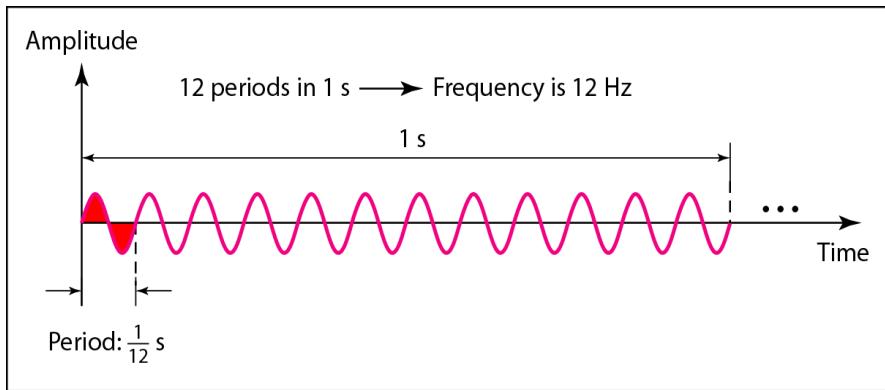


a. A signal with high peak amplitude

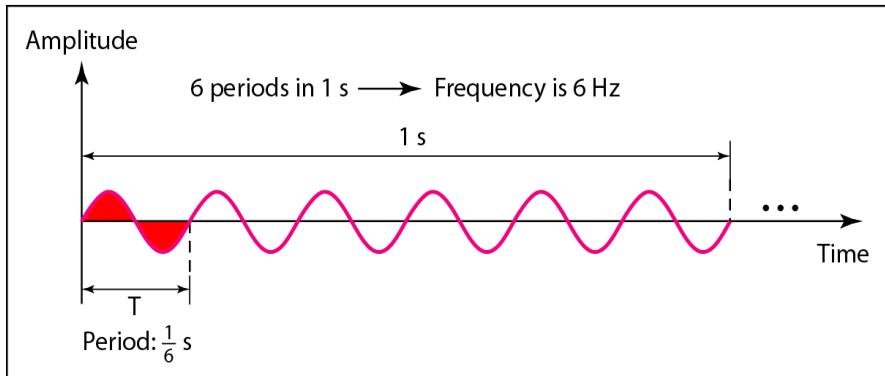


b. A signal with low peak amplitude

Different Frequencies



a. A signal with a frequency of 12 Hz



b. A signal with a frequency of 6 Hz

Different Phases

Amplitude

Time

a. 0 degrees

Amplitude

Time

b. 90 degrees

Amplitude

Time

1/2 cycle

c. 180 degrees

Amplitude

Time

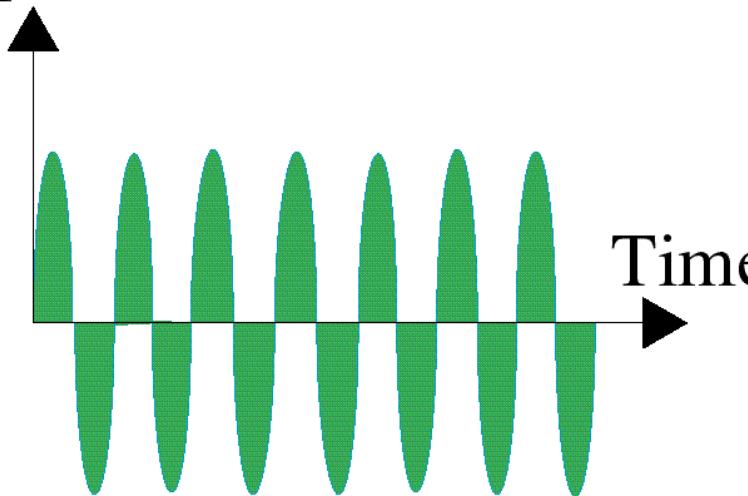
d. 270 degrees

Time and Frequency Domains

- ❖ A time-domain graph plots amplitude as a function of time
- ❖ A frequency-domain graph plots each sine wave's peak amplitude against its frequency

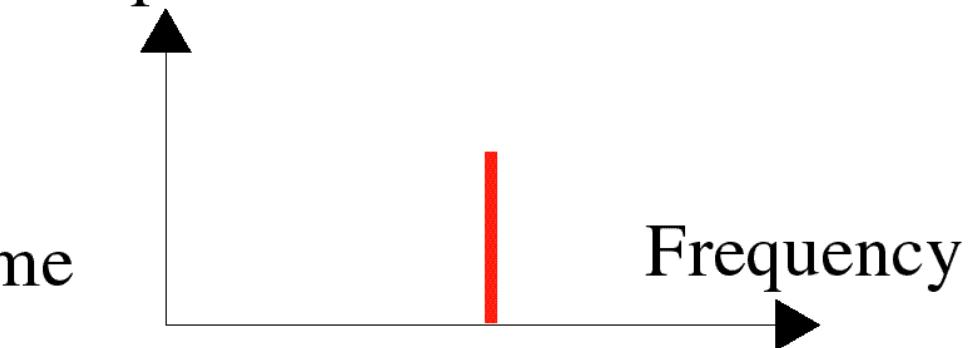
Time and Frequency Domain

Amplitude



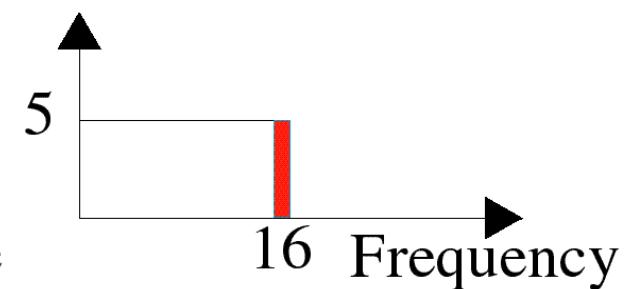
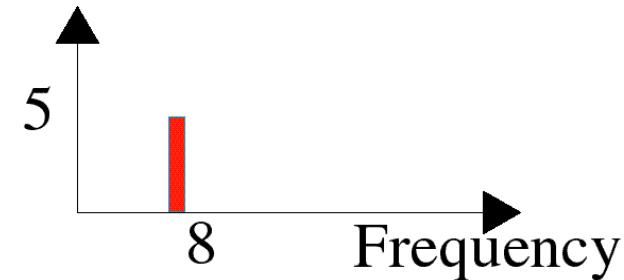
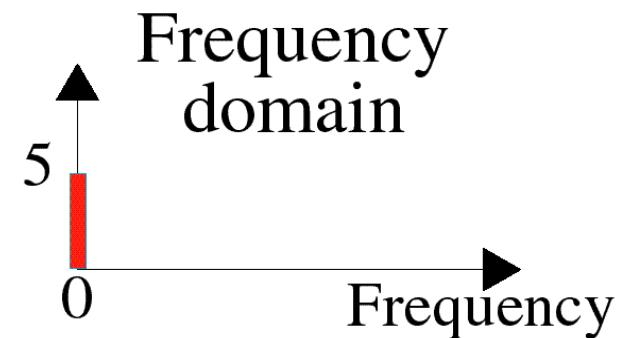
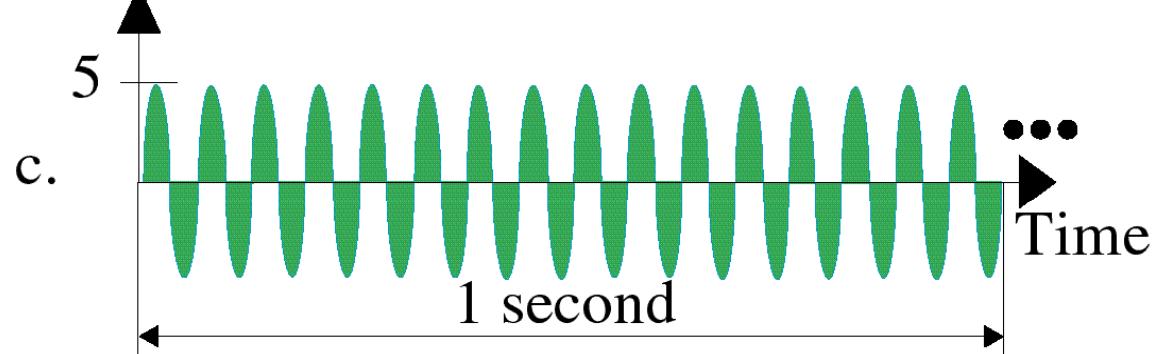
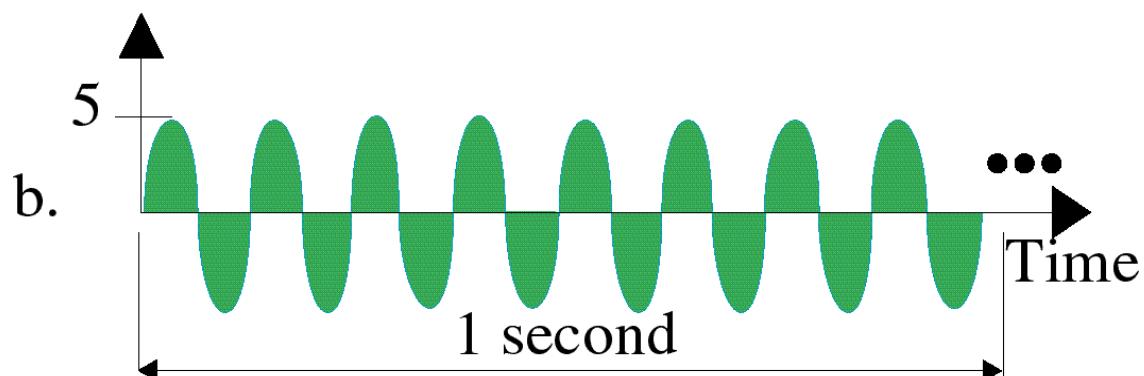
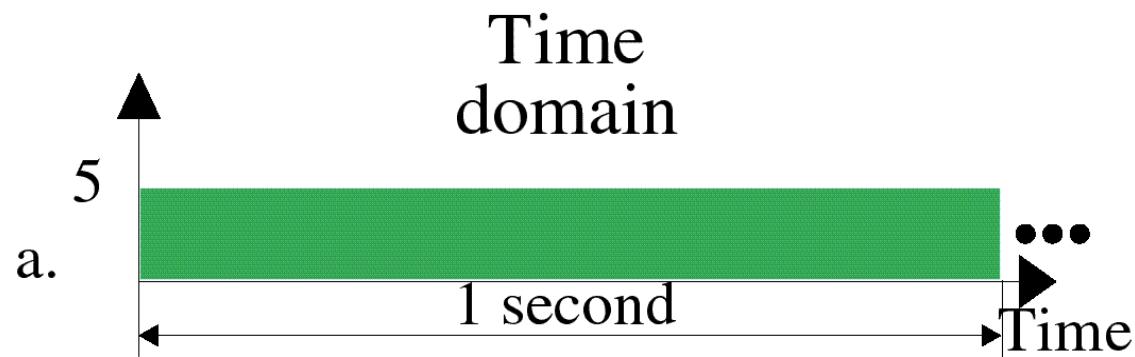
a. Time domain

Amplitude



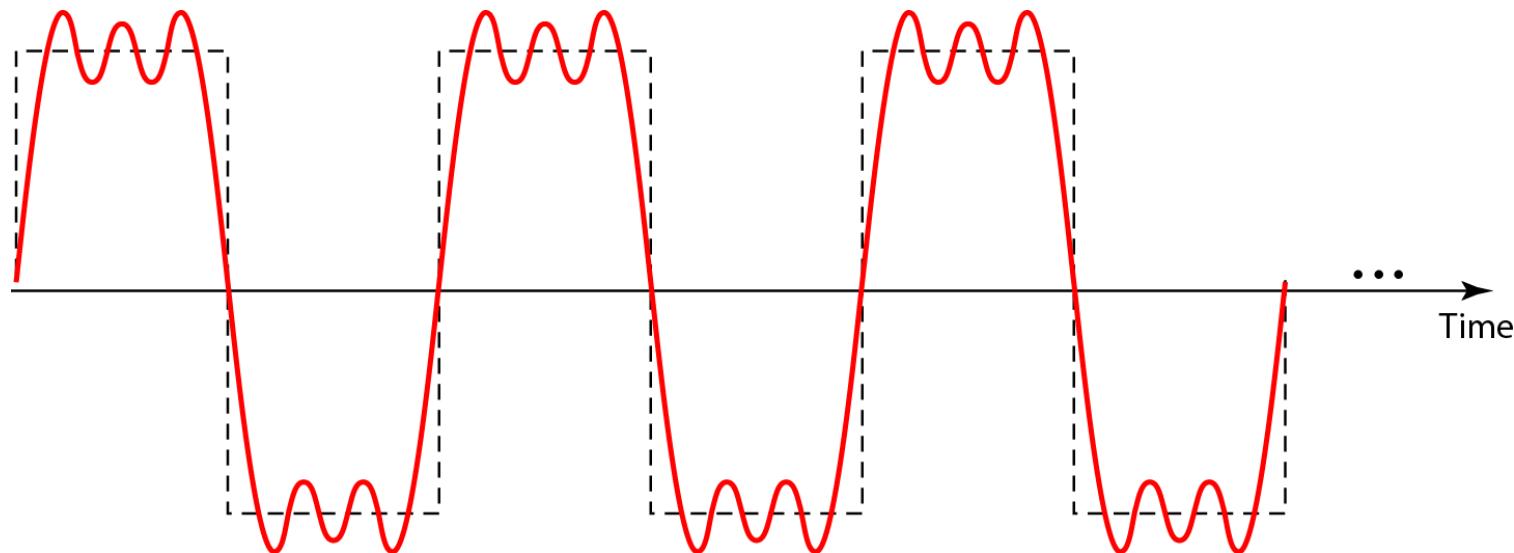
b. Frequency domain

Examples

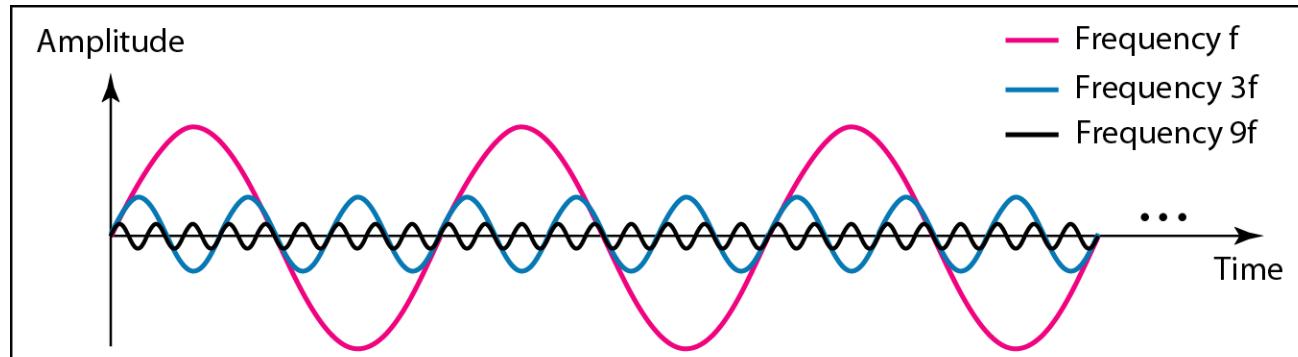


Composite Signals

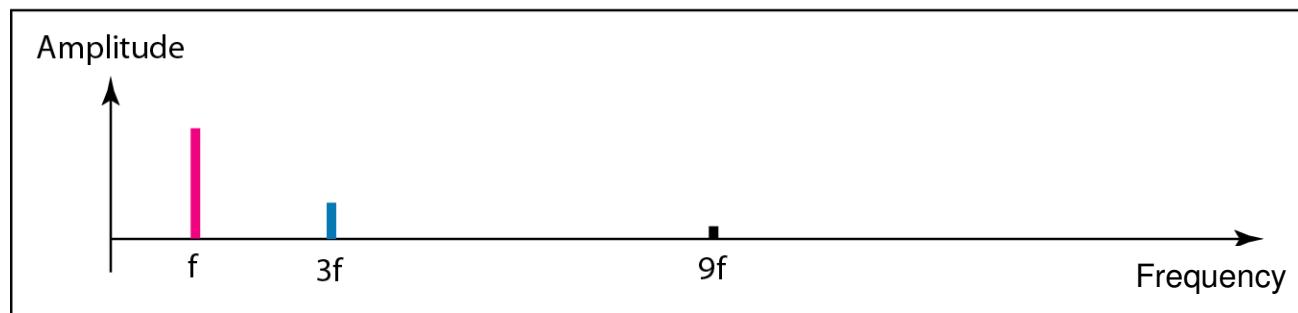
- ❖ Periodic signal can be decomposed into a set of sine waves (called **components**)
 - ❖ each has its own amplitude, frequency, and phase



Decomposition of the signal



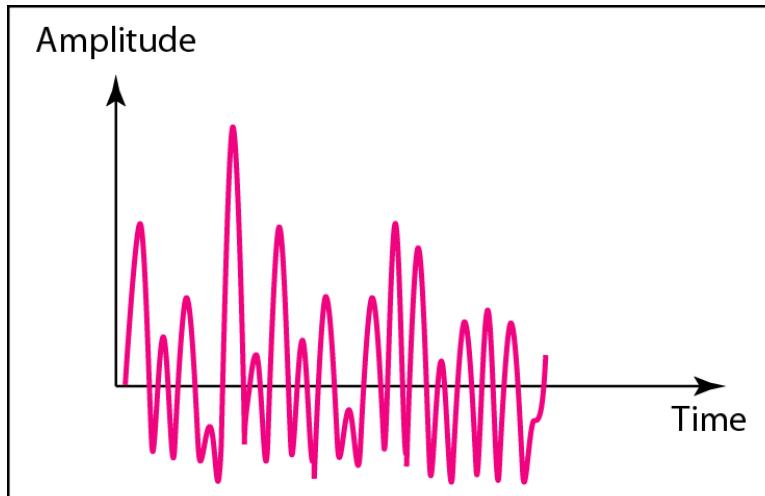
a. Time-domain decomposition of a composite signal



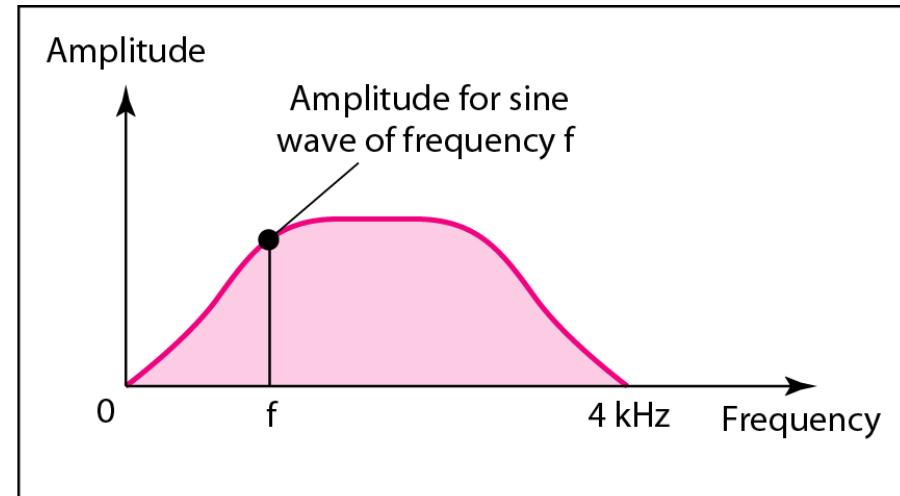
b. Frequency-domain decomposition of the composite signal

Bandwidth of a Signal

- ❖ The bandwidth of a composite signal is the **difference** between the **highest** and the **lowest** frequencies contained in that signal.



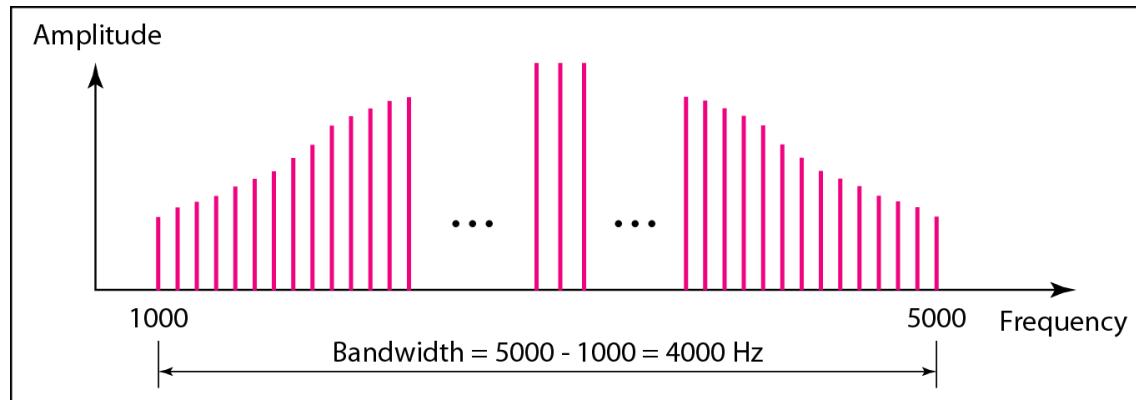
a. Time domain



b. Frequency domain

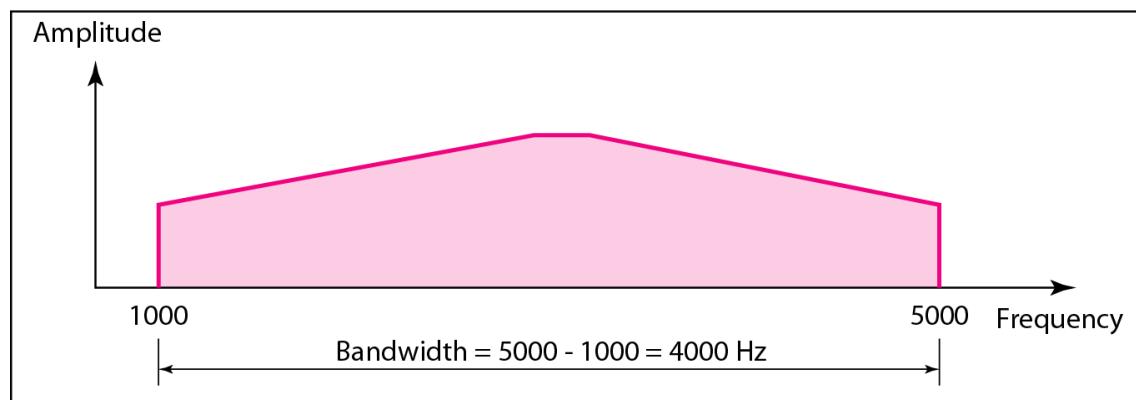
Bandwidth of periodic and aperiodic signals

Periodic Signal
contains discrete frequencies



a. Bandwidth of a periodic signal

Aperiodic Signal
with continuous frequencies



b. Bandwidth of a nonperiodic signal

Example

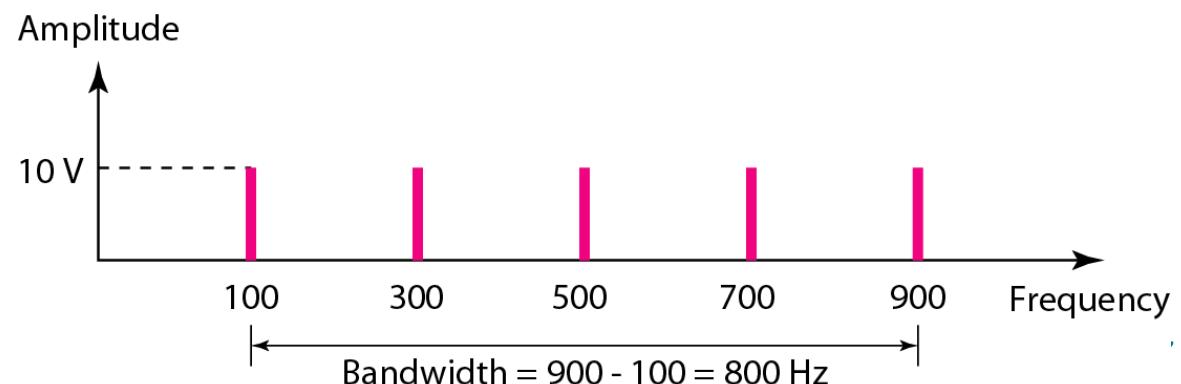
If a periodic signal is decomposed into five sine waves with frequencies of 100, 300, 500, 700, and 900 Hz, what is its bandwidth? Draw the spectrum, assuming all components have a maximum amplitude of 10 V.

Solution

Let f_h be the highest frequency, f_l the lowest frequency, and B the bandwidth. Then

$$B = f_h - f_l = 900 - 100 = 800 \text{ Hz}$$

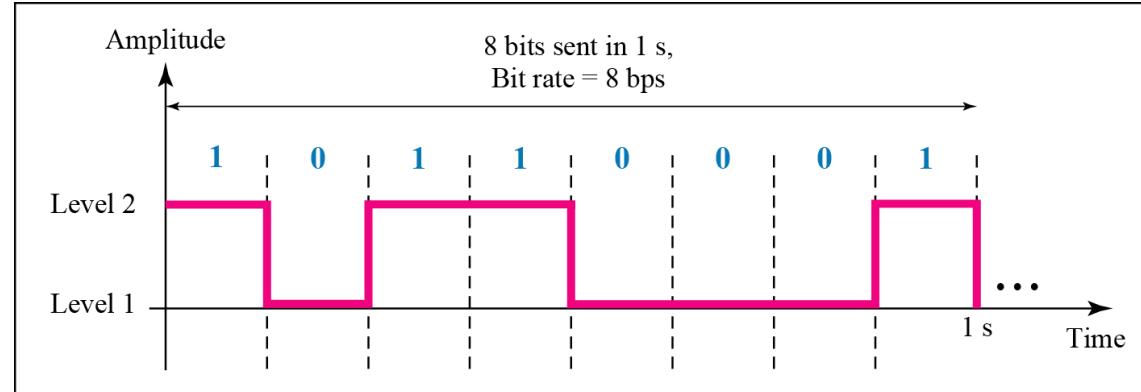
The spectrum has only five spikes, at 100, 300, 500, 700, and 900 Hz



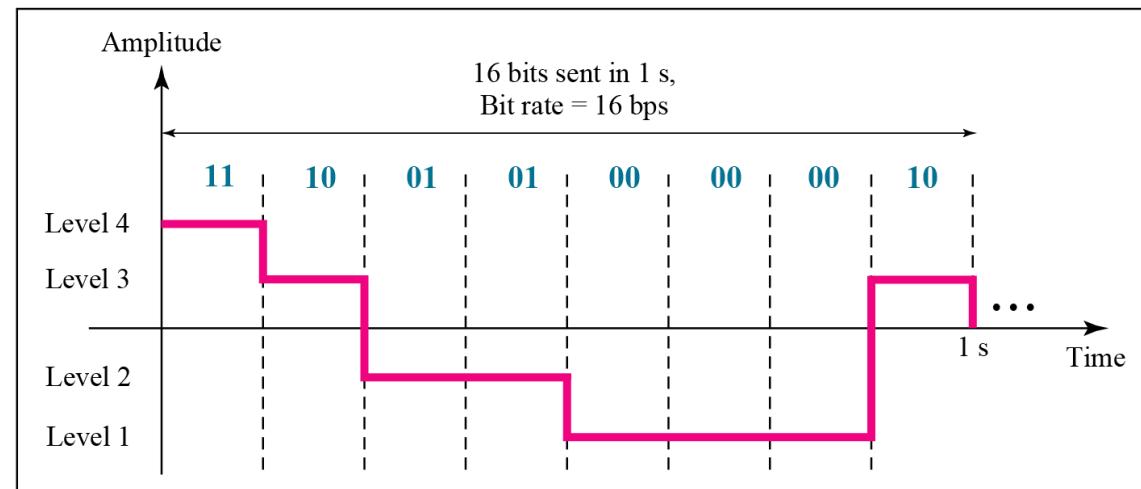
3.3 Digital Signals

- In addition to being represented by an analog signal, information can also be represented by a digital signal
- For example, a 1 can be encoded as a positive voltage and a 0 as zero voltage. A digital signal can have more than two levels. In this case, we can send more than 1 bit for each level.

Digital signal can have more than two levels, in this case, more bits can be sent in each level.



a. A digital signal with two levels



b. A digital signal with four levels

Example

A digital signal has eight levels. How many bits are needed per level?

Solution

We calculate the number of bits from the formula

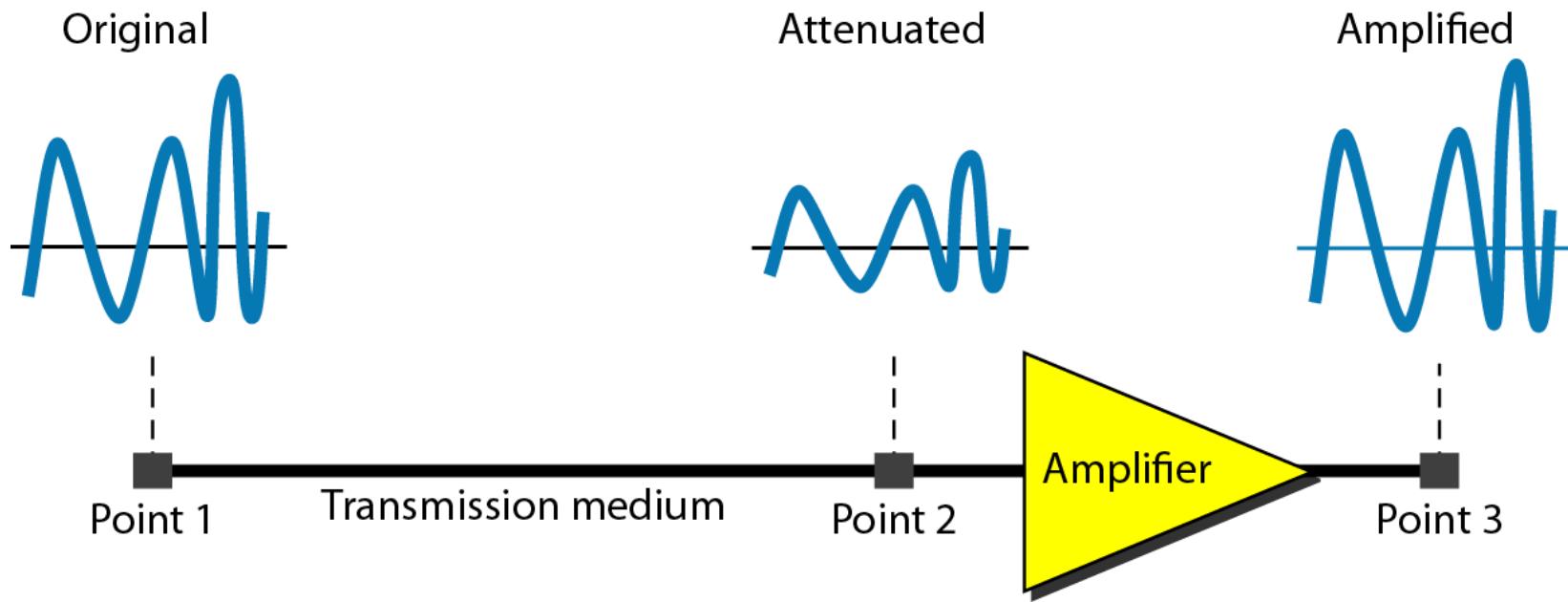
$$\text{Number of bits per level} = \log_2 8 = 3$$

Each signal level is represented by 3 bits.

3.4 TRANSMISSION IMPAIRMENT

- ❖ Signals travel through transmission media, which are not perfect. The imperfection causes signal impairment.
- ❖ The signal at the beginning of the medium is not the same as the signal at the end of the medium.
 - ❖ What is sent is not what is received.
- ❖ Three causes of impairment are
 - ❖ Attenuation
 - ❖ Distortion
 - ❖ Noise

Attenuation



❖ **Attenuation means a loss of energy.**

❖ **Measured by decibel (dB) =**

$$10 \log_{10} \frac{P_2}{P_1}$$

Example

Suppose a signal travels through a transmission medium and its power is reduced to one-half. This means that P_2 is $(1/2)P_1$. In this case, the attenuation (loss of power) can be calculated as

$$10 \log_{10} \frac{P_2}{P_1} = 10 \log_{10} \frac{0.5P_1}{P_1} = 10 \log_{10} 0.5 = 10(-0.3) = -3 \text{ dB}$$

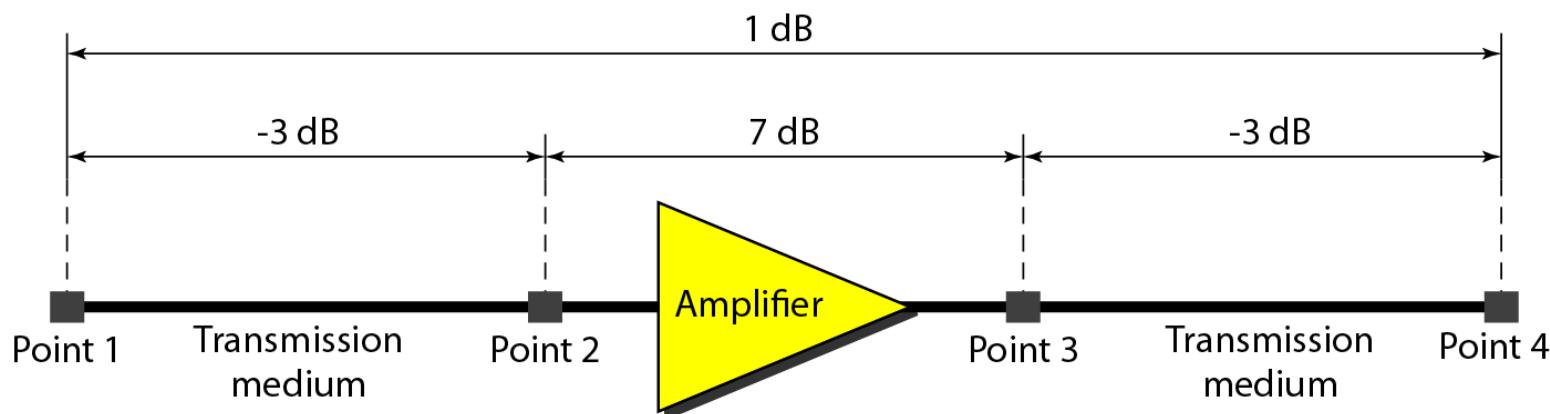
A loss of 3 dB (-3 dB) is equivalent to losing one-half the power.

Reason for using dB

Decibel numbers can be added (or subtracted) when we are measuring several points (cascading) instead of just two.

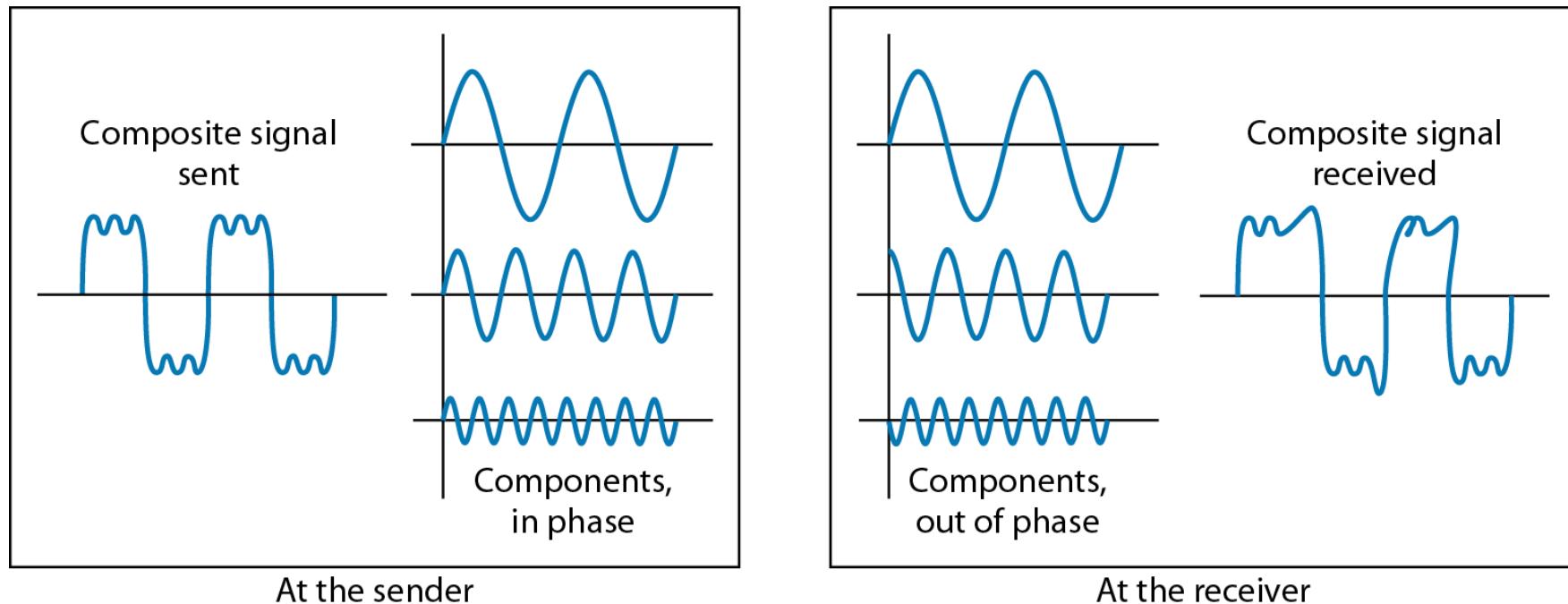
Example: A signal travels from point 1 to point 4. In this case, the decibel value can be calculated as

$$\text{dB} = -3 + 7 - 3 = +1$$



Distortion

- ❖ **Distortion** means that the signal **changes its form** or shape.
- ❖ Main cause: **Difference in delay** of different **frequency components**.



Noise

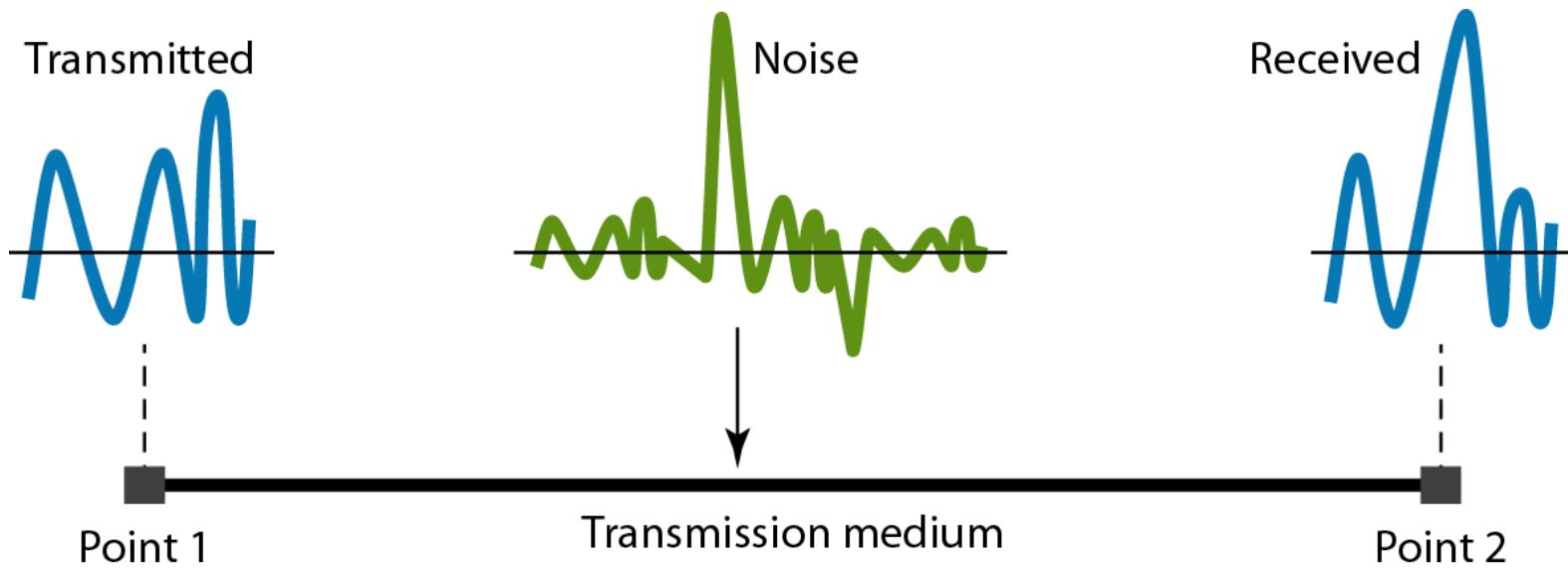
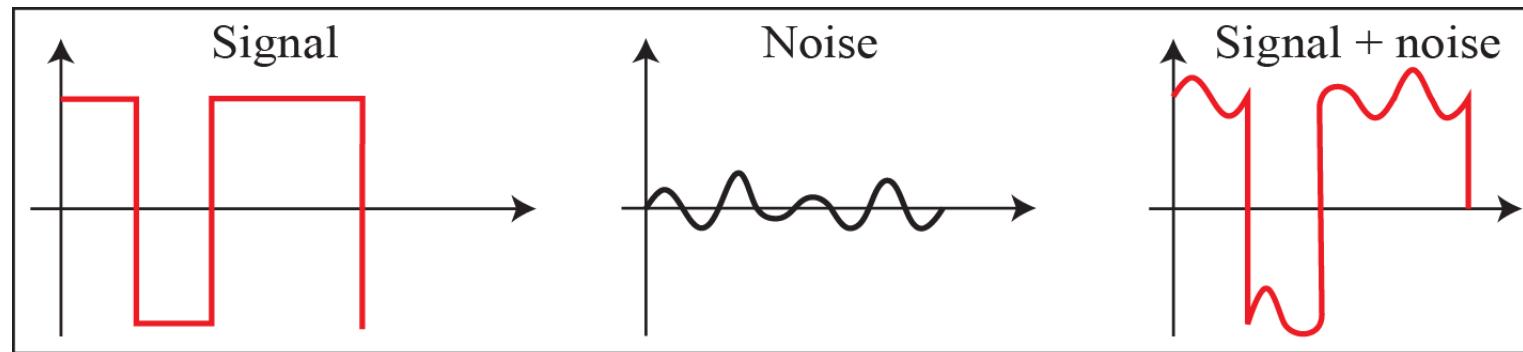


Figure 3.29

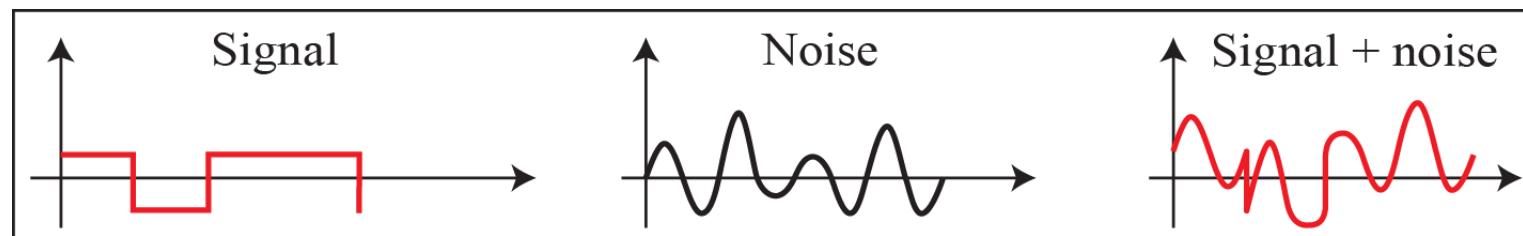
- ❖ Noise includes thermal noise, induced noise, crosstalk, and impulse noise, etc.
- ❖ Measured by **Signal-to-noise ratio (SNR)**:

$$\text{SNR} = \frac{\text{Average signal power}}{\text{Average noise power}}$$

Two cases of SNR: a high SNR and a low SNR



a. High SNR



b. Low SNR

Example

The power of a signal is 10 mW and the power of the noise is 1 μW. What are the values of SNR and SNR_{dB}?

Solution

The values of SNR and SNR_{dB} can be calculated as follows:

$$\text{SNR} = \frac{10,000 \mu\text{W}}{1 \mu\text{W}} = 10,000$$

$$\text{SNR}_{\text{dB}} = 10 \log_{10} 10,000 = 10 \log_{10} 10^4 = 40$$

3.5 DATA RATE LIMITS

A very important consideration in data communications is how fast we can send data, in bits per second, over a channel.

Data rate depends on three factors:

- 1. The bandwidth available*
- 2. The level of the signals we use*
- 3. The quality of the channel (the level of noise)*

Noiseless Channel: Nyquist Bit Rate

- ❖ For a noiseless Channel, the **Nyquist bit rate** formula defines the theoretical maximum bit rate:

$$\text{BitRate} = 2 \times \text{bandwidth} \times \log_2 L$$

Note

Increasing the levels of a signal may reduce the reliability of the system.

- ❖ Receiver becomes difficult to distinguish different levels

Examples on Nyquist Bit Rate

- ❖ Consider a noiseless channel with a bandwidth of 3000 Hz transmitting a signal with **two signal levels**. The maximum bit rate can be calculated as

$$\text{BitRate} = 2 \times 3000 \times \log_2 2 = 6000 \text{ bps}$$

- ❖ Consider the same noiseless channel transmitting a signal with **four signal levels** (for each level, we send 2 bits). The maximum bit rate can be calculated as

$$\text{BitRate} = 2 \times 3000 \times \log_2 4 = 12,000 \text{ bps}$$

Noisy Channel: Shannon Capacity

- ❖ In reality, we cannot have a noiseless channel; the channel is always noisy. In 1944, Claude Shannon introduced a formula, called the Shannon capacity.
- ❖ The theoretical maximum data rate of a noisy channel is related to SNR

$$C = B \log_2 (1 + SNR) \text{ bps}$$

- ❖ B – bandwidth of the channel (in Hz)
- ❖ C – (Shannon) Capacity of the channel (in bps)
- ❖ Actual data rate usually is smaller
- ❖ Regardless to number of signal levels

Example on Shannon Capacity

We can calculate the theoretical highest bit rate of a regular telephone line. A telephone line normally has a bandwidth of 3000. The signal-to-noise ratio is usually 3162. For this channel the capacity is calculated as

$$\begin{aligned} C &= B \log_2 (1 + \text{SNR}) = 3000 \log_2 (1 + 3162) = 3000 \log_2 3163 \\ &= 3000 \times 11.62 = 34,860 \text{ bps} \end{aligned}$$

This means that the highest bit rate for a telephone line is 34.860 kbps. If we want to send data faster than this, we can either increase the bandwidth of the line or improve the signal-to-noise ratio.

3.6. Performance Throughput

- The *throughput* is a measure of how fast we can actually send data through a network.
- Although, at first glance, bandwidth in bits per second and throughput seem the same, they are different.
- A link may have a bandwidth of B bps, but we can only send T bps, but
 T always less than B .

Throughput Example

A network with bandwidth of 10 Mbps can pass only an average of 12,000 frames per minute with each frame carrying an average of 10,000 bits. What is the throughput of this network?

Solution

We can calculate the throughput as

$$\text{Throughput} = (12,000 \times 10,000) / 60 = 2 \text{ Mbps}$$

The throughput is almost one-fifth of the bandwidth in this case.

Latency or Delay

- The latency or delay defines how long it takes for an entire message to completely arrive at the destination from the time the first bit is sent out from the source.
- We can say that latency is made of four components:
 - propagation time,
 - transmission time,
 - queuing time, and
 - processing delay.

Latency = propagation time + transmission time + queuing time + processing delay

Propagation Time and Transmission Time

- ❖ Propagation Time
 - ❖ Measures the time required for a bit to travel from the source to the destination. The propagation time is calculated by dividing distance by the propagation speed.

$$\text{Propagation Time} = \text{Distance} / (\text{Propagation Speed})$$

- ❖ We will use T_p as the short form for Propagation Time.

- ❖ Transmission Time
 - ❖ Measures the time between the first bit and the last bit leaving the sender. The Transmission time depends on the size of the message and the bandwidth of the channel.

$$\text{Transmission Time} = (\text{Message Size}) / \text{Bandwidth}$$

- ❖ We will use T_x as the short form for Transmission Time.

Propagation time Example

What is the propagation time if the distance between the two points is 12,000 km? Assume the propagation speed to be 2.4×10^8 m/s in cable.

Solution

We can calculate the propagation time as

$$\text{Propagation time} = (12,000 \times 1,000) / (2.4 \times 10^8) = 50 \text{ ms}$$

The example shows that a bit can go over the Atlantic Ocean in only 50 ms if there is a direct cable between the source and the destination.

What are the propagation time and the transmission time for a 2.5-KB (kilobyte) message if the bandwidth of the network is 1 Gbps? Assume that the distance between the sender and the receiver is 12,000 km and that light travels at 2.4×10^8 m/s.

Solution

We can calculate the propagation and transmission time as

$$\text{Propagation time} = (12,000 \times 1000) / (2.4 \times 10^8) = 50 \text{ ms}$$

$$\text{Transmission time} = (2500 \times 8) / 10^9 = 0.020 \text{ ms}$$

Note that in this case, because the message is short and the bandwidth is high, the dominant factor is the propagation time, not the transmission time.

Example 3.47

Assume that the distance between the sender and the receiver is 12,000 km and that light travels at 2.4×10^8 m/s. What are the propagation time and the transmission time for a 5-MB (megabyte) message (an image) if the bandwidth of the network is 1 Mbps?

Solution

We can calculate the propagation and transmission times as

$$\text{Propagation time} = (12,000 \times 1000) / (2.4 \times 10^8) = 50 \text{ ms}$$

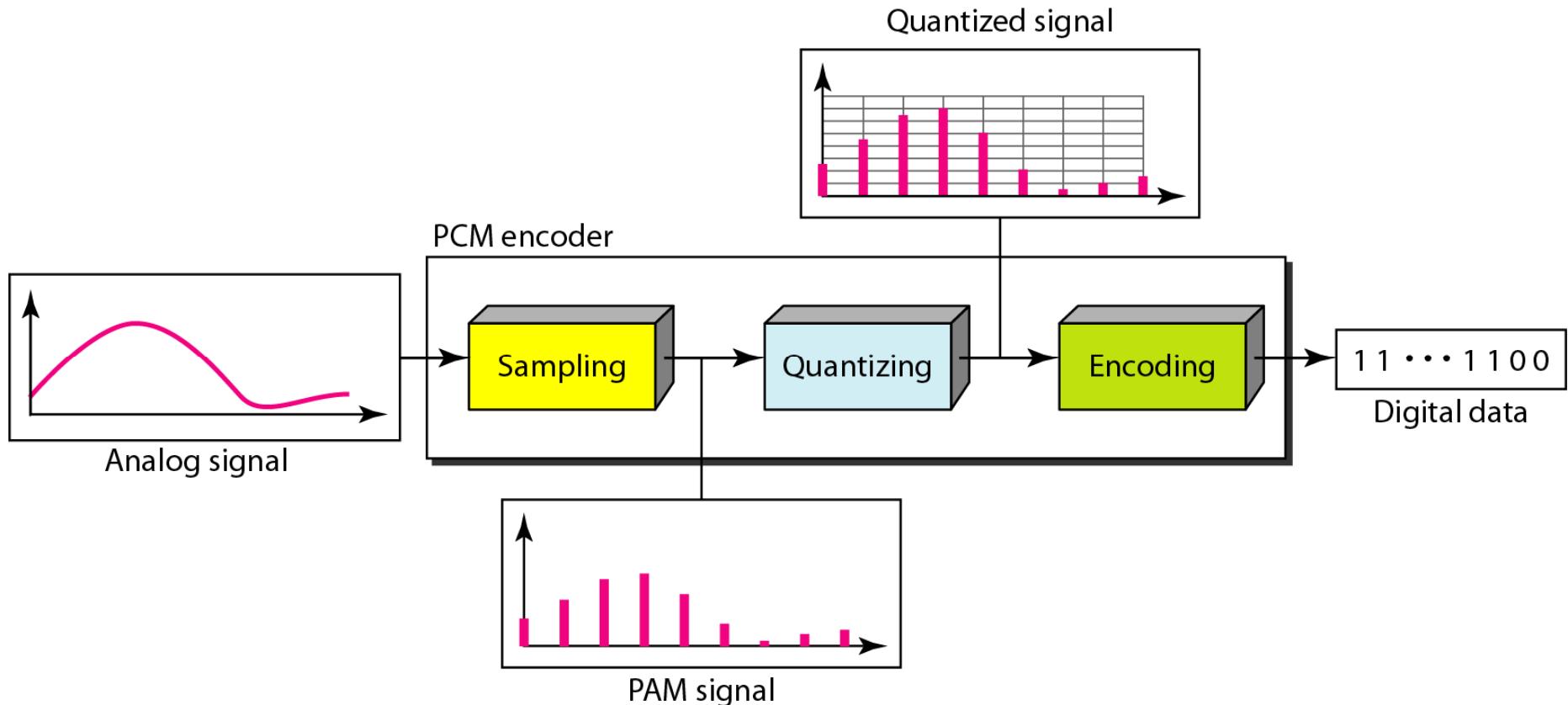
$$\text{Transmission time} = (5,000,000 \times 8) / 10^6 = 40 \text{ s}$$

The dominant factor is transmission time now.

4. Analog-to-digital Conversion

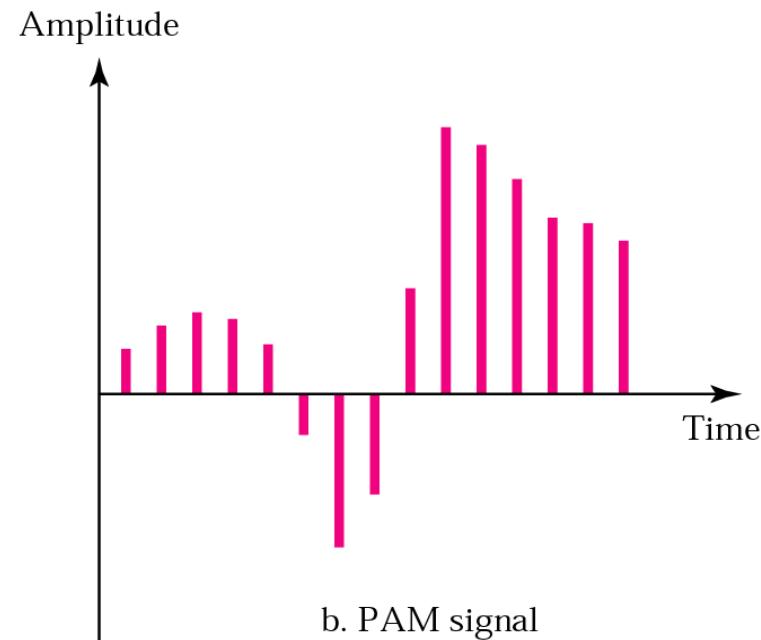
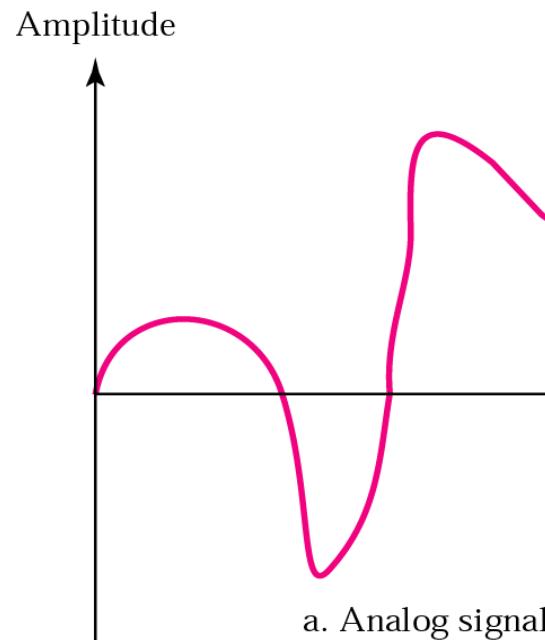
- ❖ Nowadays systems process digital data
- ❖ However, we may receive analog signal
 - ❖ Microphone, Camera, etc.
- ❖ Convert analog signal to digital data
 - ❖ Pulse Code Modulation (PCM)

Components of PCM encoder



Sampling - Pulse Amplitude Modulation (PAM)

- ❖ Analog signal is sampled every T_s
 - ❖ T_s is the sample interval
 - ❖ $f_s = 1/T_s$ is the sampling rate (or sampling frequency)





Note:

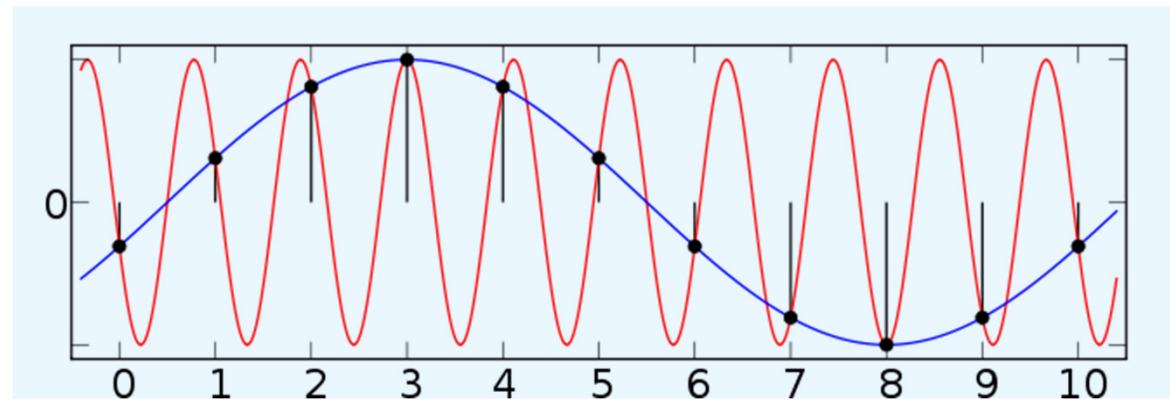
How many samples are sufficient?

According to the Nyquist theorem, the sampling rate must be at least 2 times the highest frequency contained in the signal

$$\text{Nyquist rate} = 2 \times f_{max}$$

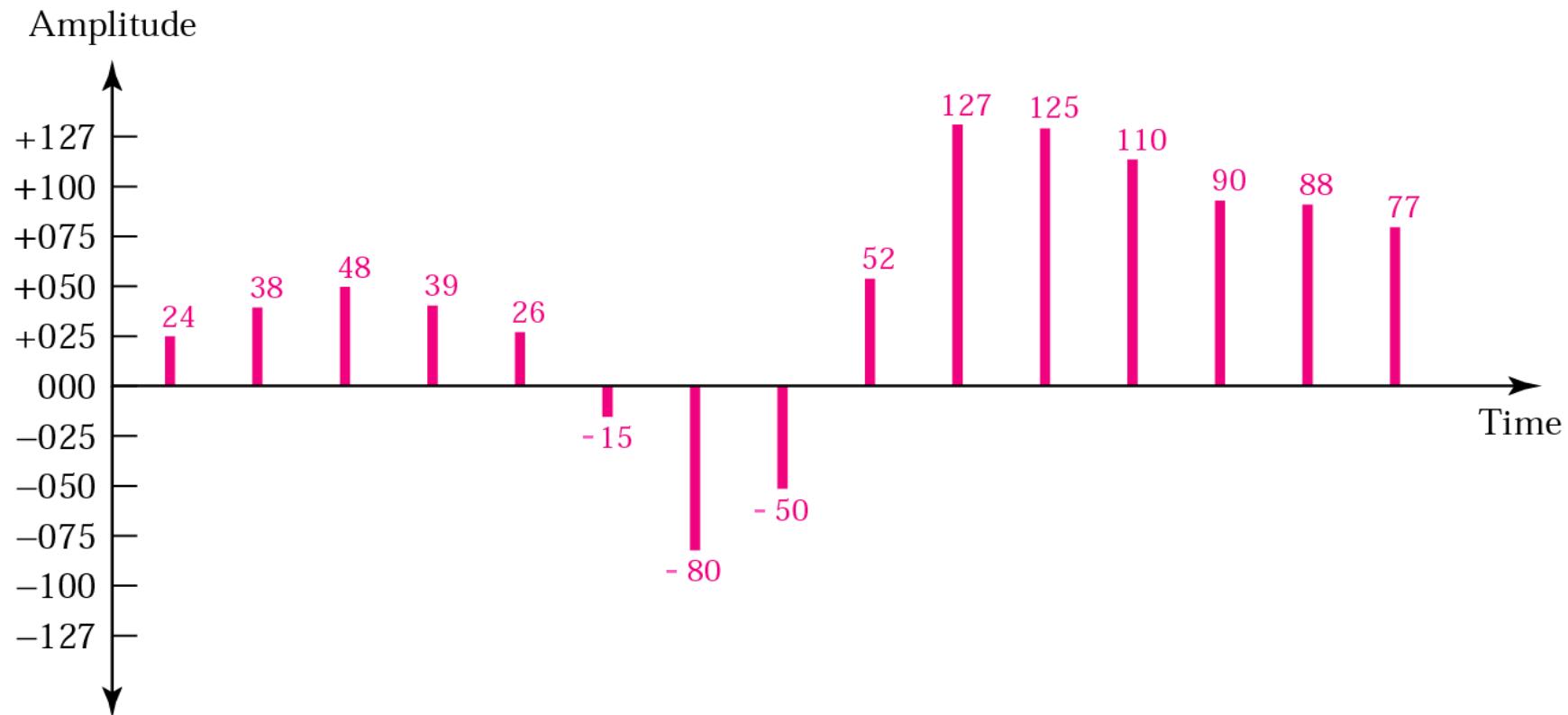
(to ensure the accurate reproduction of the original signal)

Undersampling will reproduce another signal with lower frequency



Quantization

- ❖ Assign quantized values to quantization levels
- ❖ Approximate the sample amplitude to the quantized values

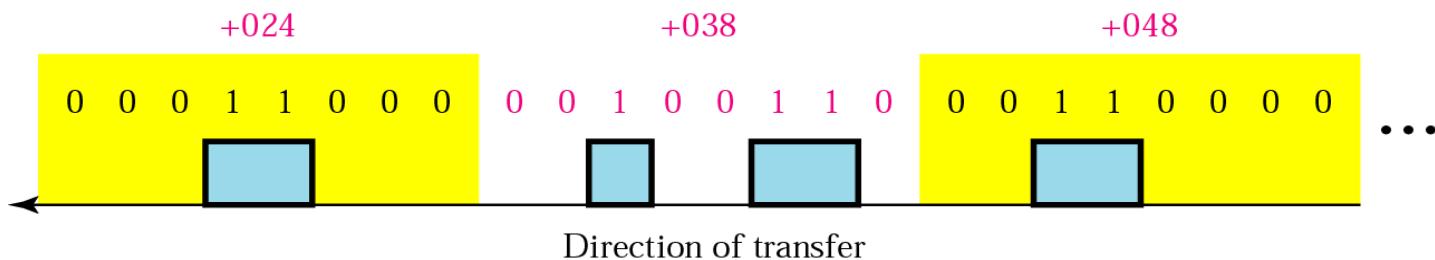


Represent the quantized samples by bits

- ❖ How many bits are required per sample?
- ❖ $n_b = \log_2 L$ (L is the number of quantization levels)

| | | | | | |
|------|----------|------|----------|------|----------|
| +024 | 00011000 | -015 | 10001111 | +125 | 01111101 |
| +038 | 00100110 | -080 | 11010000 | +110 | 01101110 |
| +048 | 00110000 | -050 | 10110010 | +090 | 01011010 |
| +039 | 00100111 | +052 | 00110110 | +088 | 01011000 |
| +026 | 00011010 | +127 | 01111111 | +077 | 01001101 |

Sign bit
+ is 0 - is 1



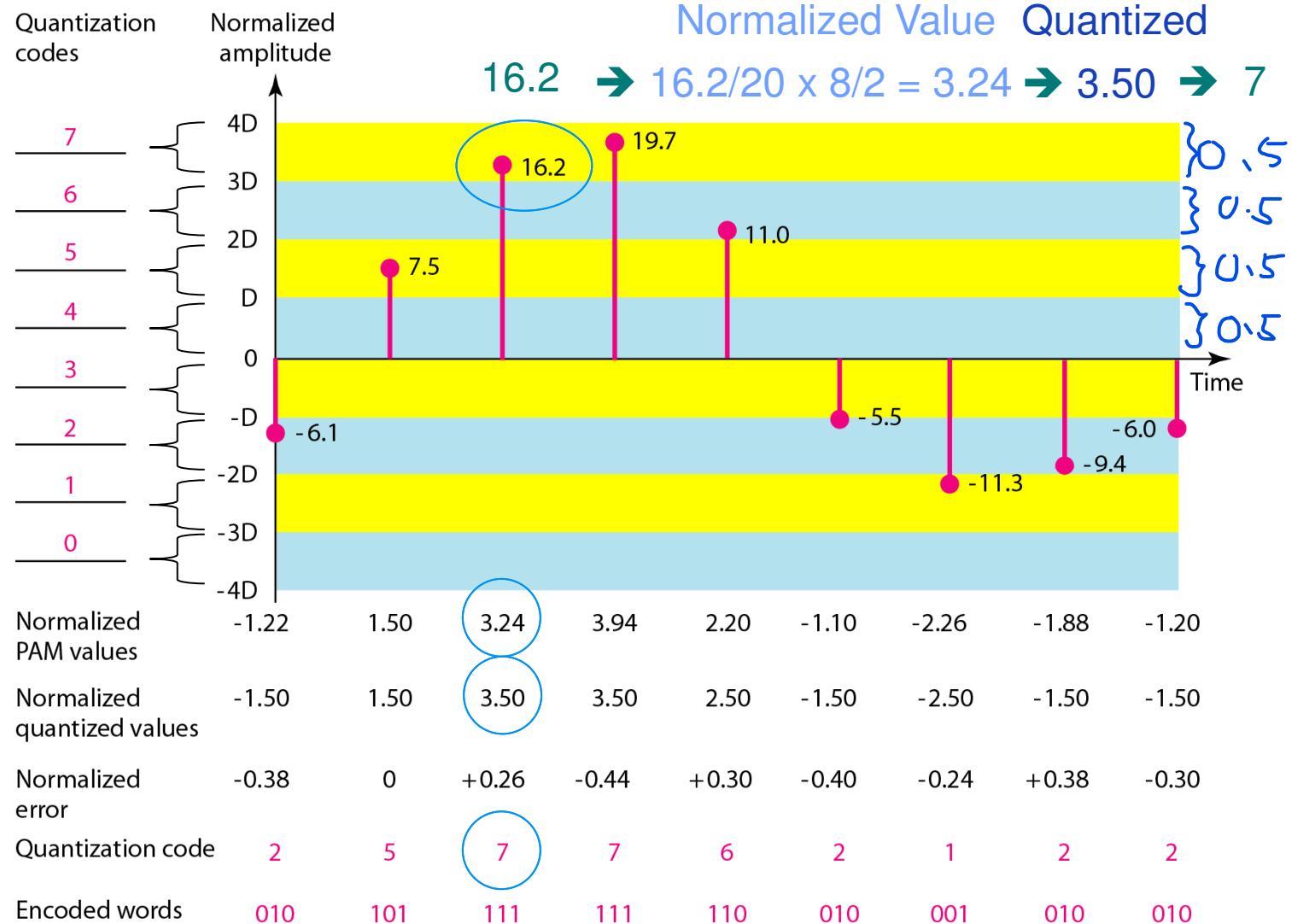
Quantization Error

- ❖ Quantization is an approximation process.
- ❖ The value of the difference of the actual value and the quantized value is the quantization error.
- ❖ This error is regarded as a noise.

- ❖ The signal-to-noise (in dB) ratio depends on the number of bits per sample (n_b), which is calculated in the following formula:

$$\text{SNR}_{\text{dB}} = 6.02n_b + 1.76 \text{ dB}$$

Quantization and encoding of a sampled signal



Example

What is the quantization error (SNR_{dB}) in the previous slide?

Solution

We can use the formula to find the quantization. We have eight levels and 3 bits per sample, so

$$SNR_{dB} = 6.02(3) + 1.76 = 19.82 \text{ dB}$$

Increasing the number of levels increases the SNR.

Example

We want to digitize the human voice. Assuming 8 bits per sample, what is the bit rate?

Solution

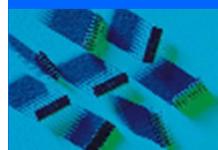
The human voice normally contains frequencies from 0 to 4000 Hz.

$$\text{Sampling rate} = 2 \times 4000 = 8000 \text{ samples/s}$$

$$\begin{aligned}\text{Bit rate} &= \text{sampling rate} \times \text{number of bits per sample} \\ &= 8000 \times 8 = 64,000 \text{ bps} = 64 \text{ Kbps}\end{aligned}$$

Summary

- ❖ **Analog Signal and Digital Signal**
 - ❖ Periodic and aperiodic
 - ❖ Time domain and frequency domain
- ❖ **Transmission Impairment**
 - ❖ Attenuation, Distortion, Noise
- ❖ **Data rate limit**
 - ❖ Nyquist bit rate (for noiseless channel)
 - ❖ Shannon capacity (for noisy channel)
- ❖ **Analog-to-digital Conversion (PCM)**
 - ❖ Sampling, quantization
- ❖ **Revision Quiz**
 - ❖ http://highered.mheducation.com/sites/0073376221/student_view0/chapter3/quizzes.html



Lecture 3 Data link - *Error Detection and* *Correction*

Textbook: Ch.9 and Ch.10

Main Topics

- ❖ Ch 9 Data Link Layer
 - ❖ 9.1 Nodes and Links
- ❖ Ch 10 Error Detection and Correction
 - ❖ 10.1 Error Detection and Correction
 - ❖ Types of Errors
 - ❖ 10.2 Linear Block Codes
 - ❖ Parity Check
 - ❖ 10.3 Cyclic Codes
 - ❖ Cyclic Redundancy Check

Data-link layer

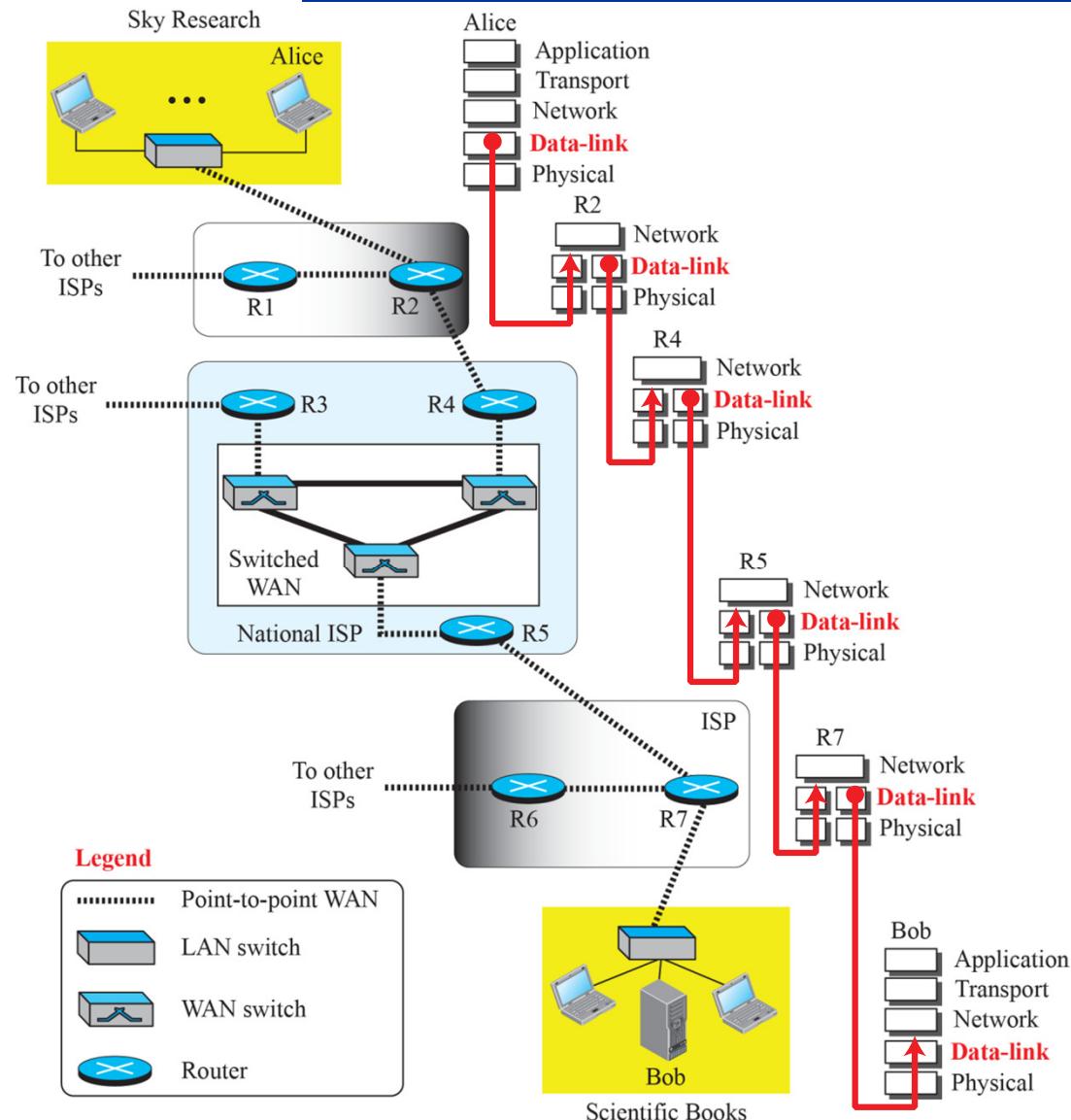
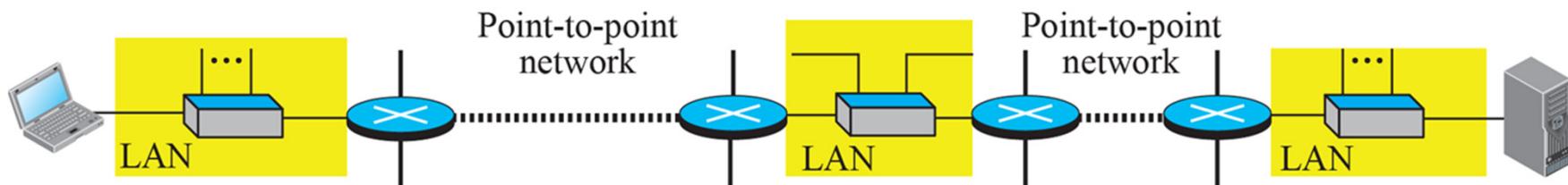


Figure 9.1: Communication at the data-link layer

9.1 Nodes and Links

- Communication at the data-link layer is ***node-to-node***.
- It is customary to refer to the two *end hosts* and the *routers* as ***nodes*** and the *networks* in between as ***links***.



a. A small part of the Internet



b. Nodes and links

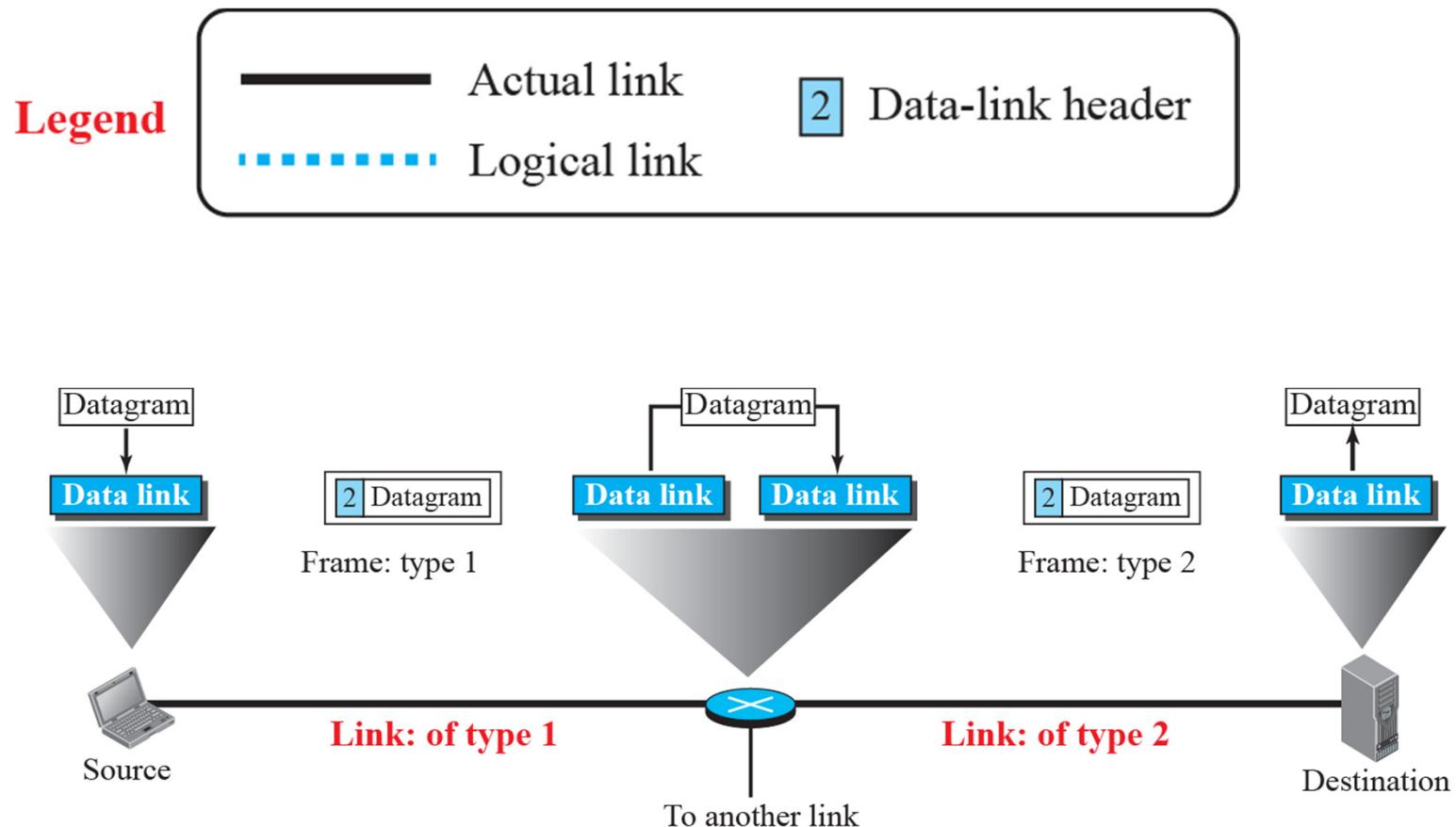
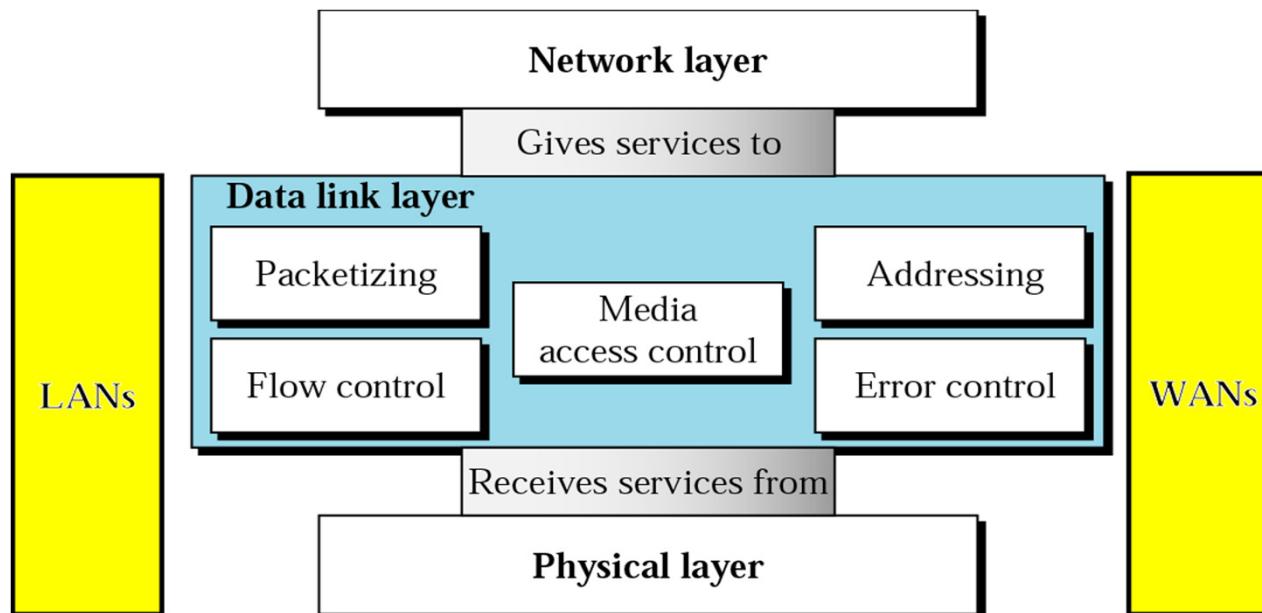


Figure 9.3: A communication with only three nodes

Service of the data-link layer

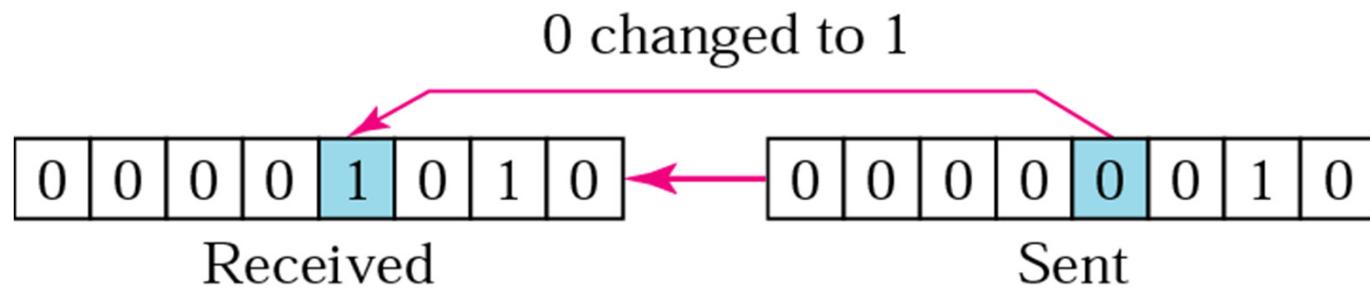
- ❖ The data-link layer is located between the physical and the network layers.
- ❖ The data-link layer provides services to the network layer; it receives services from the physical layer



Ch 10 Error Detection and Correction

10.1 Types of Errors

1. Single-bit error

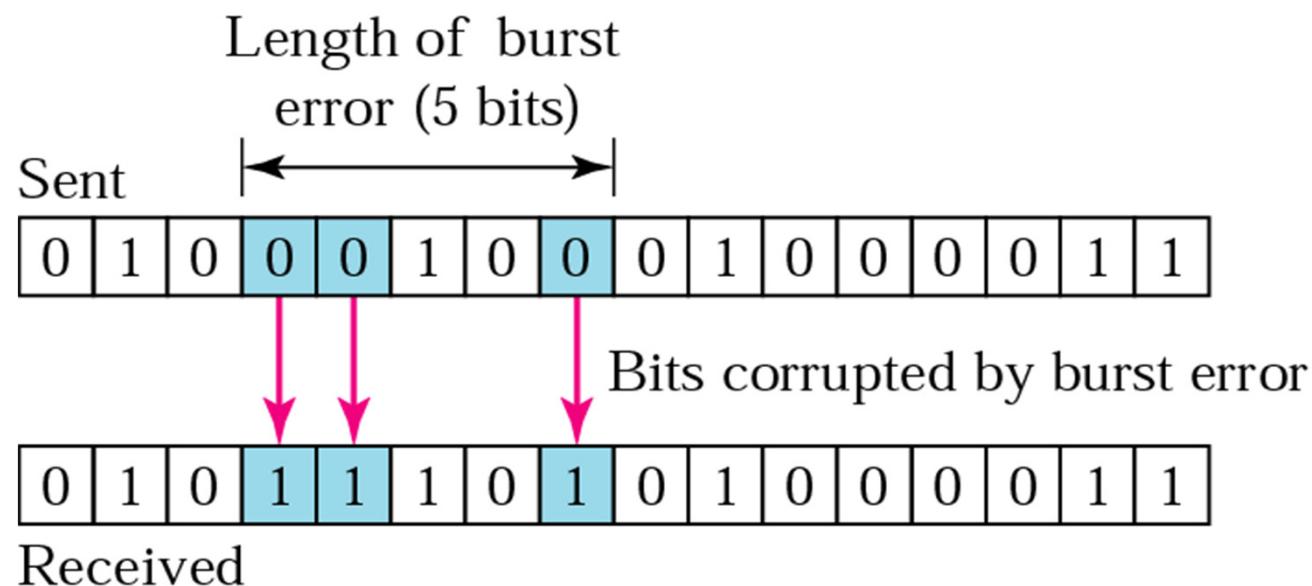


In a single-bit error, only one bit in the data unit has changed.

Types of Errors

2. Burst error

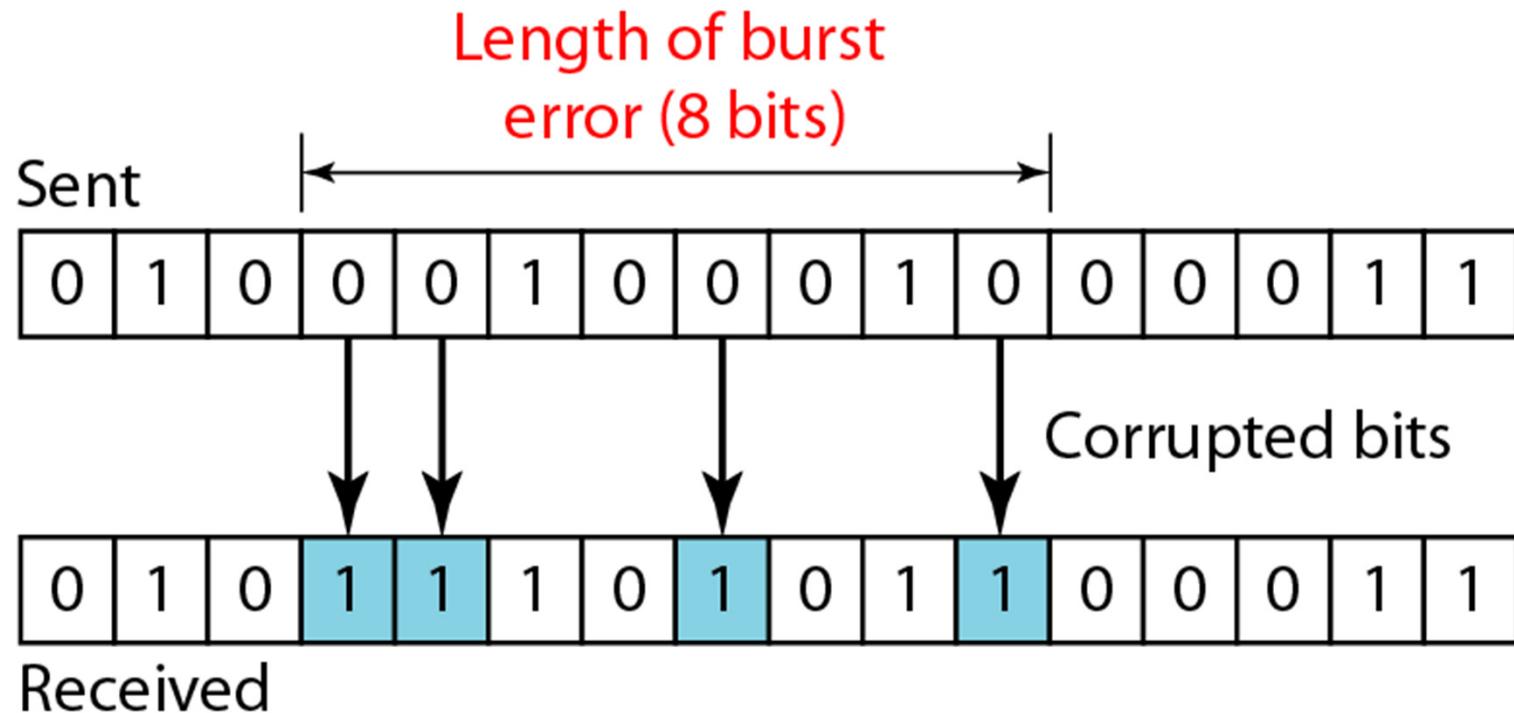
A burst error means that 2 or more bits in the data unit have changed.



Burst Error

- ❖ Burst Error - the *string* of bits between 2 successive corrupted bits including the two bits in error
 - ❖ More likely to happen due to noise duration is usually longer than the duration of 1 bit
- ❖ Burst error length B - the last corrupted bit in a burst and the first corrupted bit in the following burst must be separated by $B+1$ or more correct bits

Burst error of length 8



Assume the remaining bits are correct

Error Detection

Redundancy

- Parity Check
- Cyclic Redundancy Check (CRC)

Error detection uses the concept of redundancy, which means adding extra bits for detecting errors at the destination.

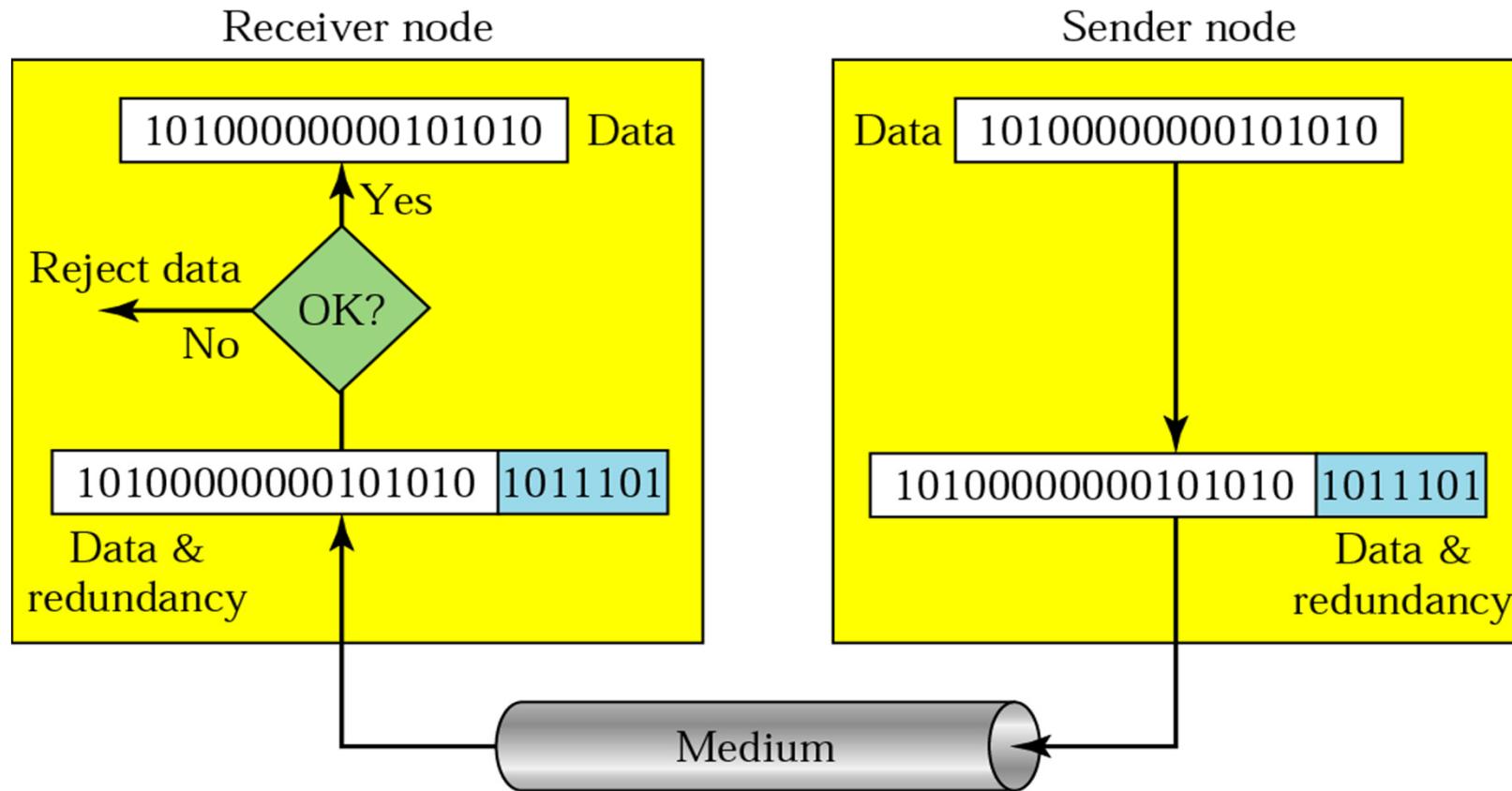
Error Detection

(Feedback Error Control)

- ❖ Each character or frame includes only sufficient additional information to enable the receiver to **detect** errors
- ❖ a **retransmission** control scheme is used
- ❖ the predominant method because of less additional bits required

Redundancy

- ❖ *Adding extra bits for detecting errors at the destination*



10.2 Linear Block Code – Parity Check Method

- ❖ Most common and simple for character-oriented transmission
- ❖ The data bits in each character are inspected prior to transmission and the **parity bit** is computed
- ❖ The parity bit is then added so that the total number of binary 1s is either odd or even:
 - ☞ If **odd parity** is used, no. of 1s is odd.
 - ☞ If **even parity** is used, no. of 1s is even.

Parity Check Method

(Sender side)

- ❖ Sender obtains data from upper layer
- ❖ Determine the parity bit (either 0 or 1) so that the total number of binary 1s is either odd or even:
 - ❖ If **odd parity** is used, no. of 1s is odd.
 - ❖ If **even parity** is used, no. of 1s is even.
- ❖ Add the parity bit to data and send it out

Parity Check Method

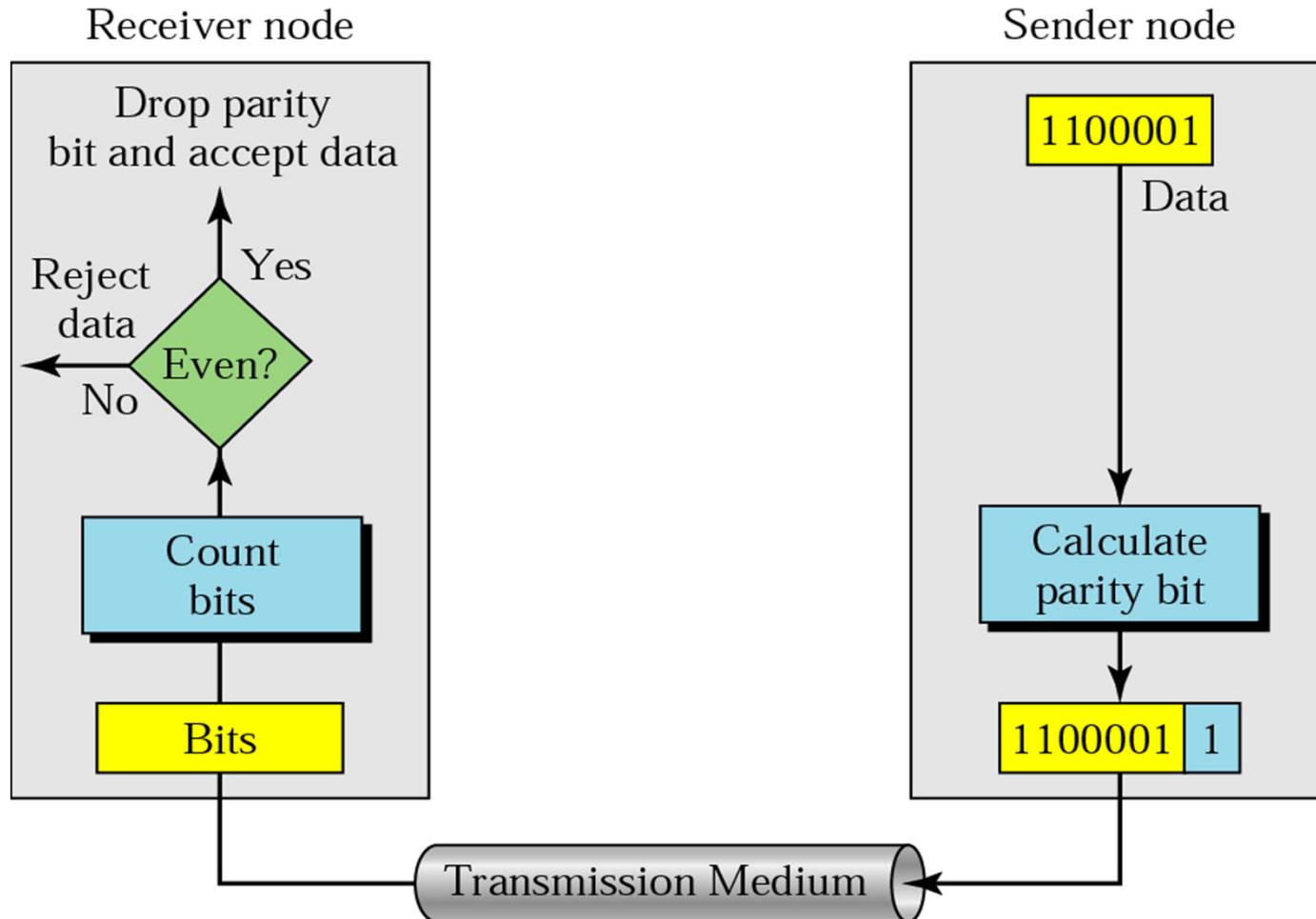
(Receiver side)

- ❖ Receiver receives bits from lower layer
- ❖ Count the number of 1s
- ❖ Error is detected if the number does not match, i.e.
 - ❖ Even number of 1s in **odd parity**
 - ❖ Odd number of 1s in **even parity**

Limitation

- ❖ Single parity bit is used
- ❖ Can only detect burst errors with odd number of error bits
- ❖ If even number of bits are corrupted, the errors cannot be detected

Even-parity concept



Example

Suppose the sender wants to send the word *world*. In 7-bit ASCII the five characters are coded as

1110111 1101111 1110010 1101100 1100100

The following shows the actual bits sent using even parity:

11101110 11011110 11100100 11011000 11001001

Example (no error)

Now suppose the word *world* in Example 1 is received by the receiver without being corrupted in transmission.

11101110 11011110 11100100 11011000 11001001

The receiver counts the number of 1s in each character and comes up with even numbers (6, 6, 4, 4, 4).

The data are accepted.

Example (with error)

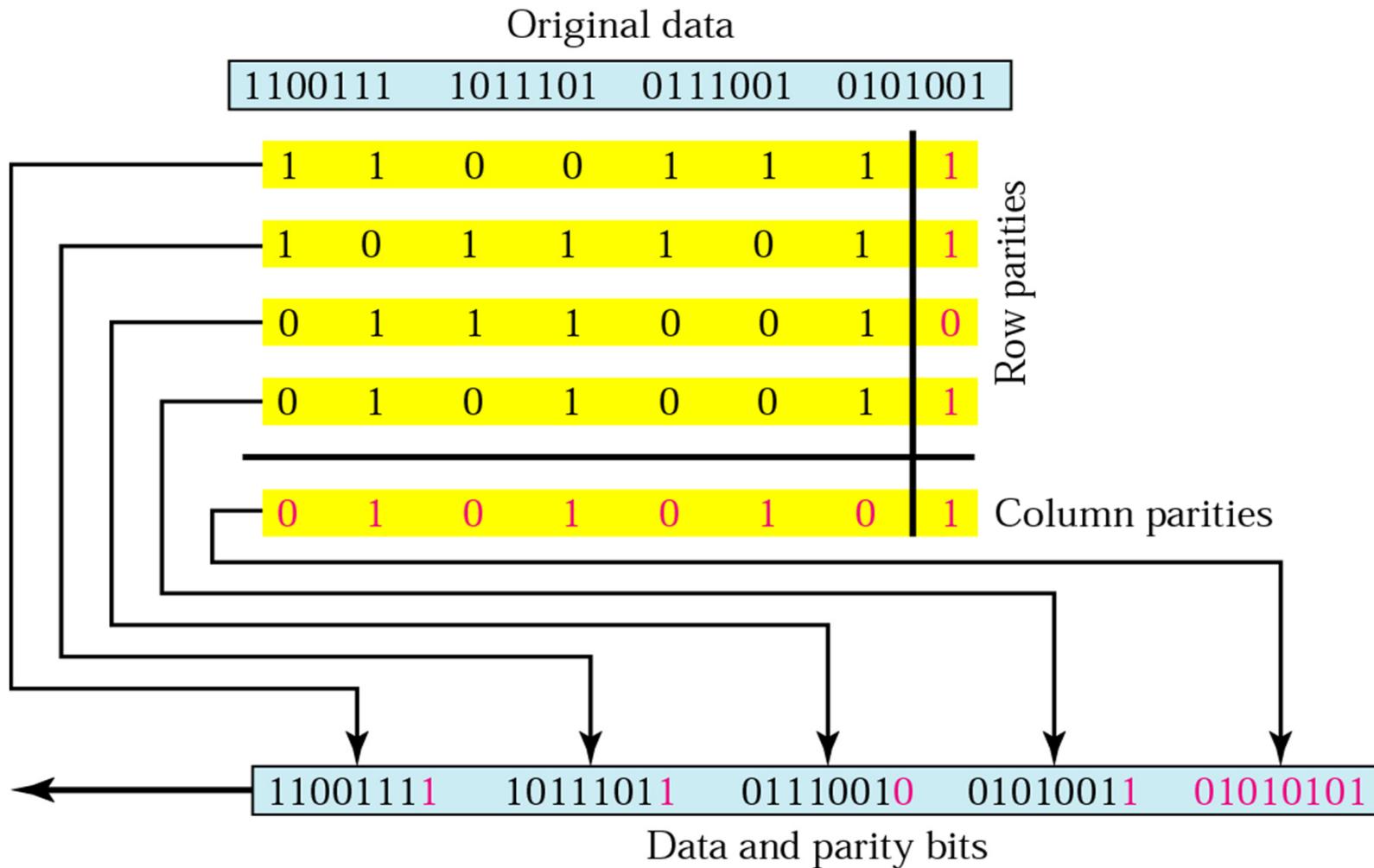
Now suppose the word *world* in Example 1 is corrupted during transmission.

1111110 1101110 11101100 11011000 11001001

The receiver counts the number of 1s in each character and comes up with even and odd numbers (7, 6, 5, 4, 4).

The receiver knows that the data are corrupted, discards them, and asks for retransmission.

Two-dimensional parity check



Two-dimensional parity check

- ❖ Organize the bits in rows and columns
- ❖ Calculate the parity bit for each row
 - ❖ same as the single-bit parity check method
 - ❖ odd parity or even parity
- ❖ Calculate the parity bit for each column
 - ❖ also for the parity bit of each row

Example

Suppose the following block is sent:

10101001 00111001 11011101 11100111 10101010

However, it is hit by a noise with 8-bit duration, and some bits are corrupted.

10100011 10001001 11011101 11100111 10101010

When the receiver checks the parity bits, some of the bits do not follow the even-parity rule and the whole block is discarded.

10100011 10001001 11011101 11100111 10101010

What kind of errors can (/cannot) be detected by two-dimensional parity check?

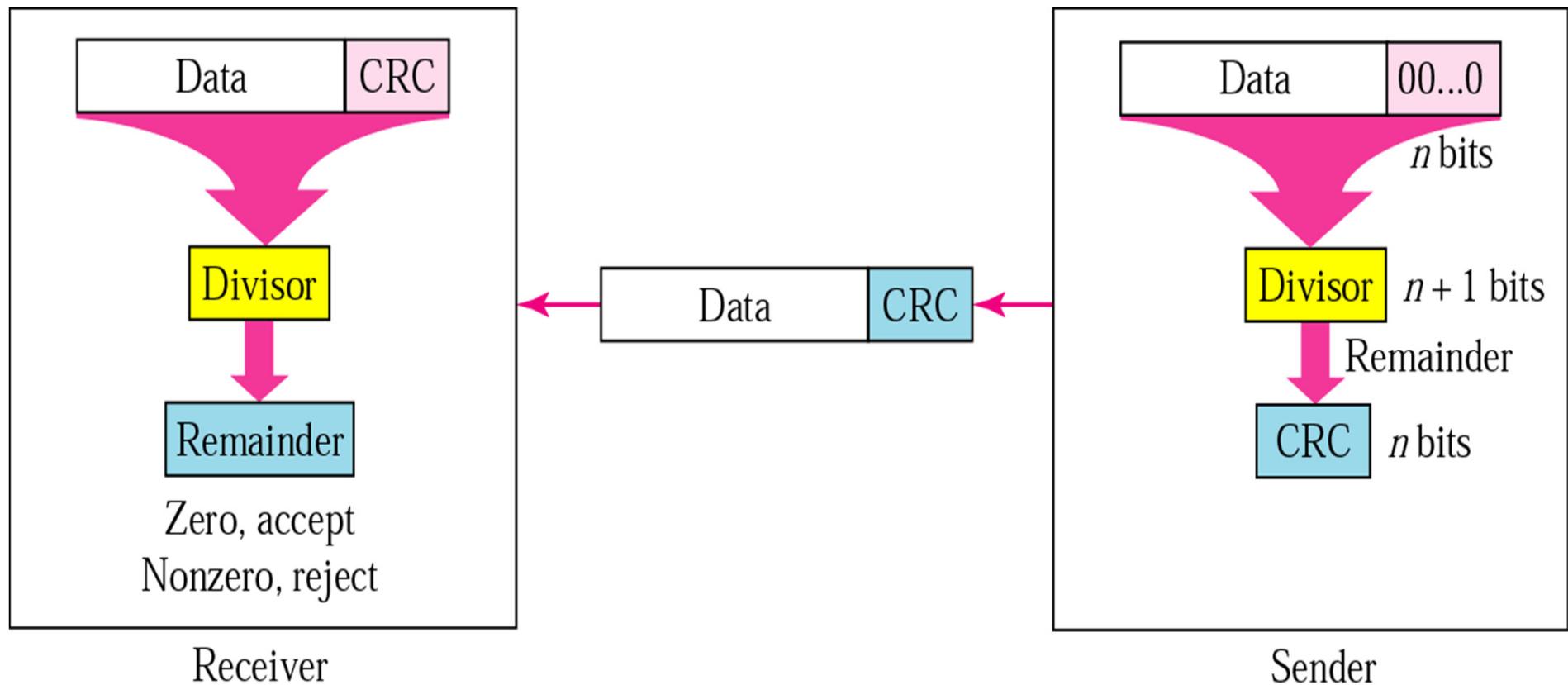
10.3 Cyclic Codes – Cyclic Redundancy Check (CRC)

- ❖ Parity check does not provide a reliable detection against error bursts
- ❖ A single set of check digits is computed and appended to the tail of each frame transmitted
- ❖ The receiver then performs a similar computation to check whether error occurs

Cyclic Redundancy Check (CRC)

- ❖ The number of check digits varies (16, 32 are most common)
- ❖ The computed check digits are called cyclic redundancy check (CRC) digits or frame check sequence (FCS) digits
- ❖ Uses property of modulo 2 arithmetic
 - ❖ uses binary addition with no carries and is effectively an XOR operation.

CRC generator and checker



CRC Procedure

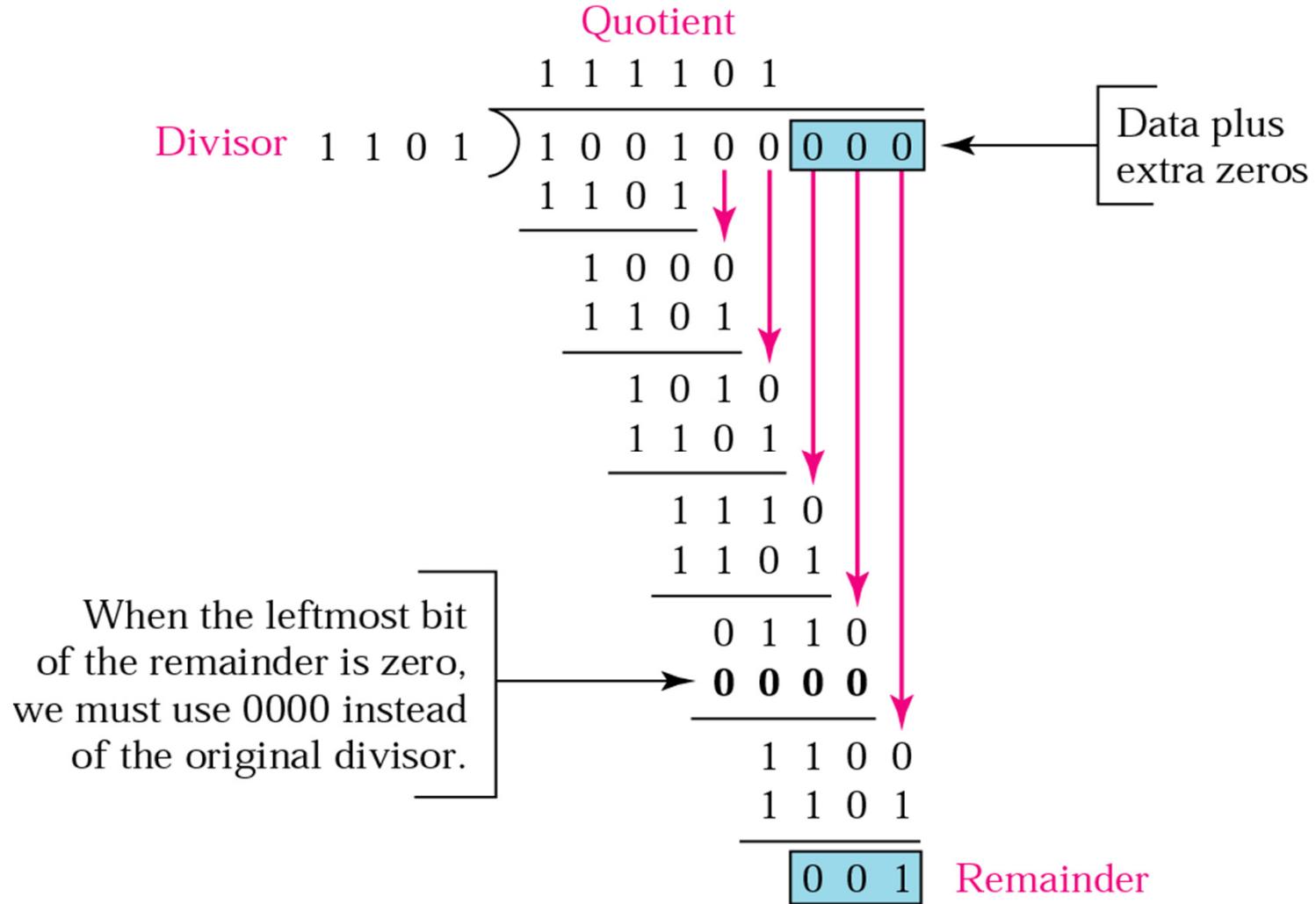
1. A set of n “0”s is appended at the end of a k -bit data frame M .
 - This is the same as multiplying the frame by 2^n so we have $(M \times 2^n)$
 - $k > n$

2. $(M \times 2^n)$ is divided modulo 2 by a $(n+1)$ -bit binary number G (the generator polynomial), giving the remainder R (n -bit) which is the CRC (or FCS) digits

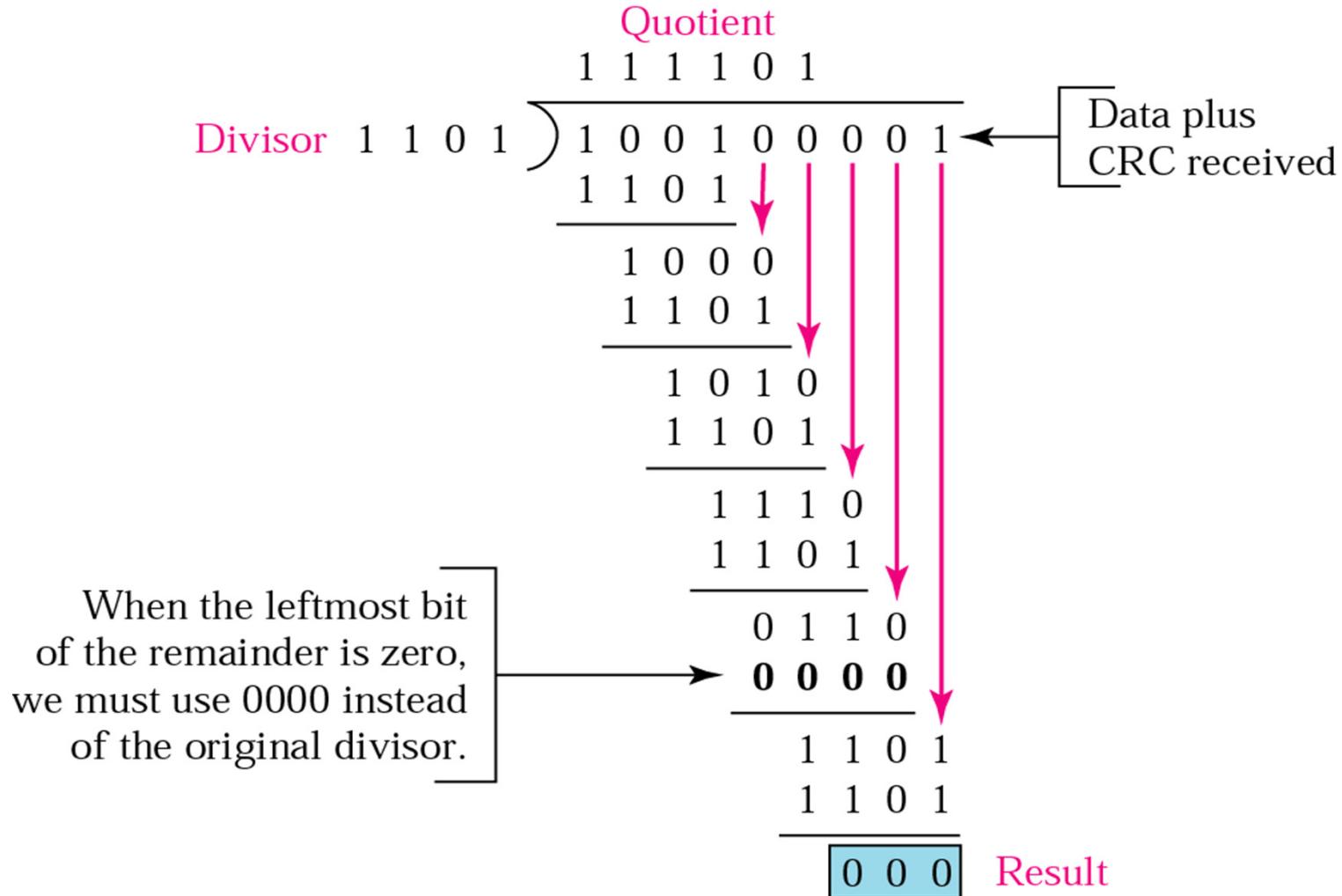
CRC Procedure

3. R is appended at the end of M and then transmit M & R together
4. At the receiver, the received bit stream including M & R is again divided by the same polynomial G, i.e. $(M \times 2^n + R)/G$
5. If the remainder = 0 then no errors;
otherwise errors occur

Binary division in a CRC generator



Binary division in CRC checker



CRC Example

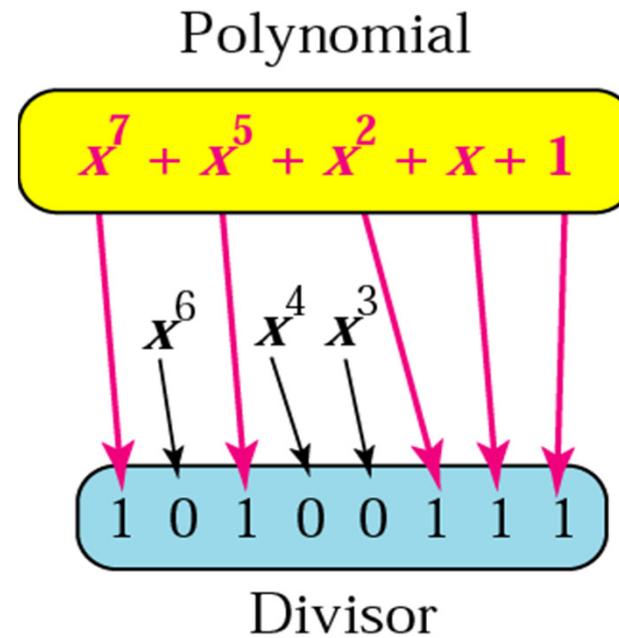
Message: 11100110

Polynomial: 11001 (i.e. $x^4 + x^3 + 1$)

What is the CRC?

- ❖ Answer: CRC should be 0110
- ❖ Division is equivalent to performing the XOR operation bit by bit in parallel as each bit in the frame is processed
- ❖ Could be implemented in hardware

A polynomial representing a divisor



The polynomial is of degree 7 but need 8 bits

Standard CRC

- ❖ Represent the generator polynomial by showing those positions that are binary 1 as powers of X
- ❖ $\text{CRC-CCITT} = X^{16} + X^{12} + X^5 + 1(X^0)$
this is equivalent to 10001000000100001
- ❖ Also call CRC-16
- ❖ it will detect all error bursts of length less than or equal to 16 bits.

Standard CRC

- ❖ CRC-CCITT (CRC-16) is used extensively with WANs
- ❖ CRC-32 is used in most LANs
- ❖ In general, a $(n+1)$ -bit generator polynomial, *with degree n*, will detect:
 - ❖ all single-bit, double-bit and odd number of bit errors
 - ❖ all error bursts with length $\leq n$
 - ❖ most error bursts with length $> n$

Standard polynomials

| Name | Polynomial | Application |
|--------|---|-------------|
| CRC-8 | $x^8 + x^2 + x + 1$ | ATM header |
| CRC-10 | $x^{10} + x^9 + x^5 + x^4 + x^2 + 1$ | ATM AAL |
| CRC-16 | $x^{16} + x^{12} + x^5 + 1$ | HDLC |
| CRC-32 | $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ | LANs |

Example

The CRC-12

$$x^{12} + x^{11} + x^3 + x + 1$$

which has a degree of 12, will detect all burst errors affecting an odd number of bits, will detect all burst errors with a length less than or equal to 12, and will detect 99.97 percent of burst errors with a length of 13 or more.

Error Correction (Forward Error Control)

- ❖ Each transmitted character or frame contains additional (redundant) information so that the receiver can **detect, locate & correct** the errors
 - ❖ e.g. Hamming code

Forward Error Control (FEC)

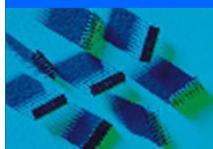
- ❖ With FEC, sufficient additional check digits are added to each transmitted message to enable the receiver
 - ❖ to **detect** the presence of one or more errors in a received message
 - ❖ and to **locate** the position of the error
- ❖ Correction is achieved simply by inverting the incorrect bit(s)

Forward Error Control (FEC)

- ❖ Number of check digits is much higher than that for just error detection
- ❖ Less efficient than retransmission with error detection
- ❖ But when the round-trip delay is long or return path is not available, FEC methods are often used

Summary

- ❖ Single-bit errors and burst errors
- ❖ Error detection by redundancy
- ❖ Parity Check
 - ❖ Odd parity and even parity
 - ❖ Two-dimentional Parity Check
- ❖ CRC / FCS
 - ❖ n-bit check digits detect burst errors $\leq n$ bits
- ❖ FEC
 - ❖ More bits are used for error location and correction
- ❖ Revision Quiz
 - ❖ http://highered.mheducation.com/sites/0073376221/student_view0/chapter_10/quizzes.html



Lecture 4 *Data Link Control & Protocols*

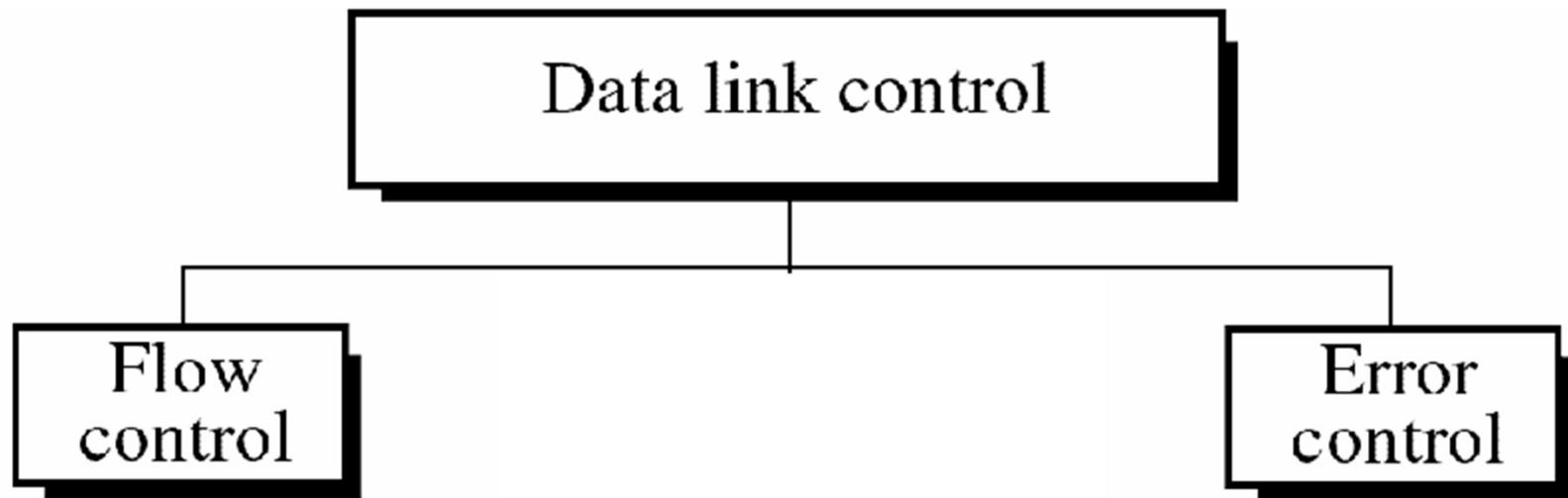
Textbook: Ch.11

Main Topics

- ❖ **11.1 Flow And Error Control**
- ❖ **11.2 Stop-and-Wait**
 - ❖ Implicit Retransmission and ARQ
 - ❖ Error-control
 - ❖ Piggybacking
- ❖ **11.3 Framing with HDLC**
 - ❖ High-level Data Link Control
- ❖ **11.4 Bit stuffing**

11.1

Data Link Control



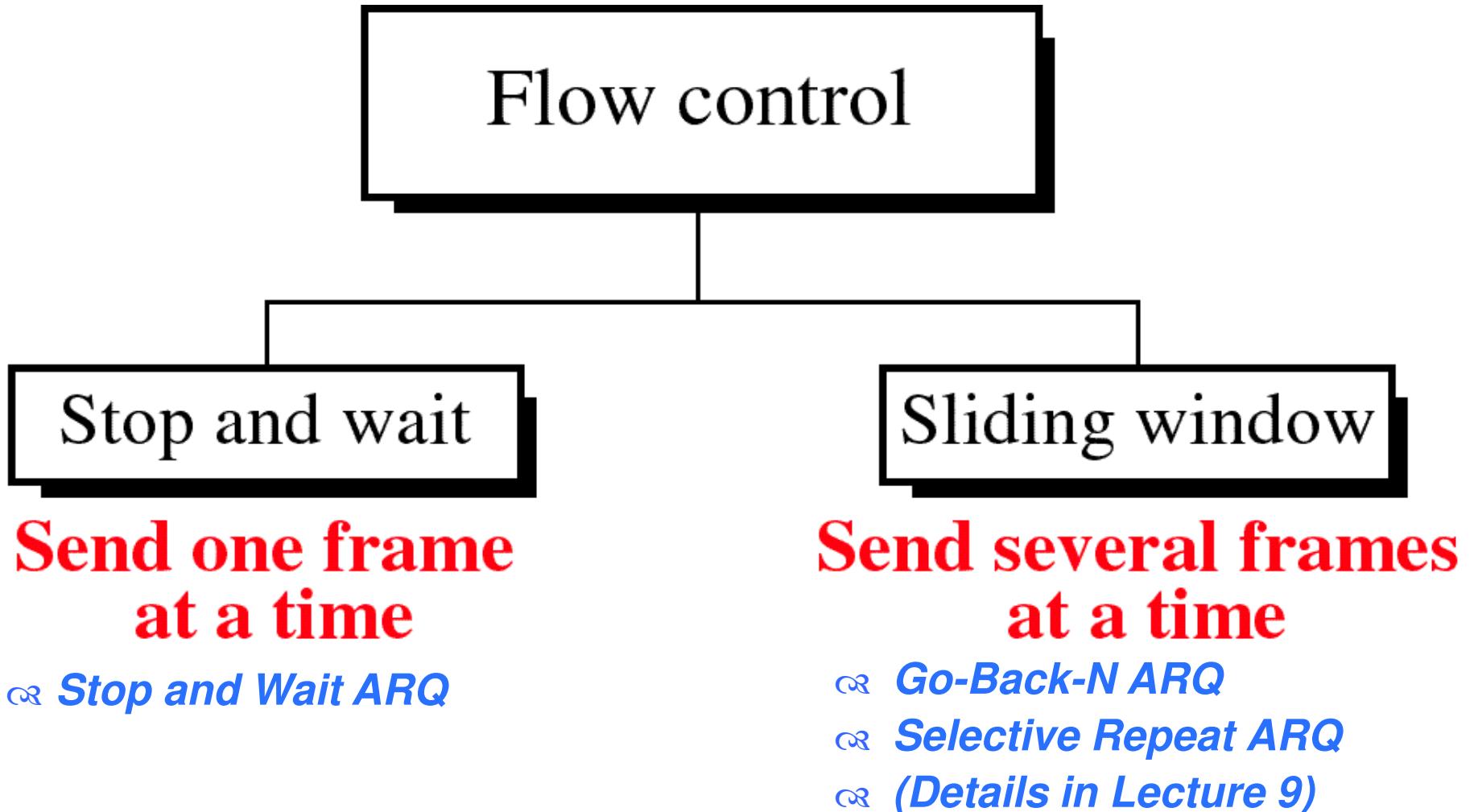
**How much data
may be sent?**

**How can errors
be corrected?**

Flow Control

- ❖ Balance between the sending rate and receiving rate
 - ❖ If sender transmits faster than the receiver can handle – data lost
 - ❖ If sender transmits too slow, receiver has to wait – less efficient
- ❖ Flow control is related to the first issue
 - ❖ Prevent data lost
 - ❖ Sender waits for acknowledgement (ACK) from receiver

Flow control Mechanisms



Error Control

- ❖ Error detection by CRC or FCS
- ❖ Error correction by retransmission
 - ❖ If error is detected, a negative acknowledgment (NAK) is returned and the specified frames are resent.
 - ❖ If no error, receiver sends acknowledgment (ACK) to sender, sender sends next frame
 - ❖ If no ACK is received after a period of time, sender retransmits

Error Control

- ❖ Automatic Repeat Request (ARQ)
 - ❖ does not use NAK
- ❖ Implicit retransmission in ARQ
 - ❖ Receiver discards the error frame and does nothing
 - ❖ Sender interprets the absence of an ACK (after a timeout) as an indication that the previous frame was corrupted or lost

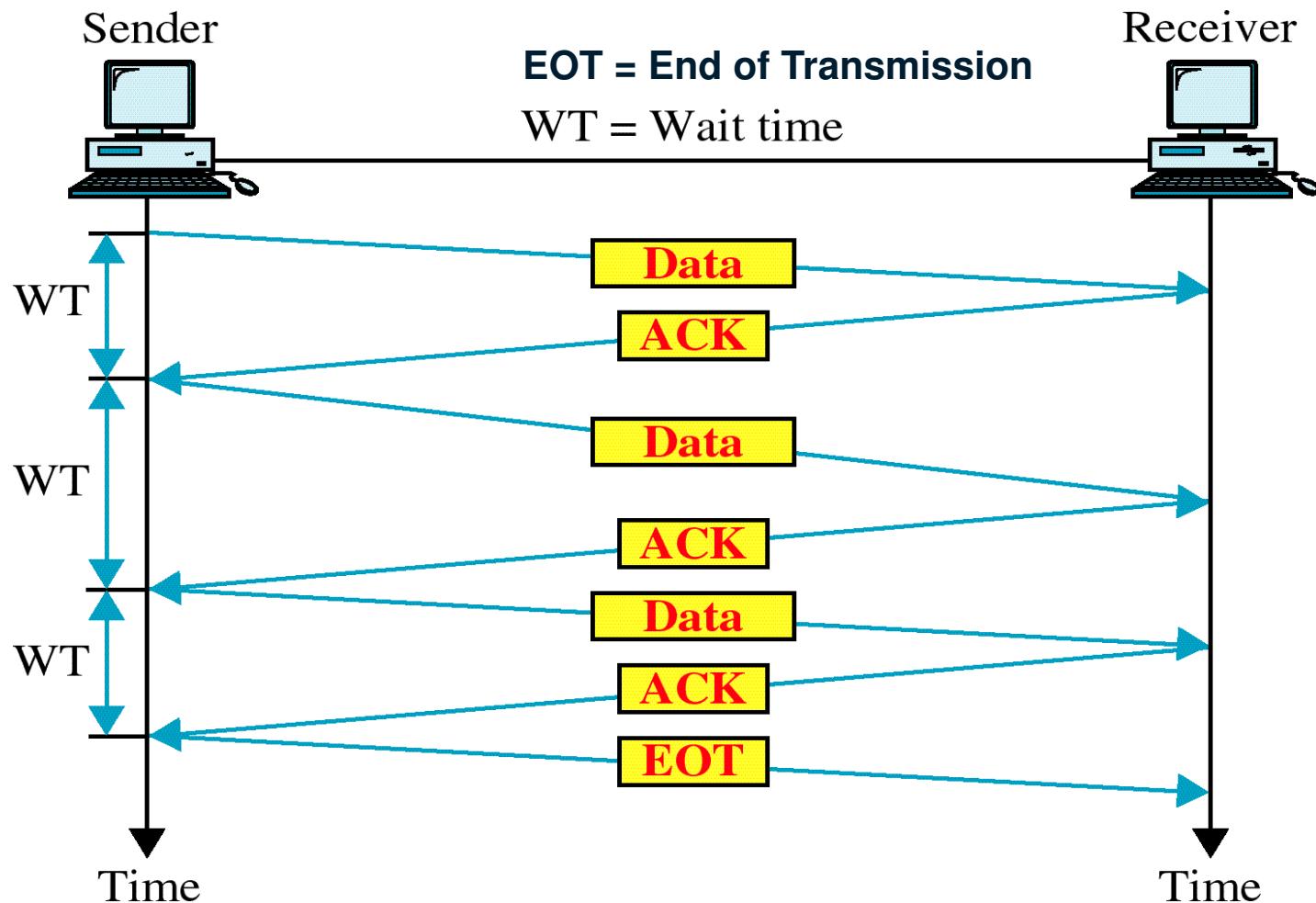
Flow control and Error control

- ❖ Can be combined
- ❖ Acknowledgement is used in both control
 - ❖ Sender waits for ACK to transmit the next frame
 - ❖ Receiver uses ACK to confirm no error
 - ❖ Sender retransmits if no ACK is received
- ❖ Stop-and-Wait ARQ
 - ❖ The simplest protocol for flow and error control

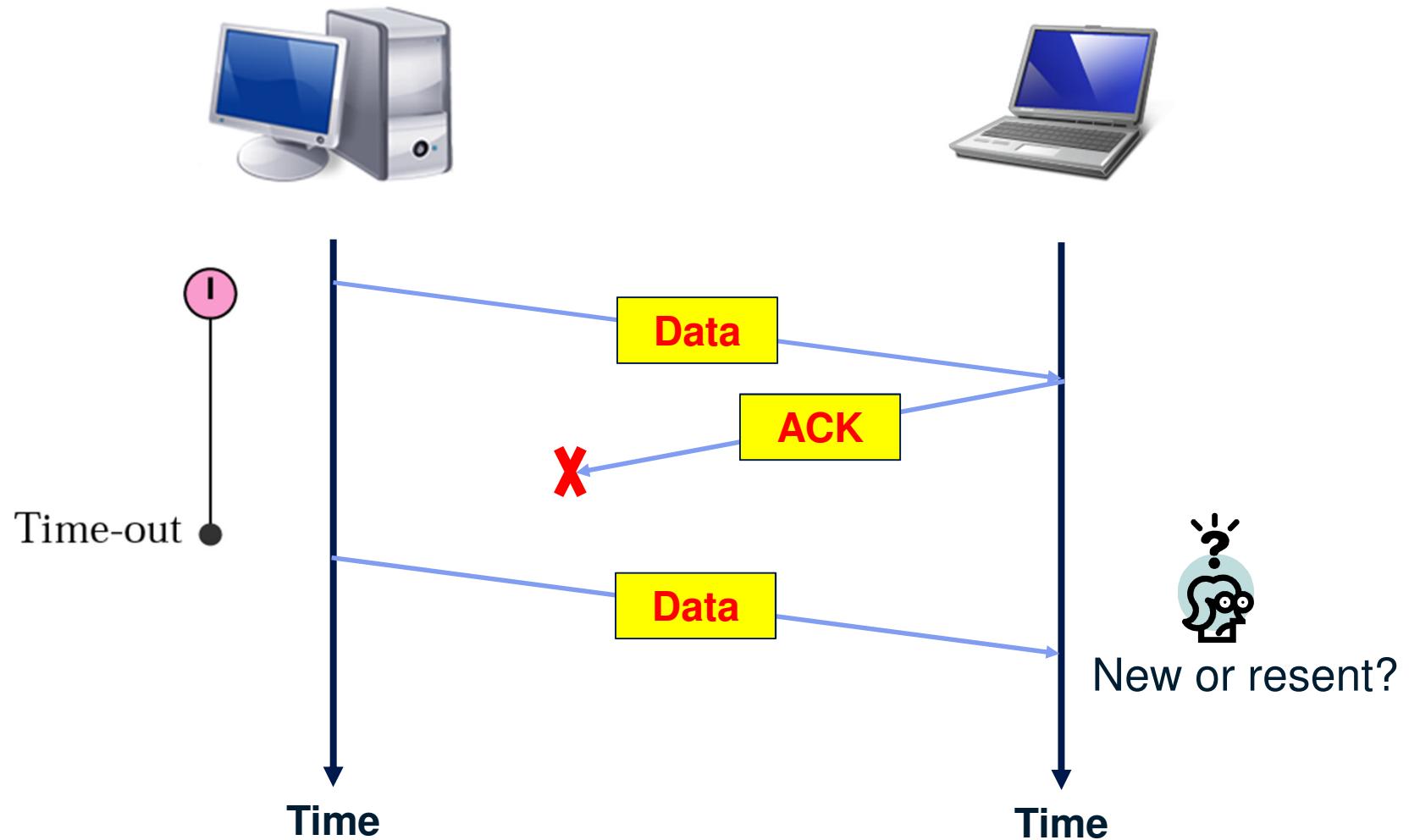
11.2 Stop-and-Wait ARQ

- ❖ The sender sends one frame and waits for an ACK before sending the next frame
- ❖ If no ACK is received after a period of time (timeout), the sender retransmits
 - ❖ Implicit retransmission in ARQ
- ❖ Advantage: Simple
- ❖ Disadvantage: Inefficient

Normal Situation



If ACK is lost...

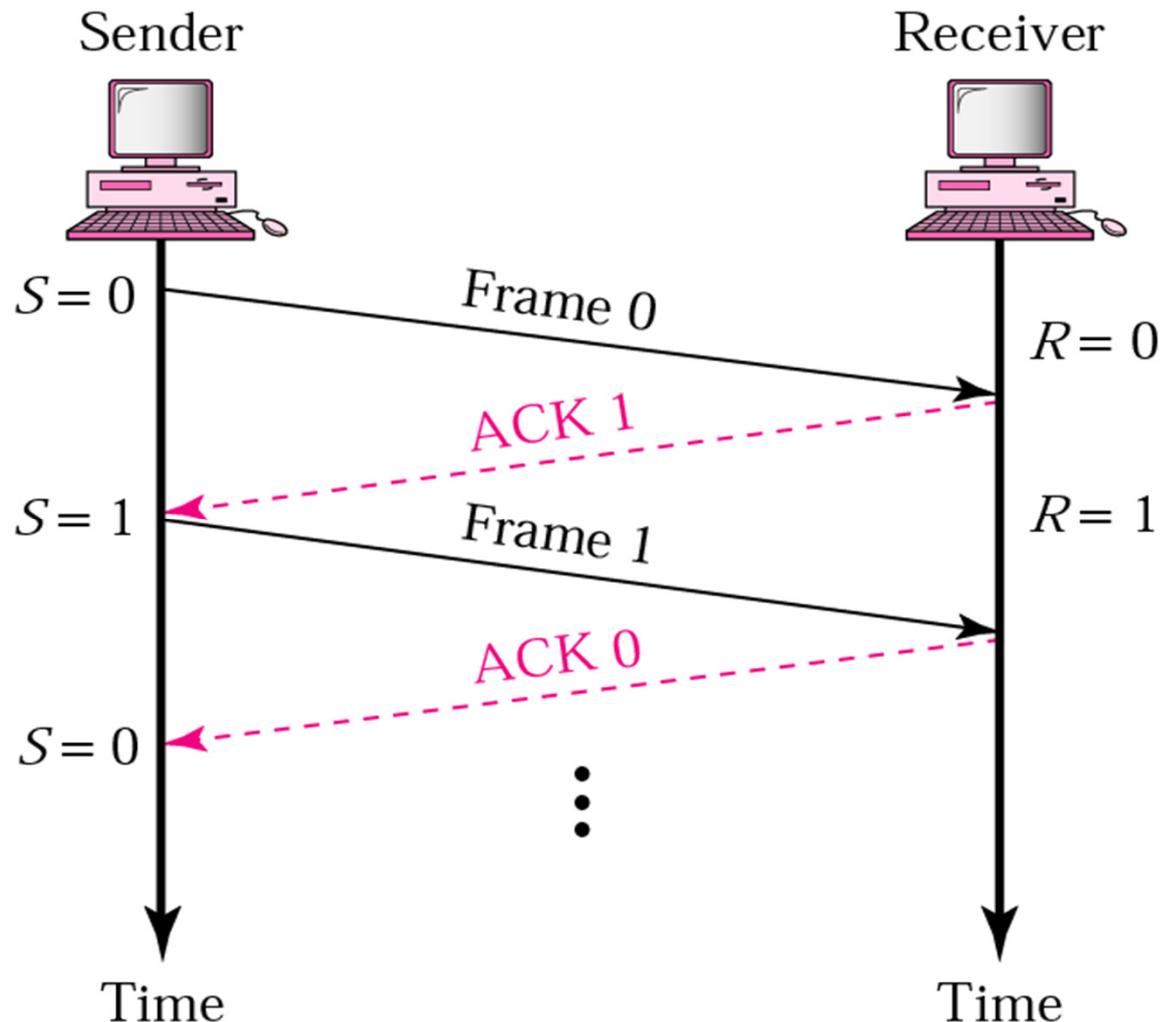


Sequence Number

- ❖ Use 1 bit sequence number to distinguish the frame is newly transmitted or resent of previous frame

- ❖ ACK confirms the correct receive of frame
- ❖ ACK also contains the sequence number of the expected frame
 - ❖ Sender knows what frame the receiver is expecting

Normal Operation



Normal Operation

- ❖ Sender can have only ***one frame ready*** to send at a time
- ❖ When sender initiates a transmission of a frame, it starts a ***timer***
- ❖ If the frame is received without error, the receiver sends ***ACK***
- ❖ If sender receives ACK, it sends ***another frame***

Implicit Retransmission

- ❖ Receiver discards the frame if it contains **errors**
 - ❖ No ACK is sent
- ❖ If sender does not receive an ACK within a predefined **time-out interval**, it retransmits the frame in the buffer
- ❖ Receiver checks the frame identifier (sequence number)
 - ❖ Accept if it is the expecting frame
 - ❖ Discard if the frame has been correctly received previously

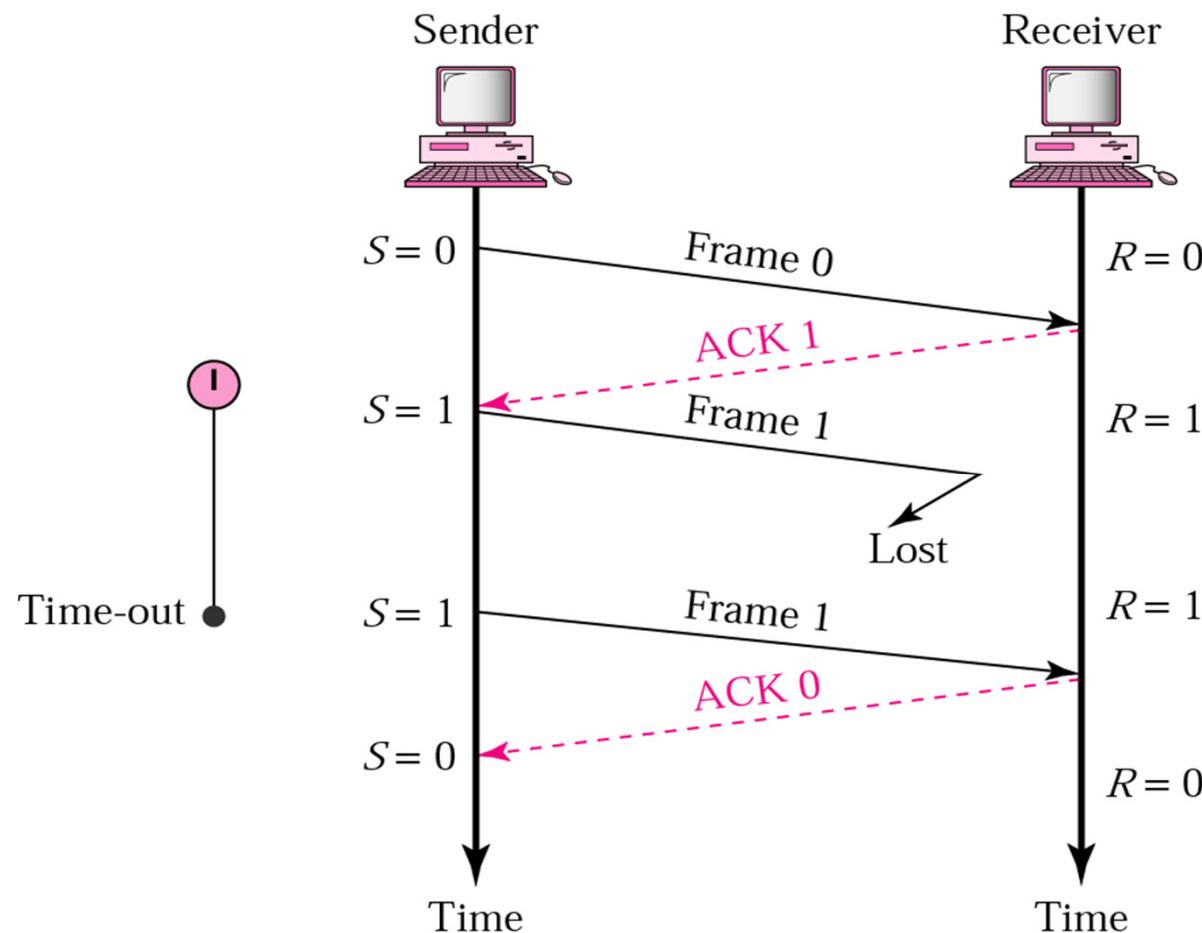


Need to send ACK?

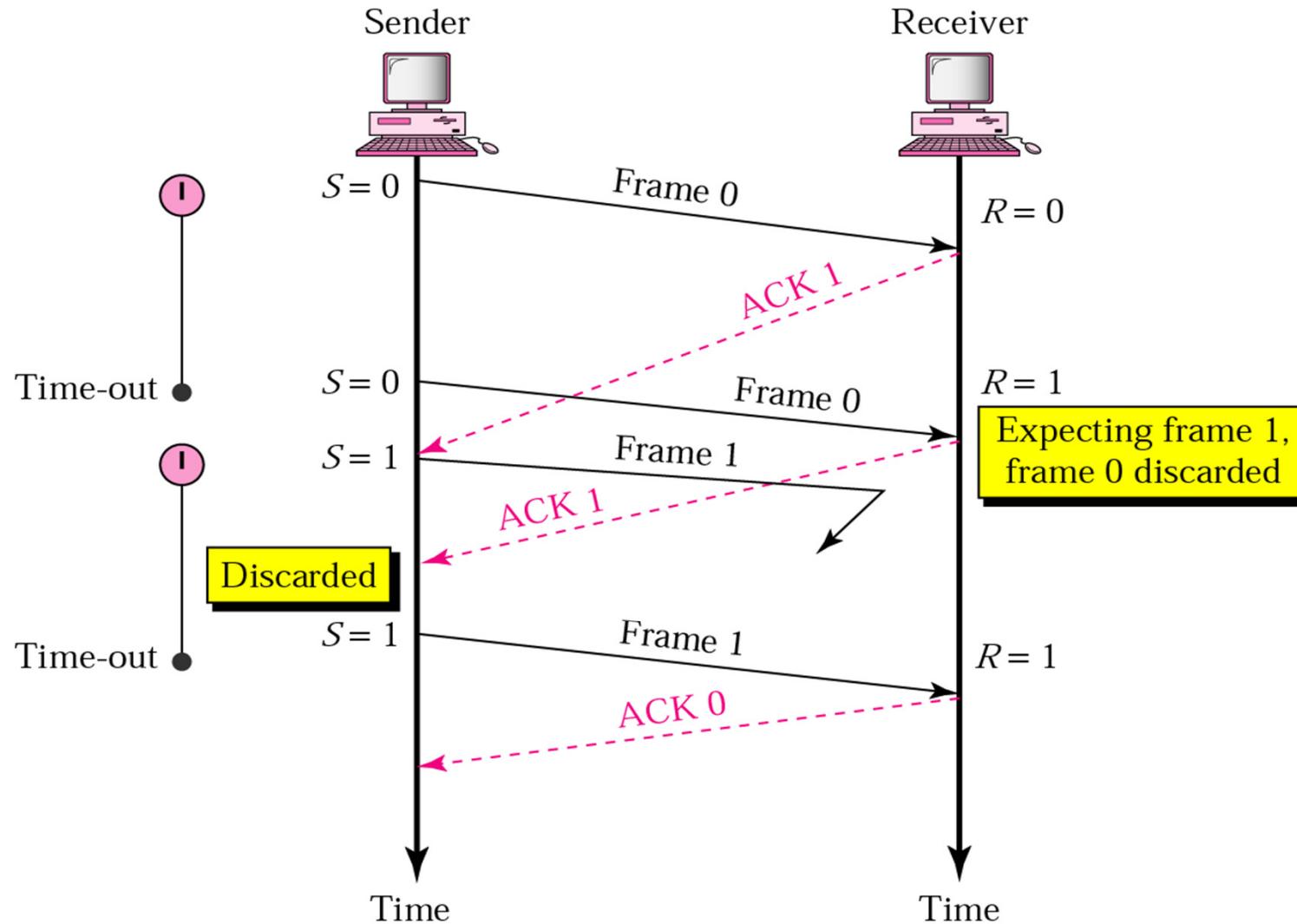
Buffer in Sender

- ❖ Error correction in Stop-and-Wait ARQ is done by keeping a copy of the sent frame
- ❖ Sender maintains a buffer with size = 1 frame
- ❖ Sender may not receive ACK because
 - ❖ Receiver detects error in the frame
 - ❖ The frame is lost before it reaches the receiver
 - ❖ The ACK is lost, delayed or corrupted
- ❖ Retransmitting the frame in the buffer when the timer expires

Stop-and-Wait ARQ, lost frame



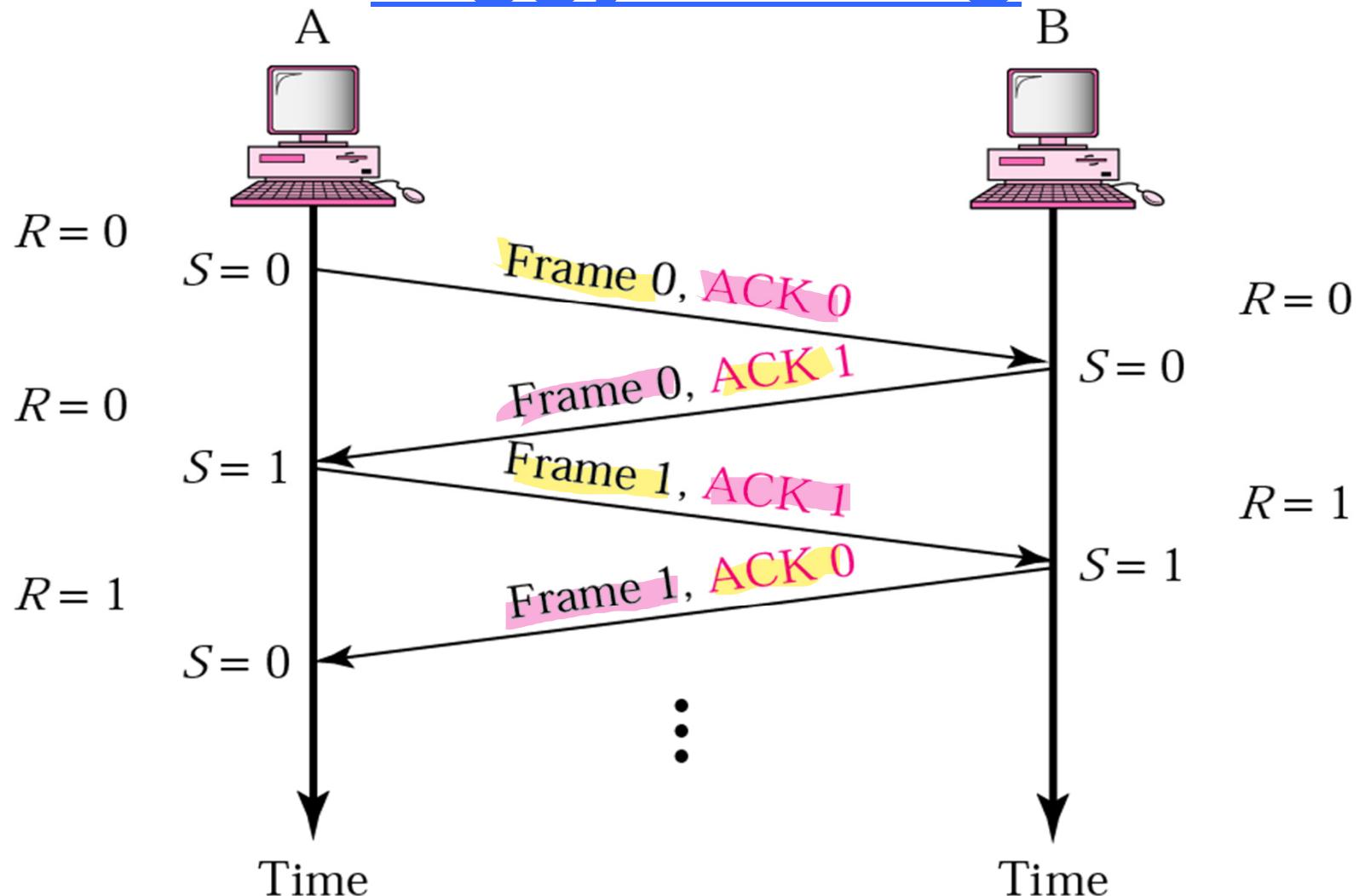
Stop-and-Wait ARQ, delayed ACK



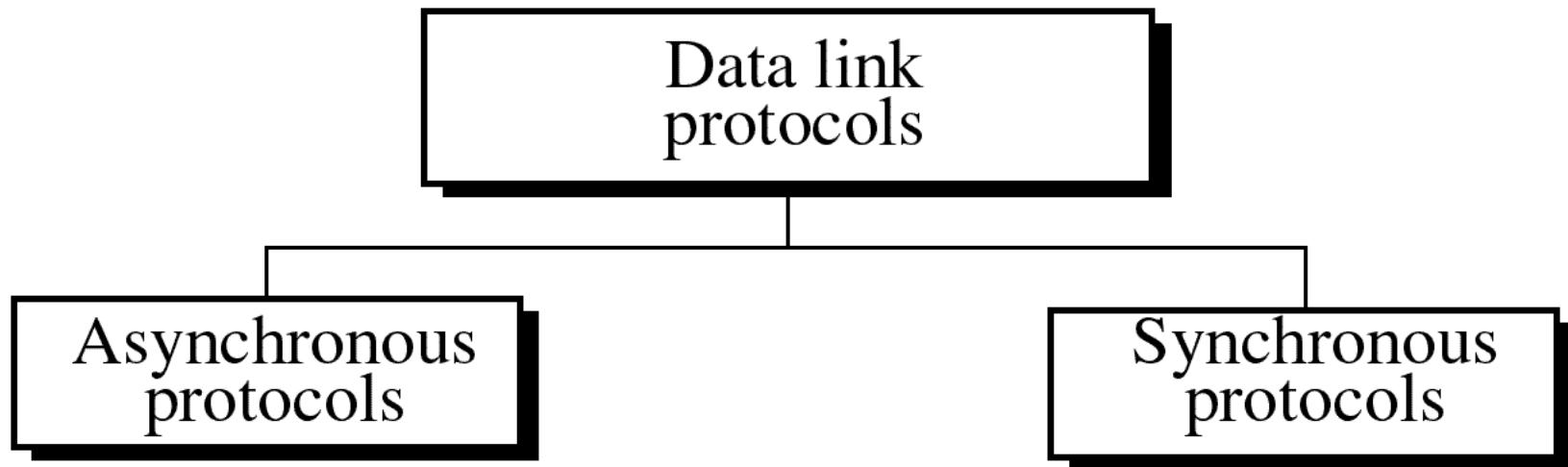
Piggybacking

- ❖ For **bidirectional** transmission
- ❖ The technique of temporarily delaying outgoing ACKs so that they can be hooked onto the next outgoing data frame
- ❖ A way of improving link utilization
- ❖ Normally most links using continuous ARQ are full-duplex and carry data frames in both directions
- ❖ Each side contains both a sender & a receiver

Piggybacking

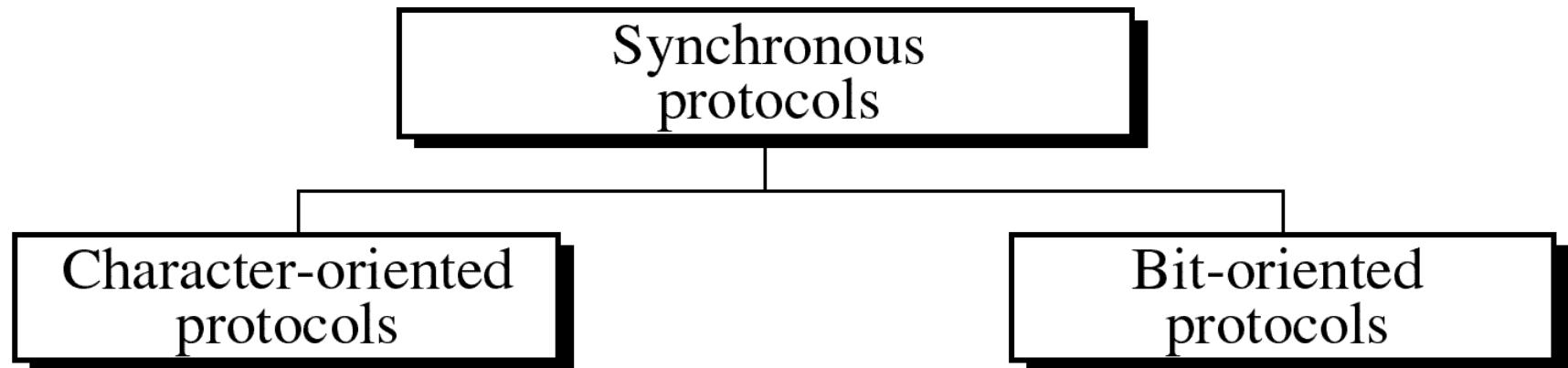


Data Link Protocols



- Asynchronous protocols, used primarily in modems, use start and stop bits and variable length gap between characters
- Due to slow data rate, they are being replaced by higher-speed synchronous protocols

Synchronous Protocols



- ❖ In character-oriented protocols, the frame is interpreted as a series of characters
 - ❖ 8-bit (e.g. ASCII), popular in old days with only text
- ❖ In bit-oriented protocols, each bit or groups of bits can have meaning

Bit-oriented Protocols

- ❖ Protocols use predefined bit patterns rather than transmission control characters to signal the start and end of a frame. (**frame delimiting**)
- ❖ The receiver searches the received bit stream on a bit by bit basis for the known start and end of frame bit pattern.
- ❖ E.g. HDLC

11.3

High-level Data Link Control (HDLC)

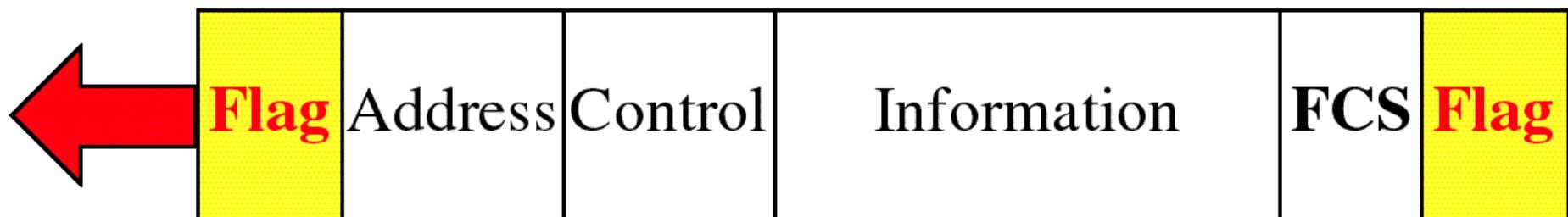
- ❖ an ISO international standard used on both point to point and multipoint (multidrop) data links
- ❖ supports both half-duplex and full-duplex with error detection
- ❖ adopts continuous ARQ with window mechanism
- ❖ used extensively in computer networks
- ❖ But many large manufacturers still use their own protocols similar to HDLC, e.g.,
 - ❖ IBM's SDLC (synchronous data link control)

HDLC Frame Formats

- ❖ Both data & control messages are carried in a standard format block

The flag is 8 bits of a fixed pattern.

01111110



HDLC Frame Formats

- ❖ **Flag field**

- ❖ (01111110) indicates start & end of a frame

- ❖ **Address Field**

- ❖ Address of the station receiving the frame

- ❖ **Control Field**

- ❖ For flow and error control (more details later)

- ❖ **Information Field**

- ❖ Contains user's data from upper layer

- ❖ **Frame Check Sequence (FCS) Field**

- ❖ For error checking similar to CRC

HDLC Frame Types

- ❖ **Information frames (I-frames)**

- ❖ carry actual data
 - ❖ also act as piggyback ACK

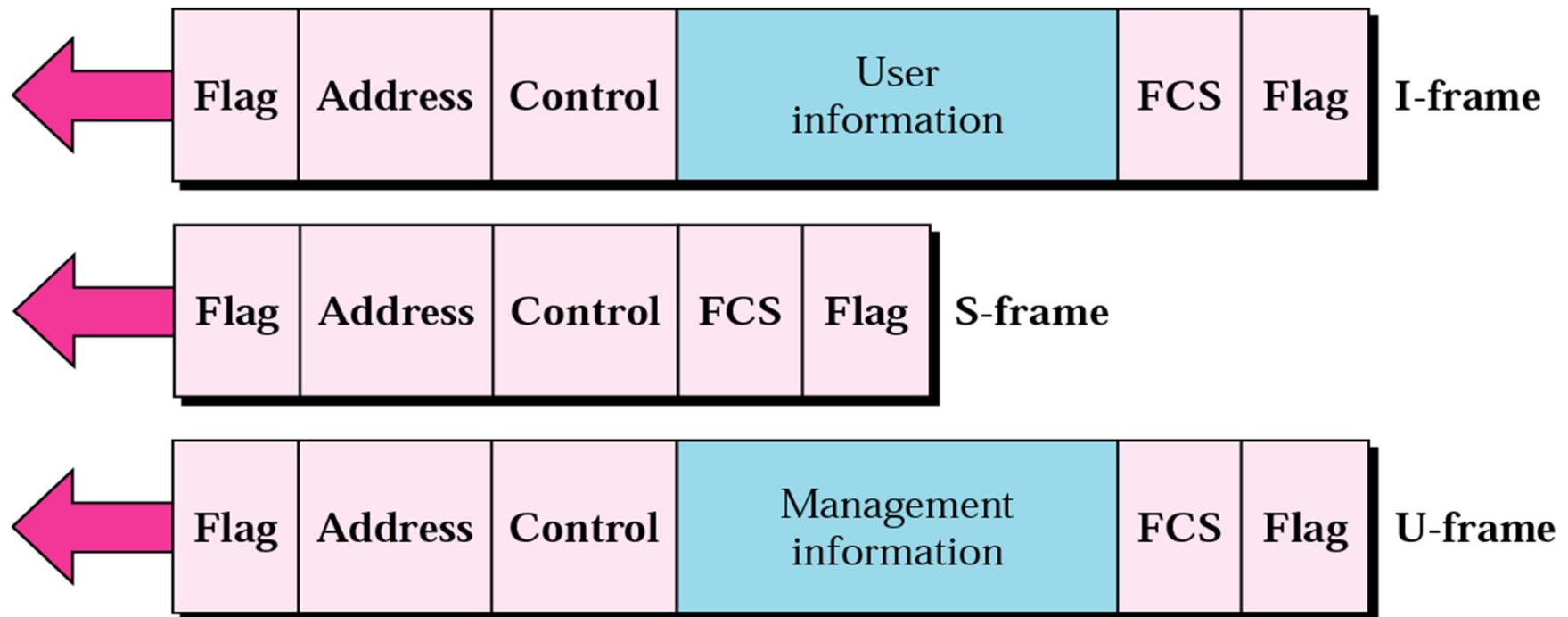
- ❖ **Supervisory frames (S-frames)**

- ❖ for transporting control information

- ❖ **Unnumbered frames (U-frames)**

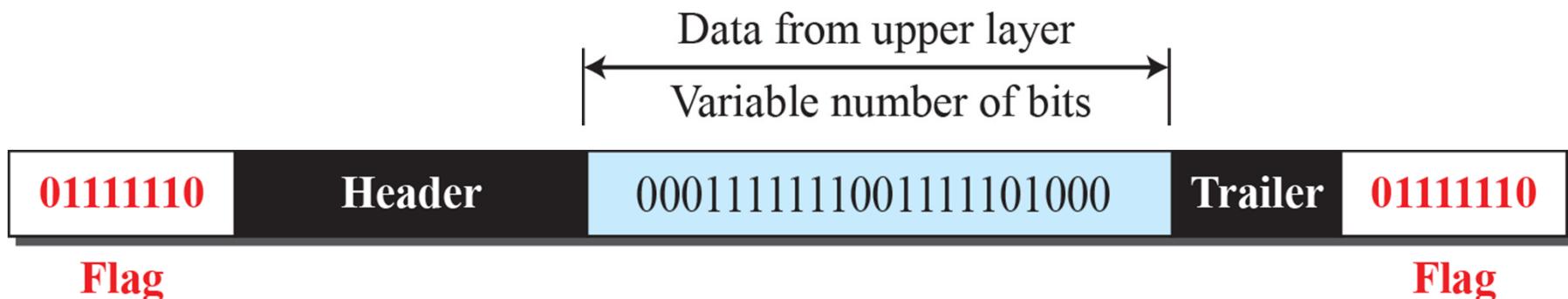
- ❖ for link set-up and disconnection

HDLC frame types



Data transparency

- ❖ Data can be any combination of bits
- ❖ Confusion between control information and data is called a **lack of data transparency**
 - ❖ E.g. data field contains 01111110



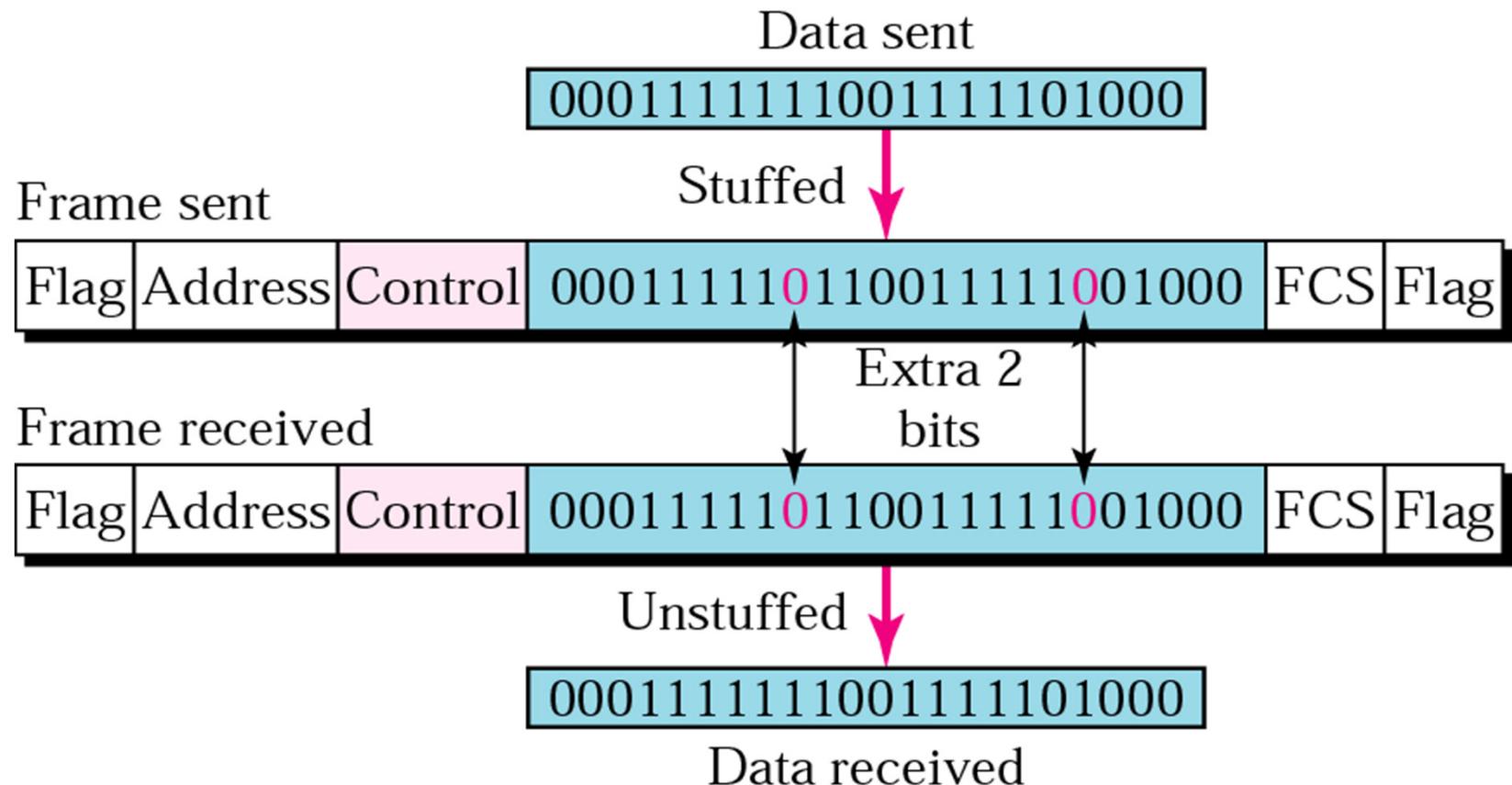
Data transparency

- ❖ when data are transparent, which means we should be able to send any combination of bits as data
- ❖ Bit Stuffing method is used in HDLC for achieving **data transparency**

Bit Stuffing

- ❖ A method used in HDLC for achieving **data transparency**
 - ❖ Ensure that the flag pattern is not present in the frame contents
- ❖ Sender inserts a “0” bit after transmitting five consecutive “1” bits
- ❖ *Exceptions:* when the bit sequence is really a flag
- ❖ Receiver removes the “0” bit after receiving five consecutive “1” bits

Bit stuffing and removal



Summary

- ❖ Flow Control and Error Control
- ❖ Stop-and-Wait ARQ
- ❖ High-level Data Link Control (HDLC)
- ❖ Bit stuffing

- ❖ Revision Quiz
 - ☞ http://highered.mheducation.com/sites/0073376221/student_view0/chapter11/quizzes.html



Lecture 5 Medium Access Control (MAC) Protocols

Textbook: Ch.12

We are still in Data Link Layer . . .

Main Topics

- ❖ 12.1 Random Access
 - ❖ ALOHA Protocol
 - ❖ Pure ALOHA and Slotted ALOHA
 - ❖ CSMA - Carrier Sense Multiple Access
 - ❖ Non-persistent CSMA
 - ❖ 1-persistent CSMA
 - ❖ CSMA/CD
 - ❖ Collision Detection Procedure
 - ❖ Collision Detection Timing

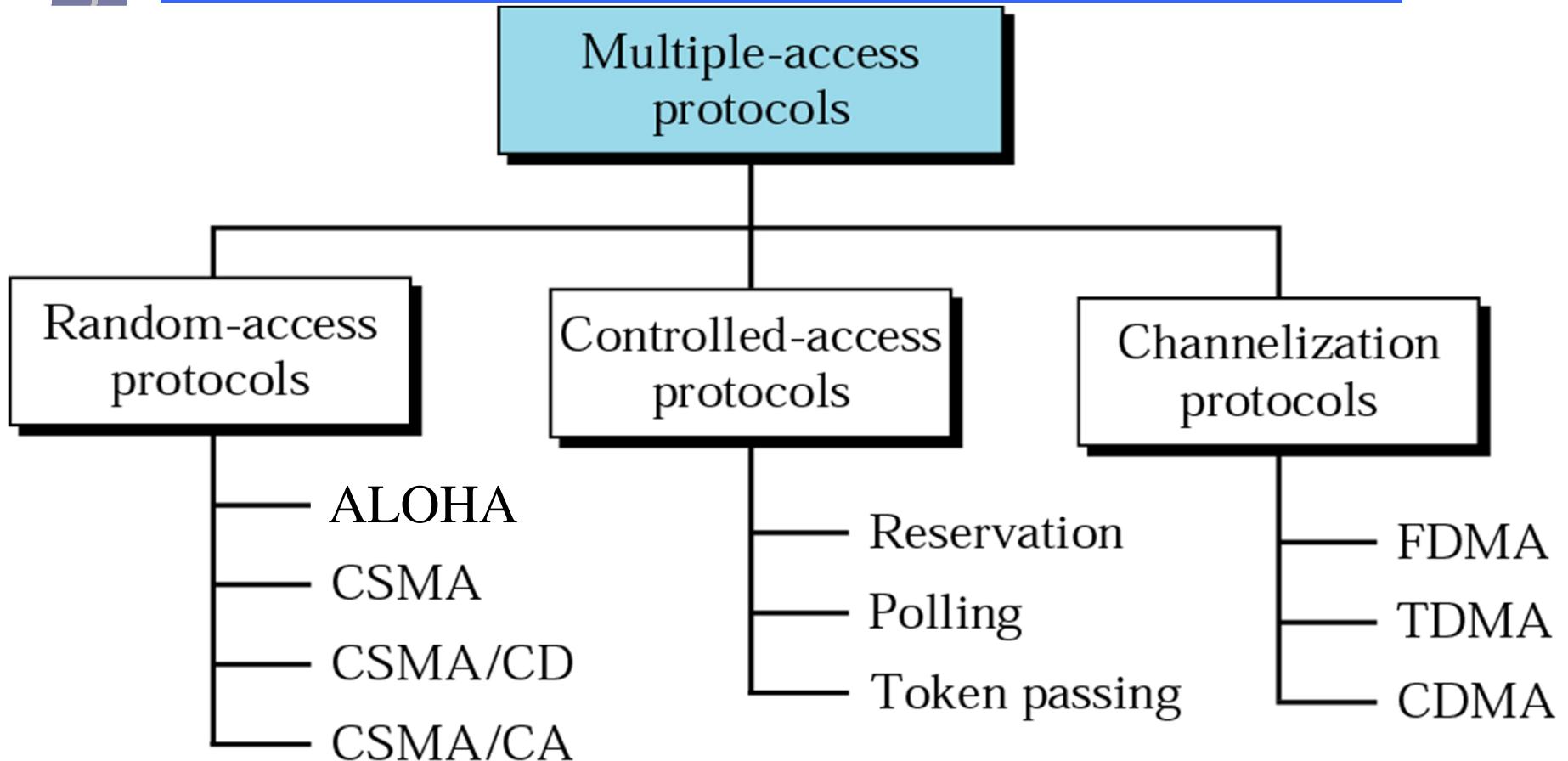
Local Area Network (LAN)

- ❖ A network used to interconnect distributed computers located within a single building or localized group of buildings ***in a limited geographic area***
- ❖ Wired LAN & wireless LAN
- ❖ Classified by
 - ❖ Topology
 - ❖ Transmission media
 - ❖ Multiple Access Control (MAC) protocols

Multiple Access Control Protocols

- ❖ The protocols used to determine who can transmit next on a ***multiaccess channel*** (i.e. the network is in a bus topology, usually also a ***broadcast channels***)
- ❖ Also called ***media access control protocols***
- ❖ A protocol must be used to ensure that the transmission medium is accessed and used in a fair way

Multiple Access Protocols



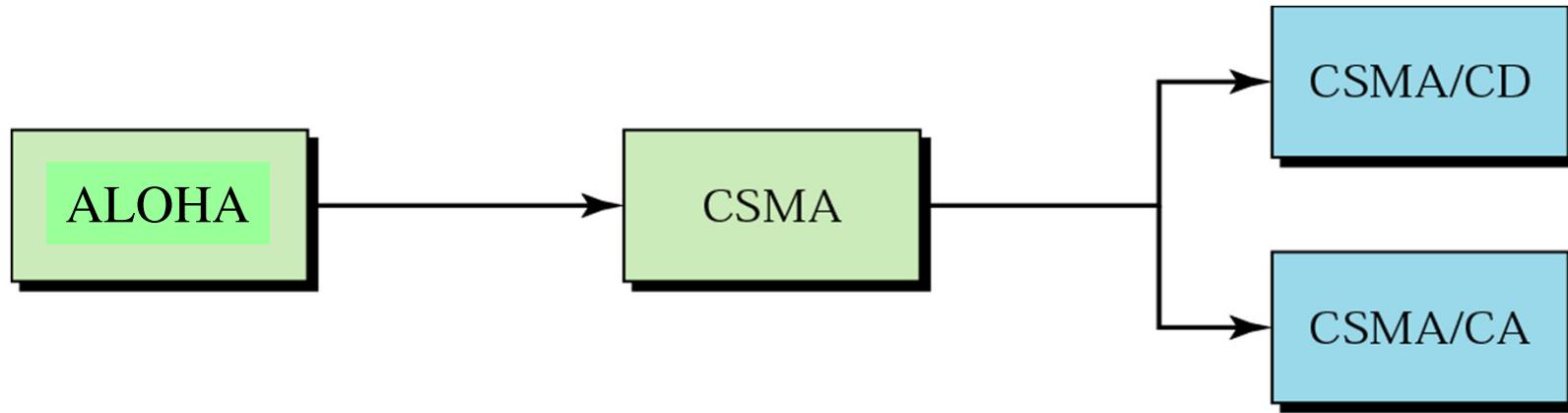
RANDOM ACCESS

In random-access or contention no station is superior to another station and none is assigned control over another.

At each instance, a station that has data to send uses a procedure defined by the protocol to make a decision on whether or not to send.

This decision depends on the state of the medium (idle or busy).

Evolution of random access protocols



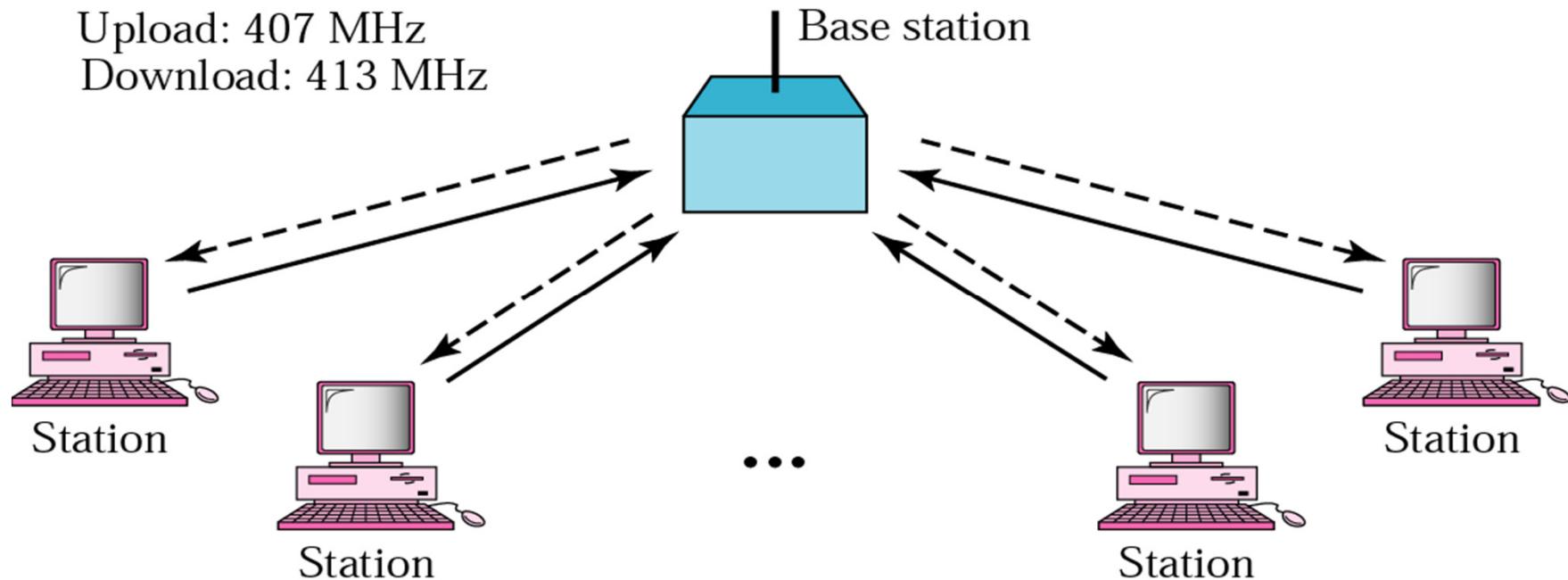
If two or more stations send at the same time, there is a **collision** *in the common channel* and all the data frames will be destroyed (since the network is a bus topology)

12.1.1

ALOHA Network

A wireless radio network in a bus topology, developed by University of Hawaii,

Upload: 407 MHz
Download: 413 MHz



- ❖ ALOHA is the earliest random access protocol (in early 1970)

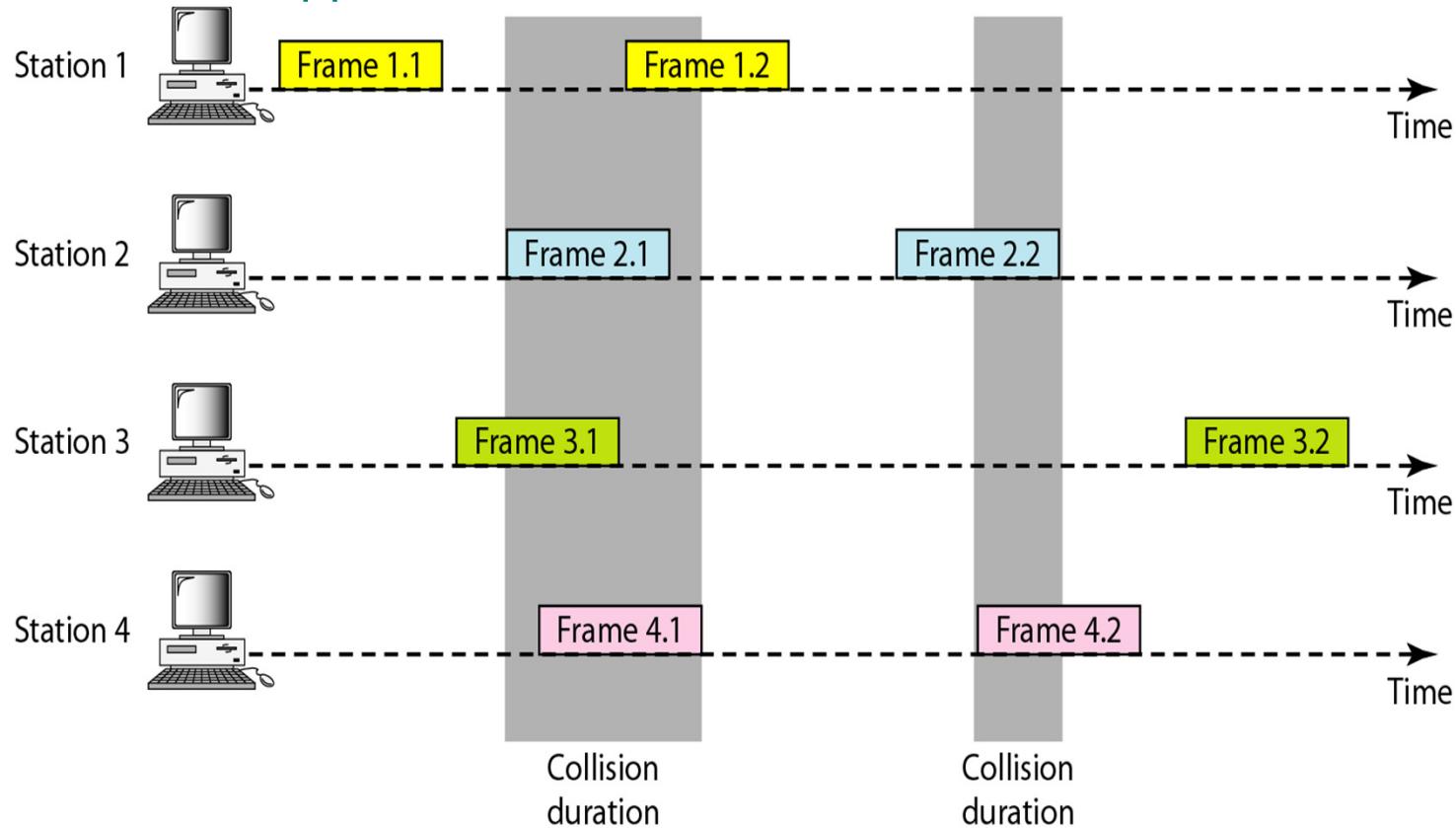
Pure ALOHA

(Random Access Protocol)

- ❖ It can be used on any shared medium
- ❖ The network is in a bus topology
- ❖ Each station makes its own decision
- ❖ Transmit whenever the data is ready
- ❖ If collided, retry (re-transmit) after a random delay (called **back-off time**)
- ❖ The collision is known by ACK operation; if no ACK receives, then assume collision

Pure ALOHA

- ❖ Idea: Each station sends a frame whenever it has a frame to send.
 - ❖ What will happen? → Collision



Procedure for ALOHA protocol

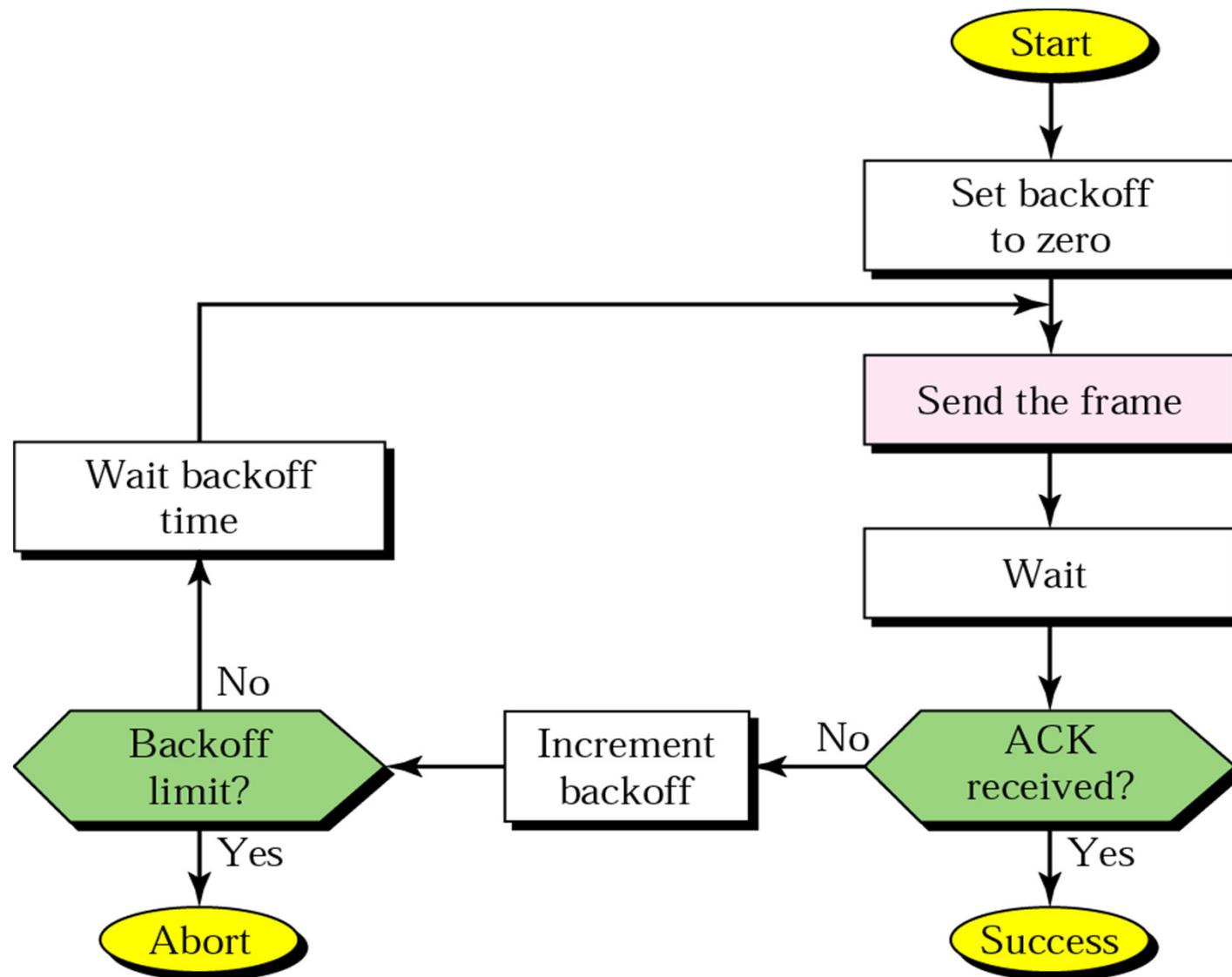
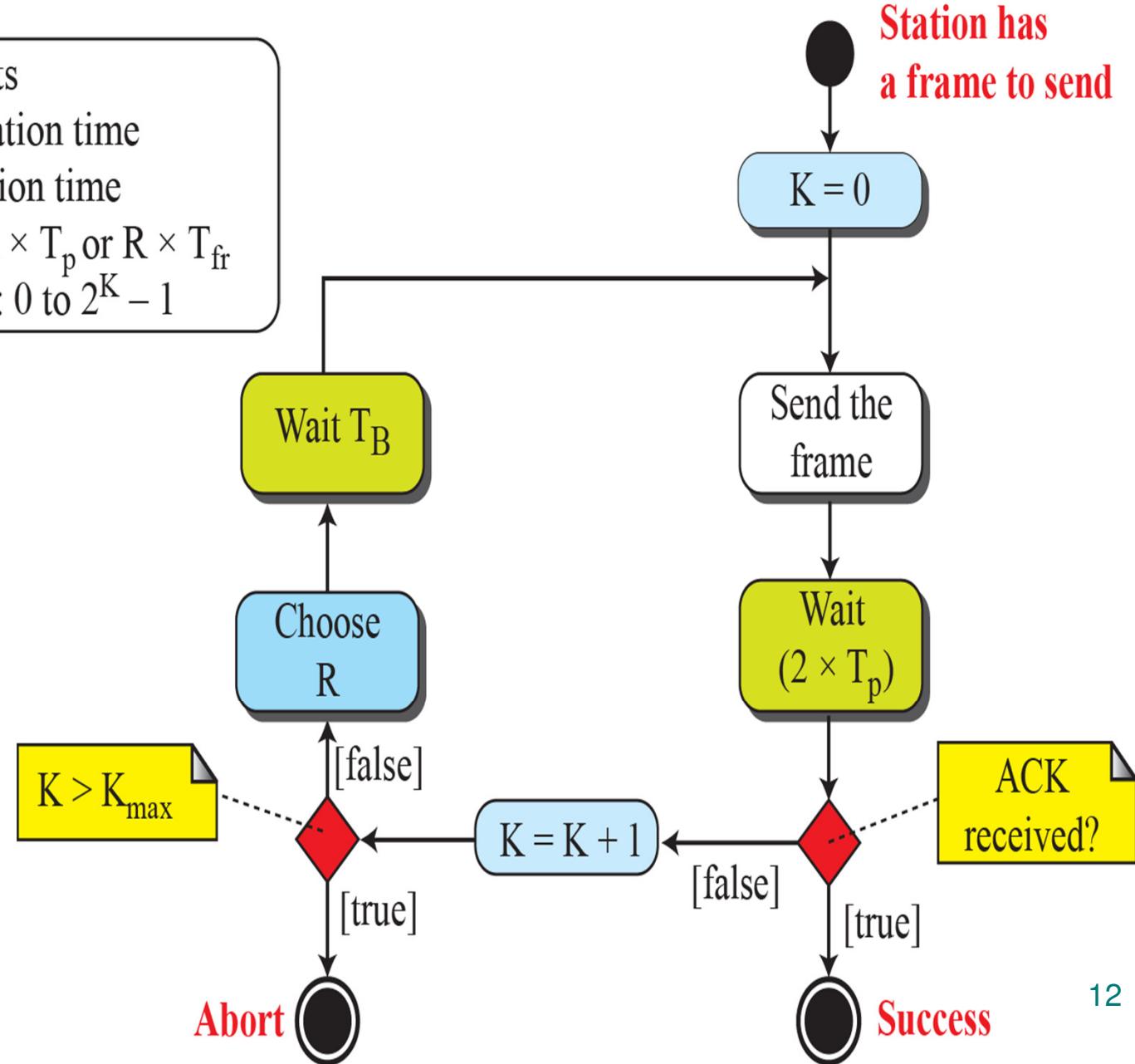


Figure 12.3: Procedure for pure ALOHA protocol

Legend

K : Number of attempts
 T_p : Maximum propagation time
 T_{fr} : Average transmission time
 T_B : (Back-off time): $R \times T_p$ or $R \times T_{fr}$
 R : (Random number): 0 to $2^K - 1$



Example 12. 1

The stations on a wireless ALOHA network are a maximum of 600 km apart.

If we assume that signals propagate at 3×10^8 m/s, we find $T_p = (600 \times 10^3) / (3 \times 10^8) = 2$ ms.

For $K = 2$, the range of R is $\{0, 1, 2, 3\}$.

This means that T_B can be 0, 2, 4, or 6 ms, based on the outcome of the random variable R .

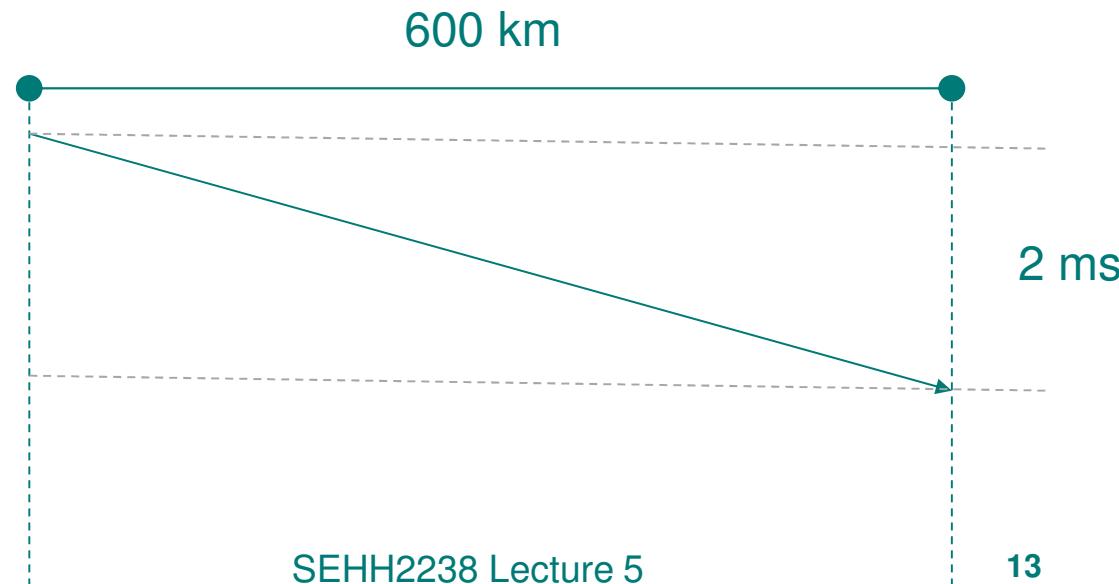
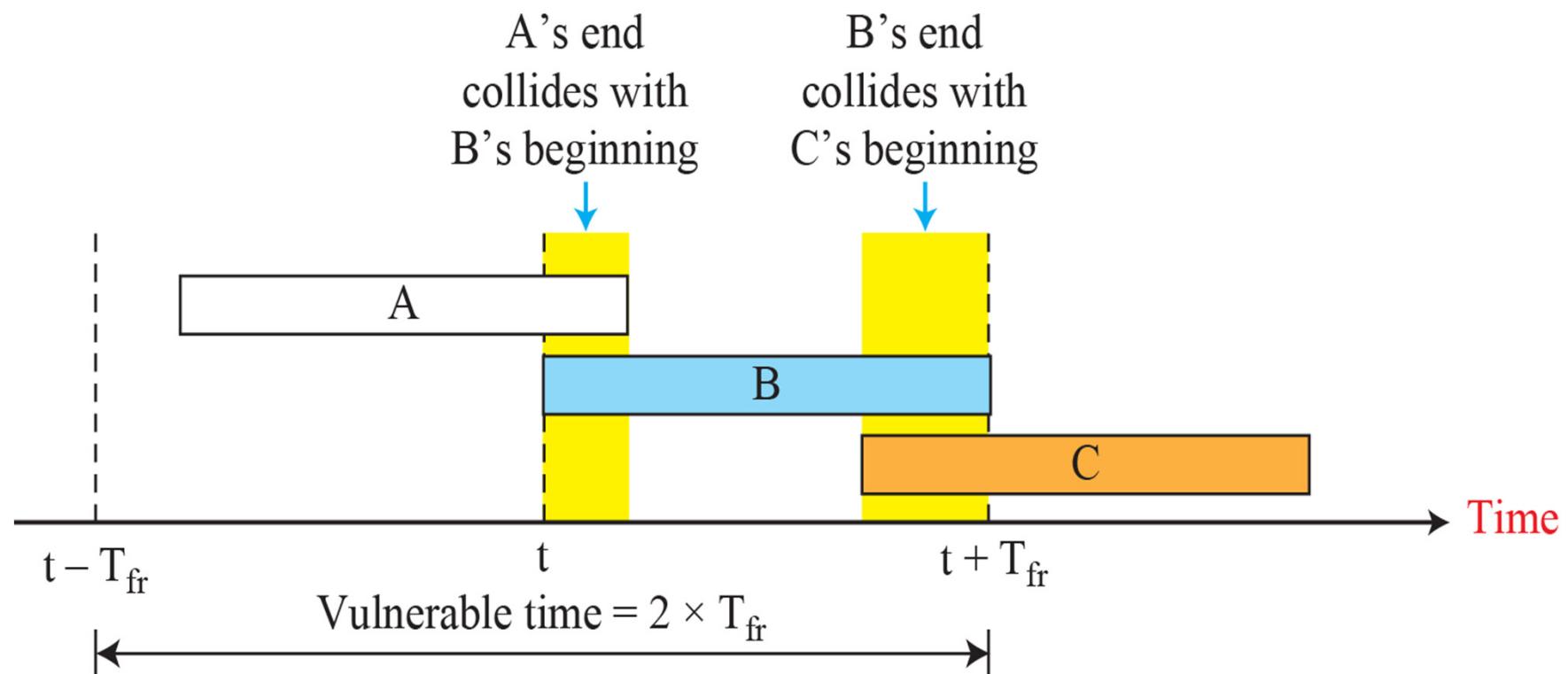


Figure 12.4 *Vulnerable time for pure ALOHA protocol*

The length of time in which collisions may occur

T_{fr} – frame size in seconds



Example 12.2

A pure ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the requirement to make this frame collision-free?

Solution

Average frame transmission time T_{fr} is 200 bits/200 kbps or 1 ms. The vulnerable time is $2 \times 1 \text{ ms} = 2 \text{ ms}$. This means no station should send later than 1 ms before this station starts transmission and no station should start sending during the period (1 ms) that this station is sending.

Throughput

- ❖ Throughput is the portion of data frames reaching the destination **successfully**
- ❖ In pure Aloha, the maximum throughput is only 0.18

Proof (optional, could be skipped):

It is known that the average number of **successful** transmission for pure Aloha is

$$S = G \times e^{-2G}$$

- ❖ where G is the average number of frames generated by the system in one frame transmission time (may be collided)
- ❖ by differentiation, we can find that S_{\max} occurs at $G = 1/2$ and the corresponding S_{\max} is 0.18

Example 12.3

A pure ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the throughput if the system (all stations together) produces

- a. 1000 frames per second?
- b. 500 frames per second?
- c. 250 frames per second?

transfer frames per second ==> frames per frame time

Solution

The frame transmission time is $200/200$ kbps or 1 ms.

- a. If the system creates 1000 frames per second, or 1 frame per millisecond, then $G = 1$. In this case $S = G \times e^{-2G} = 0.135$ (13.5 percent). This means that the throughput is $1000 \times 0.135 = 135$ frames. Only 135 frames out of 1000 will probably survive.

Example 12.3 (continued)

- b. If the system creates 500 frames per second, or 1/2 frames per millisecond, then $G = 1/2$. In this case $S = G \times e^{-2G} = 0.184$ (18.4 percent). This means that the throughput is $500 \times 0.184 = 92$ and that only 92 frames out of 500 will probably survive. Note that this is the maximum throughput case, percentage-wise.
- c. If the system creates 250 frames per second, or 1/4 frames per millisecond, then $G = 1/4$. In this case $S = G \times e^{-2G} = 0.152$ (15.2 percent). This means that the throughput is $250 \times 0.152 = 38$. Only 38 frames out of 250 will probably survive

Slotted ALOHA

- ❖ Adopt the same **fixed packet length** (data frame length) for all stations
- ❖ Divide time into discrete intervals (**slots**) of duration equals to the packet length (in terms of transmission time)
- ❖ All stations follow the same synchronized time system
- ❖ Transmit only at the **beginning** of the next time slot
- ❖ If collided, retry after a random delay
- ❖ *Maximum throughput is doubled to 0.36*

Figure 12.5 *Frames in a slotted ALOHA network*

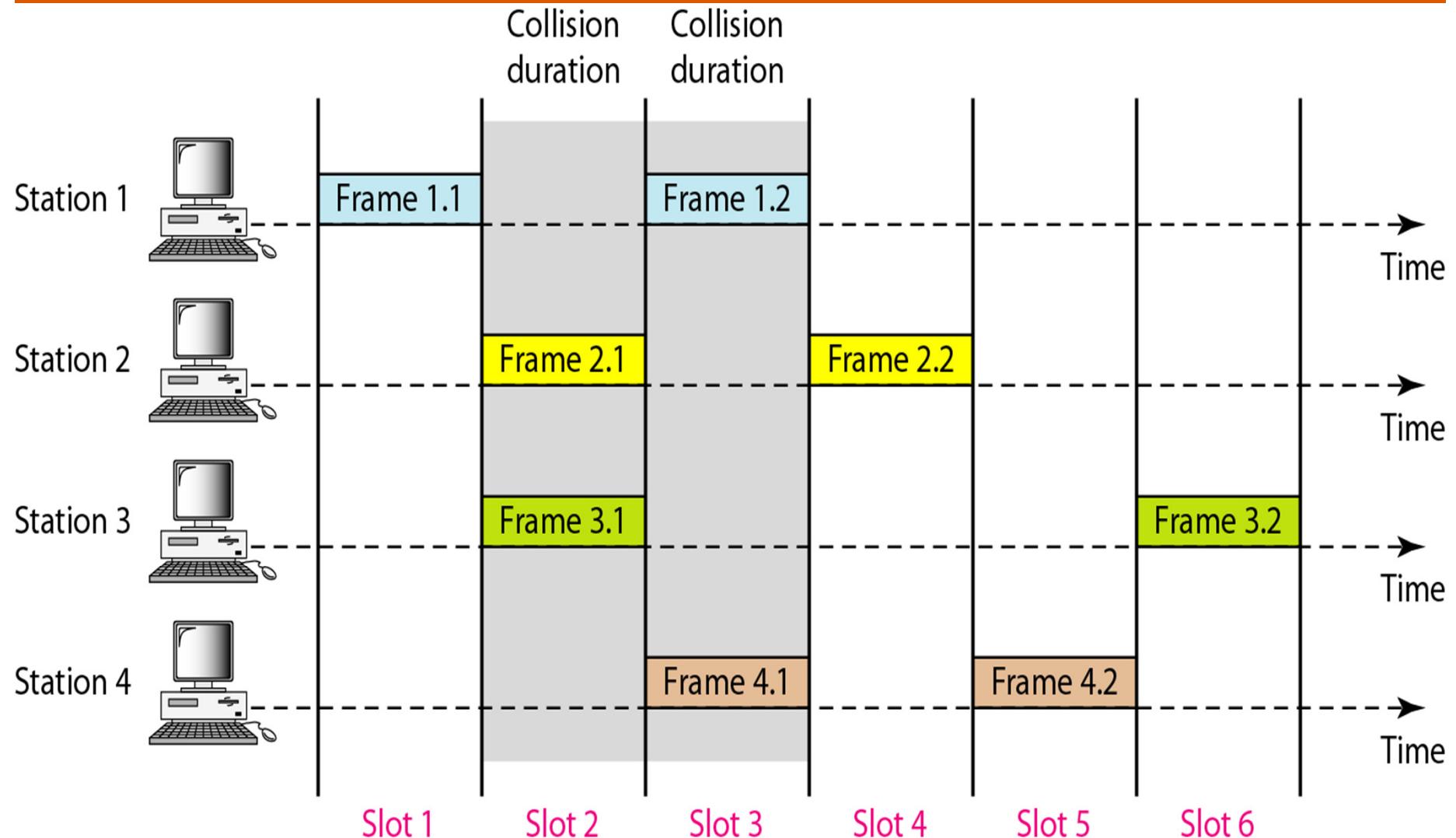
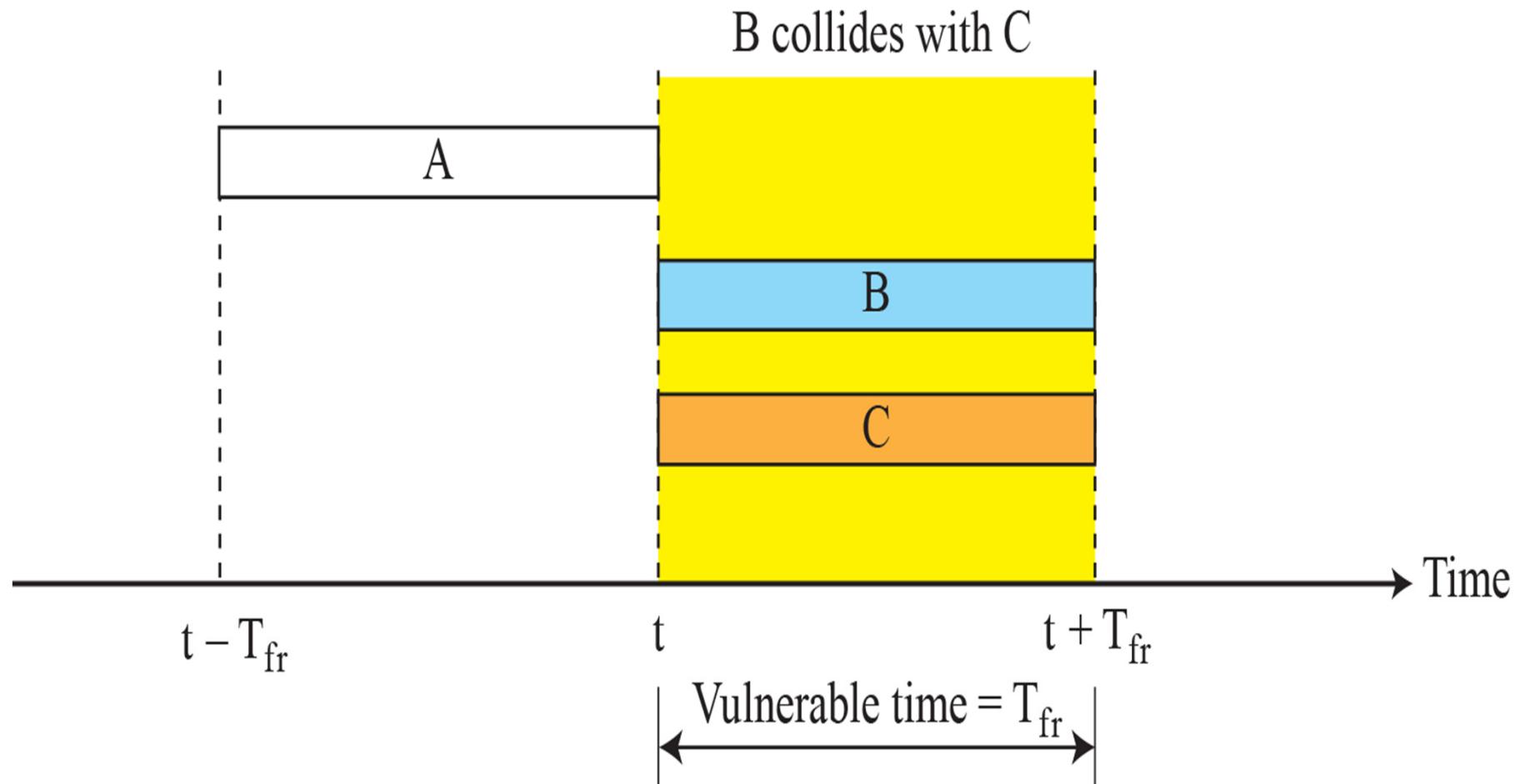


Figure 12.6 *Vulnerable time for slotted ALOHA protocol*



12.1.2 Carrier Sense Multiple Access (CSMA)

- ❖ Allow **variable packet length**
- ❖ Listen to the channel (sense the carrier) before transmission
- ❖ If the channel is idle then transmit *otherwise*
 - ❖ **non-persistent CSMA**
 - (if channel is busy) retry after a random delay
 - ❖ **1-persistent CSMA**
 - (if channel is busy) wait until the channel becomes idle and then transmit
- ❖ If collided, retry after a random delay

Collision in CSMA

(The collision is known by receiving and checking the ACK)

The network should be a
bus topology

B starts
at time t_1

C starts
at time t_2

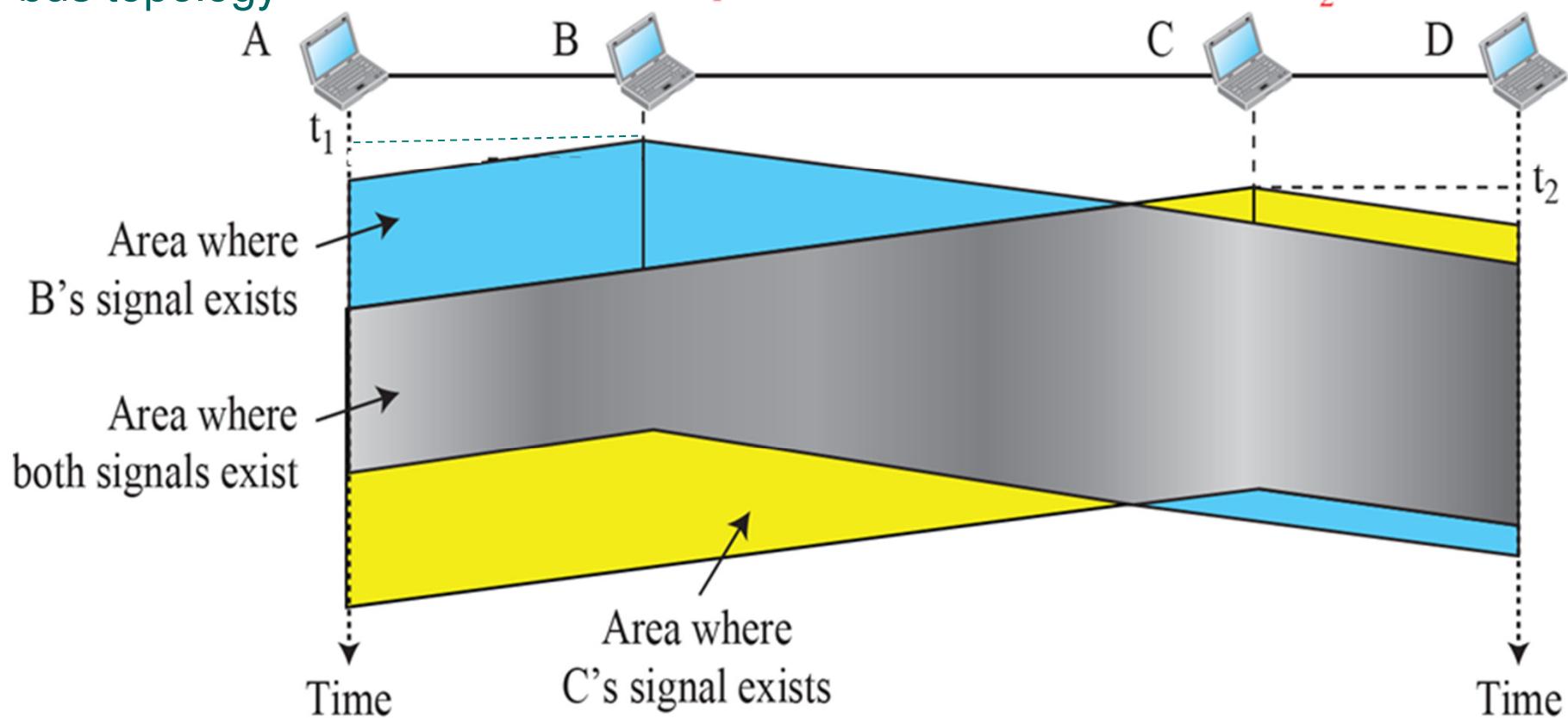
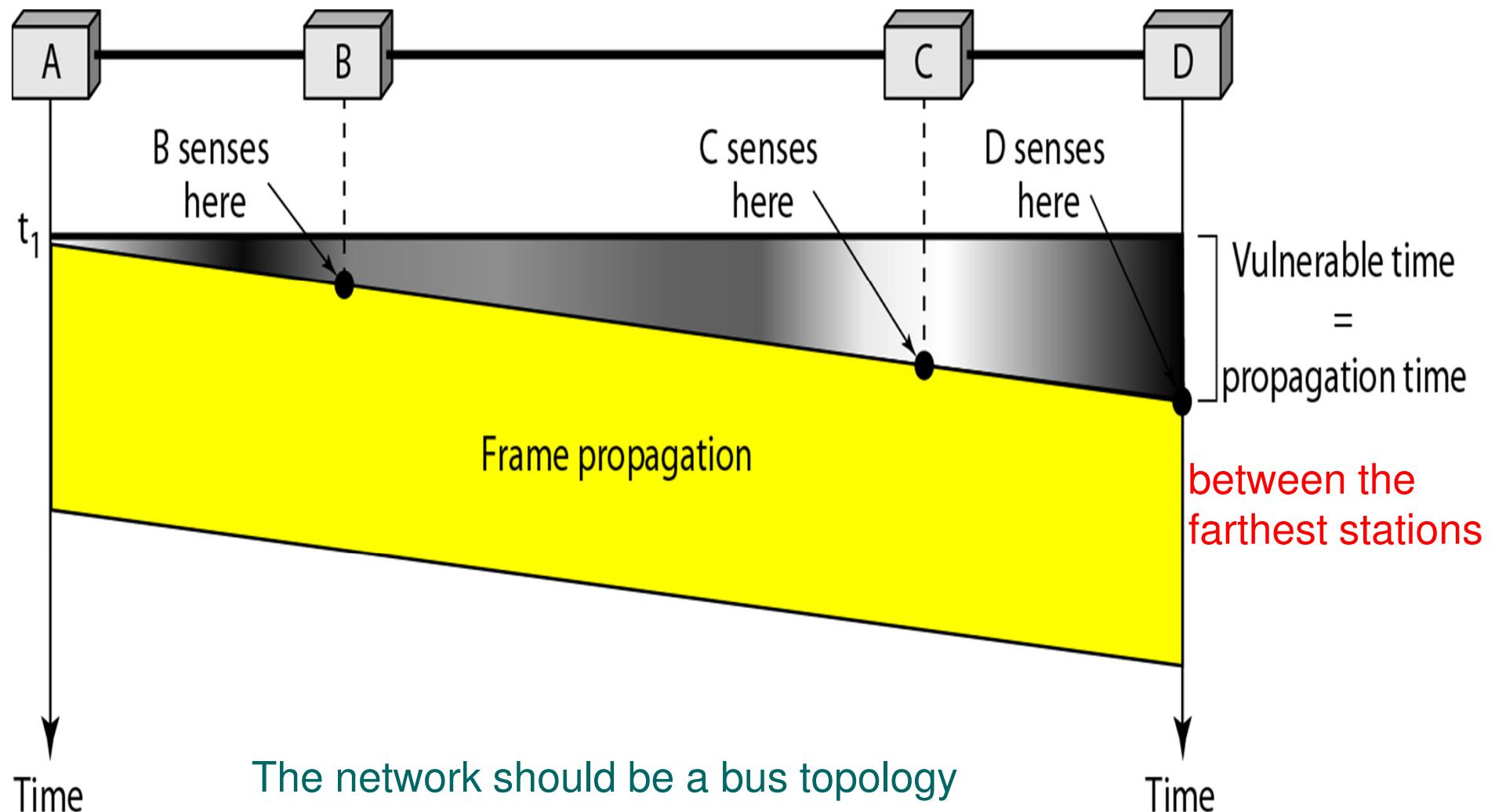


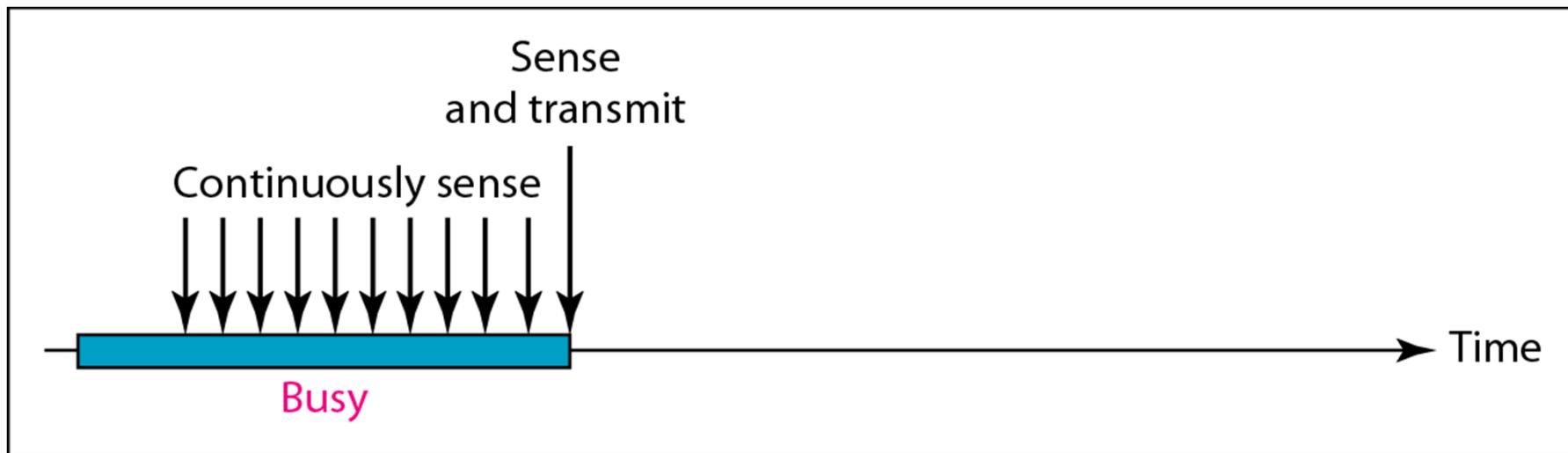
Figure 12.8 *Vulnerable time in CSMA*



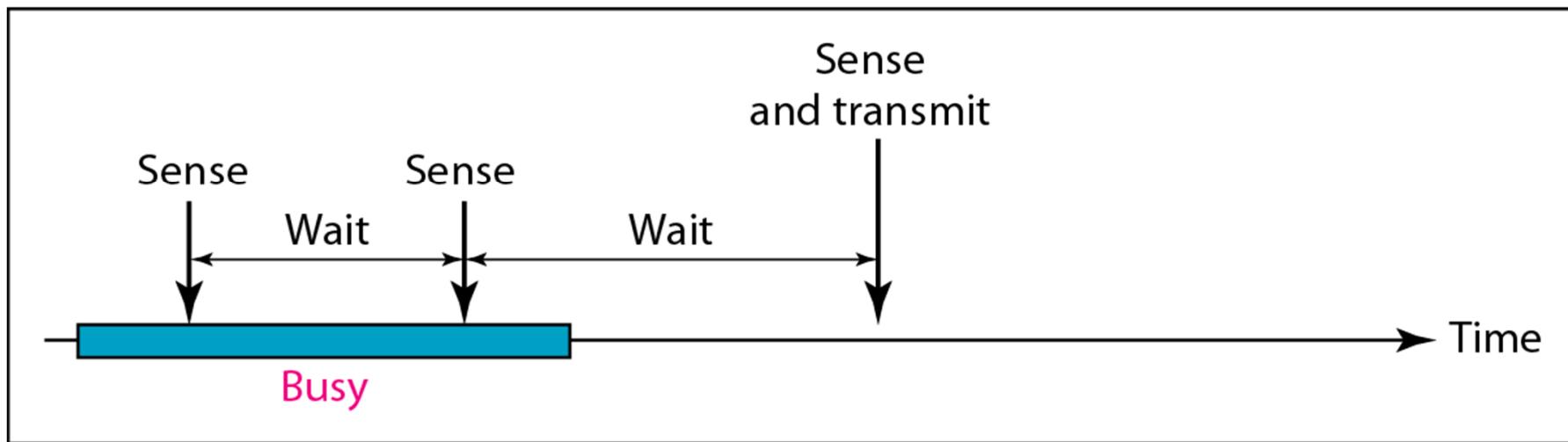
Persistence Methods

- ❖ Persistence method defines the procedure for a station that senses a busy medium
- ❖ **non-persistent CSMA**
- ❖ **1-persistent CSMA**
- ❖ **p-persistent CSMA (skip)**

Figure 12.9 *Behavior of persistence methods*

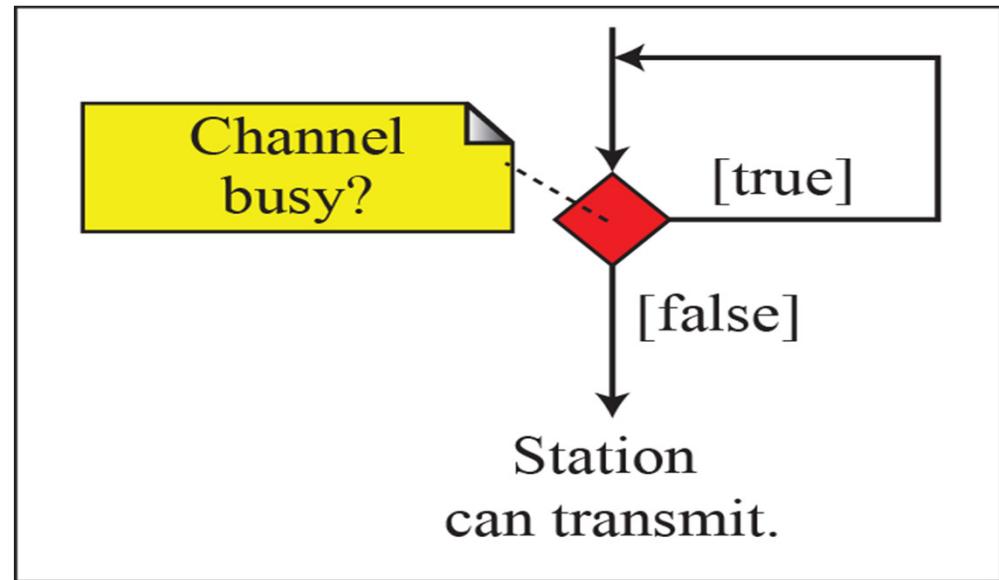


a. 1-persistent

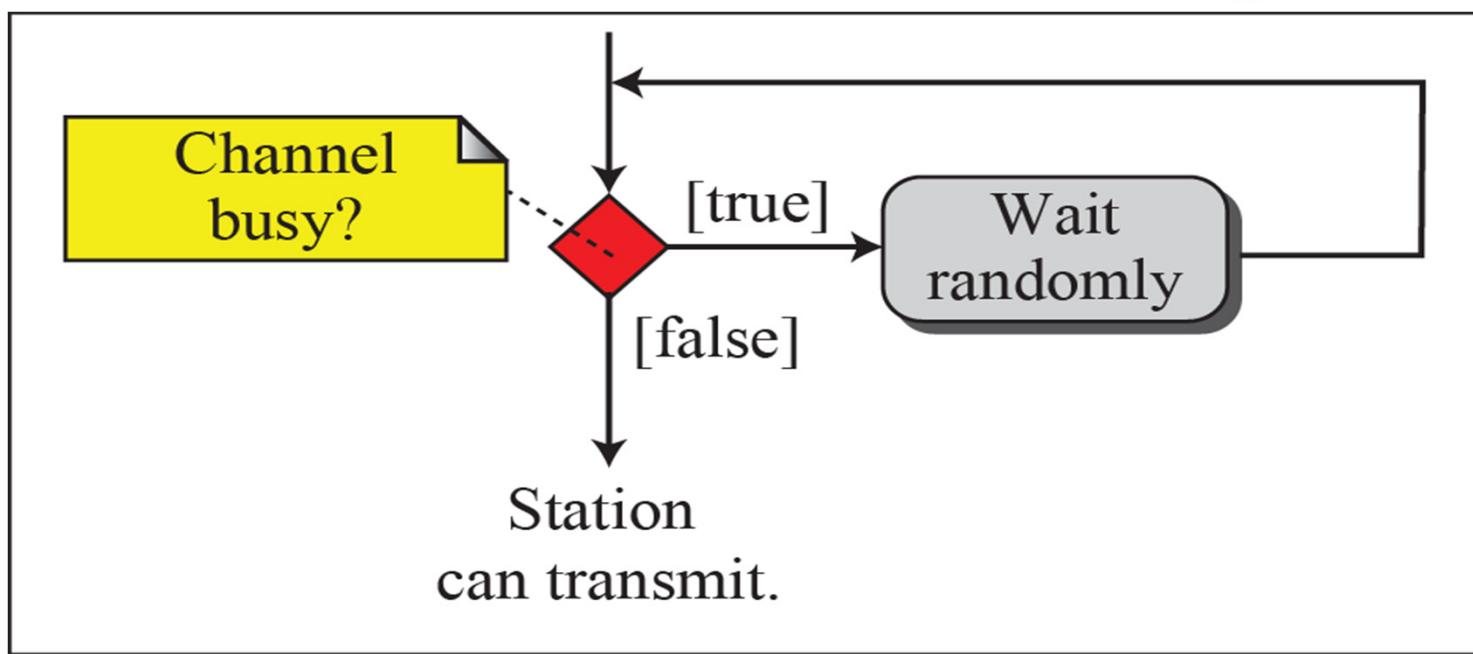


b. Nonpersistent

Figure 12.10: Flow diagram for persistence methods



a. 1-persistent



b. Nonpersistent

Non-persistent CSMA

- ❖ A station senses the channel when it has a frame ready to send
- ❖ If the channel is idle, the station sends the frame immediately
- ❖ If the channel is busy, it waits a random period of time and senses the channel again (and repeat the process)
- ❖ If collided, retry after a random delay

1-Persistent CSMA

- ❖ A station senses the channel when it has a frame ready to send
- ❖ If the channel is busy, the station senses the channel again and again until the channel becomes idle
- ❖ When the channel is idle, the station sends the frame immediately
- ❖ If collided, retry after a random delay

12.1.3

CSMA/CD Protocol

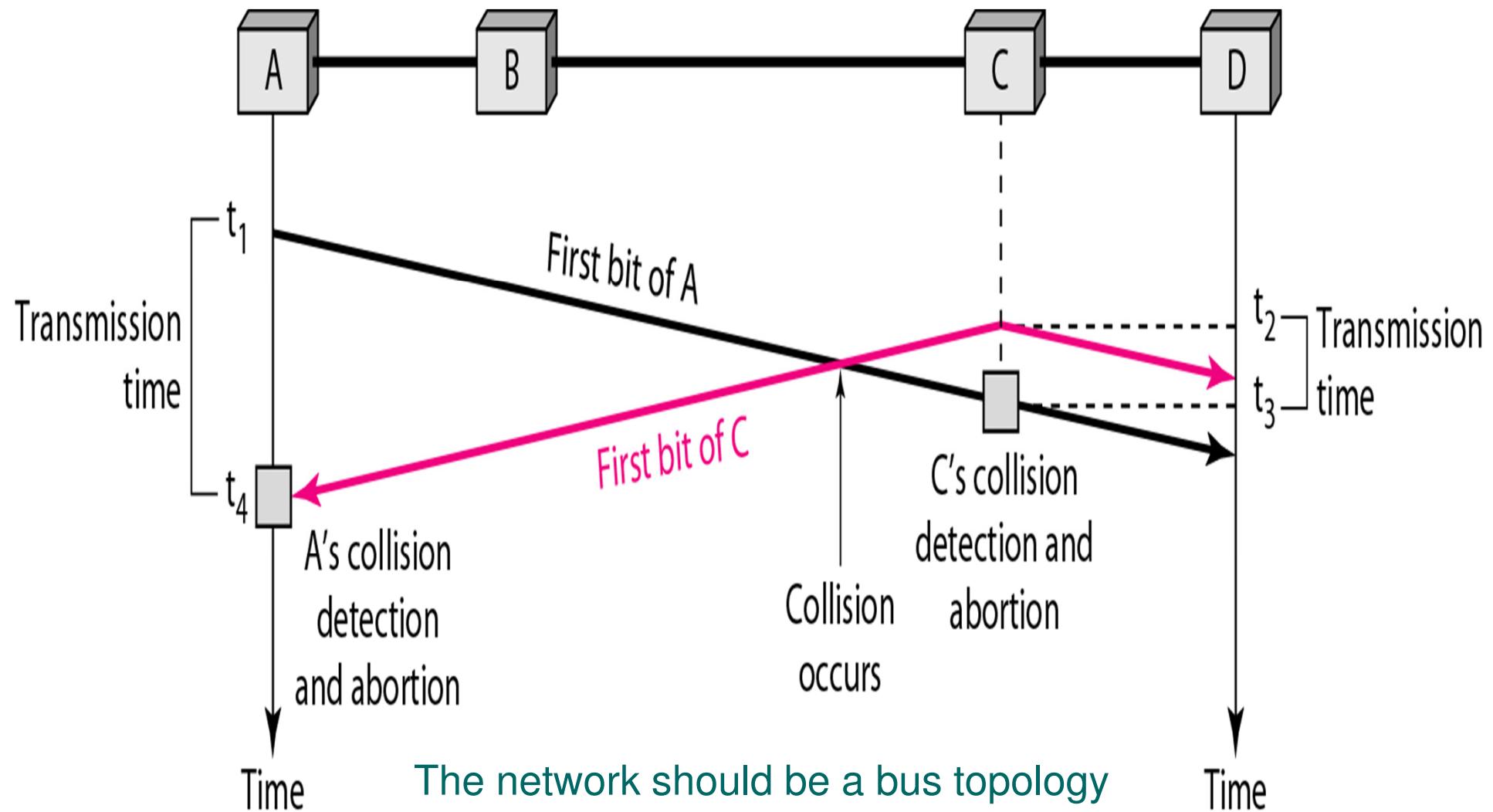
- ❖ Carrier Sense Multiple Access **with Collision Detection**
- ❖ Listen *before & while* transmission
- ❖ Before transmission, the source station first listen to the channel
- ❖ If the channel is idle, transmit
- ❖ If collided, retry after a random delay
- ❖ What is more ...

Collision Detection in CSMA/CD

- ❖ It is possible that 2 stations detect the channel idle at the same time and start transmission simultaneously and hence data collided
- ❖ *To check whether collision occurs, the station simultaneously monitors the data signal actually present on the channel when transmitting a frame*
- ❖ *If transmitted & monitored signals are different then collision detected (CD). The station then stops transmitting the current frame immediately*

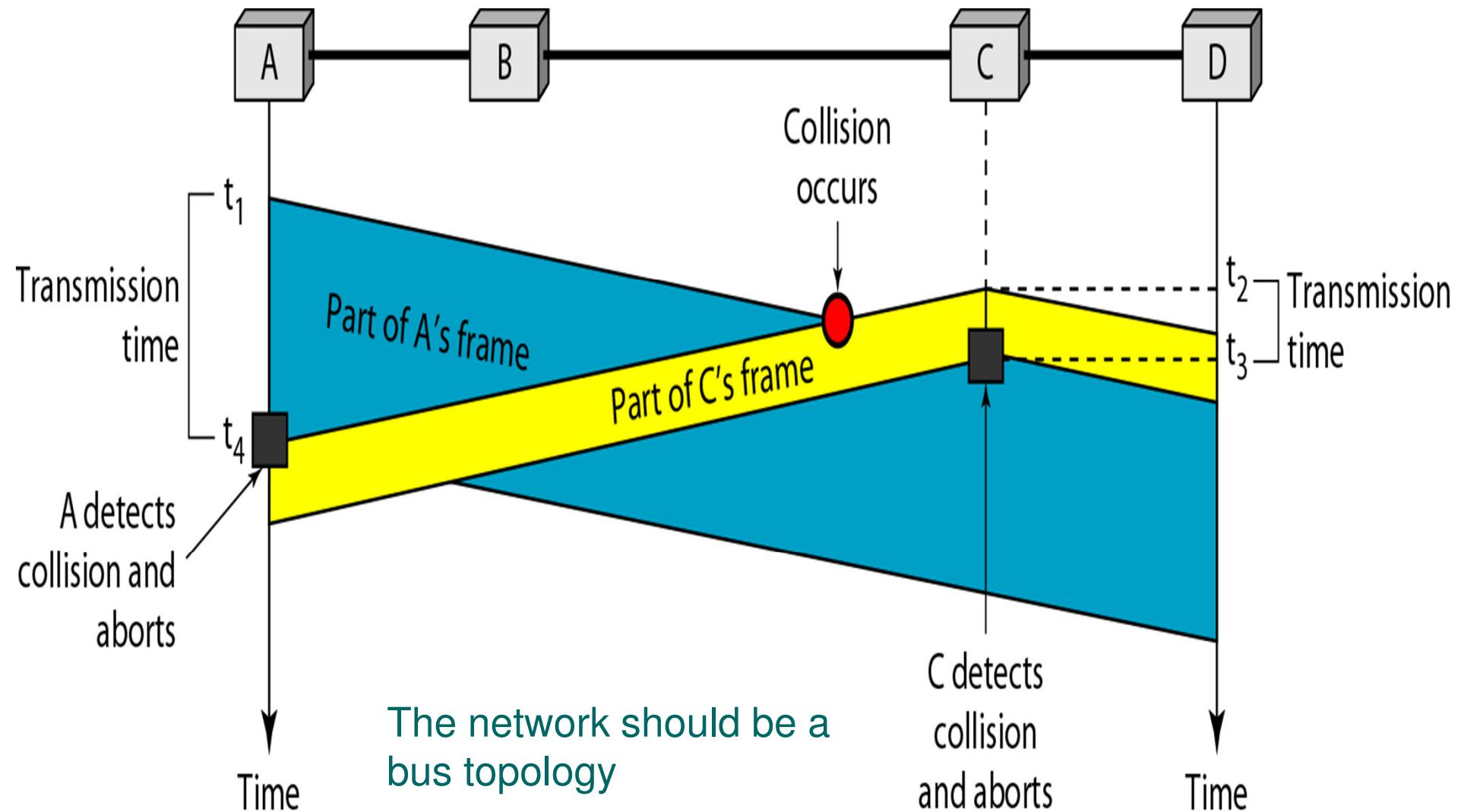
Collision Detection

Figure 12.11 *Collision of the first bit in CSMA/CD*



Collision Detection

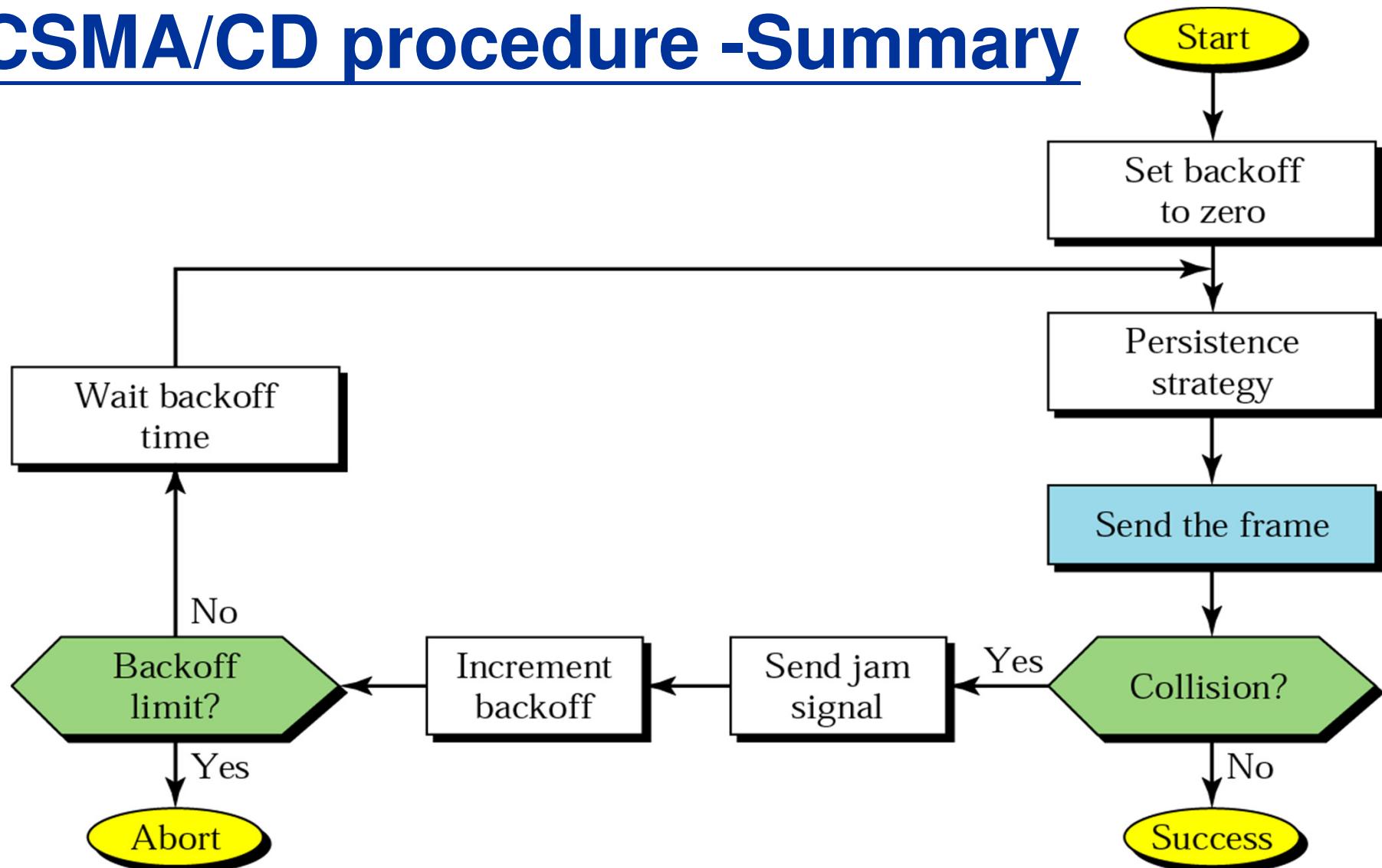
Figure 12.12 *Collision and abortion in CSMA/CD*



CSMA/CD (Cont.)

- ❖ To tell other stations that collision has occurred, the station continues to send a random bit pattern (known as **jam sequence/signal**) for a short period of time before stopping transmission
- ❖ The stations involved then wait for a random delay (**the back-off time**) before trying to retransmit the affected frames (i.e. those collided)
- ❖ For 1-persistent method the max throughput of using CSMA/CD is around 0.5
- ❖ For non-persistent method, the max throughput of CSMA/CD can go up to 0.9

CSMA/CD procedure -Summary



Legend

T_{fr} : Frame average transmission time

K : Number of attempts

R : (random number): 0 to $2^K - 1$

T_B : (Back-off time) = $R \times T_{fr}$

Fig12.13 Flow diagram for the CSMA/CD

Station has a frame to send

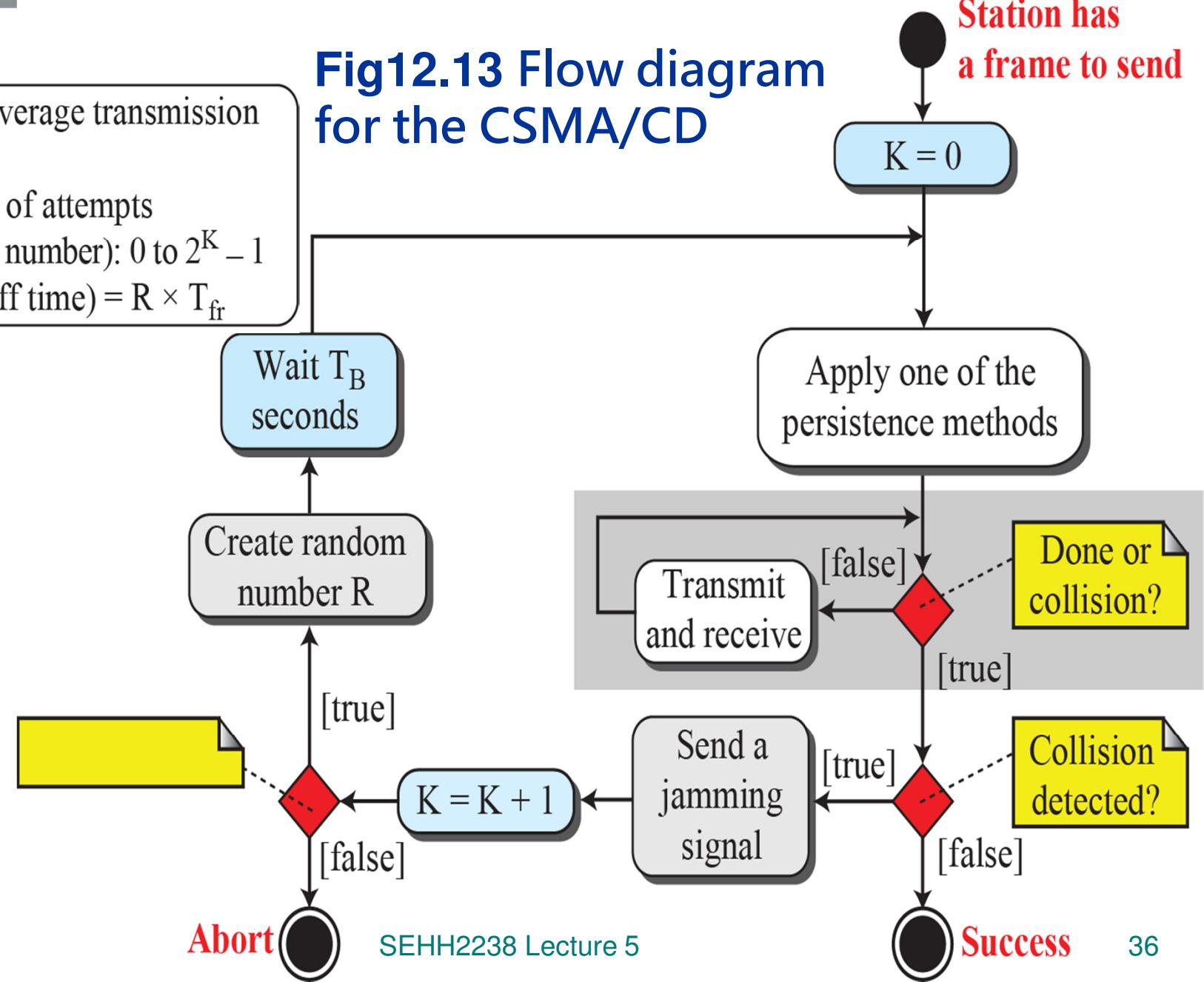
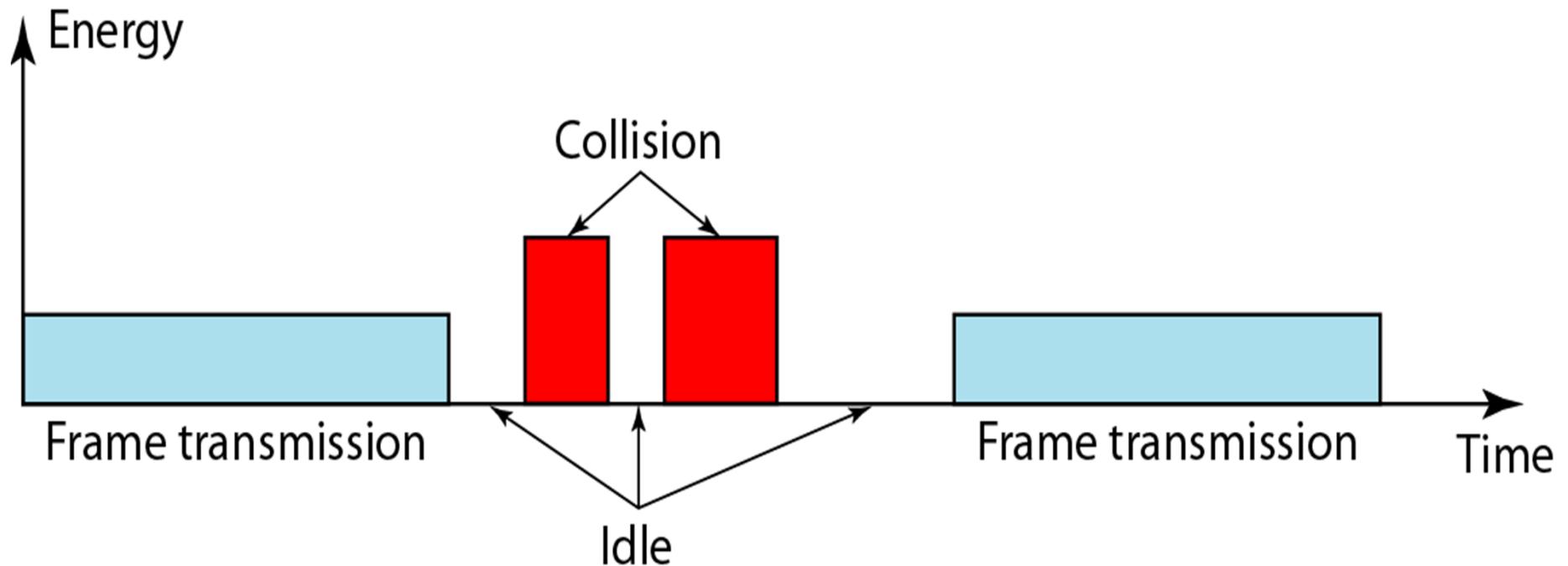


Figure 12.14 *Energy level during transmission, idleness, or collision*



A station can also monitor the energy level to determine the channel is idle, busy, or in collision

Time for detecting collision

- ❖ T_p is the time for a signal to propagate between the **farthest** stations
- ❖ In the **worst case** a station cannot be sure that it has seized the channel until it has transmitted for $2T_p$ without hearing a collision
- ❖ Therefore the longest time to detect collision is **the maximum round trip delay = $2T_p$**
- ❖ **$2T_p$ is also the minimum frame size (transmission time) required for proper operation of CSMA/CD**

Example 12.5

A network using CSMA/CD has a bandwidth of 10 Mbps. If the maximum propagation time (including the delays in the devices and ignoring the time needed to send a jamming signal) is 25.6 µs, what is the minimum size of the frame?

Solution

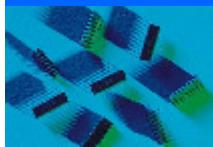
- The frame transmission time is $T_{fr} = 2 \times T_p = 51.2 \mu s$.
- This means, in the worst case, a station needs to transmit for a period of 51.2 µs to detect the collision.
- The minimum size of the frame is $10 \text{ Mbps} \times 51.2 \mu s = 512 \text{ bits or } 64 \text{ bytes}$.
- This is actually the minimum size of the frame for **Standard Ethernet**.

Back-off Time (optional, skip)

- ❖ How much is enough?
- ❖ Simplest: just double the back-off time if collide again
- ❖ Exponential back-off
 - ❖ In K^{th} attempt,
 - ❖ the station waits a random amount of time between:
0 to $(2^K - 1) \times T_{\text{fr}}$
 - ❖ where T_{fr} is the average frame transmission time
- ❖ Back-off Limit
 - ❖ If the number of retry exceeds the pre-set limit in back-off (usually 15), the station has tired enough and abort the procedure

Summary

- ❖ MAC protocols to be use in a bus channel
 - ❖ Pure ALOHA – max throughput 0.18
 - ❖ Slotted ALOHA – max throughput 0.36
 - ❖ CSMA - Carrier Sense Multiple Access
 - ❖ Listen before transmission
 - ❖ 1-persistent CSMA/CD – max throughput ≈ 0.5
- ❖ All – If collided, retry after a random delay
- ❖ **Revision Quiz**
 - ❖ http://highered.mheducation.com/sites/0073376221/student_view0/chapter12/quizzes.html



Lecture 6

Ethernet, Wireless LAN & Internetworking

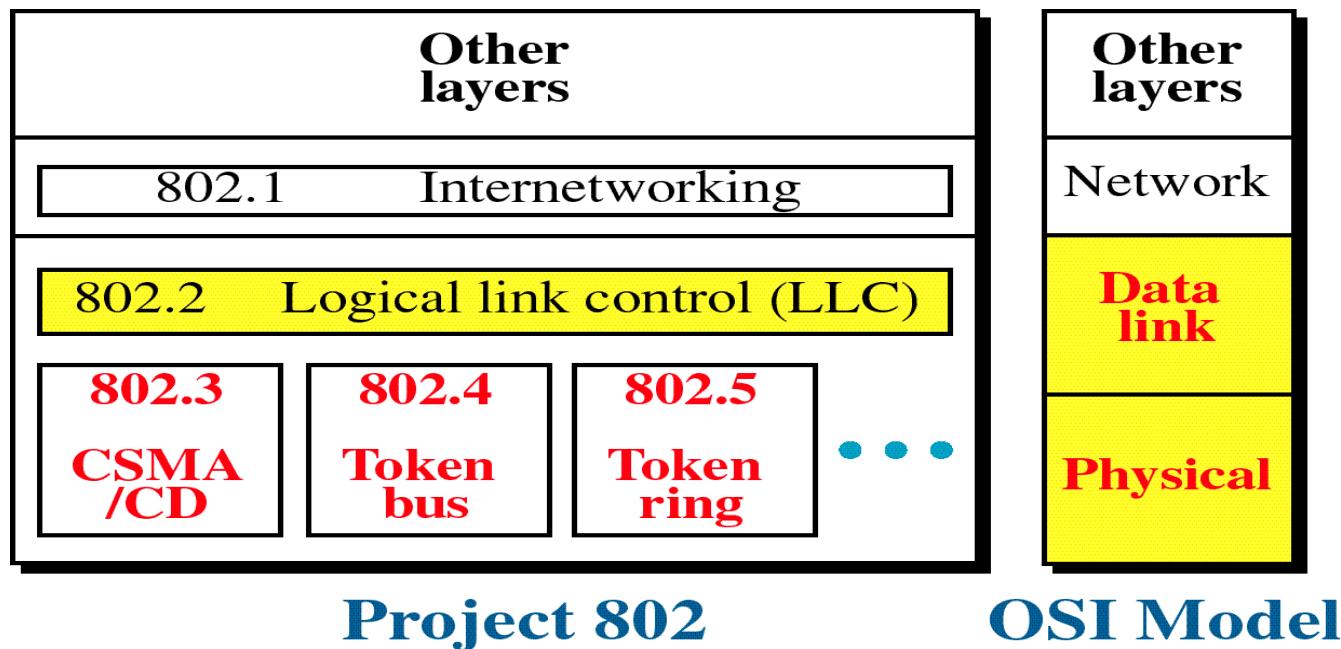
Textbook: Ch.8, 13 and 15

Main Topics

- ❖ IEEE 802.3 Ethernet
- ❖ IEEE 802.11 Wireless LAN
- ❖ WAN & Internetworking
- ❖ Circuit-switched Network
 - ❖ Phases
 - ❖ Delay in a circuit-switched network
- ❖ Packet Switching
- ❖ Datagram Approach
 - ❖ Store-and-Forward operation
 - ❖ Delay in a datagram network
- ❖ Virtual Circuit Approach

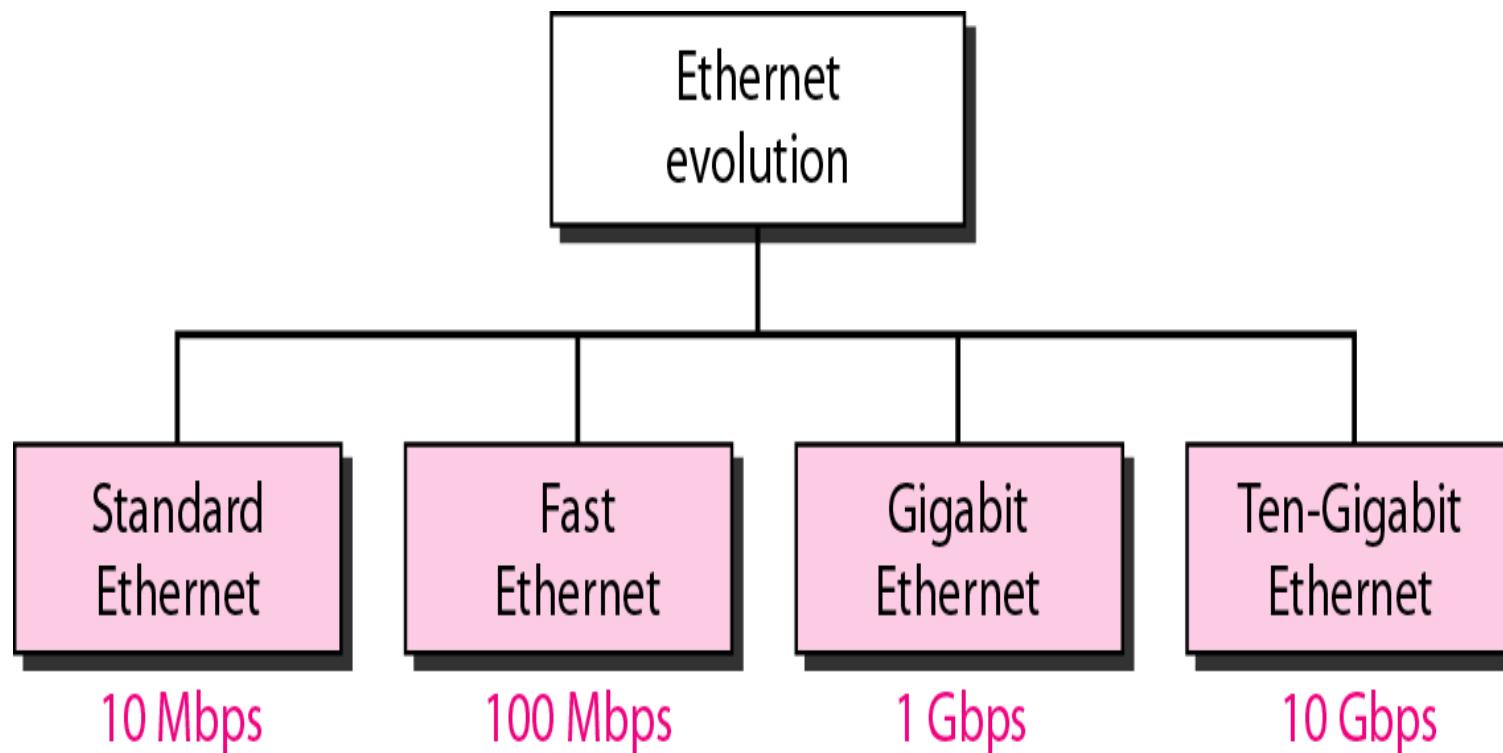
IEEE 802 Standard for LANs

- *In 1985, the Computer Society of the IEEE started a project, called Project 802, to set standards to enable intercommunication among equipment from a variety of manufacturers.*
- *Project 802 is a way of specifying functions of the physical layer and the data link layer of major LAN protocols.*



Standard Ethernet

❖ *Ethernet evolution through four generations*



Network Components of Ethernet

- ❖ **Communication controller card (*NIC, network interface card*)** in the station (computer) contains:
- ❖ **MAC unit** for such functions as encapsulation, error detection & execution of MAC algorithm
- ❖ **Transceiver** : transmitter and receiver in one unit (*also called MAU, medium attachment unit*) - part of the NIC
 - ❖ send & receive data from cable
 - ❖ detect occurrence of collisions

MAC Protocol of Ethernet

- ❖ Bus topology with a broadcast channel (usually a coaxial cable)
- ❖ Access Method
 - ❖ 1-persistent CSMA/CD Bus (*IEEE 802.3*)

Frame Format of Ethernet

Preamble: 56 bits of alternating 1s and 0s.

SFD: Start frame delimiter, flag (10101011)

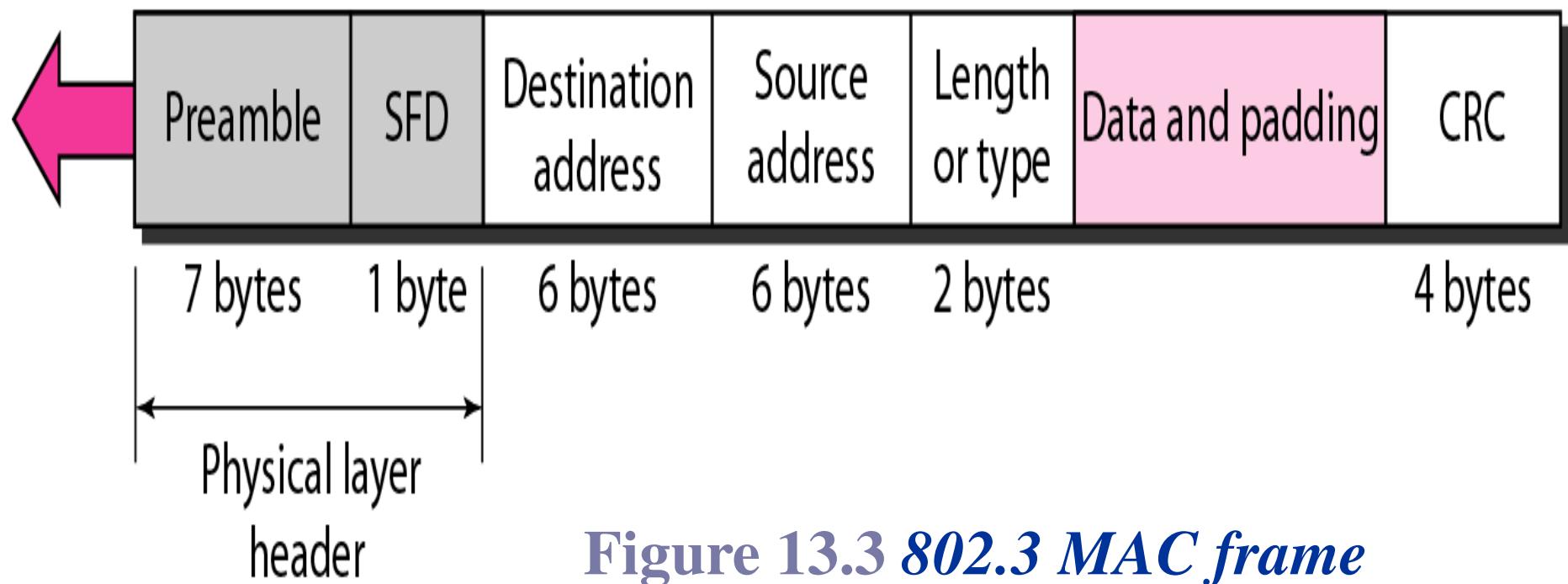


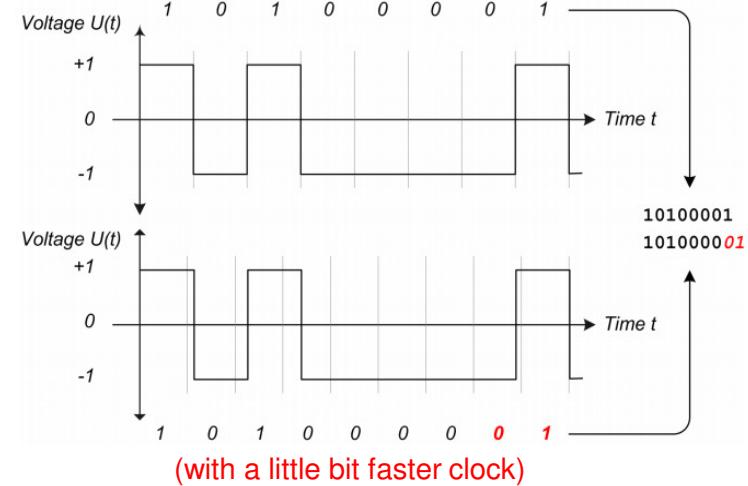
Figure 13.3 802.3 MAC frame

Ethernet Frame Format & Parameters

- ❖ Ethernet Frame contains seven fields:

1. Preamble

- ❖ contains $7 \times (10101010)$ for bit synchronization



2. Start of Frame Delimiter (SFD)

- ❖ 10101011 (also as a last chance for synchronization)
- ❖ signals the beginning of the frame

Ethernet Frame Fields

3. Length (or type)

- ❖ the packet length **in bytes (excluding preamble and SFD)**

4. Destination Address (DA)

- ❖ 6 bytes containing the **physical address** of the destination station or stations to receive the packet

5. Source Address (SA)

- ❖ 6 bytes containing the physical address of the sender of the packet

Ethernet Frame Fields

6. Data

- ❖ carries data encapsulated from the upper-layer protocols
- ❖ **data length: a *minimum* of 46 bytes and a *maximum* of 1500 bytes**
- ❖ if length < minimum frame size, then dummy bytes are added (known as **padding**) in the data field

7. CRC

- ❖ CRC-32 for error detection

Ethernet Frame Length

- ❖ Minimum length:

- ❖ Remember in CSMA/CD, a minimum length restriction is required for correct operation

- ❖ Maximum length:

- ❖ Reduce the size of buffer in memory

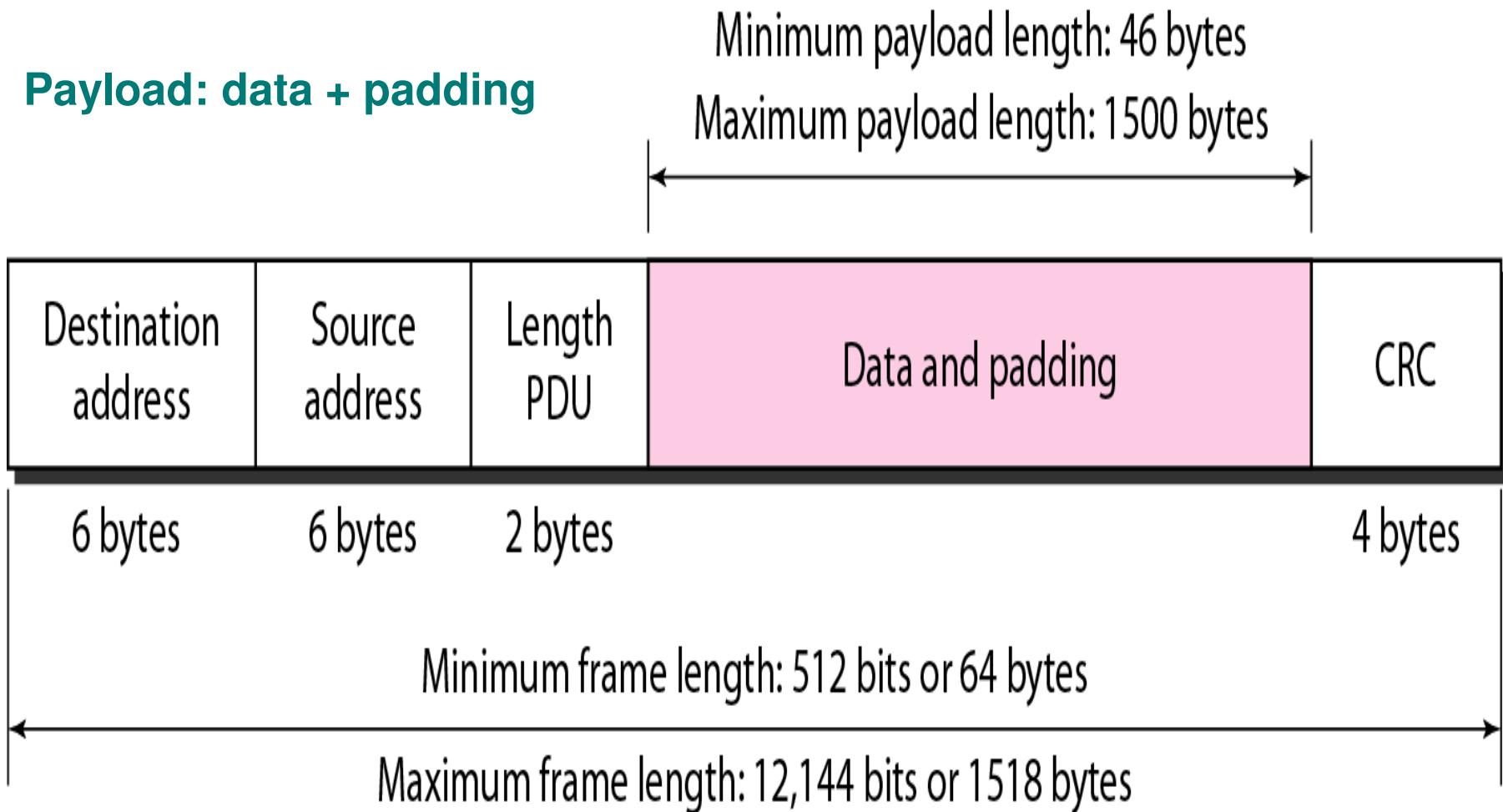
- ❖ Prevent one station from monopolizing the shared channel (using the channel too long)

Frame length:

Minimum: 64 bytes (512 bits)

Maximum: 1518 bytes (12,144 bits)

Figure 13.5 Minimum and maximum lengths



Wireless LAN - IEEE 802.11

- ❖ IEEE 802.11 is the **IEEE** specifications for a wireless LAN.
- ❖ **Infrastructure** (architecture): uplink and downlink via the **access points** (base stations)
- ❖ **Transmission media**: Infrared or radio signal using spread spectrum techniques
- ❖ Use **CSMA/CA** (collision avoidance) protocol to organize the transmissions from mobile stations
- ❖ Wireless (radio) networks *cannot* use the CSMA/CD protocol (skip the details:15.1.3, p.438-439)

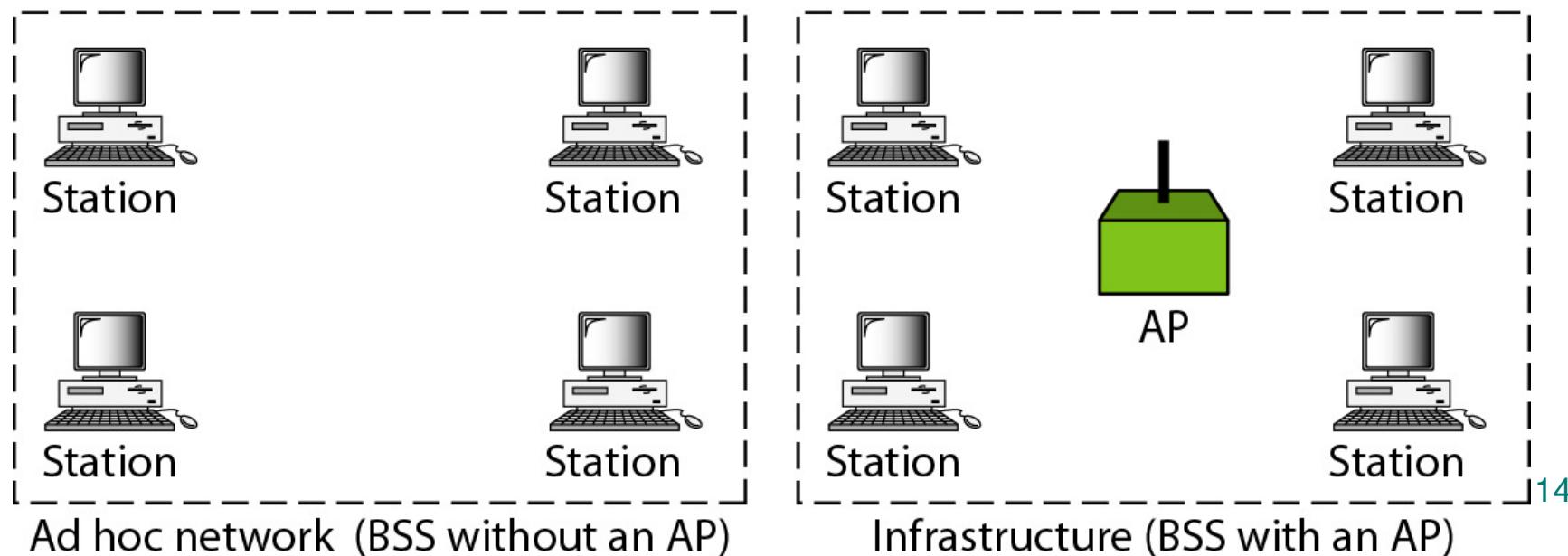
Architecture of Wireless Network

- ❖ Two kinds of services: BSS and ESS
 - ❖ Basic Service Set (BSS)
 - ❖ It made up of stationary or mobile wireless stations.

A BSS without an AP is called an **ad hoc network;**
a BSS with an AP is called an **infrastructure network.**

BSS: Basic service set

AP: Access point



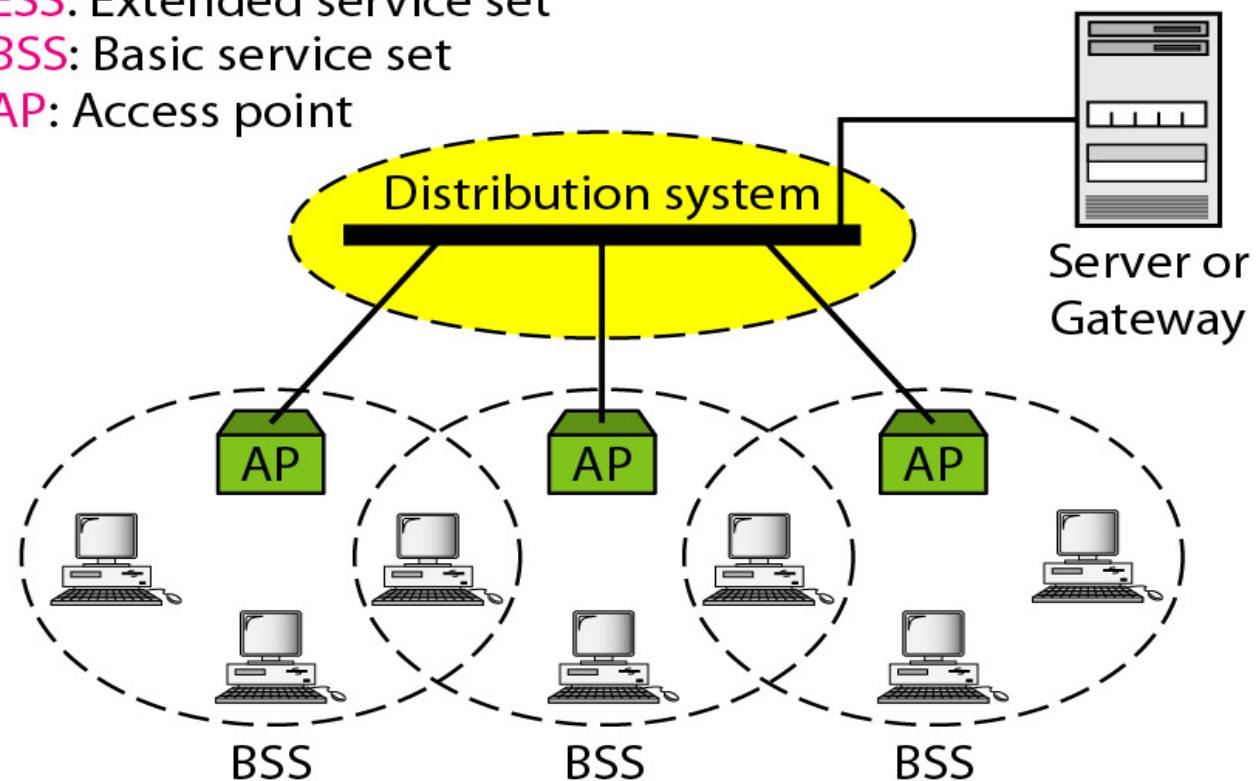
Extended Service Set (ESS)

- ❖ It made up of two or more BSSs with **Access Points (AP)**.
- ❖ BSSs are connected through a *distribution system* (usually Wired LAN)

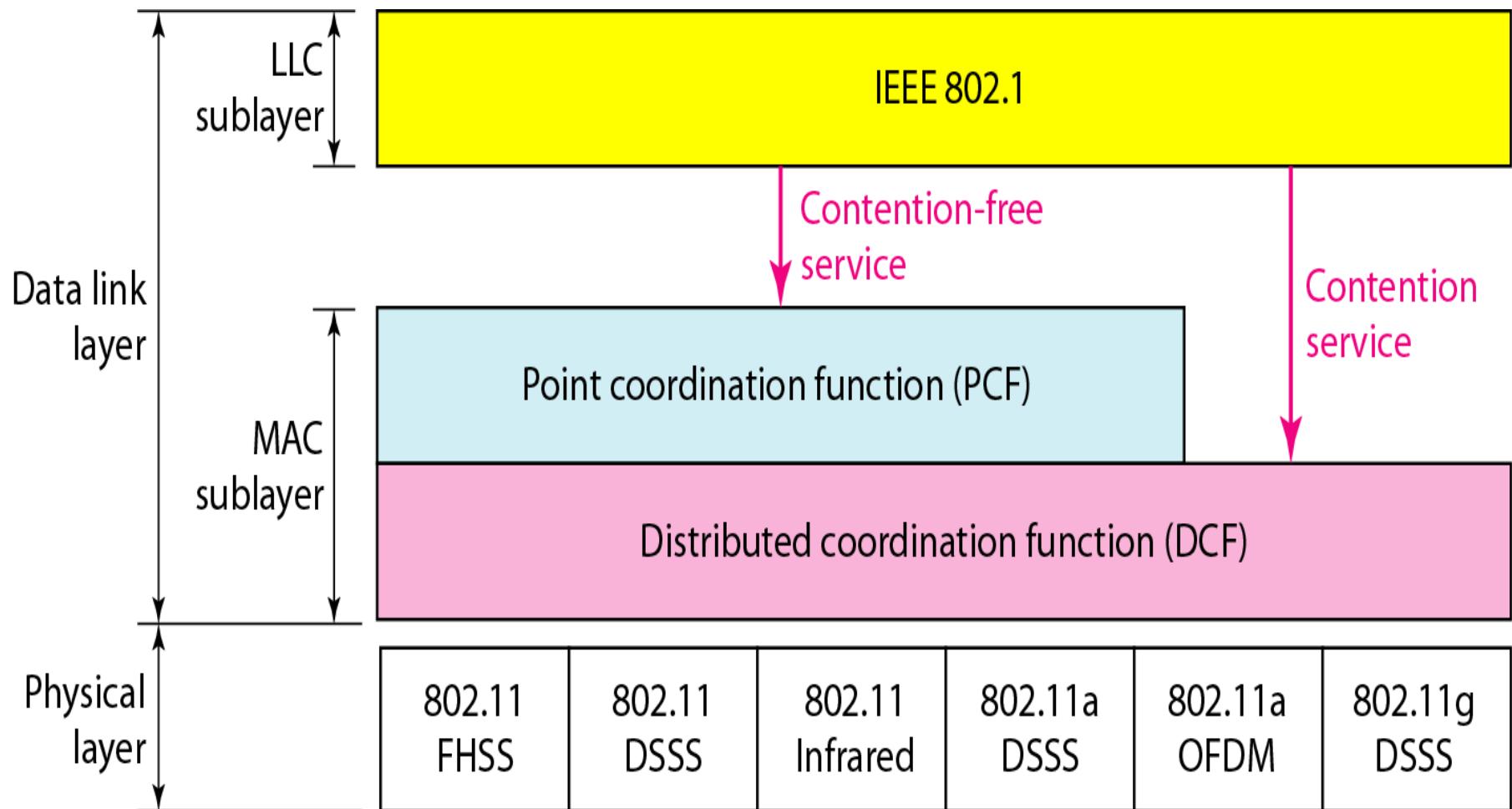
ESS: Extended service set

BSS: Basic service set

AP: Access point



MAC layers in IEEE 802.11 standard



(Skip the details)

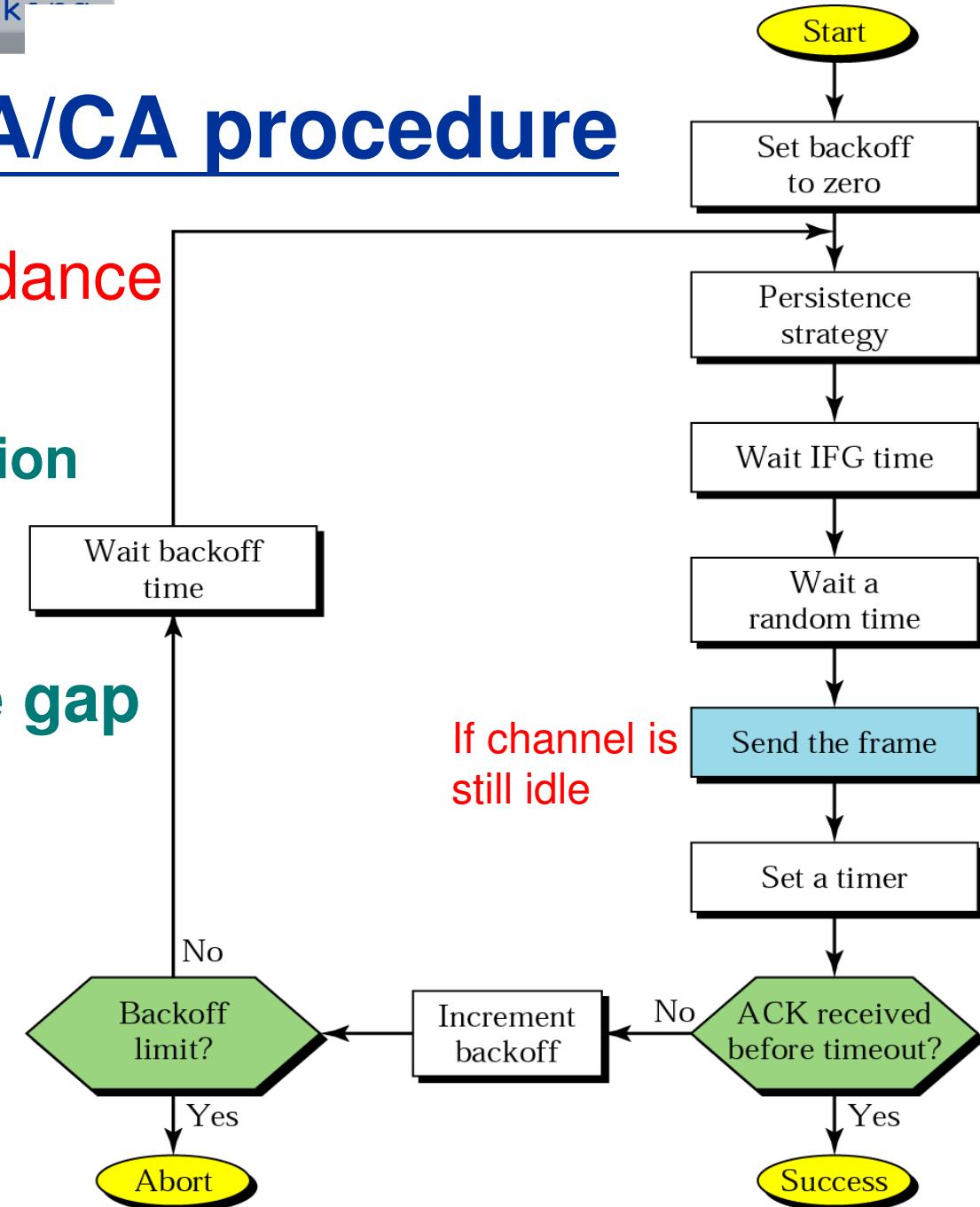
SEHH2238 Lecture 6

Figure 15.6

16

CSMA/CA procedure

- ❖ CA: Collision Avoidance
- ❖ Key difference:
 - ❖ There is no collision detection
- ❖ IFG is inter-frame gap



(Skip the details)

Wide Area Network (WAN)

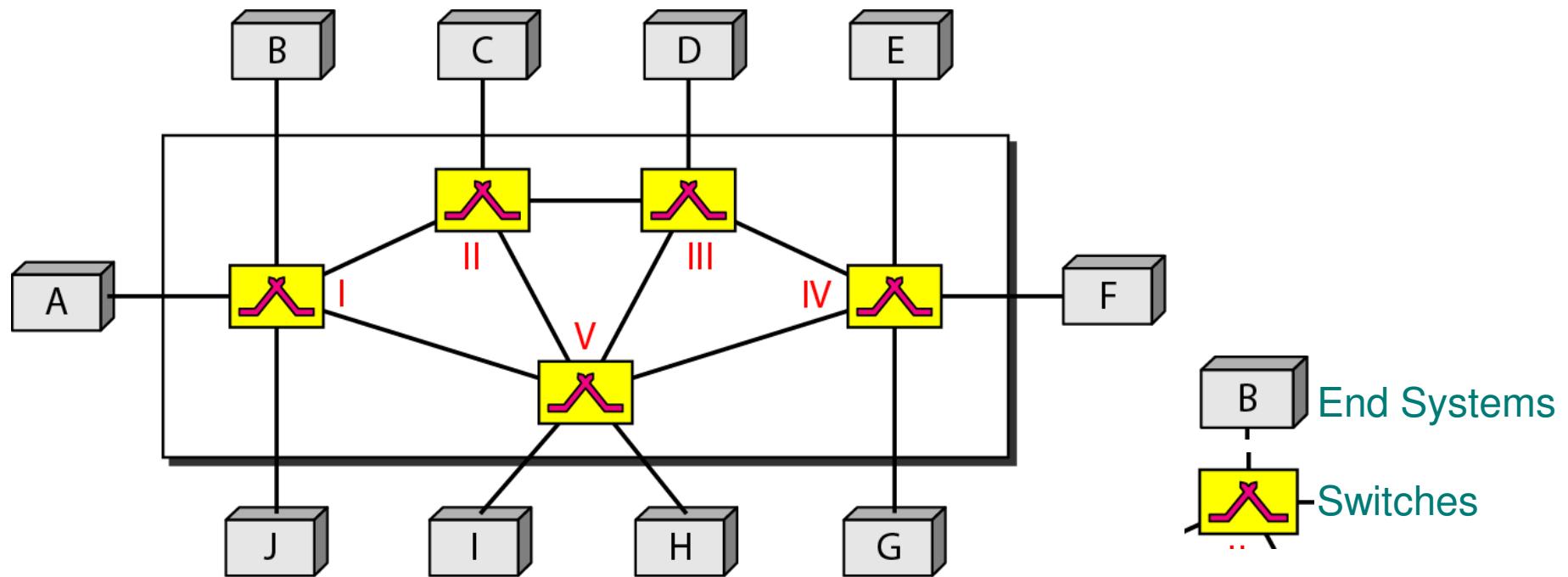
- ❖ A network that links stations and LANs that are physically located in different geographic areas
- ❖ Include both public data networks and enterprise wide private data networks
- ❖ Three major concerns/functions in internetworking:
 - ❖ Routing
 - ❖ Congestion Control
 - ❖ Flow Control

Three major concerns on WANs

- ❖ **Routing**
 - ❖ determine how packets are routed from source to destination (i.e. select the best path)
- ❖ **Congestion Control**
 - ❖ make sure the network is able to carry the offered traffic
 - ❖ a ***global issue*** involves all stations and routers
- ❖ **Flow Control**
 - ❖ make sure that a fast sender cannot continually transmit data faster than the receiver can accept
 - ❖ a ***local issue*** between a given sender and a given receiver

Ch.8 Switching

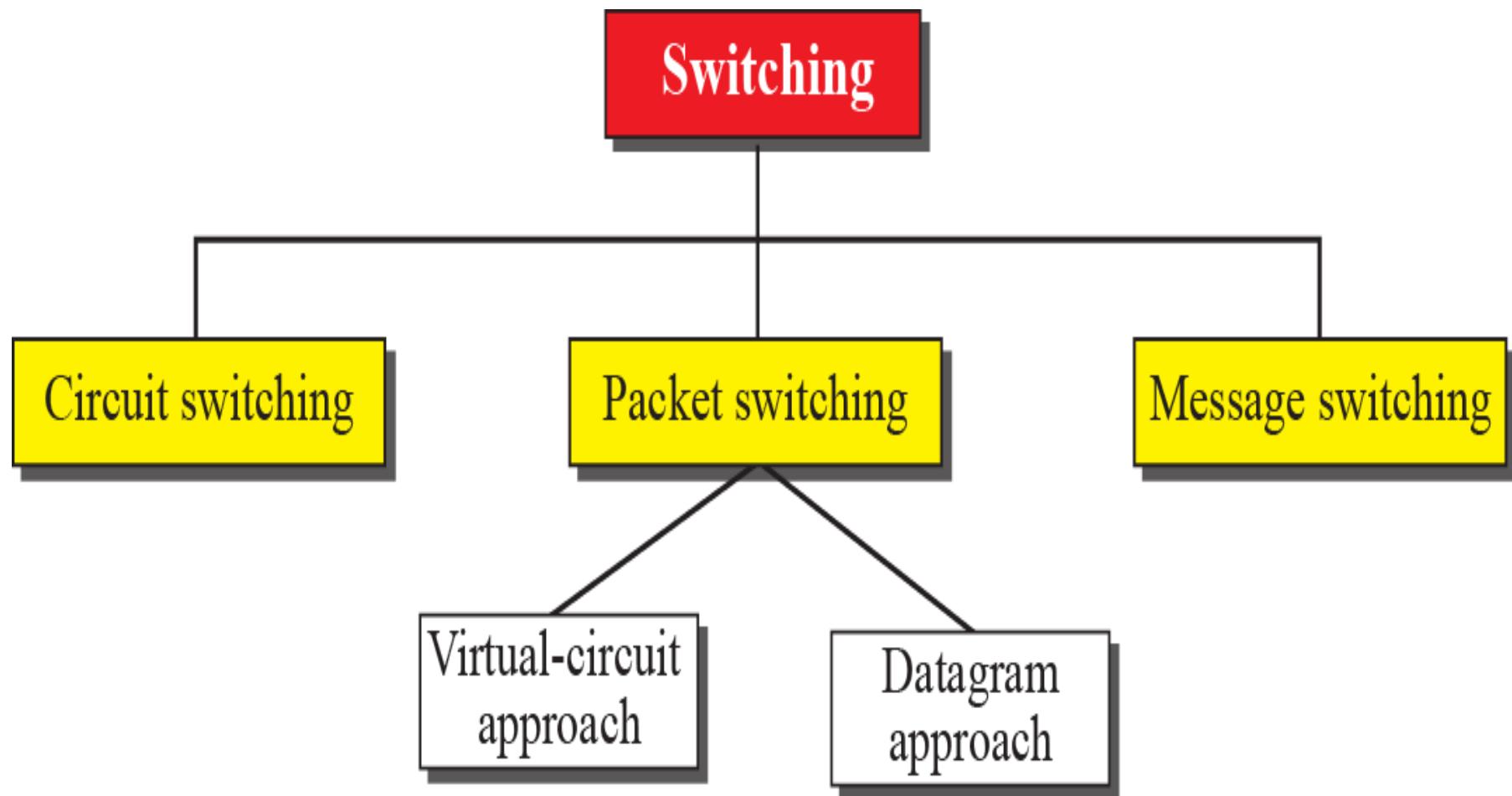
- ❖ A switched network consists of a series of interlinked nodes, called **switches**.
- ❖ Switches are devices capable of creating temporary connections between two or more devices linked to the switch.



Three Methods of Switching

- ❖ Switching is a method in which communication devices are connected to one another efficiently.
- ❖ Traditionally three methods of switching: ***circuit switching, packet switching, and message switching.***
- ❖ The first two are commonly used today. The third has been phased out in general communications but still has applications.
- ❖ Packet switching can further be divided into two subcategories, ***virtual-circuit approach and datagram approach.***

Figure 8.2: Taxonomy of switched networks



Circuit-switched Network

- ◆ It consists of a set of switches connected by physical links.
- ◆ A connection between two stations is a **dedicated path** made of one or more links.
- ◆ However, each connection uses only one dedicated channel on each link (which is normally divided into n channels by using FDM or TDM).
- ◆ In ***circuit switching***, the **resources** need to be reserved during the **setup phase**.
- ◆ The **resources** remain **dedicated** for the entire duration of data transfer until the **teardown phase**.

Phases in Circuit-switched Network

❖ Setup phase

- ❖ A channel is reserved on each link and the dedicated path is defined.

❖ Data Transfer phase

- ❖ Two parties can transfer data.

❖ Teardown Phase

- ❖ When one of the parties needs to disconnect, a signal is sent to each switch to release the resource.

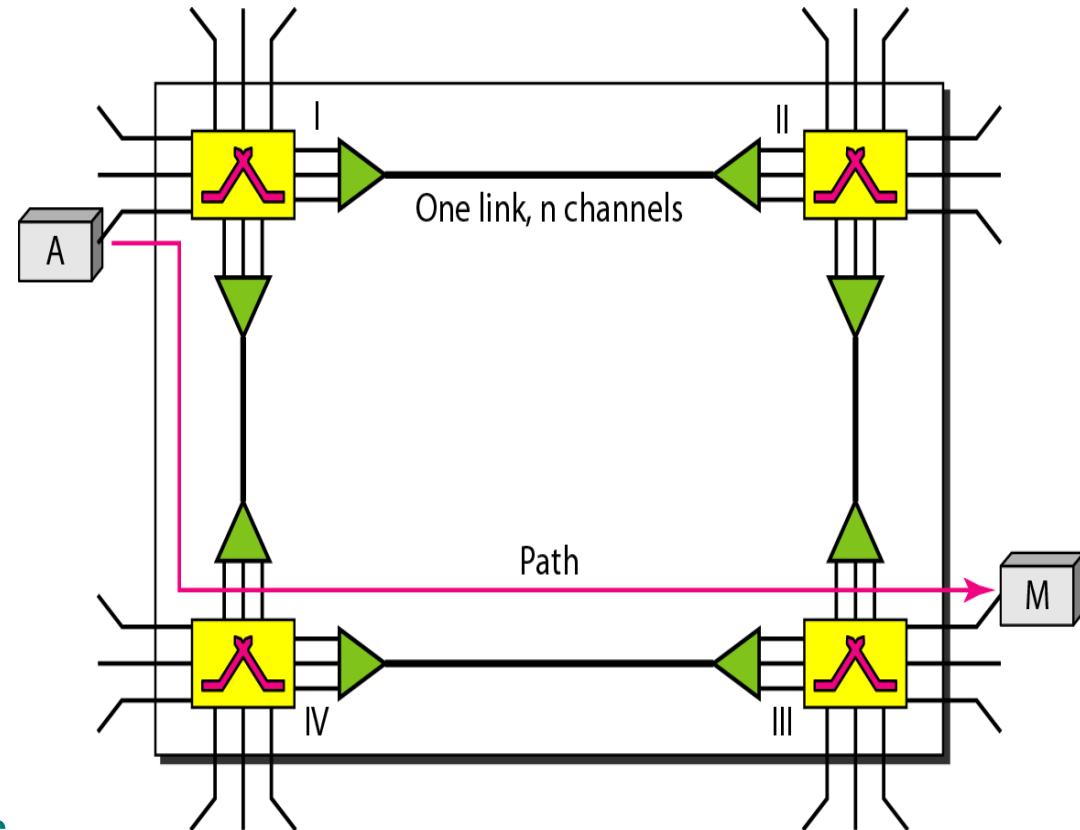


Figure 8.3 *A trivial circuit-switched network*

Circuit Switching

- ❖ A (**temporary**) **dedicated path** (just like a point-to-point link) between the source and the destination is provided for the duration of data transmission (called **session**). It is similar to a telephone call.
- ❖ Advantage
 - ❖ Throughput and delay characteristics are predictable
- ❖ Disadvantages:
 - ❖ Waste the capacity of the links (when no data within a session)
 - ❖ Connection establishment and disconnection are relatively time-consuming
 - ❖ Possibility of blocking (stop new data input) when traffic is heavy

Delay in a circuit-switched network

- ❖ Note that during data transfer the data are not delayed at each switch, as **no waiting time is required inside each switch**

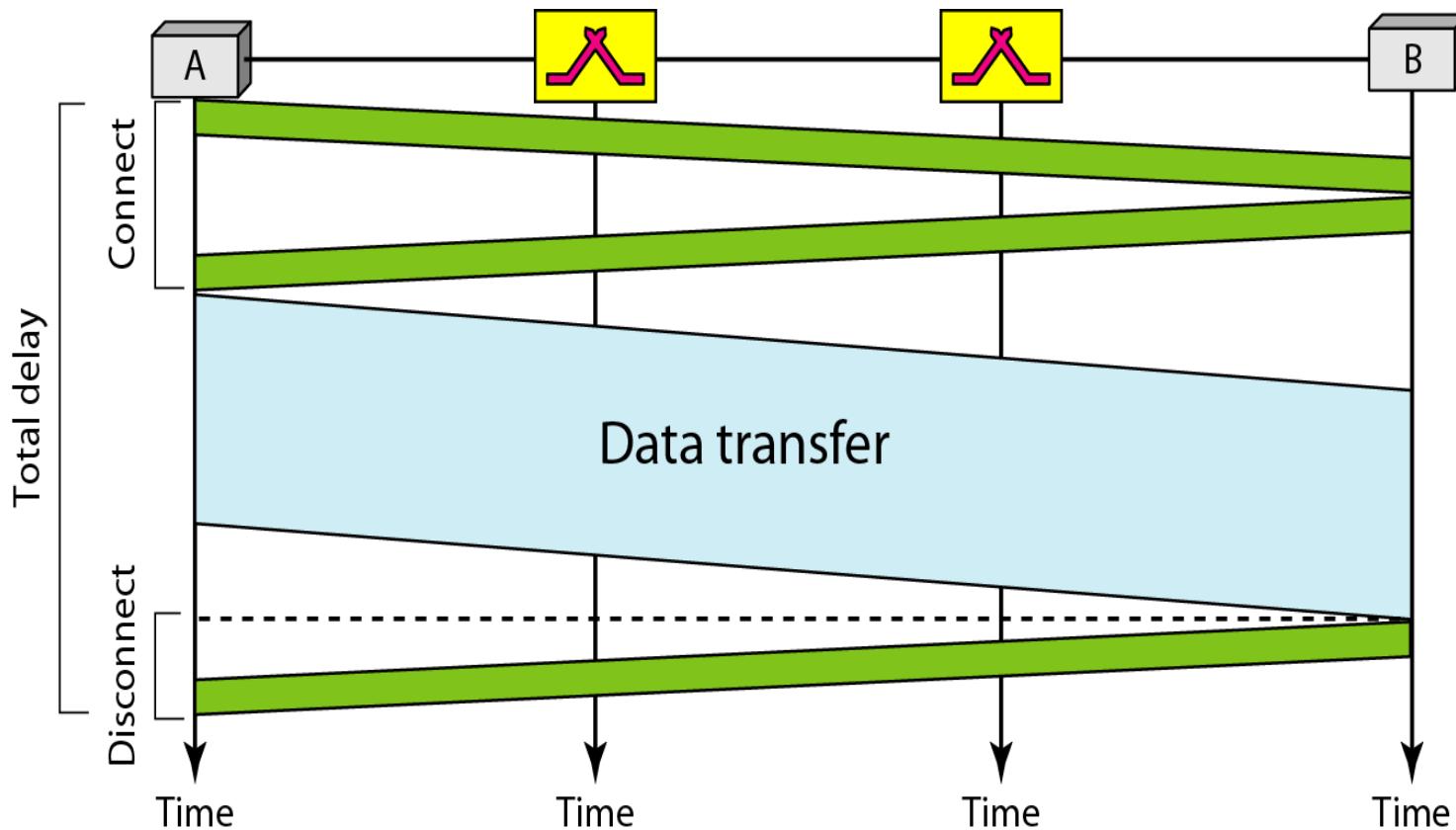


Figure 8.6

Packet Switching

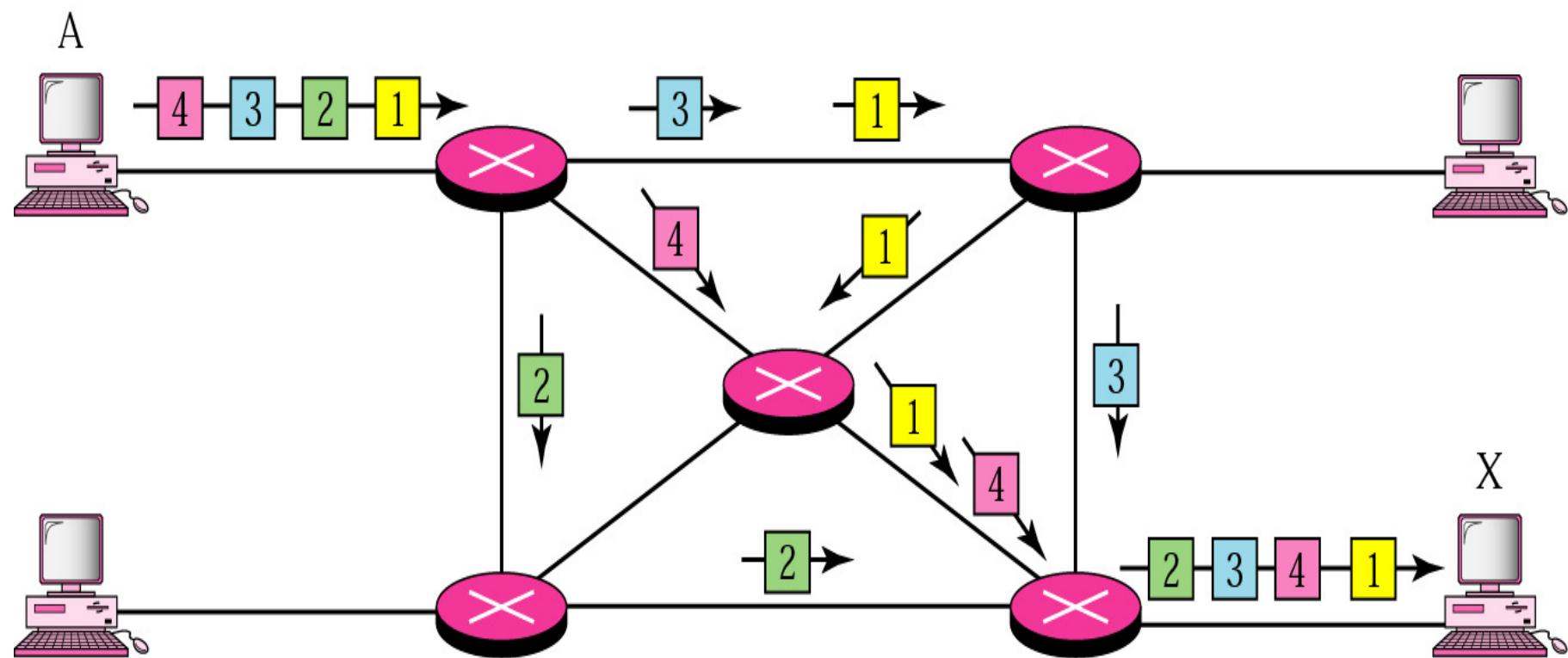
- ❖ **Packetizing:** the data message needs to be divided into packets of fixed or variable size
- ❖ The size of the packet is determined by the network and the governing protocol
- ❖ **Encapsulating** the payload in a network-layer packet *at the source*
- ❖ **Decapsulating** the payload from the network-layer packet *at the destination*
- ❖ Use **Store-and Forward** operation
- ❖ **Virtual-circuit approach and Datagram approach**

Datagram Networks

- ❖ It is a **connectionless service** of packet switching
- ❖ Each packet (of the same message) is sent out **independently**
- ❖ **No connection set up** is required
- ❖ Does not guarantee delivery of error-free and sequenced data
- ❖ Packets of the same message may travel along different paths via different intermediate nodes (thus **re-sequencing** is needed at the destination node)
- ❖ Users must handle error & flow control themselves
- ❖ Packets in this approach are referred to as **datagrams**

Main Features of Datagram Packet Switching

- ❖ Message is divided into **packets of fixed (maximum) size**
- ❖ There is **no resource reservation**
- ❖ Resources are allocated on demand
- ❖ Packets (of the same message) are only reassembled at the destination node



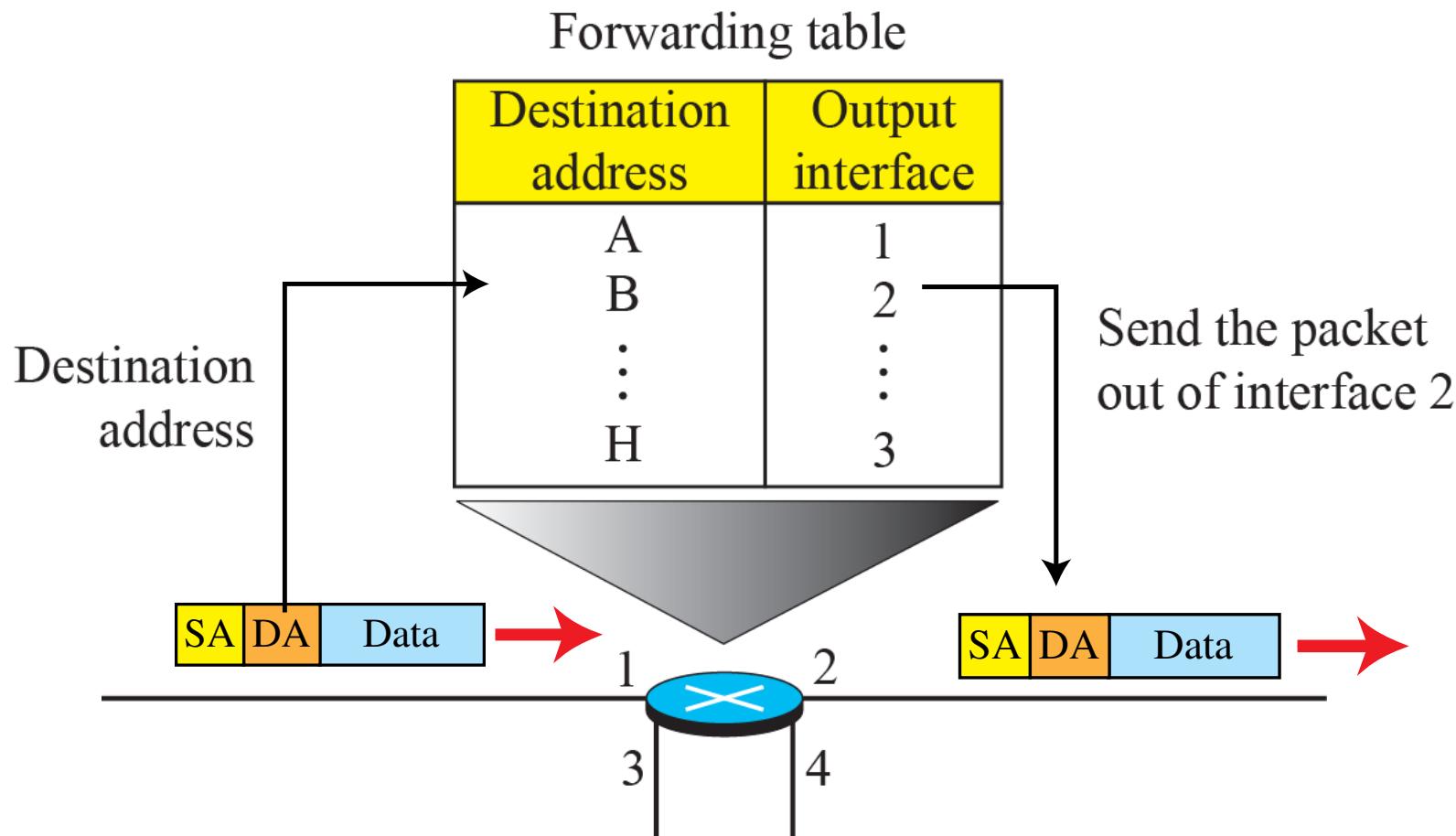
Store-and-Forward Operation

- ❖ Each cable (channel/link) connects a pair of nodes (**Point-to-Point Channel**)
- ❖ If no direct link between two nodes, they must communicate indirectly (via other nodes)
- ❖ The packet is received at each intermediate node, be stored there until the output link is free, and then be forwarded to another node
- ❖ A routing decision is made to select the next intermediate node before forwarding

Forwarding Process & Routing Table

- A switch (router) uses a routing table (*forwarding table*) to determine the output port
- It is ***based on the destination address, (for datagram approach)*** which ***remains the same*** during the entire journey of the packet.

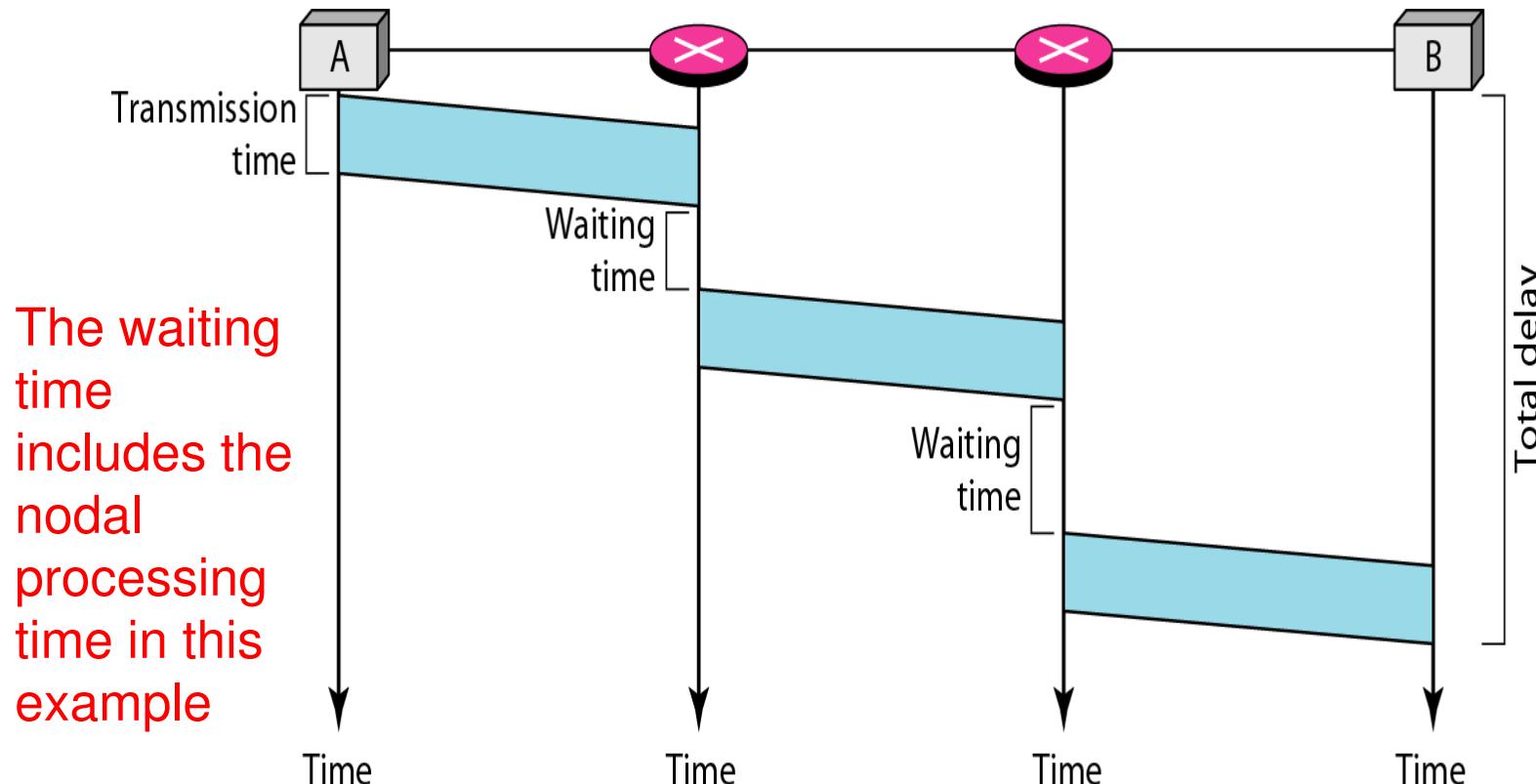
Figure 18.4: Forwarding process in a router when used in a connectionless datagram network



Delay in a Datagram Network

- ❖ *E.g. A packet travels two switches. There are*
 - ❖ *3 transmission times ($3 T_x$)*
 - ❖ *3 propagation delays ($3 T_p$)*
 - ❖ *2 waiting times (w_1 , and w_2)*
 - ❖ *The total delay = $3T_x + 3 T_p + w_1 + w_2$*

Figure 8.9



Circuit Switching and Datagram

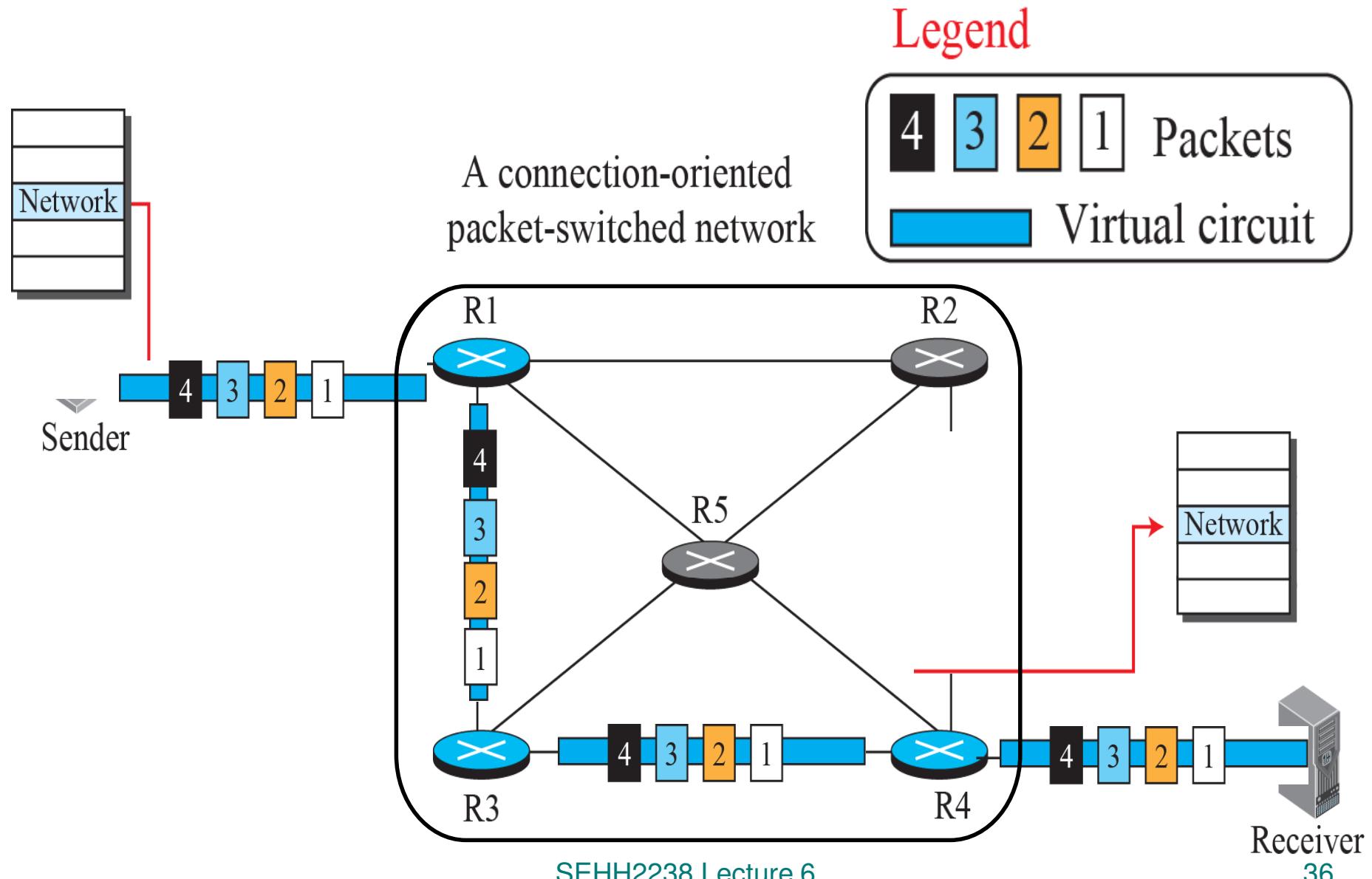
Switching at the *physical layer* in the traditional telephone network uses the circuit-switching approach.

Switching in the Internet is done by using the datagram approach of packet switching at the *network layer*.

Virtual Circuit (VC) Approach

- In a **connection-oriented service** (also called virtual-circuit approach), there is a relationship between all packets belonging to a message.
- Before all datagrams in a message can be sent, a **virtual connection** should be set up to define the path for the datagrams.
- After connection setup, the datagrams can **all follow the same path using store-and-forward operation**.
- In this type of service, not only must the packet contain the source and destination addresses, it must also contain a flow label, a **virtual circuit identifier (VCI)** that defines the virtual path the packet should follow.

Figure 18.5: A virtual-circuit packet-switched network



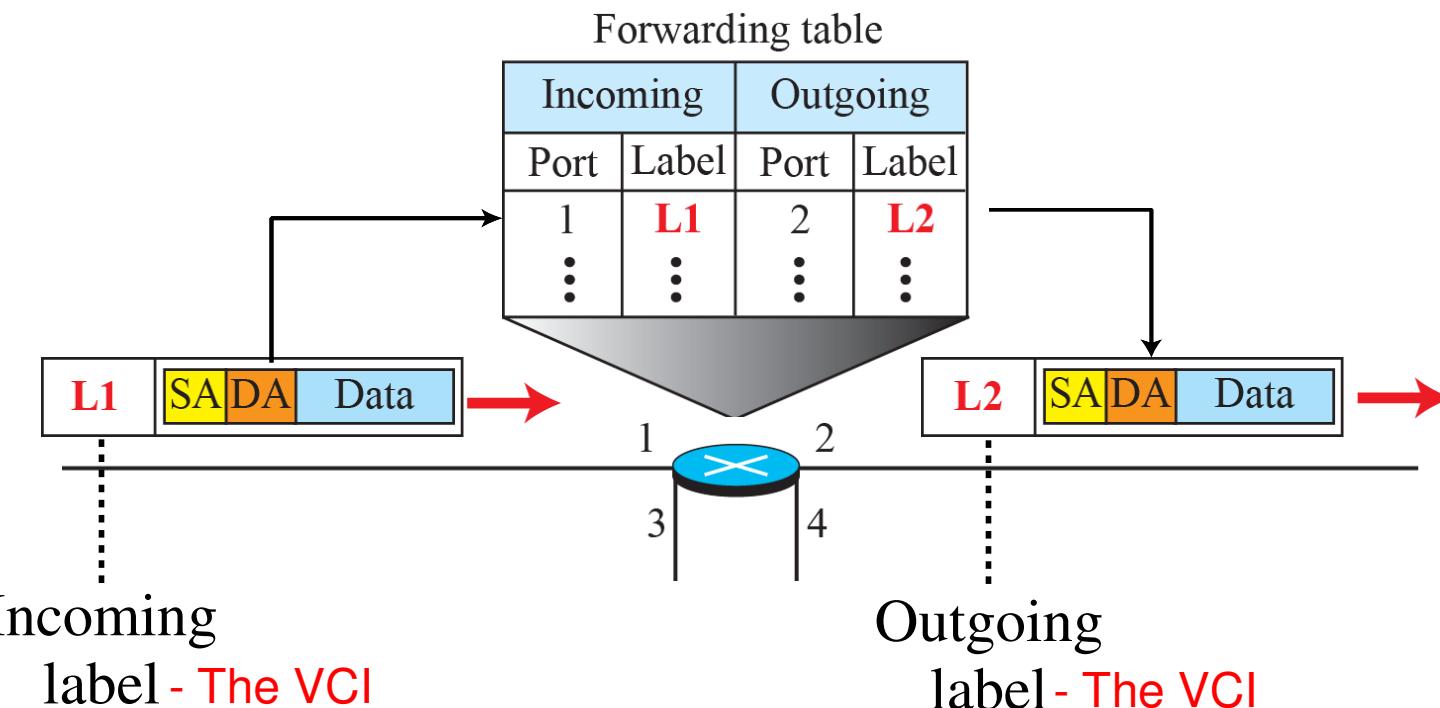
Virtual Circuit Approach

- ❖ **Virtual circuit** approach (in packet switching) can be considered as a mix of *circuit switched* (CS) and *packet switched* (PS) networks.
 - ❖ Phase: Setup, data transfer, teardown (CS)
 - ❖ A “virtual” path is set up (**to book the resource**) before data transfer
 - ❖ Data are packetized. Each packet carries an address (and VCI) in the header (PS)
 - ❖ All packets of the same message **follow the exact (same) route** (indicated by the VCI) (CS)
 - ❖ But the physical path is **not dedicated** (and may be shared by other connections) (PS)

Virtual Circuit Approach

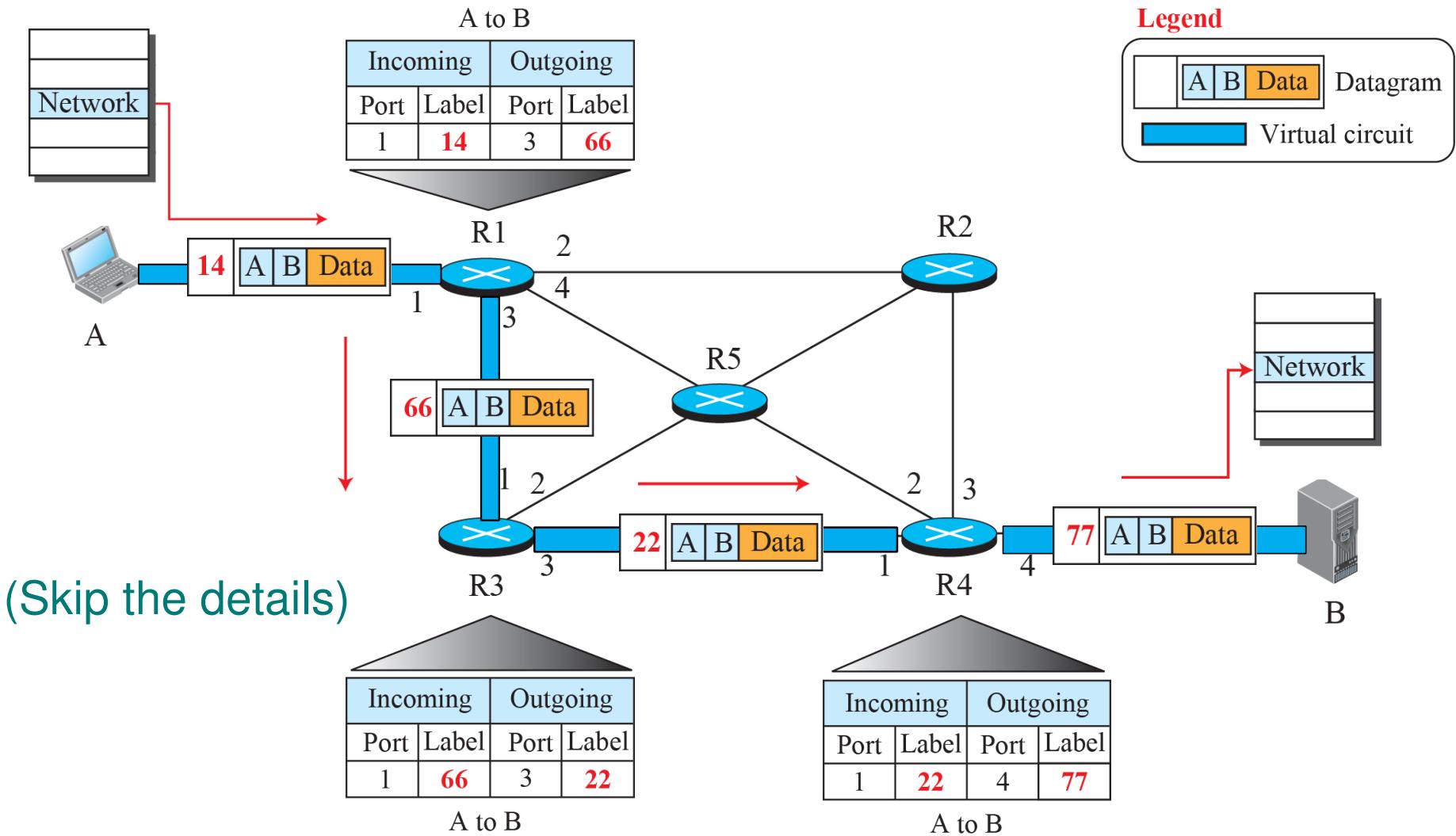
- ❖ This “**service**” provides with a “**perfect channel**” and **guarantees error-free and sequenced data**
- ❖ The packets may arrive at the destination with different delays
- ❖ The complicated communication issues (e.g. error and flow control, re-sequencing of data packets) are handled by the (VC) service provider

Figure 18.6: Forwarding process in a router when used in a virtual circuit network



(Skip the details)

Figure 18.9: Flow of one packet in an established virtual circuit



Delay in a virtual-circuit network

- ❖ *E.g. A packet travels two switches.*
 - ❖ *The total delay = $3T_x + 3 T_p + \text{Setup delay} + \text{teardown delay}$*
 - ❖ *(The nodal processing time usually can be neglected. Also assume no waiting time at each node in this example)*

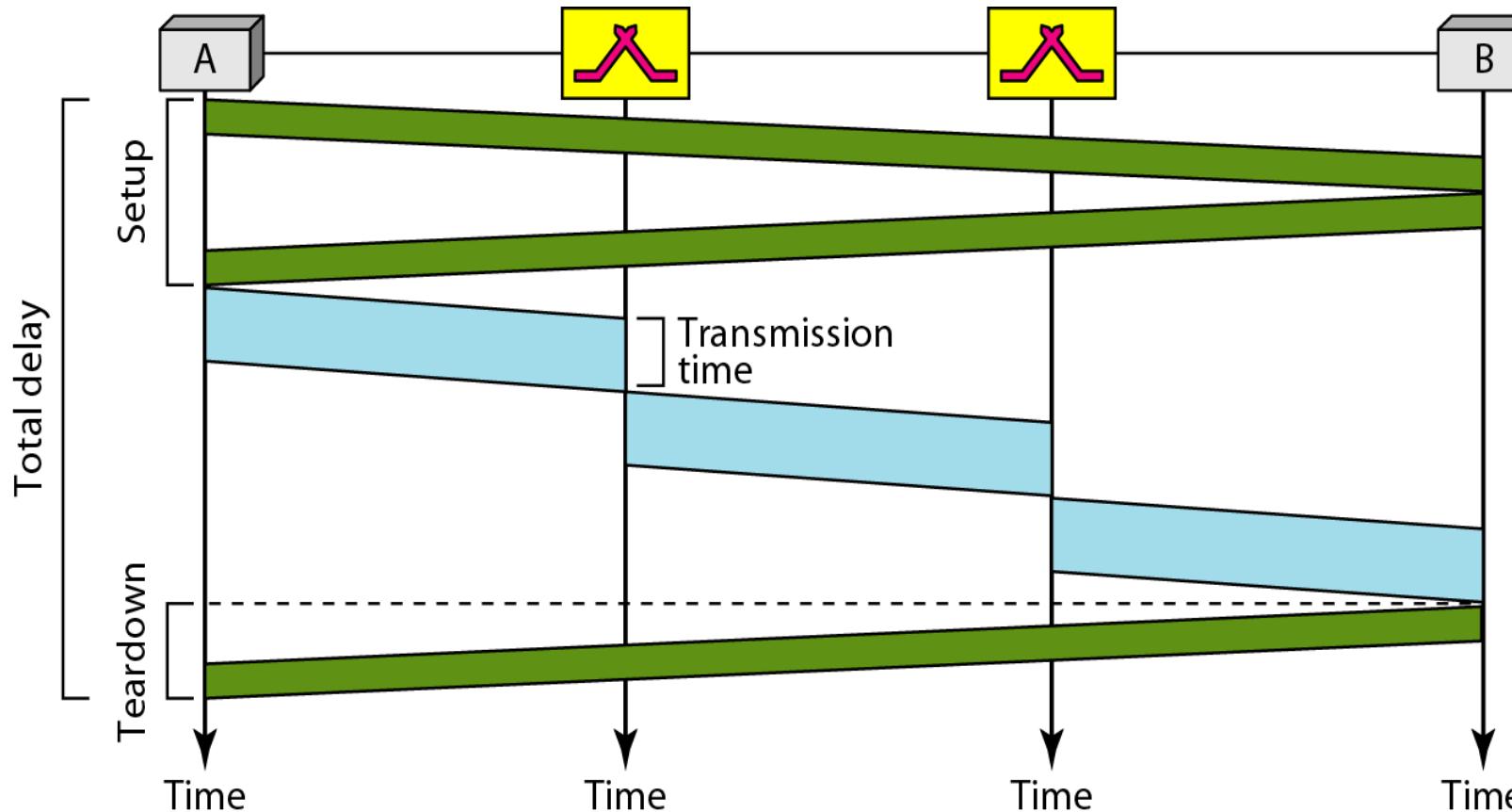
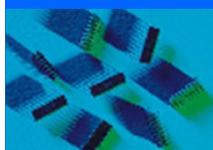


Figure 8.16

Summary

- ❖ IEEE 802.3 Ethernet - 1-persistent CSMA/CD Bus
- ❖ IEEE 802.11 Wireless LAN – CSMA/CA
- ❖ Circuit Switching - A (temporary) dedicated path
- ❖ Packet Switching - Store-and-Forward operation
- ❖ Datagram Approach - Connectionless, Datagrams
- ❖ Virtual Circuit Approach - Connection-oriented
 - ☞ Packets travel along the same path
- ❖ Revision Quiz
 - ☞ http://highered.mheducation.com/sites/0073376221/student_view0/chapter8/quizzes.html



Lecture 7

IP Addressing and Subnets

Textbook: Ch.18

Main Topics

1. IPv4 Address
 - ❖ Classful and Classless Addressing
2. Network/Address Mask
3. Subnets
4. Design Examples

1) Internet Protocol (IPv4) Address

- ❖ Internet Protocol version 4 (IPv4)
- ❖ 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. (It is a logical address.)*
- ❖ IPv4 address is commonly called **IP address**
- ❖ The IP address is the **address of the connection**, not the host or the router
- ❖ All hosts on a given network have a **common prefix** in their IP address

IP Address

An Internet address is made of four bytes (32 bits) that define a host's connection to a network.

Class :
Type :

Netid

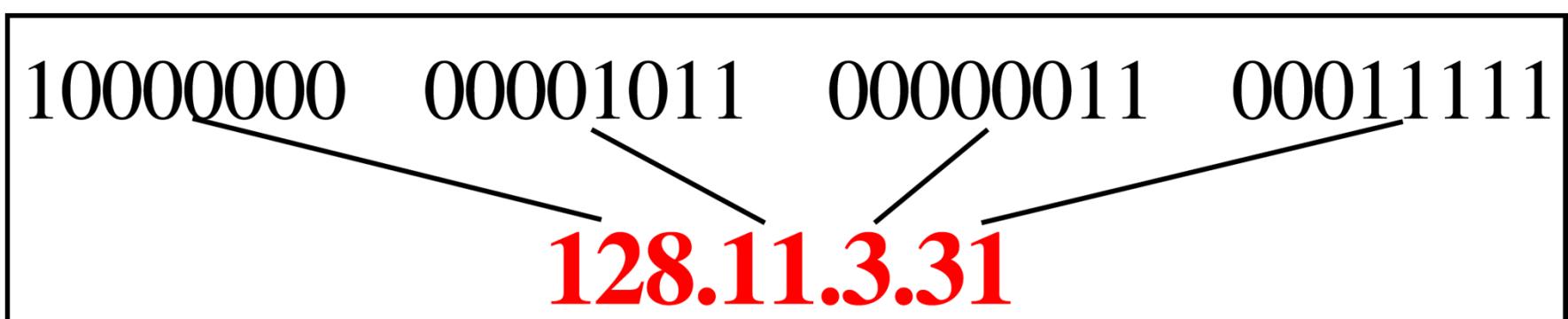
Hostid

An IPv4 address is 32 bits long.

**The address space of IPv4 is
 2^{32} or 4,294,967,296.**

IP Addresses in Dotted-Decimal Notation

- ❖ **Dotted Decimal Notation** is for **human memory and communication**



Binary

10000001 00001011 00001011 11101111

11111001 10011011 11111011 00001111

Dotted-Decimal Notation

129.11.11.239

249.155.251.15

Example 1

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

- a. 111.56.45.78
- b. 75.45.34.79

Solution

We replace each decimal number with its binary equivalent:

- a. 01101111 00111000 00101101 01001110
- b. 01001011 00101101 00100010 01001111

Example 2

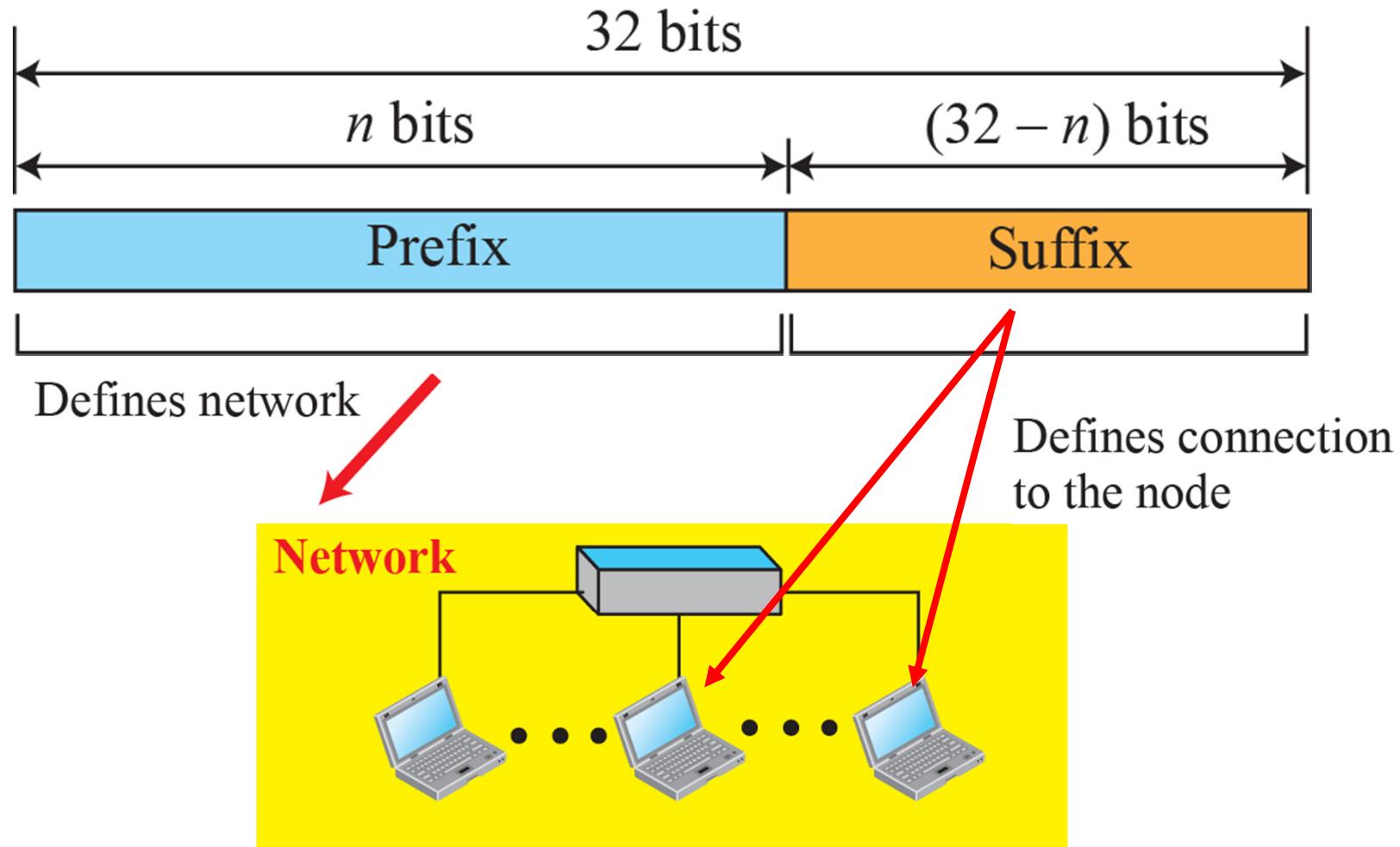
Find the error, if any, in the following IPv4 addresses.

- a. 111.56.045.78
- b. 221.34.7.8.20
- c. 75.45.301.14
- d. 11100010.23.14.67

Solution

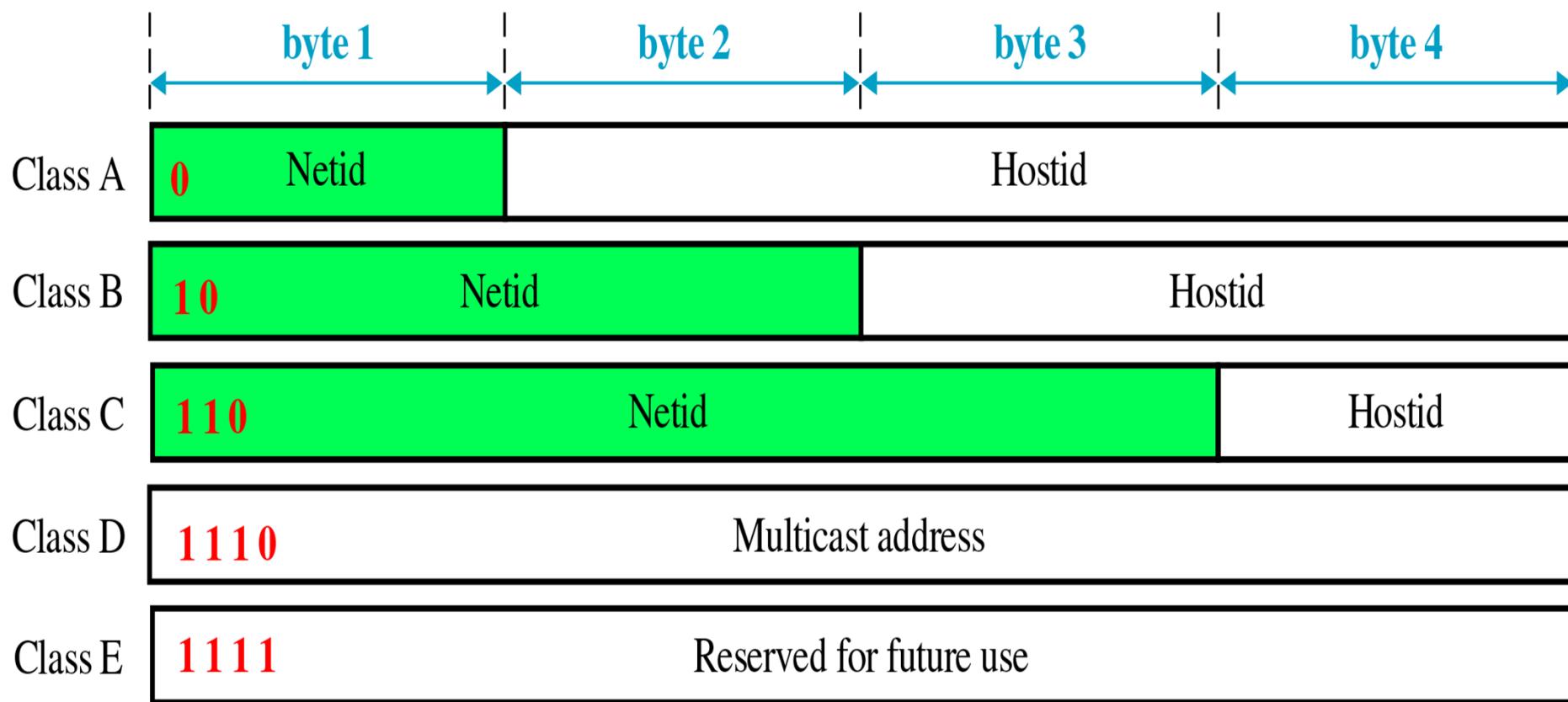
- a. *There must be no leading zero (045).*
- b. *There can be no more than four numbers.*
- c. *Each number needs to be less than or equal to 255.*
- d. *A mixture of binary notation and dotted-decimal notation is not allowed.*

Figure 18.17: Hierarchy in addressing



Classful Addressing

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



Class Ranges of Internet Addresses

(IP for hosts in Class A: 1.0.0.1 to 126.255.255.254)

(127.X.X.X is for *loopback testing* and should never be used for normal IP address)

| | 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

| <i>Class</i> | <i>Number of Blocks</i> | <i>Block Size</i> | <i>Application</i> |
|--------------|-------------------------|-------------------|--------------------|
| 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 |

*Number of blocks and block size
in classful IPv4 addressing*

Example 3

Find the class of each address:

- a. Binary: 00000001 00001011 00001011 11101111
- b. DDN: 252.5.15.111
- c. DDN: 134.11.78.56

Solution

- a. The first bit is 0; this is a class A address.
- b. The first byte is 252 (between 240 and 255); the class is E.
- c. The first byte is 134 (between 128 and 191); the class is B.

Another method (besides memorizing) is to convert the first number to binary digits and then check the first few bits

Network Address and Broadcast Address

- The **network address** (*which represents the whole network*) is assigned by the *Internet Corporation for Assigned Names and Addresses (ICANN)*
 - ❖ Note: a network address is different from a netid
 - ❖ A network address has both netid and hostid, with all “0”s for the hostid.
- ❖ **Broadcast Address**
 - ❖ In the destination address, if the hostid of the IP address contains all “1”s, it means that all hosts within this network are the target destinations.

2) Network/Address Mask

- ❖ A 32-bit number made of contiguous 1s followed by contiguous 0s
- ❖ It is used to find the netid and hostid

Default masks for classful addressing

| Class | Binary | Dotted-Decimal | CIDR |
|-------|-------------------------------------|----------------|------|
| A | 11111111 00000000 00000000 00000000 | 255.0.0.0 | /8 |
| B | 11111111 11111111 00000000 00000000 | 255.255.0.0 | /16 |
| C | 11111111 11111111 11111111 00000000 | 255.255.255.0 | /24 |

Example 4

Given the address, find the default class, the network mask, the network address and the broadcast address of :

a) 23.56.7.91

b) 132.6.17.85

Solution

a) The class is A.

The network mask :**255.0.0.0** (Only the first byte defines the netid.)

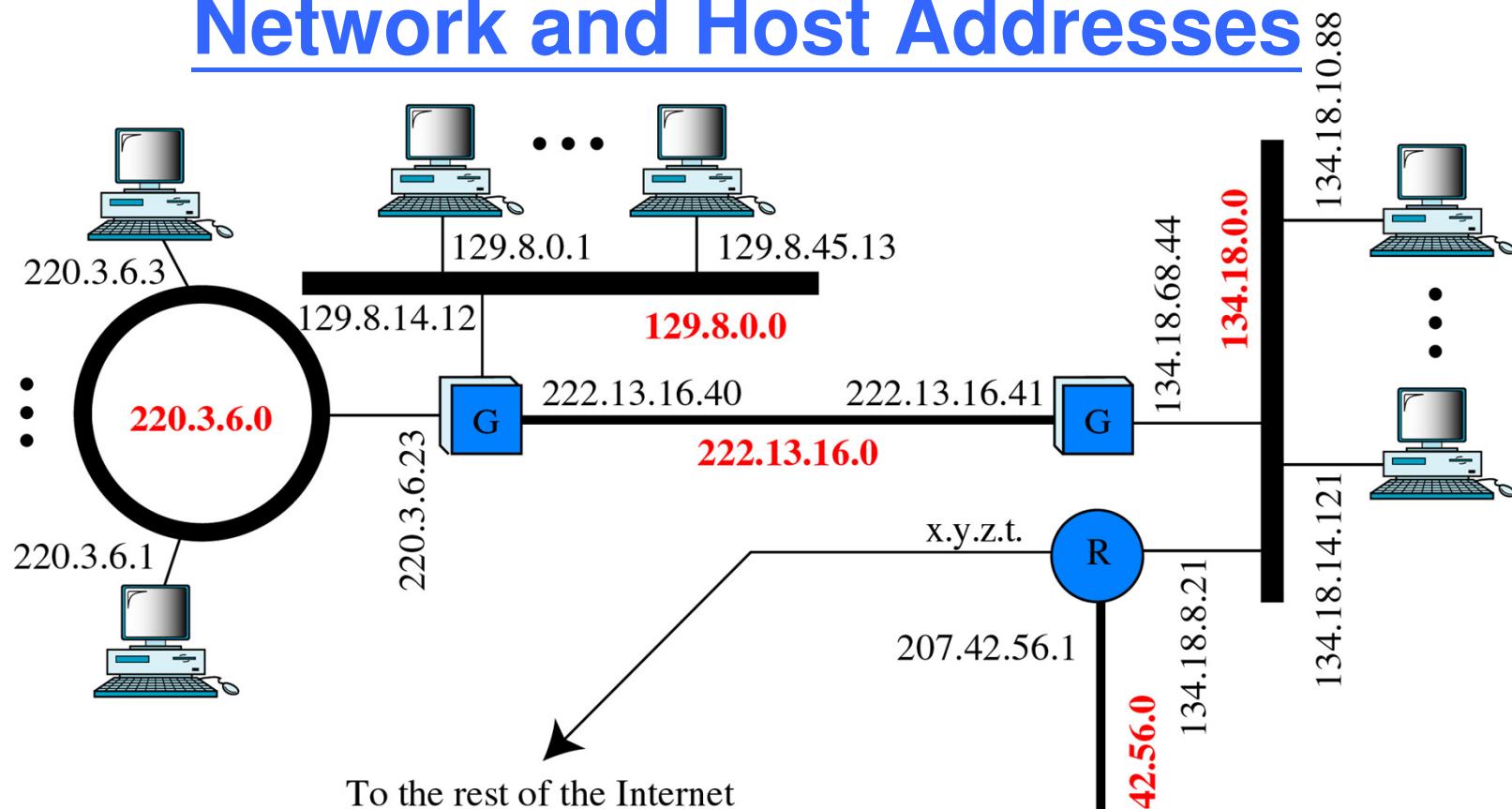
By replacing the hostid bytes (**56.7.91**) with 0s, the network address is **23.0.0.0**. The broadcast address is **23.255.255.255**

b) The class is B.

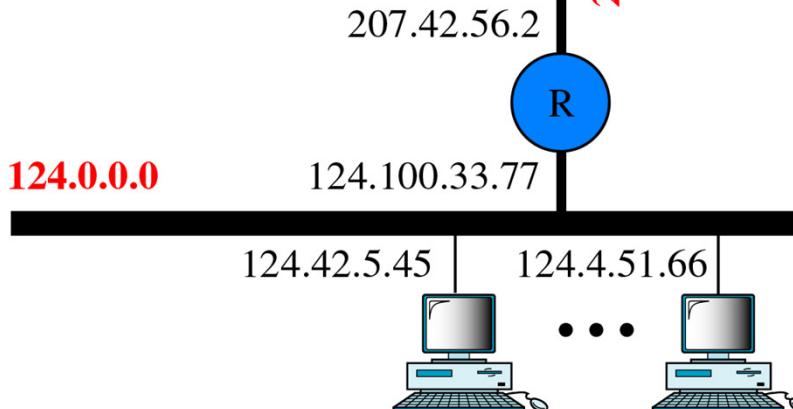
The network mask :**255.255.0.0** (The first 2 bytes defines the netid.)

By replacing the hostid bytes (**17.85**) with 0s, the network address is **132.6.0.0**. The broadcast address is **132.6.255.255**

Network and Host Addresses



The IP address in red color is the
network address



Problems of Classful Addressing

- ❖ **Small no.** of Class A and B addresses, but too **large in size** in each class.

In classful addressing, a large part of the available addresses were wasted.

- ❖ Class C block is too small for most mid-size organization.

Classful addressing is replaced with classless addressing.

Classless Addressing

- ❖ With the growth of the Internet, it was clear that a larger address space was needed
- ❖ It requires that the length of IP addresses be increased, which means the format of the IP packets needs to be changed
- ❖ Although the long-range solution has already been devised and is called IPv6 (128-bit address)
- ❖ A short-term solution, classless addressing, was also devised to use the same IPv4 address space but to change the distribution of addresses to provide a fair share to each organization

Classless Addressing

- ❖ The addresses are still granted in blocks.
- ❖ There are no classes. The network mask is more flexible.
- ❖ An **address block** is a group of IP address, in which:
 1. Addresses in a block must be contiguous, one after another.
 2. Number of addresses in a block must be a power of 2 (1, 2, 4, 8..).
 3. ... (skip the details)

(Classless) Network Mask

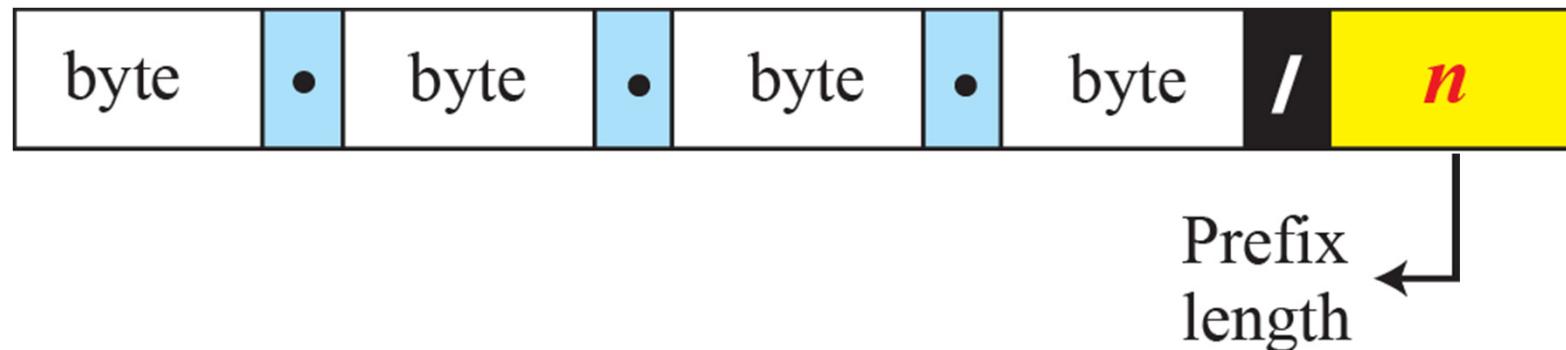
In IPv4 addressing, a block of addresses can be defined as

x.y.z.t /n

in which x.y.z.t defines one of the addresses and the /n defines the mask.

- ❖ A mask is used to define an address block
- ❖ n can be 1 to 30 - *it is the length of the net-id*
- ❖ The notation (/n) is called **Classless Interdomain Routing (CIDR)** notation

Figure 18.20: Slash notation (CIDR)



Examples:

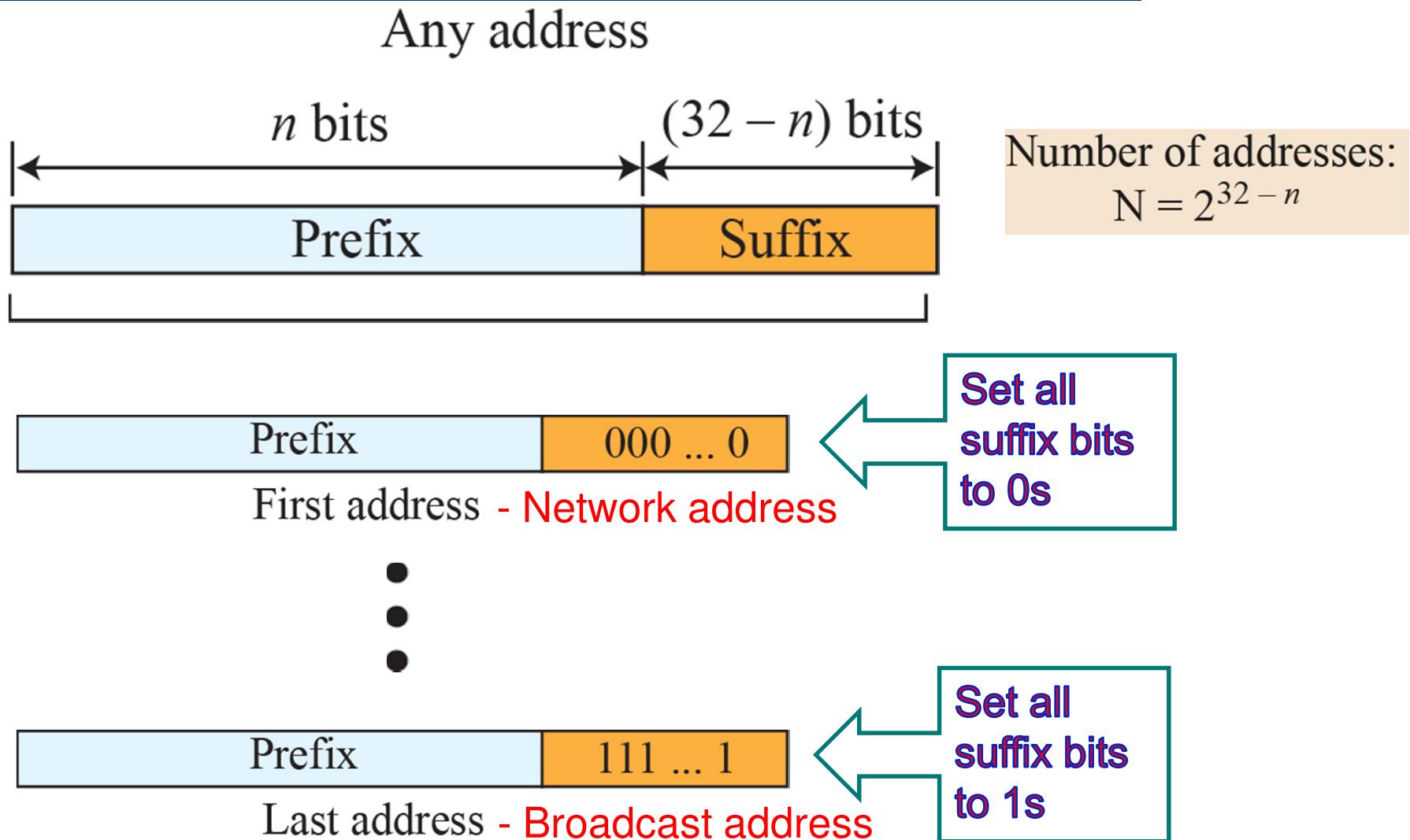
12.24.76.8/8

23.14.67.92/12

220.8.24.255/25

Figure 18.21:

Information extraction in classless addressing



Example 5

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

Solution

The binary representation of the given address is

11001101 00010000 00100101 00100111

We get the mask of 28 “1”s at the leftmost,

11111111 11111111 11111111 11110000

We get the network address (1st 28 bits + “0000”)

11001101 00010000 00100101 00100000

or

205.16.37.32

(What is the last address? What is the usage?)

Example 18.1

A classless address is given as 167.199.170.82/27.

The number of addresses in the network is $2^{32-n} = 2^5 = 32$ addresses. The first address can be found by keeping the first 27 bits and changing the rest of the bits to 0s.

| | | | | |
|----------------------------|----------|----------|----------|----------|
| Address: 167.199.170.82/27 | 10100111 | 11000111 | 10101010 | 01010010 |
|----------------------------|----------|----------|----------|----------|

| | | | | |
|----------------------------------|----------|----------|----------|----------|
| First address: 167.199.170.64/27 | 10100111 | 11000111 | 10101010 | 01000000 |
|----------------------------------|----------|----------|----------|----------|

The last address can be found by keeping the first 27 bits and changing the rest of the bits to 1s.

| | | | | |
|----------------------------|----------|----------|----------|----------|
| Address: 167.199.170.82/27 | 10100111 | 11000111 | 10101010 | 01010010 |
|----------------------------|----------|----------|----------|----------|

| | | | | |
|---------------------------------|----------|----------|----------|----------|
| Last address: 167.199.170.95/27 | 10100111 | 11000111 | 10101010 | 01011111 |
|---------------------------------|----------|----------|----------|----------|

(Classless) Network Mask

- ❖ For /n, the address mask is a 32-bit number in which
 - ❖ the n leftmost bits are 1s and the rest are 0s
- ❖ Use **bit-wise logical operation** to extract information from the IP address in a block
- ❖ **First Address = IP address AND mask**
- ❖ **Last Address = IP address OR (NOT mask)**

Example 18.2

We repeat Example 18.1 using the mask. The mask in dotted-decimal notation is 255.255.255.224. The AND, OR, and NOT operations can be applied to individual bytes using calculators and applets at the book website.

Number of addresses in the block: $N = \text{NOT}(\text{mask}) + 1 = 0.0.0.31 + 1 = 32 \text{ addresses}$

First address: $\text{First} = (\text{address}) \text{AND}(\text{mask}) = 167.199.170.64$

Last address: $\text{Last} = (\text{address}) \text{OR}(\text{NOT mask}) = 167.199.170.95$

(The mask should be 255.255.255.224 !)

The AND, OR, NOT operation should be done in binary numbers “bit-by-bit”

Example 18.2 (Cont.)

Solution

IP address 167.199.170.82/27

The binary representation of the given address is

10100111 11000111 10101010 01010010

We get the mask of 27 “1”s at the leftmost,

11111111 11111111 11111111 11100000

(NOT mask) is

00000000 00000000 00000000 00011111

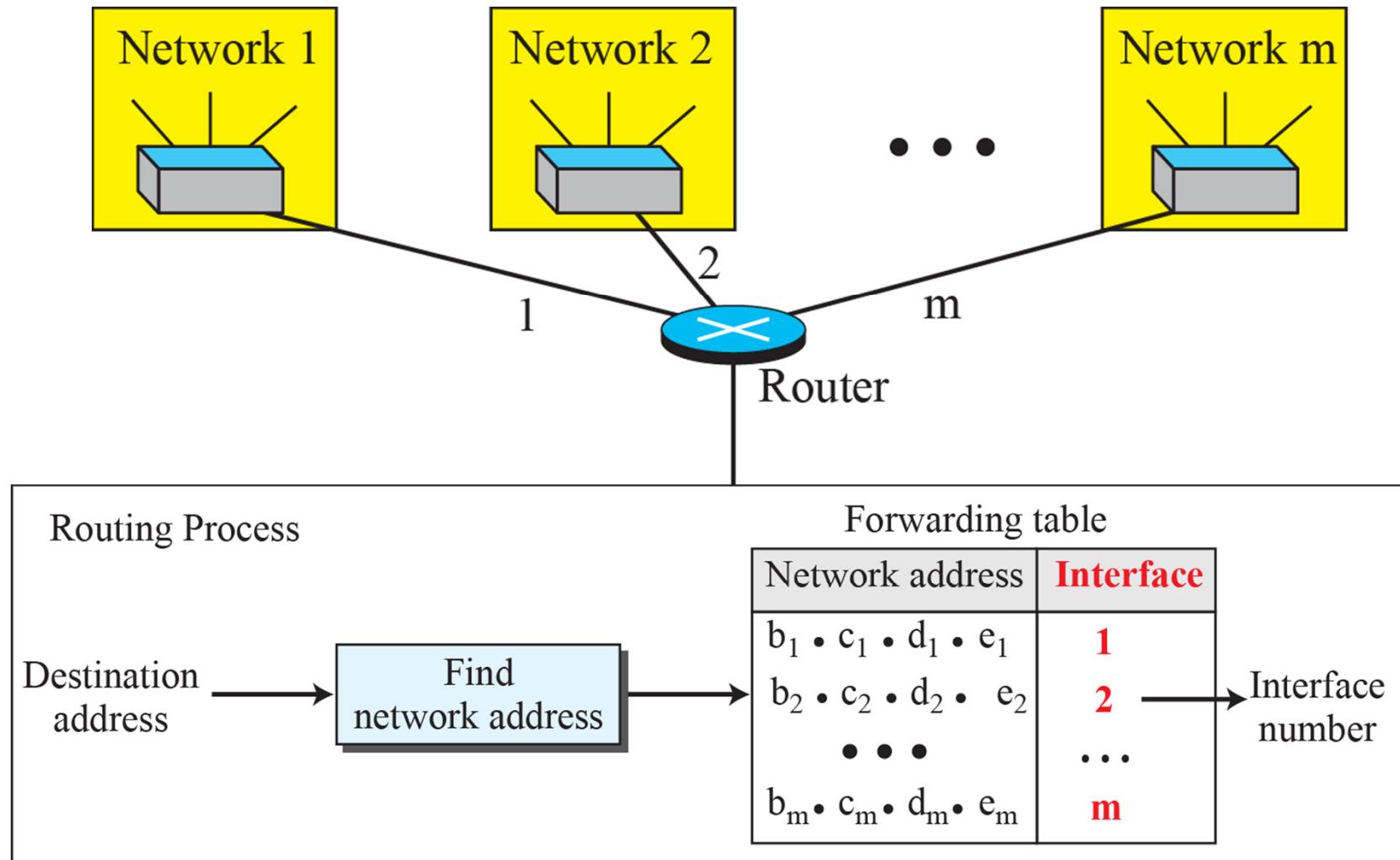
OR 10100111 11000111 10101010 01010010 (IP)

10100111 11000111 10101010 01011111

After (address)OR(NOT mask)operation

We get the broadcast address 167.199.170.95/27

Figure 18.22: Network address



The network address is the identifier of the network and it is used by the forwarding (or routing) table in the Internet

How does a *host* get IP address

- ❖ Hard-coded by system admin in a file
- ❖ The network administrator can manually assign addresses to the individual hosts or routers

OR

- ❖ **DHCP: Dynamic Host Configuration Protocol**
- ❖ An application-layer program which allows a host to dynamically (and automatically) get address from a server
- ❖ Useful when the no. of hosts is more than the no. of available IP addresses but the no. of “active” hosts is not high

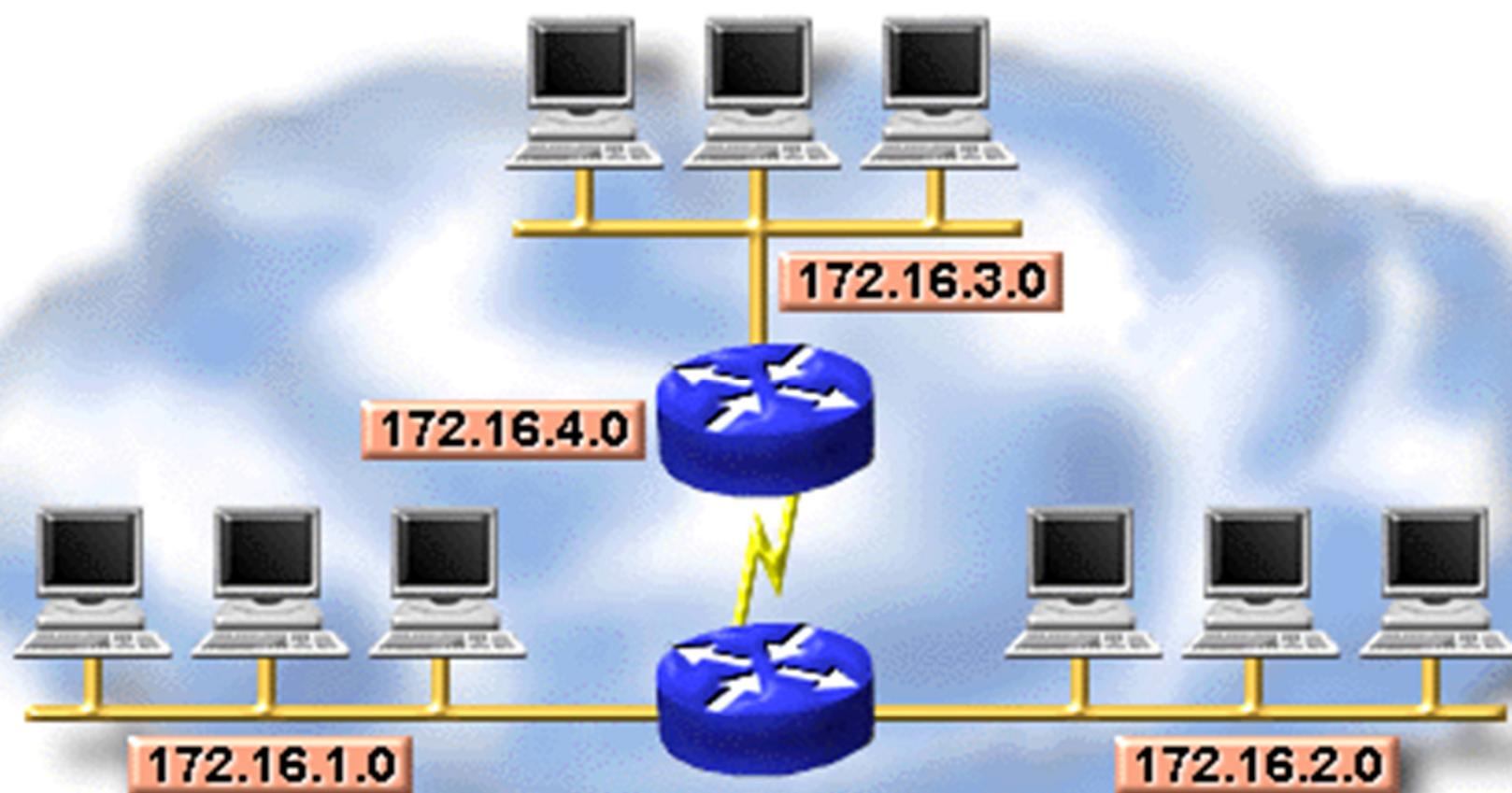
3) IP Subnetworks (Subnets)

- ❖ For transferal of data on the Internet, one network sees another as a single network and has no detailed knowledge of its internal structure
- ❖ The reason for this is that it helps keep routing tables small
- ❖ However large networks are often divided into smaller networks called subnetworks (subnets)

Subnets

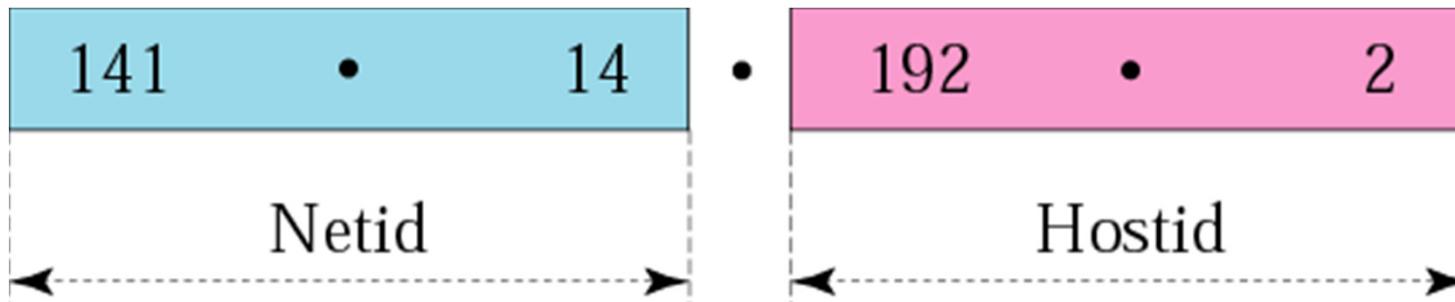
- ❖ The primary reason for using a subnet is to reduce the size of a broadcast domain (and hence reduce collisions)
- ❖ Subnets also improve the efficiency of addressing
- ❖ Adding subnets does not change how the *outside world* sees the network. Thus, a device on an outside network only sees the network ID and host ID of a device on another network.
- ❖ However, *internally*, networks can view themselves as being a series of smaller subnets.

Addressing with Subnets

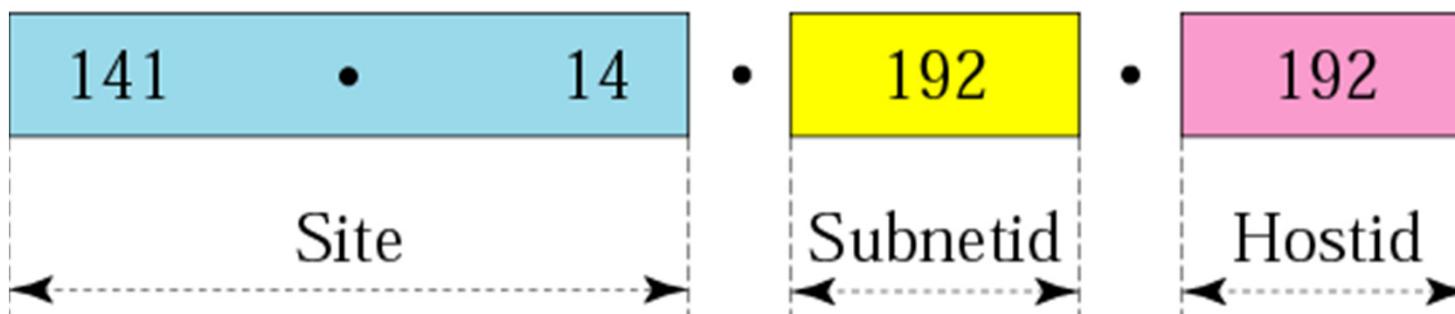


Network 172.16.0.0

Addresses in a network with and without subnetting



a. Without subnetting



b. With subnetting

Subnet Address

- ❖ Subnet addresses include a *network ID*, a *subnet ID* within the network, and a *host ID* within the subnet.
- ❖ To create a subnet address, a network administrator "borrows" bits from the **host field** and designates them as the subnet field.
- ❖ Network devices use **subnet masks** to identify which part of the address is considered as subnet ID and which part represents host addressing.

Length of Subnet ID (1)

- ❖ For a network without subnets, when the host field of an IP address is all 0s, it is a network address.
- ❖ When the host field is all 1s, it is a broadcast address.
- ❖ Therefore, when we consider a network with subnets, **the combined field {<subnet> <host>}** **should NOT be all 0s or all 1s**

Length of Subnet ID (2)

- ❖ The **minimum** number of bits that can be borrowed for subnet is **2**
- ❖ The **maximum** number of bits that can be borrowed for subnet can be any number that **leaves at least 2 bits for hosts**
 - ☞ (If only 1 bit for host, a 0-host number or a 1-host number will occur, which may cause a all-0 network address or a all-1 broadcast address.)

Example Subnets

- ❖ With each subnet, you cannot use the first and last address (i.e. the network and the broadcast address).
- ❖ A **class B** network has 16 bits in the host field. Therefore **up to 14 bits** can be borrowed to create subnets.
- ❖ A **class C** network has only 8 bits in the host field. Therefore, only **up to 6 bits** can be borrowed to create subnets.

Subnet Mask

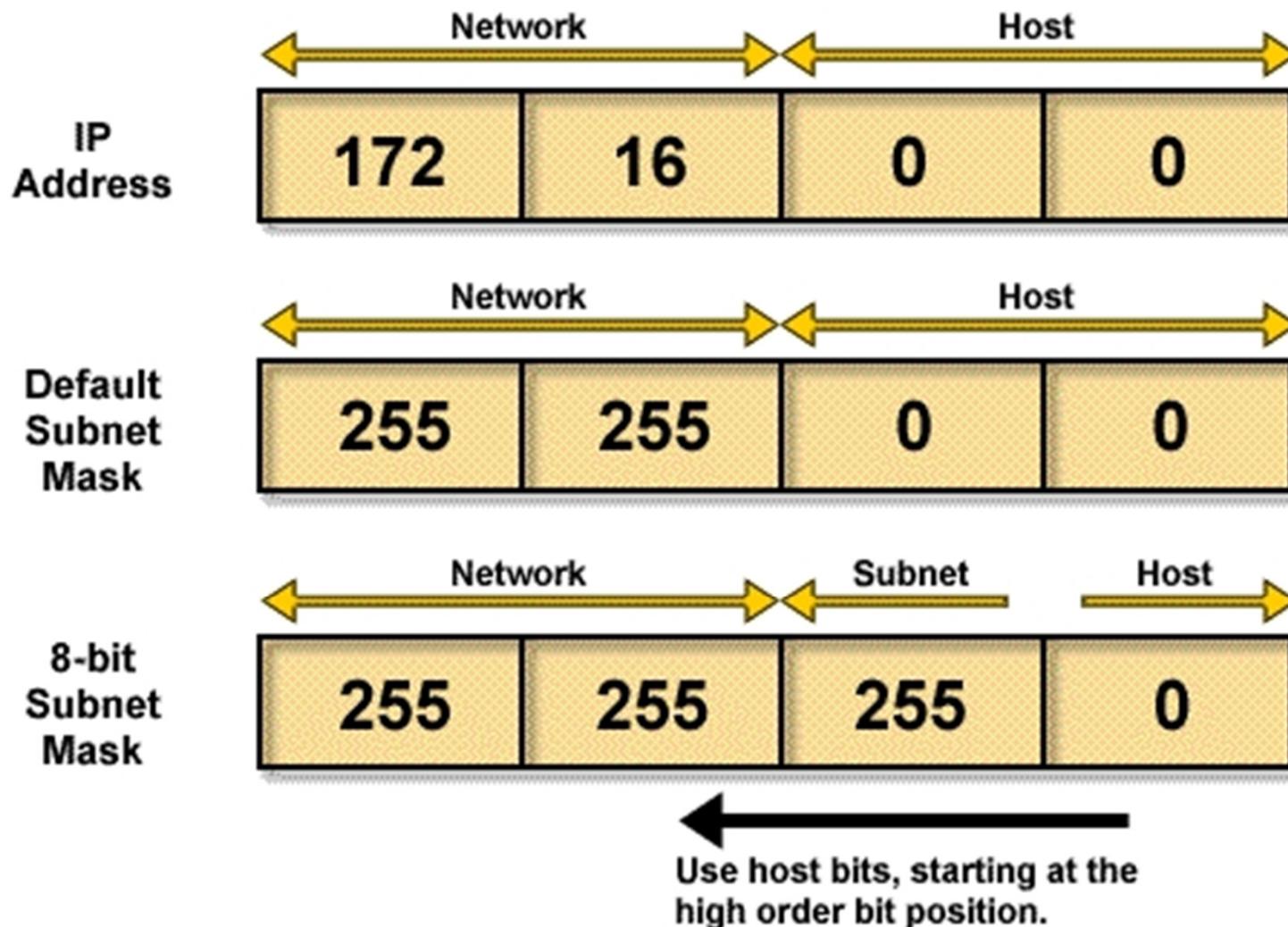
- ❖ Once a packet has arrived at an organization's gateway with its unique network ID, it can be routed within the organization's internal routers using the subnet ID, which is found by a subnet mask.
- ❖ The subnet mask identifies which part of an IP address is the network address (net-ID), which part is the subnet ID, and which part is the host ID.
- ❖ **A subnet mask is 32 bits long.**
- ❖ *It contains all 1s in the network portion and the subnet portion, and contains all 0s in the host portion.*

Subnet Mask Examples

- ❖ By default, if no bits are borrowed (i.e. no subnet), the subnet mask for a class "B" network would be 255.255.0.0
- ❖ However, if 8 bits are borrowed to form subnets, the subnet mask for the same class "B" network would be 255.255.255.0
- ❖ All networks have **default subnet masks** (for no subnet).
 - ❖ E.g. For the network, 172.16.0.0, the default subnet mask is 255.255.0.0



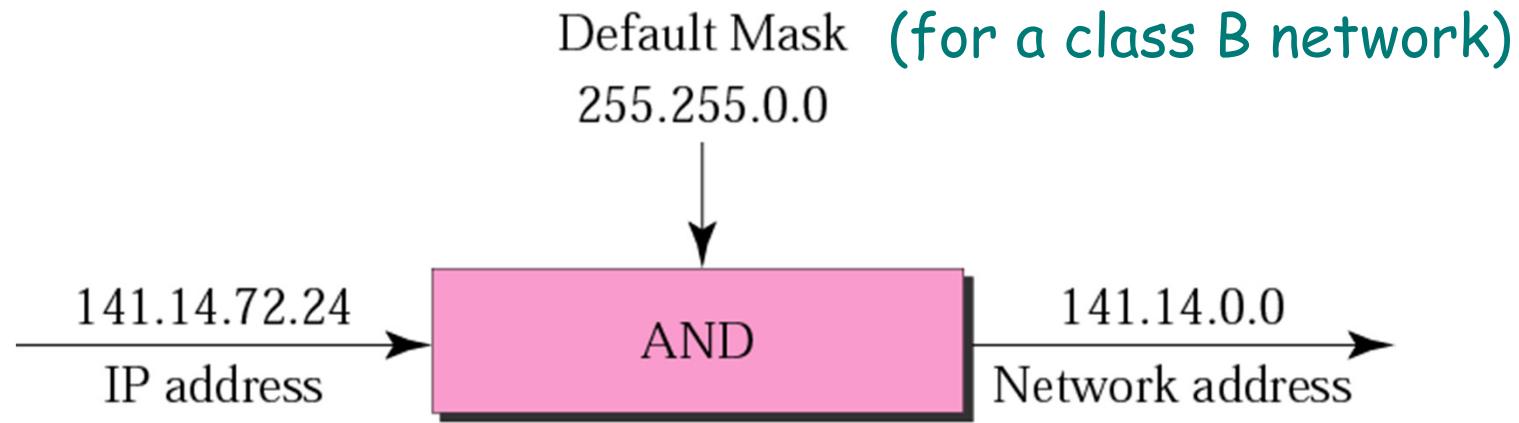
Subnet Mask



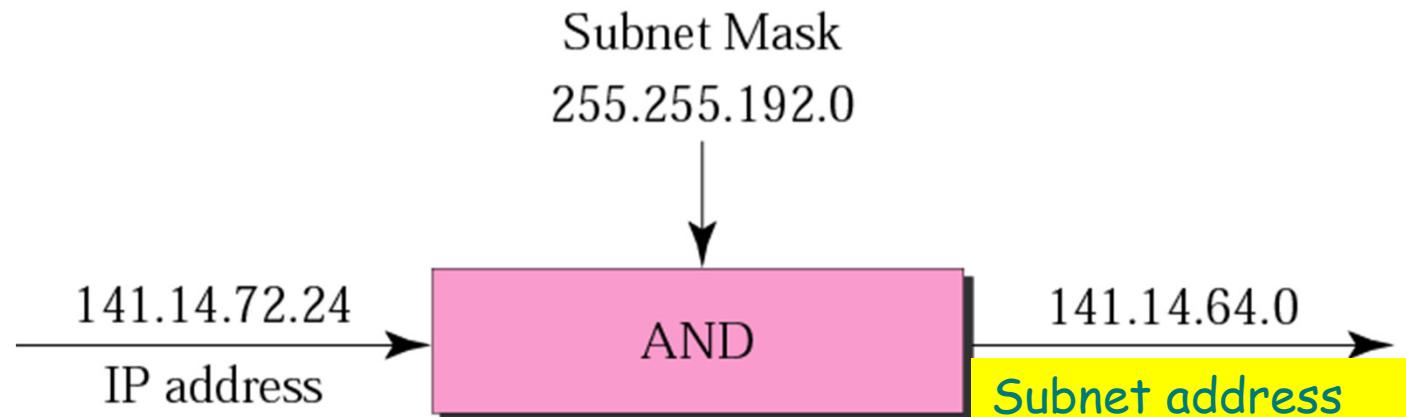
Use of Subnet Mask

- ❖ In order to find the address of a subnet, a router must take the incoming IP address and the subnet mask, and then apply the **logical “AND” operation** between their binary forms.
- ❖ The resulting number is the network or subnet address.

4)) Default mask and subnet mask

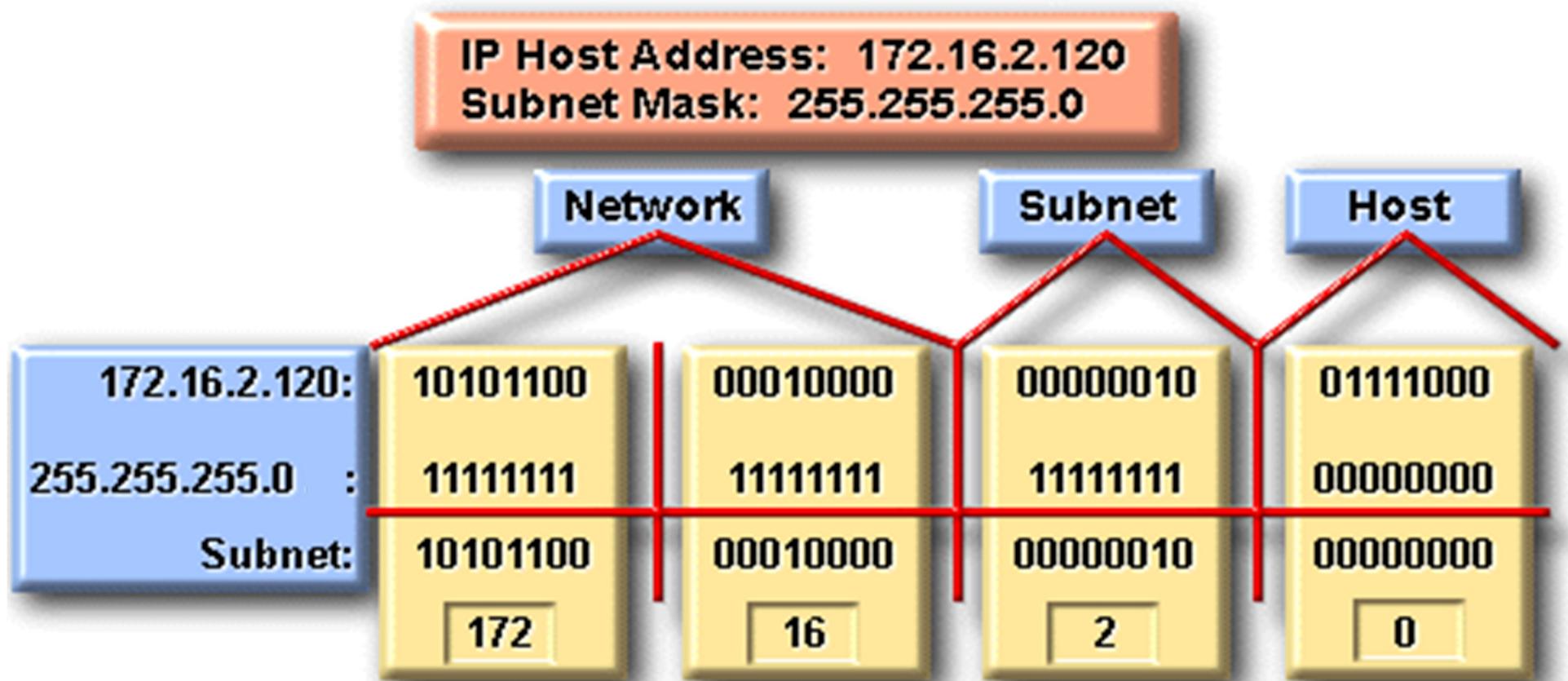


a. Without subnetting



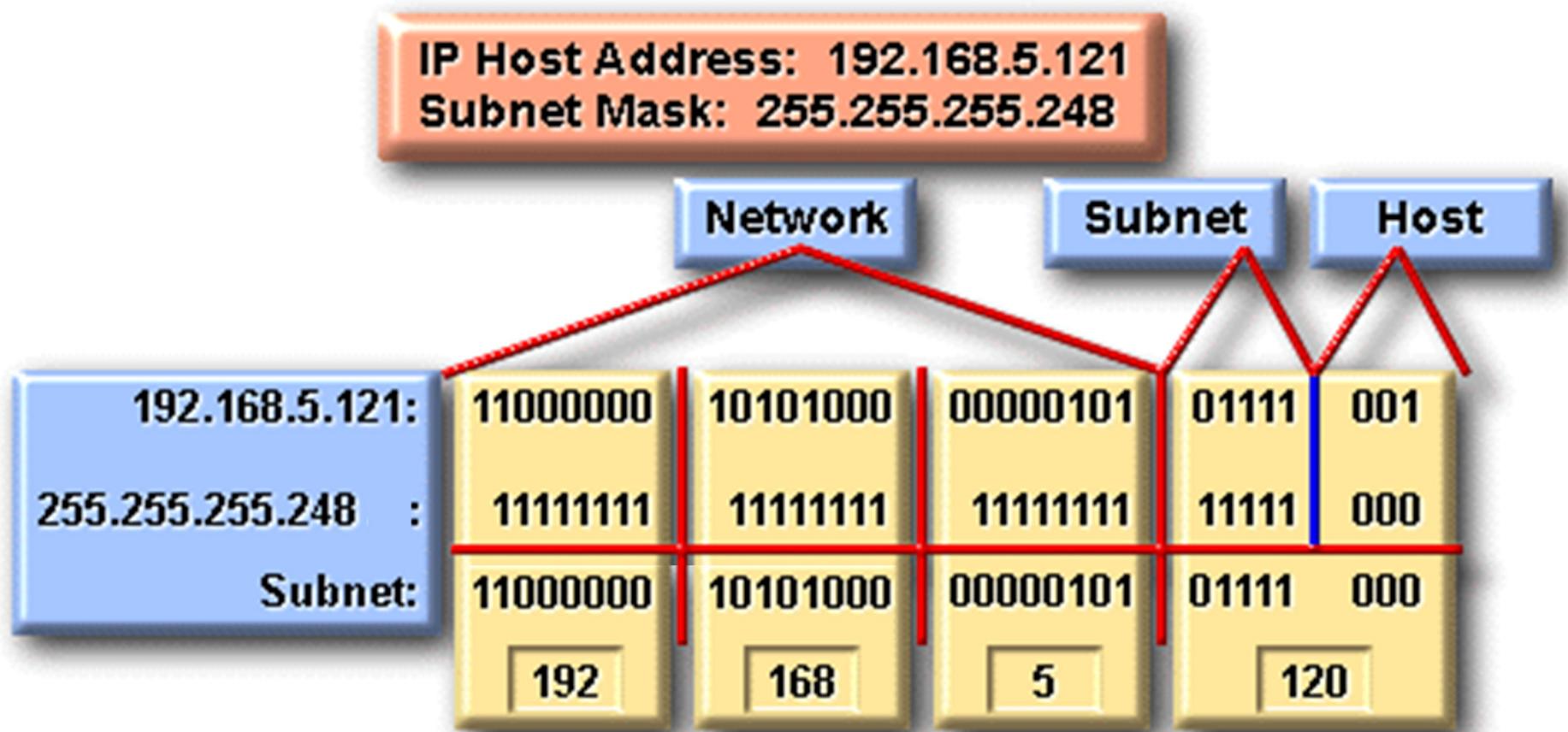
b. With subnetting

Class B Subnet Planning Example



- Subnet Address = 172.16.2.0
- Host Addresses = 172.16.2.1 - 172.16.2.254
- Broadcast Address = 172.16.2.255
- Eight bits of subnetting

Class C Subnet Planning Example



- Subnet Address = 192.168.5.120
- Host Addresses = 192.168.5.121 - 192.168.5.126
- Broadcast Address = 192.168.5.127
- Five bits of subnetting

Consider a Class B network, 16 bits are used for network ID, and the remaining 16 bits are used for subnet and host ID:

| #bits borrow | subnet mask |
|--------------|---|
| 0 | 255.255.0.0 |
| 1 | n/a (<i>min. 2 bits</i>) |
| 2 | 255.255. <u>192</u> .0 |
| 4 | 255.255. <u>240</u> .0 |
| 6 | 255.255. <u>252</u> .0 |
| 8 | 255.255. <u>255</u> .0 |
| 14 | 255.255.255. <u>252</u> |
| 15 | n/a (<i>at least 2 bits left for host id</i>) |
| 16 | n/a (not applicable) |

*In each subnet, the total number of hosts is **2 less than** the maximum number of possible host IDs.*

Finding the Subnet Address

Straight Method

Use binary notation for both the address and the mask and then apply the AND operation to find the subnet address

Example 1

What is the subnet address if the destination address is 200.45.34.56 and the subnet mask is 255.255.240.0?

11001000 00101101 00100010 00111000

11111111 11111111 11110000 00000000

11001000 00101101 00100000 00000000

The subnet address is **200.45.32.0**.

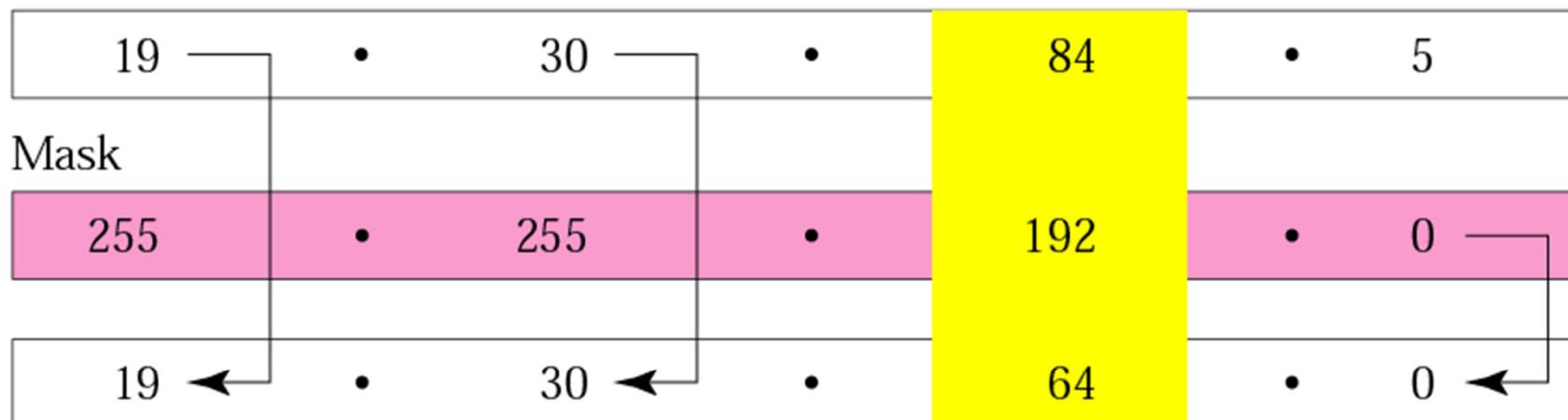
Short-Cut Method

- ** If the byte in the mask is 255, copy the byte in the address.
- ** If the byte in the mask is 0, replace the byte in the address with 0.
- ** If the byte in the mask is neither 255 nor 0, we write the mask and the address in binary and apply the AND operation.

Example 2

What is the subnet address if the destination address is 19.30.80.5 and the mask is 255.255.192.0?

IP Address



Subnet Address

| | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|
| 84 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 192 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 64 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

Example 3

A company is granted the site (network) address 201.70.64.0 (class C). The company needs six subnets. Design the subnets.

Solution

The number of 1s in the default mask is 24 (class C).

The company needs six subnets. This number 6 is not a power of 2. The next number that is a power of 2 is 8 (2^3). We need 3 more 1s in the subnet mask. The total number of 1s in the subnet mask is 27 ($24 + 3$).

The total number of 0s is 5 ($32 - 27$). The mask is

E.g. 3 Solution (Continued)

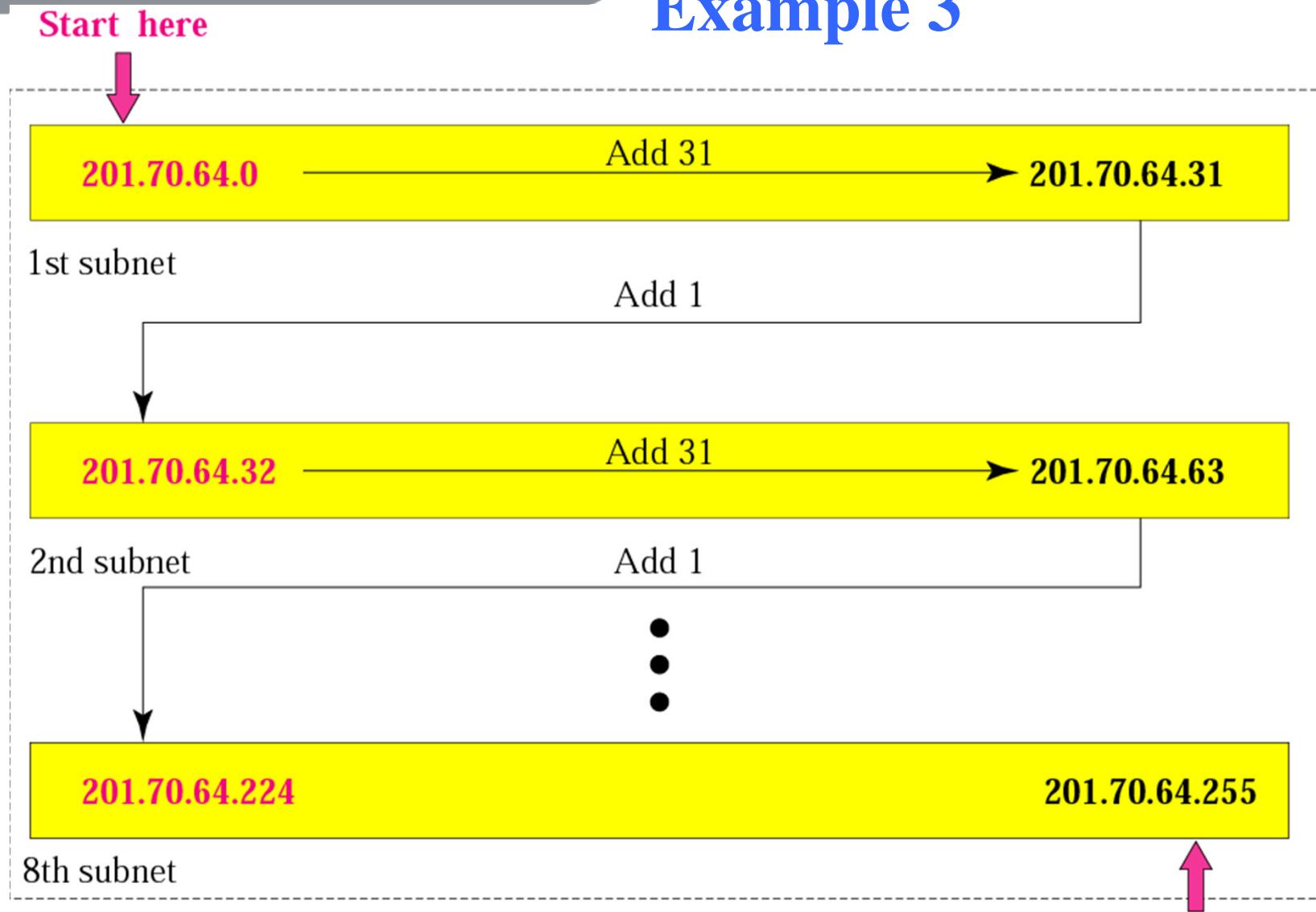
11111111 11111111 11111111 11100000
or **255.255.255.224**

The number of subnets is 8.

The number of addresses in each subnet is 2^5
(5 is the number of 0s) or 32 (***but only 30 are host addresses***).

See next slide

Example 3

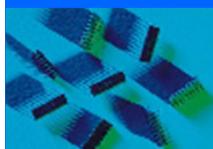


Usually the first and the last subnets are not used

Summary

- ❖ IPv4 Address – 32 bits
 - ❖ Classful: 5 classes A to E (phase out)
 - ❖ Classless: \n (currently use)
 - ❖ IPv6: 128 bits (long term solution)
- ❖ Network/Address Mask
 - ❖ Use logical “AND” to find the network address
- ❖ Subnets – smaller networks within the same organization (network)

- ❖ Revision Quiz
 - ❖ http://highered.mheducation.com/sites/0073376221/student_view0/chapter18/quizzes.html



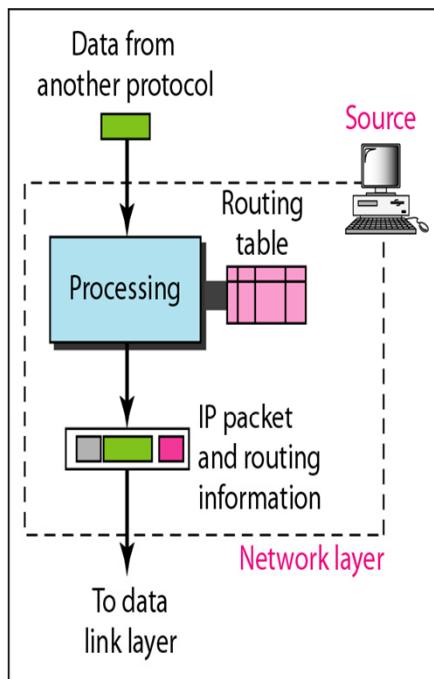
Lecture 8 Internet Protocol (IP) & Routing

Textbook: Ch.19, 20

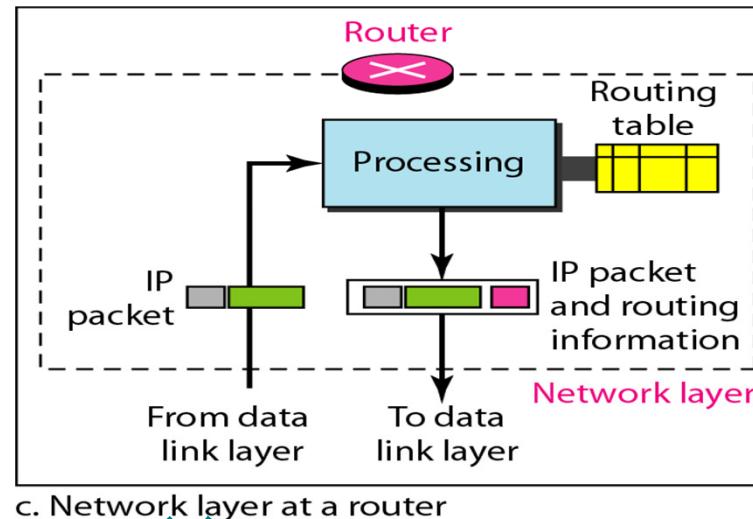
Main Topics

- A. Network Layer Functions
- B. Internet Protocol (IP)
 - ❖ TCP/IP
 - ❖ IP Datagram Format
 - ❖ IP Fragmentation
- C. Routing
 - ❖ Routing Algorithms
 - ❖ Next-hop Routing Table
- D. LAN Addressing
 - ❖ Address Resolution Protocol (ARP)

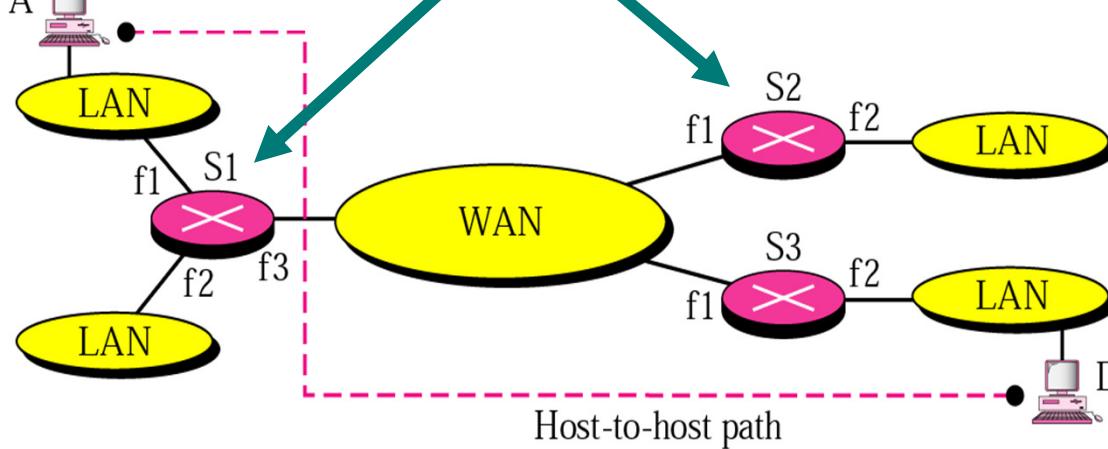
A. Network Layer Functions



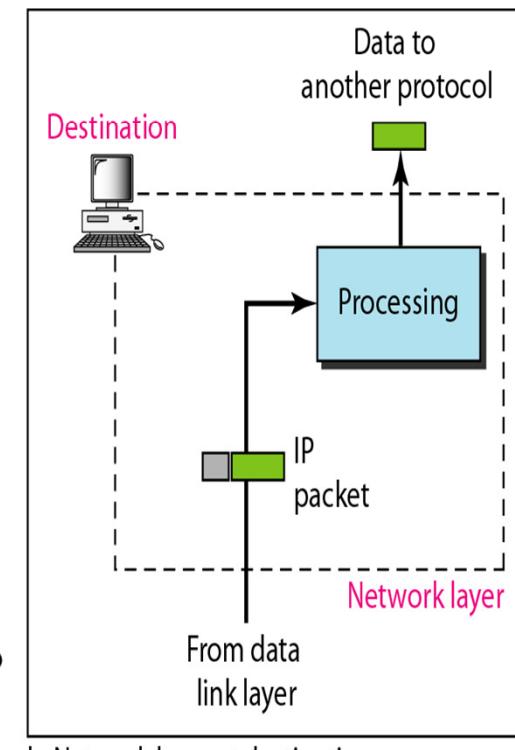
a. Network layer at source



c. Network layer at a router



Host-to-host path

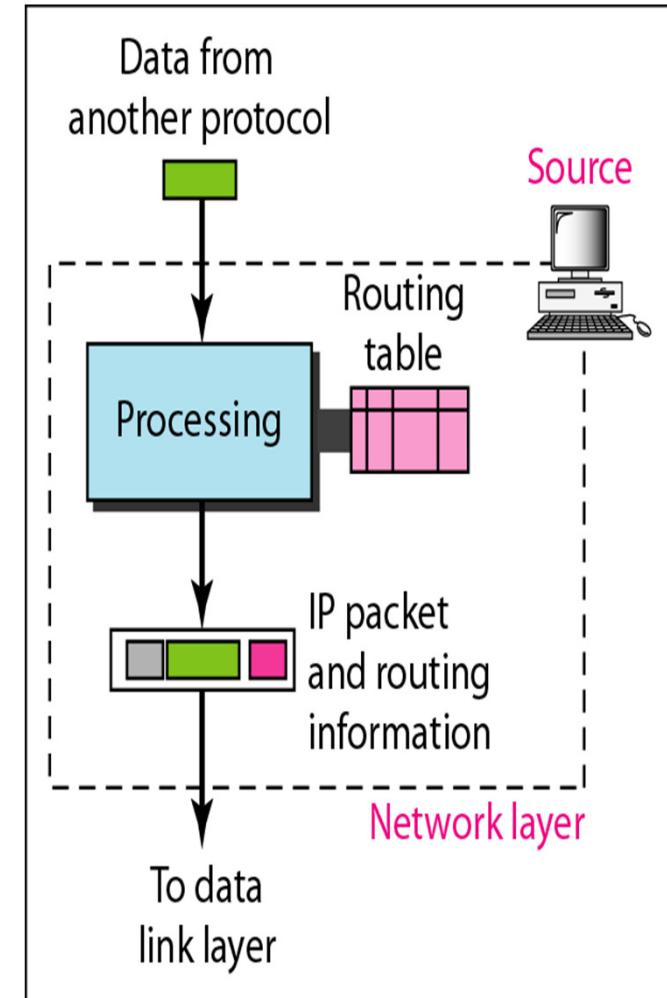


b. Network layer at destination

Network Layer Functions

❖ Source

- ❖ **Creating a packet** from the data coming from another protocol (usually the upper layer, TCP or UDP).
- ❖ The **header** of packet contains the logical address (e.g. IP address) of the source and destination.
- ❖ **Identify** the first **routing information**. (E.g. the address of next hop, such as default gateway.)
- ❖ If the packet is too large, the packet is **fragmented**. (Similar to data size in the Ethernet)

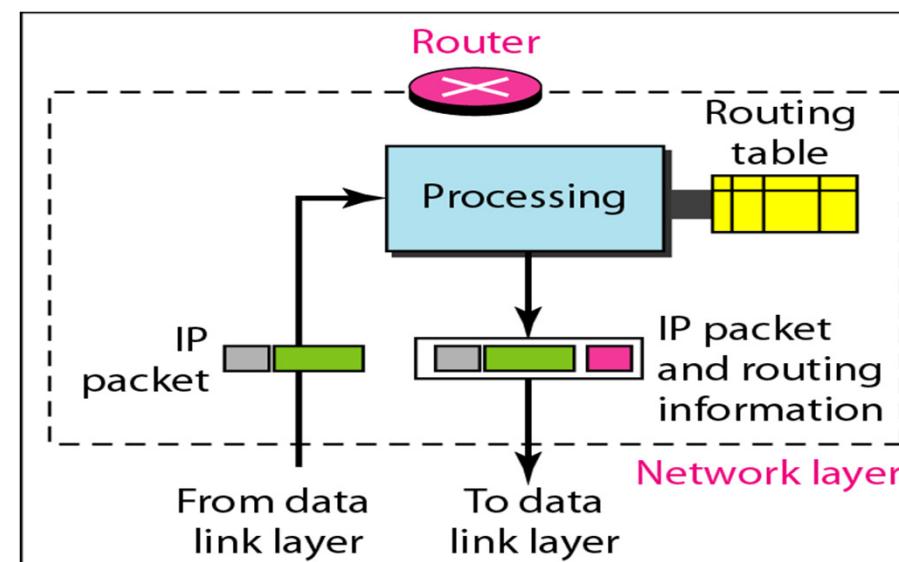


a. Network layer at source

Network Layer Functions

❖ Router (or L3 Switch)

- ❖ Routing the packet.
- ❖ It looks up the **routing table** find the interface that the packet should be sent.
- ❖ The packet is sent to data link layer with the appropriate port.

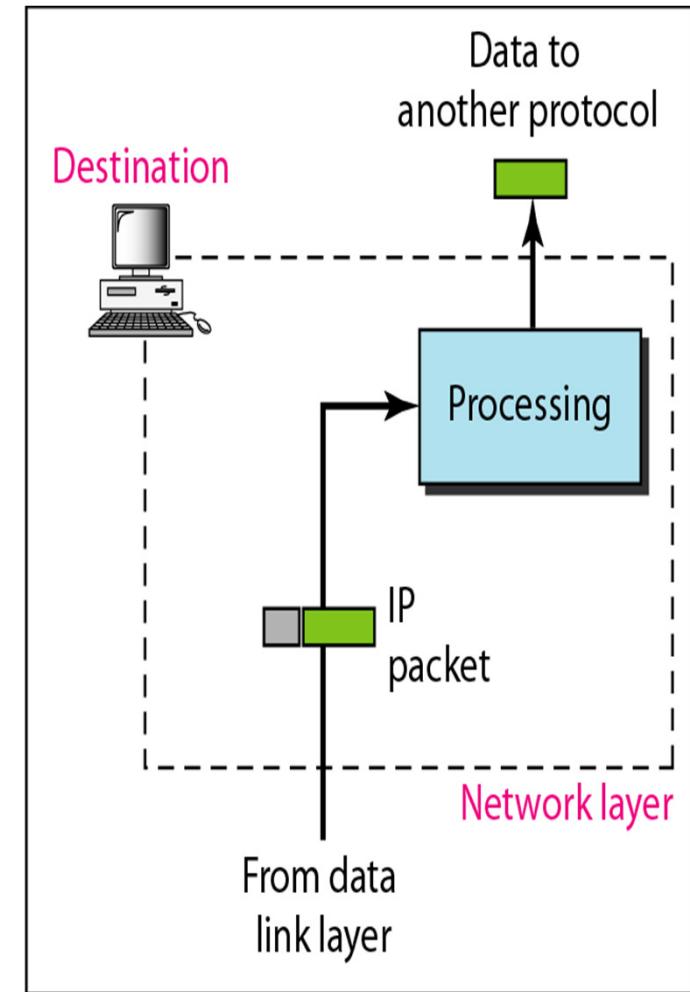


c. Network layer at a router

Network Layer Functions

❖ Destination

- ❖ Send the packet to the upper layer.
- ❖ It performs **address verification** to ensure that the destination address on the packet is the same as the address of the host.
- ❖ If the packet is a fragment, it waits until all fragments have arrived and **reassembles** them.

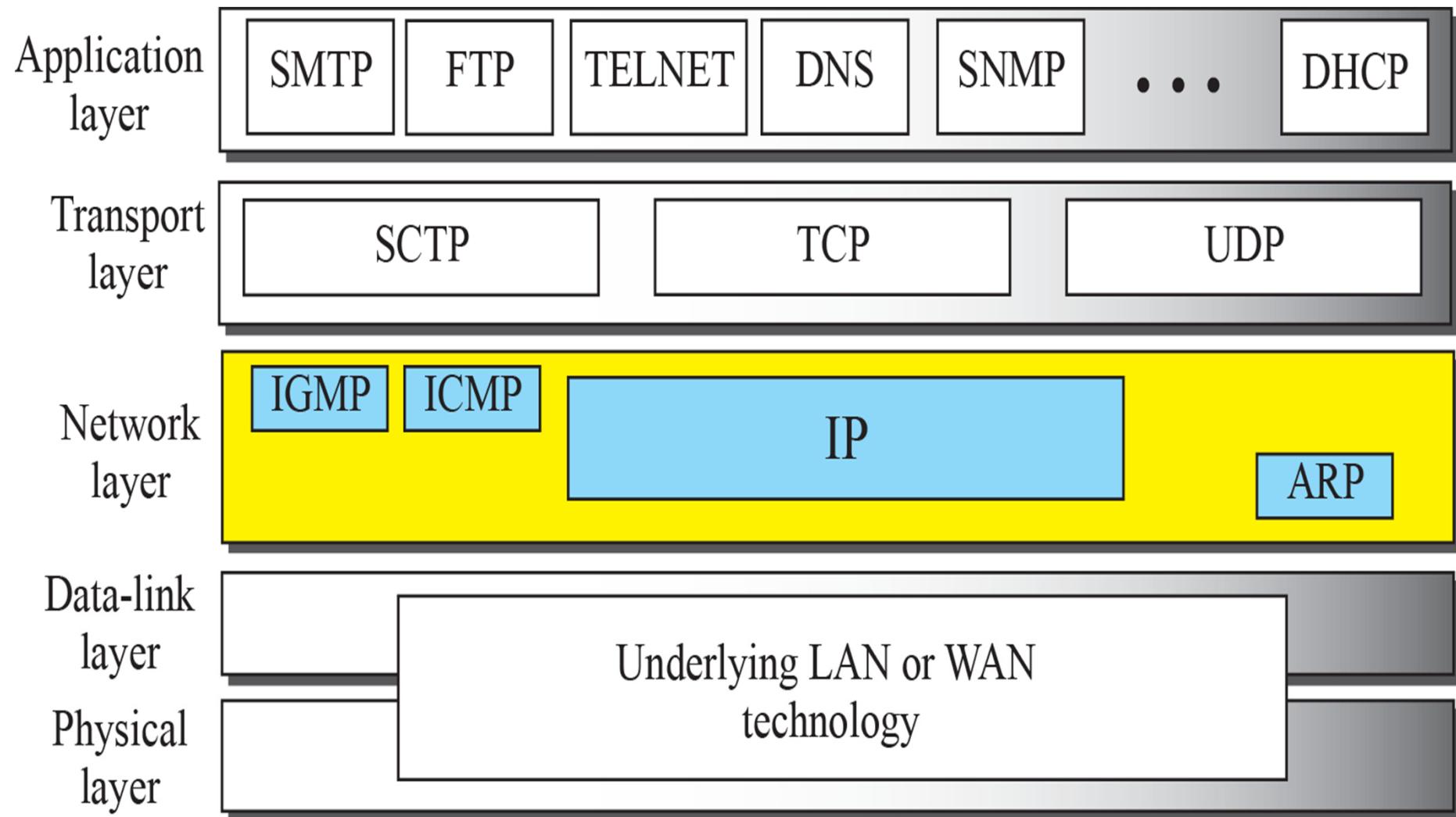


b. Network layer at destination

B. TCP/IP

- ❖ **A set of protocols**, or protocol suite, that defines how all transmissions are exchanged across the Internet
- ❖ TCP/IP is a **five-layer** protocol: physical, data link, network, transport and application
- ❖ ***Transport layer***: (2 protocols/services)
 - ☞ **Transmission Control Protocol (TCP)** - data unit is called TCP segment (VC service)
 - ☞ **User Datagram Protocol (UDP)** – data unit is called user datagram (Datagram service)
- ❖ ***Network layer***: **Internet Protocol (IP)**
- ❖ At least 6 protocols in the application layer

TCP/IP Protocol Suite



Internet Protocol (IP)

- ❖ The Internet Protocol version 4 (IPv4) is the network layer protocol used by TCP/IP
- ❖ Provides an unreliable, connectionless datagram best-effort delivery service

❖ **Unreliable**

packet may be lost, duplicated, delayed, out of order

❖ **Connectionless datagram**

sequence of packets (belong to the same data file)
may travel along different paths

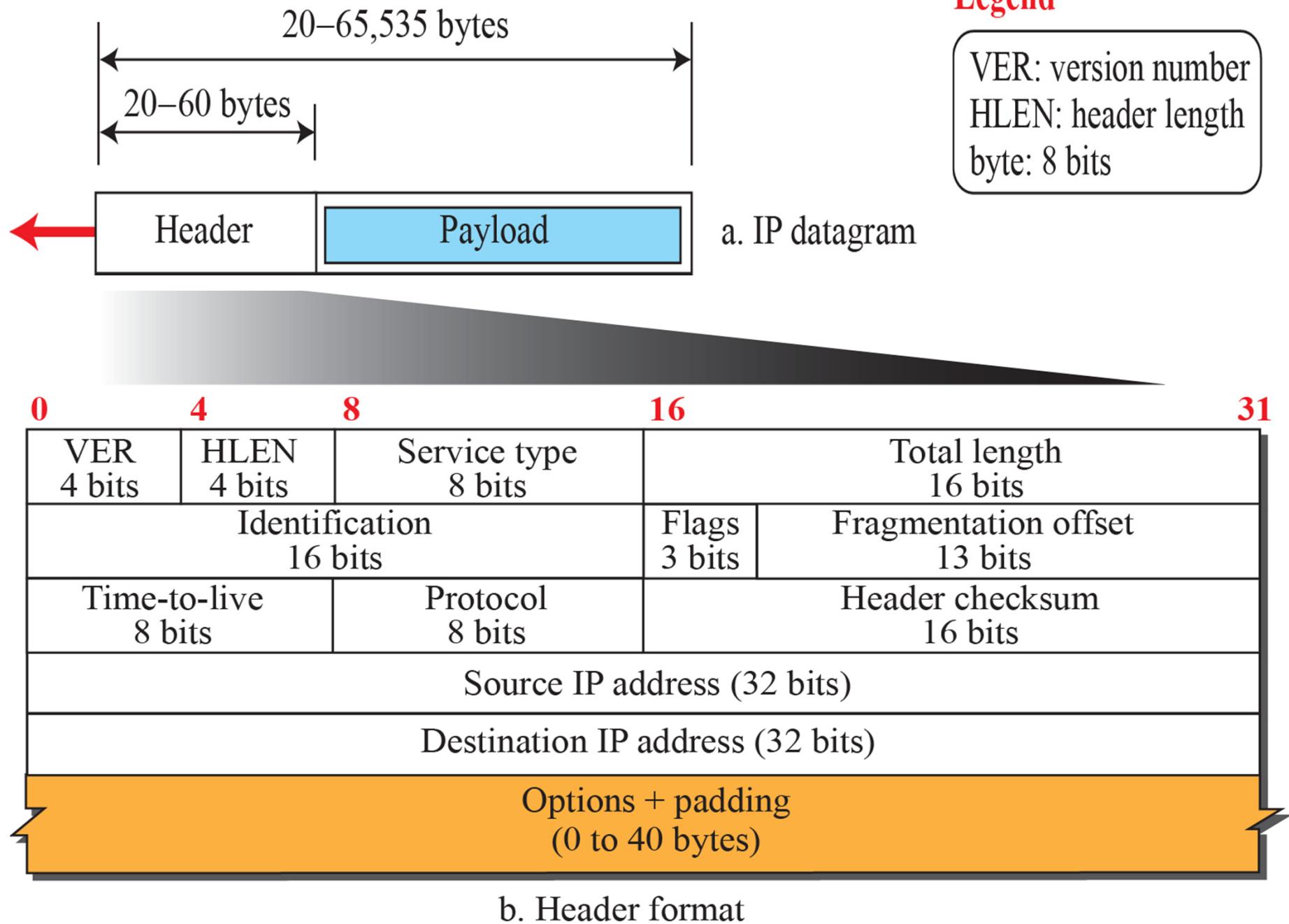
❖ **Best-effort delivery**

- Internet does not discard packets
- unreliability arise only when resources are fully loaded or underlying networks fail

Internet Protocol (IP)

- ❖ Defines the format of the basic unit of data transfer
- ❖ The IP packet, called **IP datagram**, consists of variable-length header and data (*payload*) fields
- ❖ Setting IP address and routing tables
- ❖ Handle the routing decision and operation

Figure 19.2: IP datagram



IP Datagram Format

- ❖ **Version Number (VER)**
 - ❖ for IPv4, it is 4
- ❖ **Header Length (HLEN)**
 - ❖ min. 20 bytes to max. 60 bytes
 - ❖ (in 4-byte words) the header length in bytes is divided by 4 to get the value of this field
- ❖ **Total Length**
 - ❖ the total length (header plus data) in bytes
 - ❖ min. 20 bytes to max. 65535 bytes
- ❖ **Identification, Flags, and Fragment Offset**
 - ❖ for fragmentation (when the datagram is too long) and reassembly (*discuss later*)

IP Datagram Format

❖ Time-to-live

- ❖ max. no. of remaining hops (or routers) allowed to visit
- ❖ initially set to two times the path length
- ❖ being decremented at each router
- ❖ if it is zero, the datagram will be discarded

❖ Protocol

- ❖ a number to indicate which upper layer protocol will get the data

❖ Header Checksum

- ❖ just for detecting errors in the header

Example 19.1

An IPv4 packet has arrived with the first 8 bits as $(01000010)_2$. The receiver discards the packet. Why?

Solution

There is an error in this packet. The 4 leftmost bits $(0100)_2$ show the version, which is correct. The next 4 bits $(0010)_2$ show an invalid header length ($2 \times 4 = 8$). The minimum number of bytes in the header must be 20. The packet has been corrupted in transmission.

Example 19.2

In an IPv4 packet, the value of HLEN is $(1000)_2$. How many bytes of options are being carried by this packet?

Solution

The HLEN value is 8, which means the total number of bytes in the header is 8×4 , or 32 bytes. The first 20 bytes are the base header, the next 12 bytes are the options.

Example 19.3 (modified)

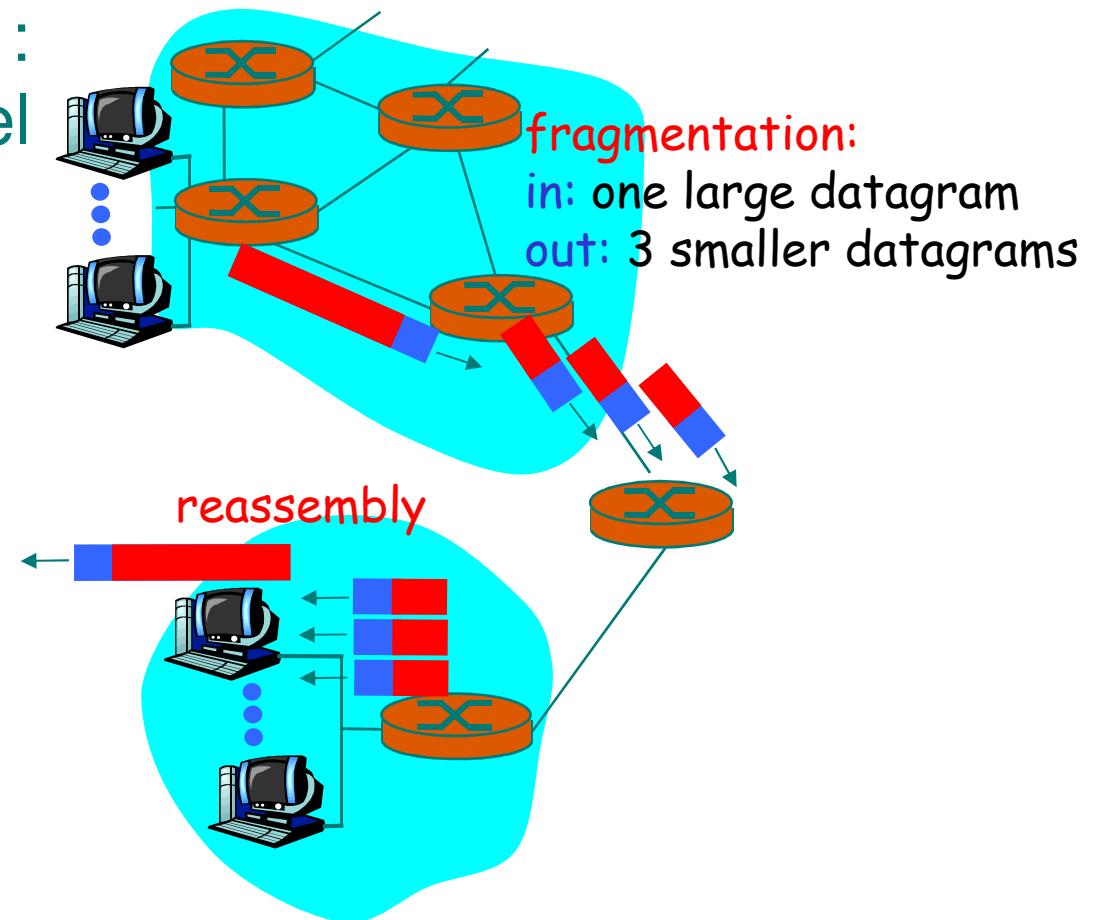
In an IPv4 packet, the value of HLEN is 5, and the value of the total length field is $(0038)_{16}$. How many bytes of data are being carried by this packet?

Solution

The HLEN value is 5, which means the total number of bytes in the header is 5×4 , or 20 bytes (no options). The total length is $(0038)_{16}$ or 56 bytes, which means the packet is carrying 36 bytes of data ($56 - 20$).

IP Fragmentation & Reassembly

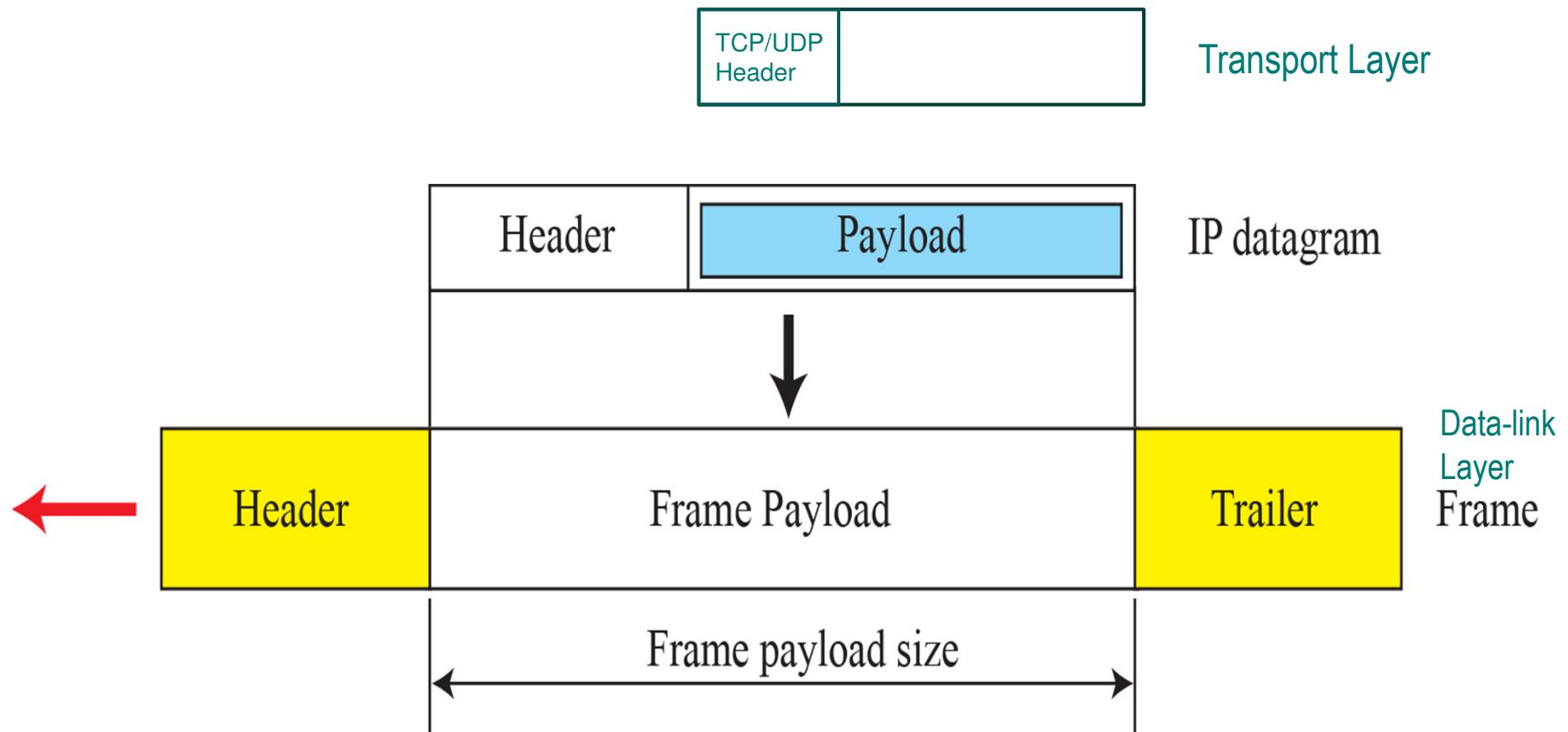
- ❖ Network links have **MTU** (maximum transfer unit) : largest possible link-level frame payload size
- ❖ Different link types, different sizes of MTUs
- ❖ The size of MTU includes both data and header of the IP datagram



IP Fragmentation & Reassembly (cont.)

- ❖ A datagram can travel through different networks. Each router decapsulates the IP datagram from the frame it receives, processes it, and then encapsulates it in another frame.
- ❖ The format and size of the sent frame depend on the protocol used by the physical network through which the frame is going to travel.
- ❖ For example, if a router connects a LAN to a WAN, it receives a frame in the LAN format and sends a frame in the WAN format.

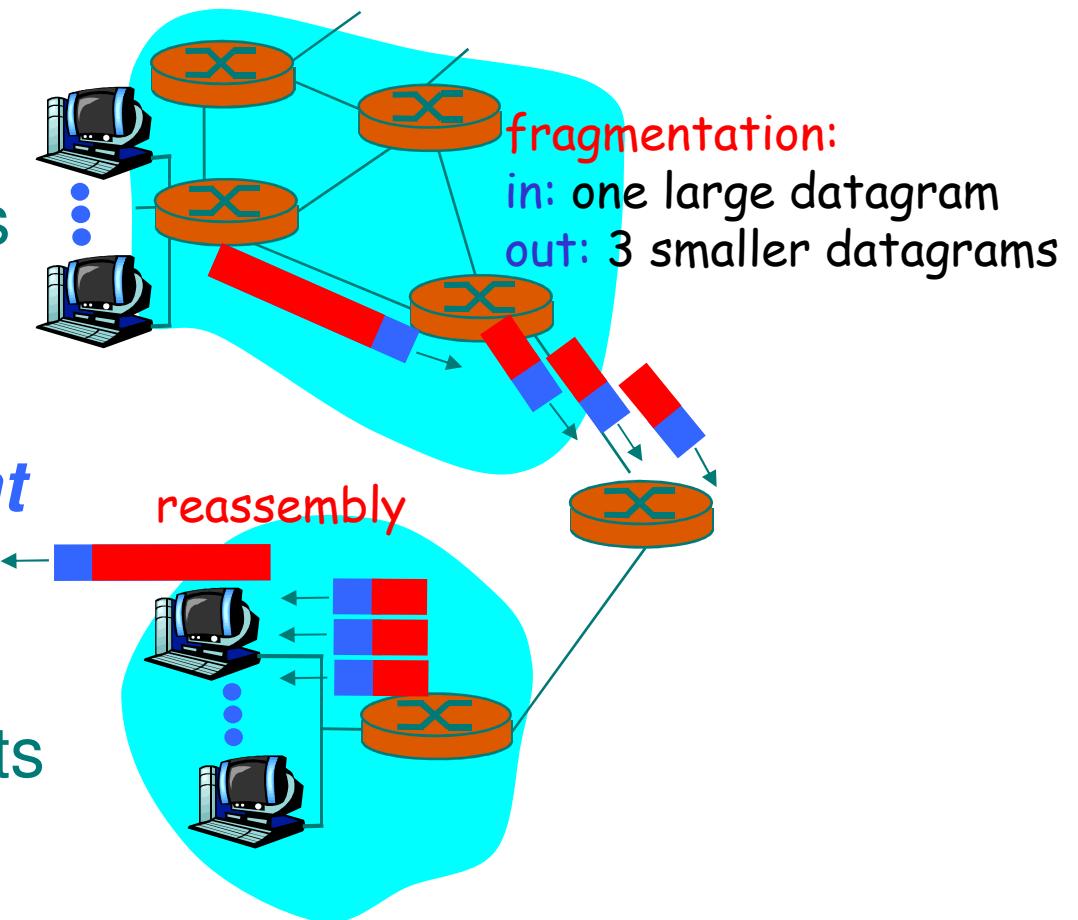
Figure 19.5: Maximum transfer unit (MTU)



MTU: Maximum size of frame payload

IP Fragmentation & Reassembly (cont.)

- ❖ Large IP datagram is divided (“fragmented”) within a network
- ❖ One datagram becomes several datagrams (fragments)
- ❖ ***“Reassembled” only at final destination***
- ❖ IP header bits used to identify related fragments and put them in order



IP Fragmentation Fields

- ❖ The 16-bit **Identification** field identifies a datagram originating from the source
- ❖ When a datagram is fragmented, the ID is copied into all fragments
- ❖ The 3-bit **Flags** field defines three flags:
 - ❖ 1st bit – not used
 - ❖ 2nd bit (**D-bit**), “*do not fragment*” bit - if 1, means must not fragment; if 0, means can be fragmented if needed
 - ❖ 3rd bit (**M-bit**), “*more fragment*” bit – if 1, means more fragments after this one; if 0, means this is the last or only fragment
- ❖ The 13-bit **Fragmentation Offset** field shows the offset (position) of this fragment in the original datagram ***in units of 8 bytes***

IP Fragmentation

Example

- Assume no optional fields in header
- 4000 byte datagram
- MTU = 1500 bytes
- Data size = $4000 - 20 = 3980$
- $= 1480 + 1480 + 1020$

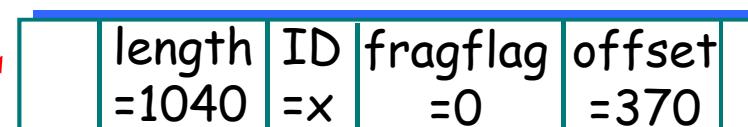
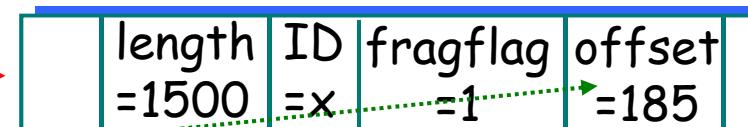
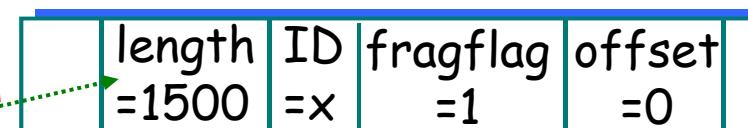
1480 bytes in data field

$$\text{offset} = \frac{1480}{8}$$

The fragflag is the M-bit



One large datagram becomes several smaller datagrams



The offset value is specified in units of **8 bytes**

IP Fragmentation

- ❖ (M-bit, offset)

- ❖ (0, 0) =>

no fragmentation

- ❖ (1, 0) =>

the first fragment

- ❖ (1, > 0) =>

a middle fragment and
more to be followed

- ❖ (0, > 0) =>

the last fragment

The fragflag is the M-bit

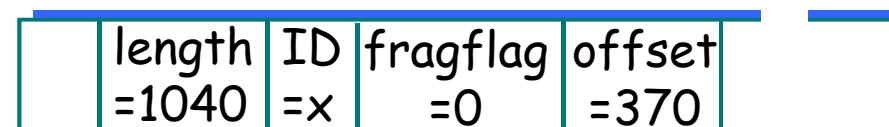
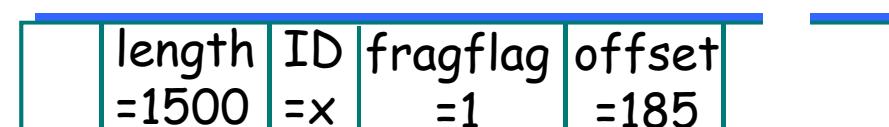
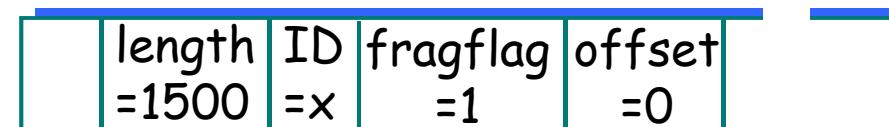
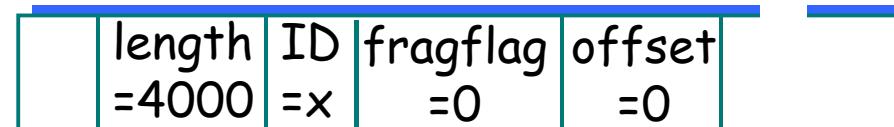
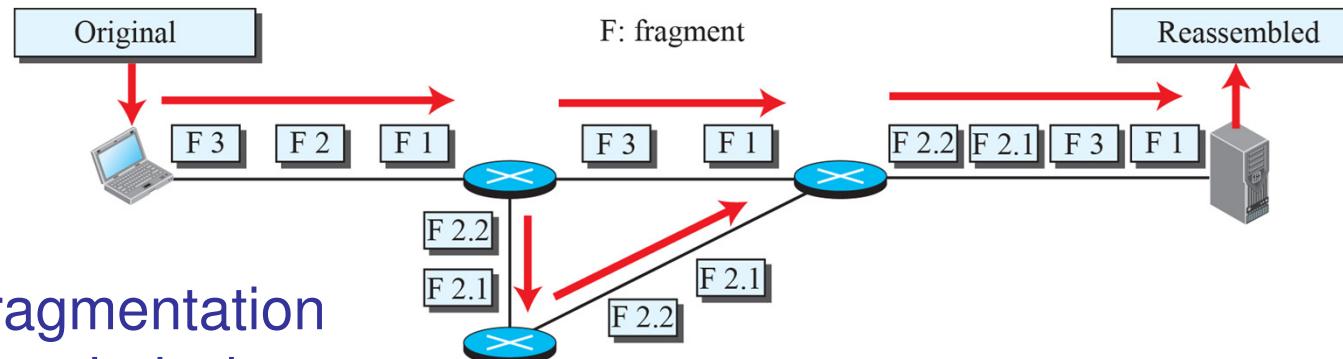
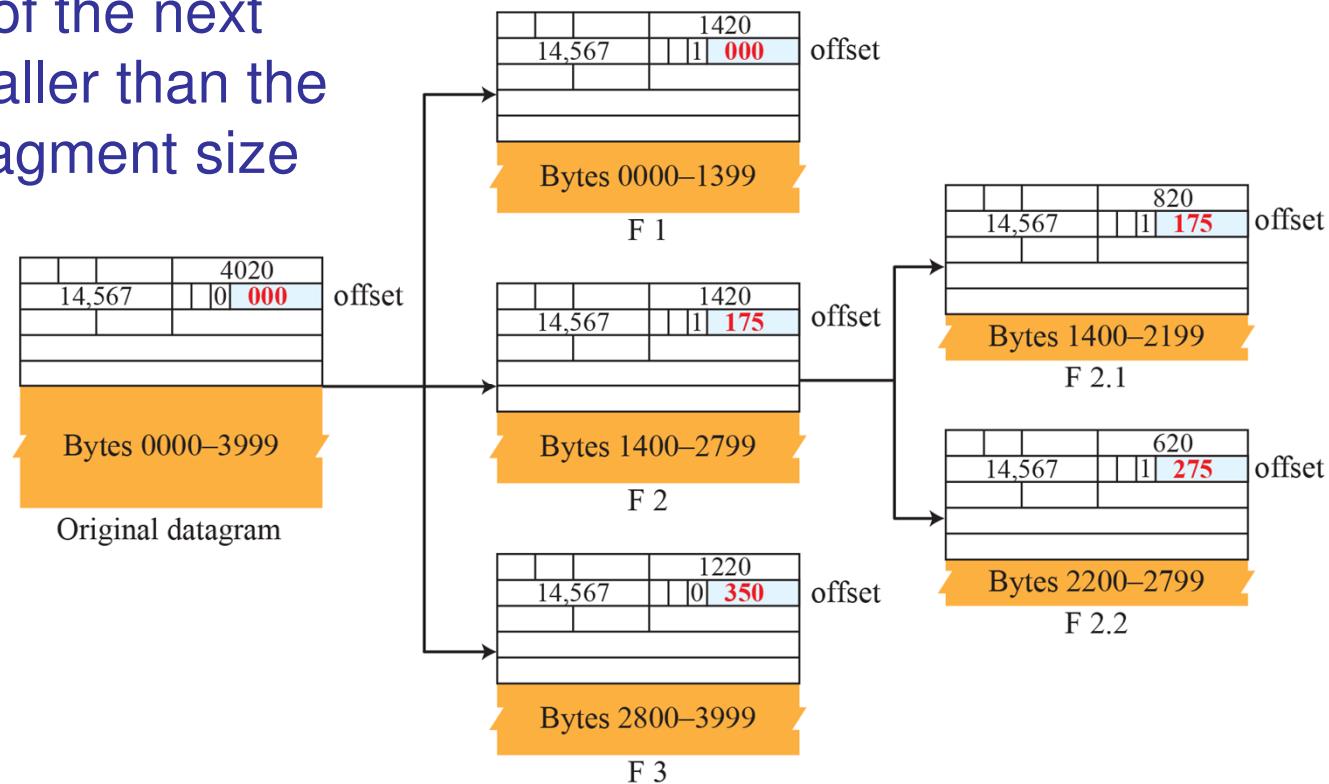


Figure 19.7: Detailed fragmentation example



Further fragmentation may be needed when the MTU of the next link is smaller than the current fragment size



Example 19.6

A packet has arrived with an M bit value of 0. Is this the first fragment, the last fragment, or a middle fragment? Do we know if the packet was fragmented?

Solution

If the M bit is 0, it means that there are no more fragments; the fragment is the last one.

However, we cannot say if the original packet was fragmented or not. A nonfragmented packet is considered the last fragment.

Example 19.7

A packet has arrived with an M bit value of 1. Is this the first fragment, the last fragment, or a middle fragment? Do we know if the packet was fragmented?

Solution

If the M bit is 1, it means that there is at least one more fragment.

This fragment can be the first one or a middle one, but not the last one. We don't know if it is the first one or a middle one; we need more information (the value of the fragmentation offset).

Example 19.9

A packet has arrived in which the offset value is 100. What is the number of the first byte? Do we know the number of the last byte?

Solution

To find the number of the first byte, we multiply the offset value by 8.

This means that the first byte number is 800. We cannot determine the number of the last byte unless we know the length of the data.

Example 19.10

A packet has arrived in which the offset value is 100, the value of HLEN is 5, and the value of the total length field is 100. What are the numbers of the first byte and the last byte?

Solution

The first byte number is $100 \times 8 = 800$.

The total length is 100 bytes, and the header length is 20 bytes (5×4),

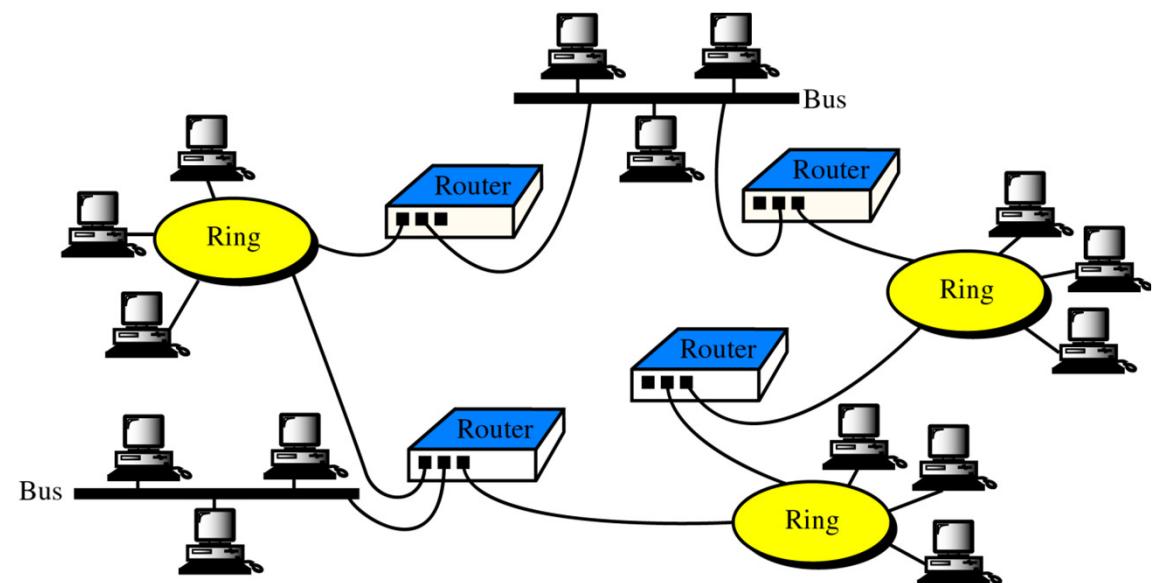
which means that there are 80 bytes in this datagram.

If the first byte number is 800, the last byte number must be 879.

Routers



- ❖ They connect networks and decide the path a packet should take
- ❖ The routing table are normally dynamic and are updated using routing protocols



C. Routing Algorithms

- ❖ Decide which **output link (path)** a packet should be transmitted in order to reach the destination
- ❖ There are many different routing algorithms:
 - ❖ It is used to exchange information among routers to build and maintain their **routing tables**
 - ❖ Decision may base on some criteria: e.g. shortest path, least cost (Compare it with how you reach the campus from home)
 - ❖ In routing the term **shortest** can mean the combination of many factors including shortest, cheapest, fastest, most reliable and so on.

Routing Algorithm Classification

Global or decentralized?

Global:

- ❖ All routers have complete topology, link cost information
- ❖ e.g. “link state” algorithms

Decentralized:

- ❖ Router knows neighbors' information, link costs to neighbors only
- ❖ Iterative process of computation, exchange of information with neighbors
- ❖ e.g. “distance vector” algorithms

Static or dynamic?

Static:

- ❖ Routes (the routing tables) change slowly over time

Dynamic:

- ❖ Routes change more quickly
 - ❖ periodic update
 - ❖ in response to link cost changes

Next-Hop Routing

- ❖ Each host (source/destination) and router have their own (local) routing tables.
- ❖ Look at this table to find the route to the final destination (i.e. the next router to be visited)
- ❖ The table holds only the information that leads to the next hop (neighbor)
- ❖ The entries of the routing tables are generated by a selected algorithm and must be consistent with each other

Next-Hop Routing

Routing table for host S based on host-specific routing

| Destination | Next Hop |
|-------------|----------|
| A | R1 |
| B | R1 |
| C | R1 |
| D | R1 |

Also see the example:

SEHH2238_T8_IP_RoutingTableExample.pdf

Routing table for host S based on network-specific routing

| Destination | Next Hop |
|-------------|----------|
| N2 | R1 |

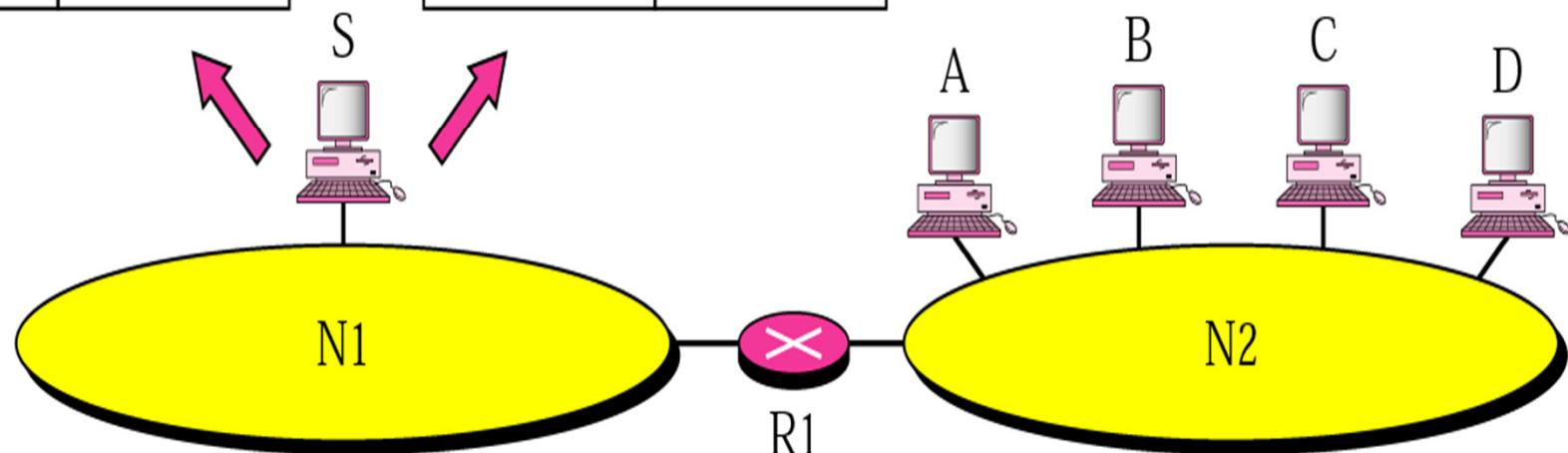
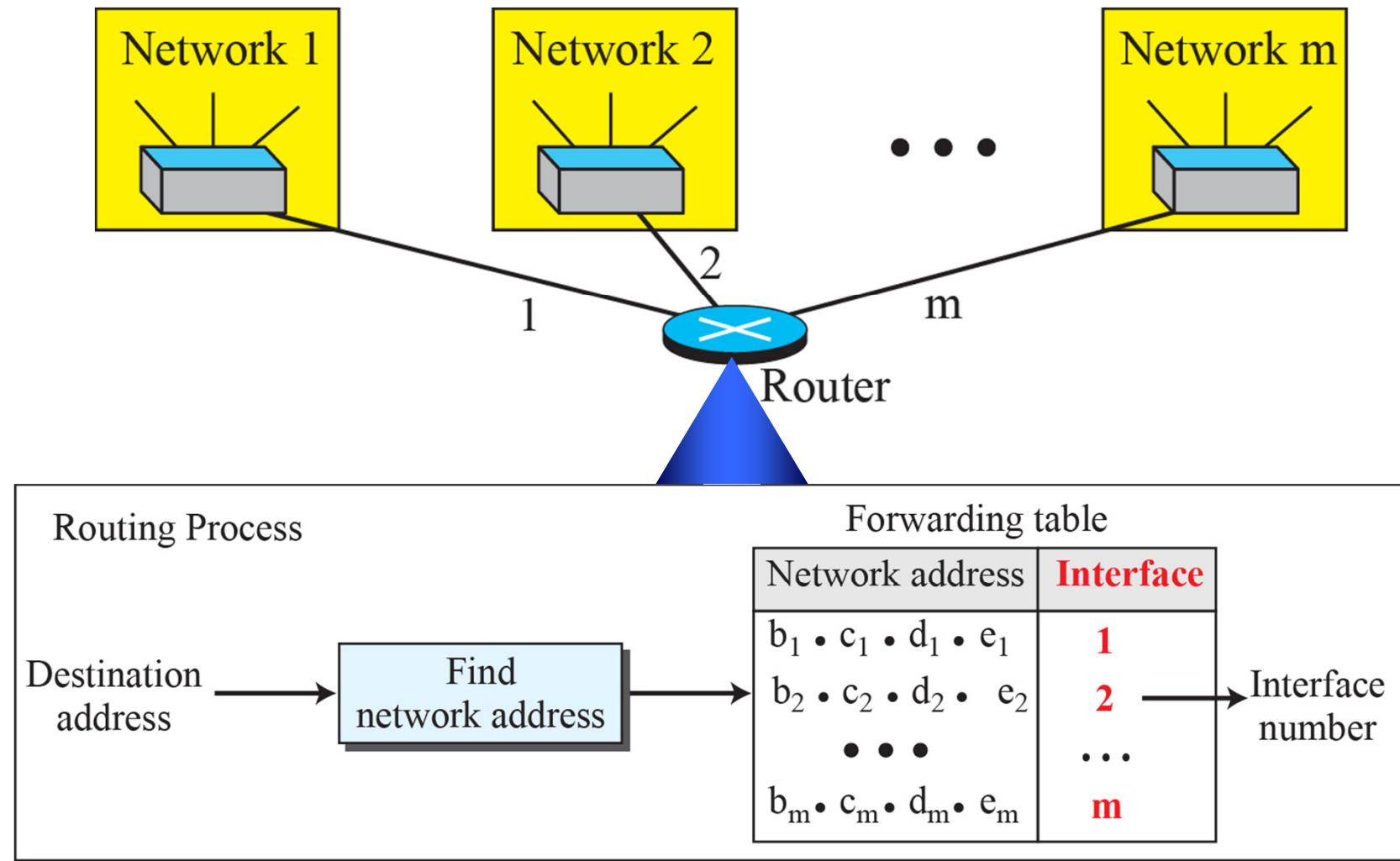


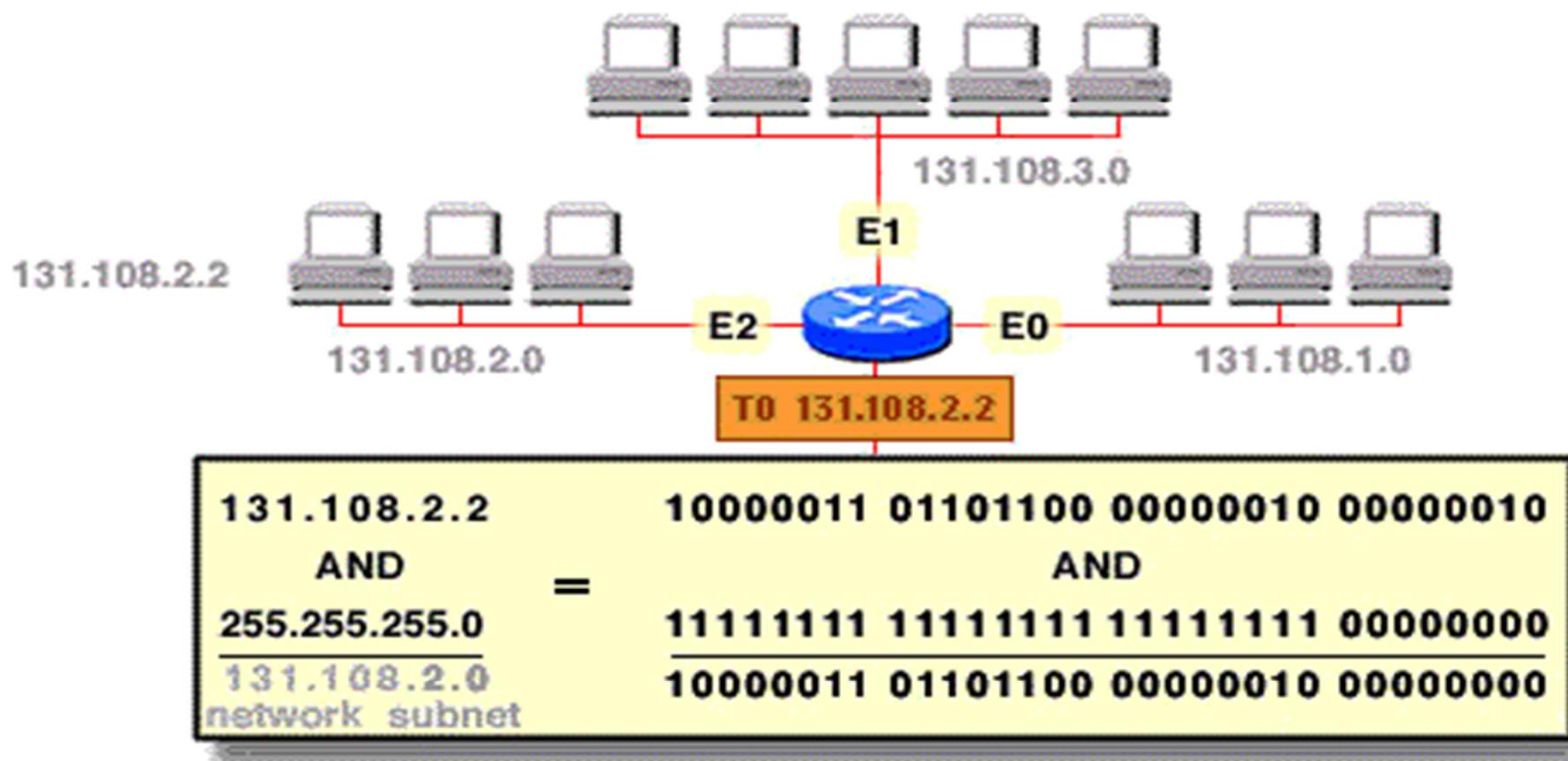
Figure 18.22: Network address

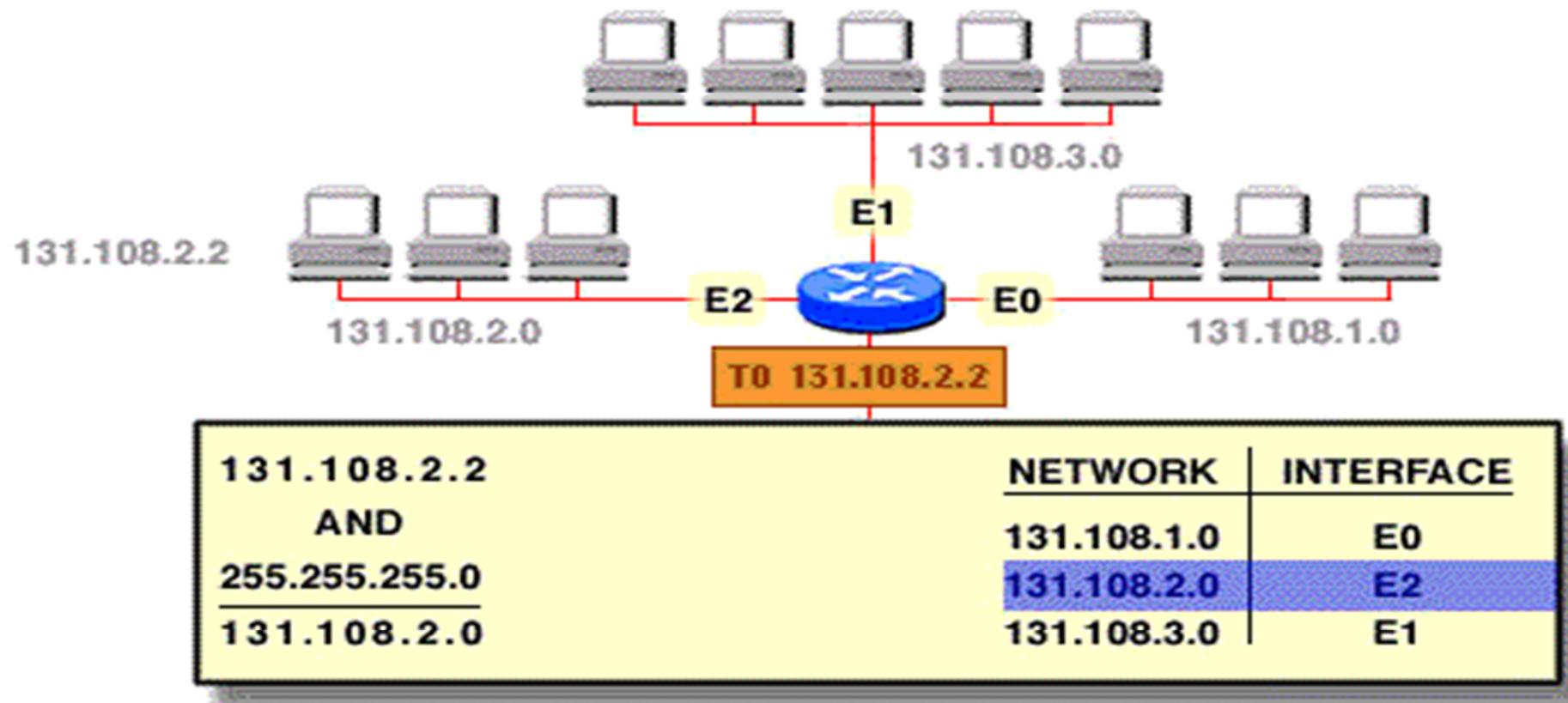


The network address is the identifier of the network and it is used by the forwarding (or routing) table in the Internet

Routing Example

- ❖ Assume that a station wants to send data with destination IP address of 131.108.2.2
- ❖ The data is sent out over the Internet until it reaches the router that is attached to the network
- ❖ The router in the destination network will determine which one of the subnets the data should be routed to
- ❖ The router knows that the subnet mask is 255.255.255.0





Forwarding table
at the router

D. LAN Addresses and Address Resolution Protocol

32-bit IP address:

- ❖ *Network-layer* address
- ❖ Used to send datagram to destination IP network
- ❖ A logical address

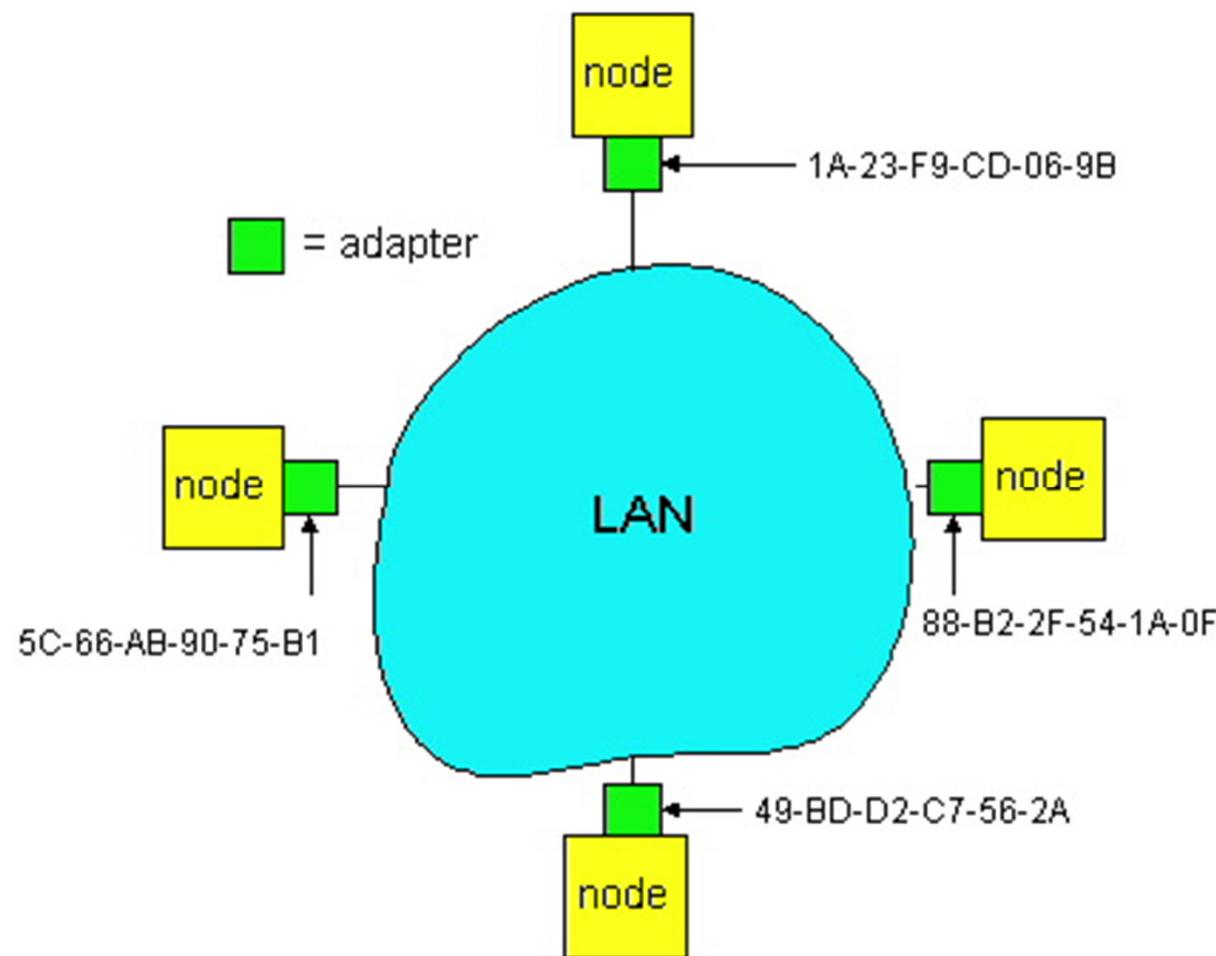
LAN (or MAC or Ethernet) address:

- ❖ A physical address
- ❖ Used to send datagram from one node to another physically-connected node (in the same network)
- ❖ **48 bit** MAC address (for most LANs) which is burned (hard-coded) in the adapter ROM

Physical Address. : 00-25-26-56-8D-30

LAN Addresses and ARP

Each adapter on LAN has unique LAN address



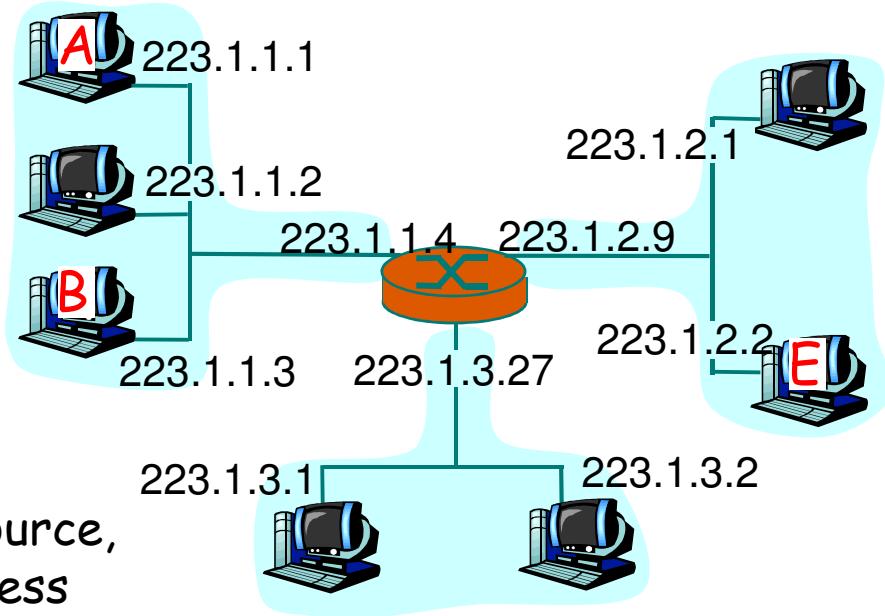
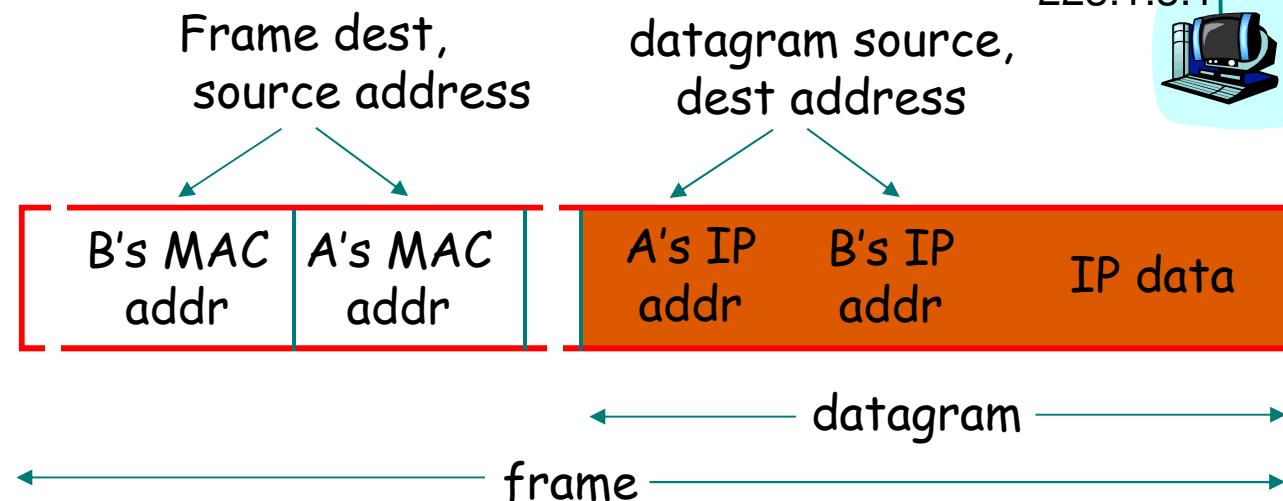
LAN Address (more)

- ❖ MAC address allocation administered by IEEE
- ❖ Manufacturer buys MAC address space (to assure uniqueness)
- ❖ **MAC address** => **portable**
 - ☞ can move LAN card from one LAN to another LAN (the station (LAN card) still uses the same MAC address)
- ❖ **IP address** (structured) => **NOT portable**
 - ☞ depends on the IP of the attached network
 - ☞ when moving to another network, the station needs to change its IP address (without changing the MAC address if the same LAN card is used)

Recall earlier routing discussion

Starting at A, given IP datagram addressed to B:

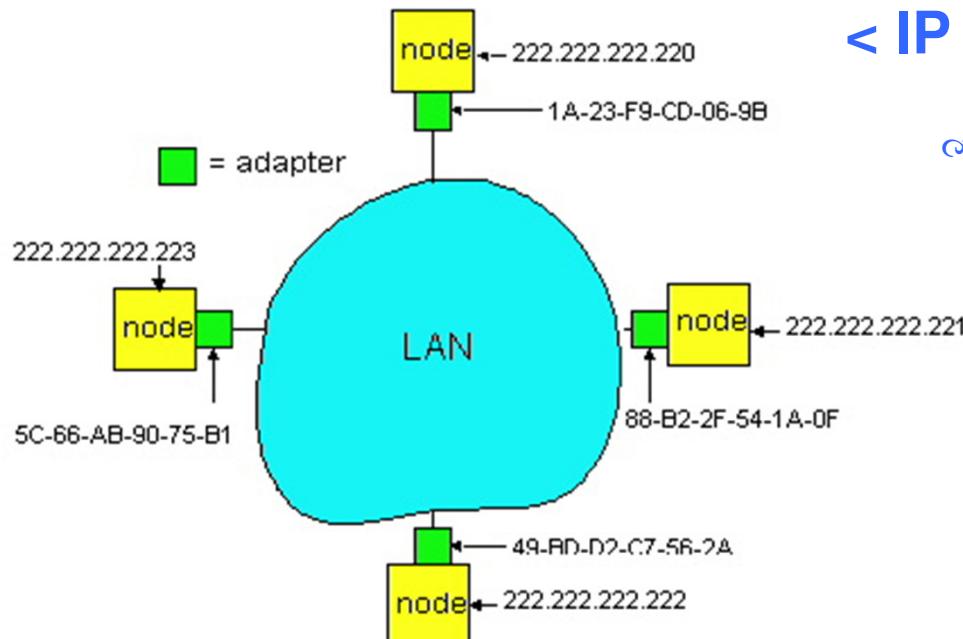
- look up net. address of B, find B on same net. as A
- *link layer send datagram to B inside link-layer frame*



ARP: Address Resolution Protocol

How to determine MAC address of B knowing B's IP address?
- Done by ARP

- ❖ Each IP node (Host, Router) on LAN has an ARP table
- ❖ **ARP Table: IP/MAC address mappings** for some LAN nodes



<IP address; MAC address; TTL>

- ❖ TTL (Time To Live): time after which address mapping will be forgotten (typically 20 min)

ARP Protocol

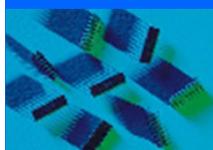
- ❖ “A” wants to send datagram to “B”, and A knows B’s IP address.
- ❖ Suppose B’s MAC address is not in A’s ARP table.
- ❖ An ***ARP packet*** contains the sending and receiving IP and MAC addresses
- ❖ “A” **broadcasts *ARP query*** packet, containing B’s IP address (as well as A’s IP & MAC addresses)
 - ☞ all nodes on LAN receive ARP query

ARP Protocol (cont.)

- ❖ “B” receives ARP query packet, then replies to “A” with its (B's) MAC address
 - ❖ *ARP reply* packet sent to A's MAC address only
- ❖ After receiving the ARP reply packet, “A” **caches** (saves) IP-to-MAC address pair of “B” in its ARP table until information becomes old (times out- TTL expires)

Summary

- ❖ Network Layer Functions
- ❖ Internet Protocol (IP)
 - ❖ Connectionless, unreliable, best-effort
- ❖ Routing
 - ❖ Routing Algorithms, Next-hop Routing Table
- ❖ LAN Addressing
 - ❖ Address Resolution Protocol (ARP)
- ❖ Revision Quiz
 - ❖ http://highered.mheducation.com/sites/0073376221/student_view0/chapter_19/quizzes.html



Lecture 9

Automatic Repeat Request (ARQ)

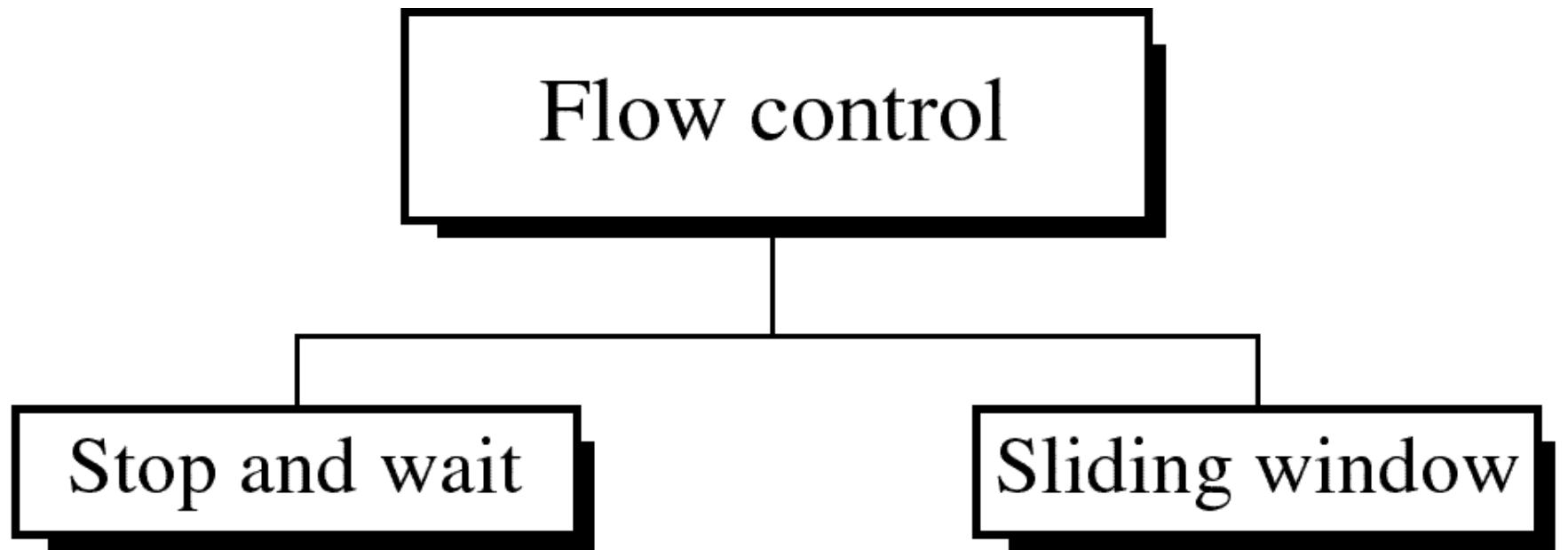
Protocols

Textbook: Ch.23

Main Topics

- A. Sliding Window
- B. Revision on Stop-and-Wait ARQ
- C. Go-Back-N ARQ
 - ❖ Cumulative ACK
 - ❖ Window size
- D. Selective Repeat ARQ
 - ❖ Individual Acknowledgement
 - ❖ Sender and Receiver Window Size

Flow & Error Control Mechanisms



**Send one frame
at a time**

☞ *Stop and Wait ARQ*

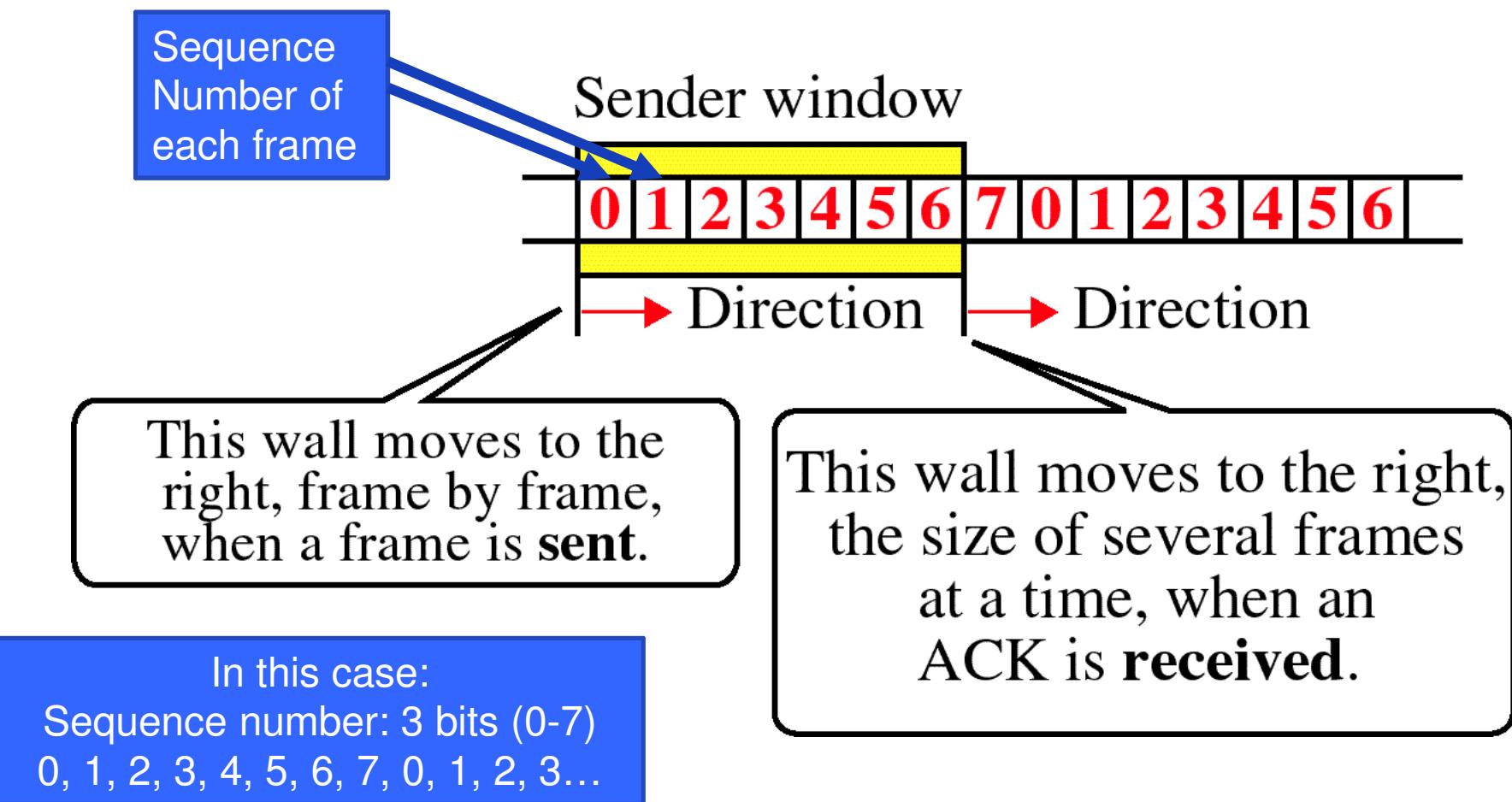
**Send several frames
at a time**

- ☞ *Go-Back-N ARQ*
- ☞ *Selective Repeat ARQ*

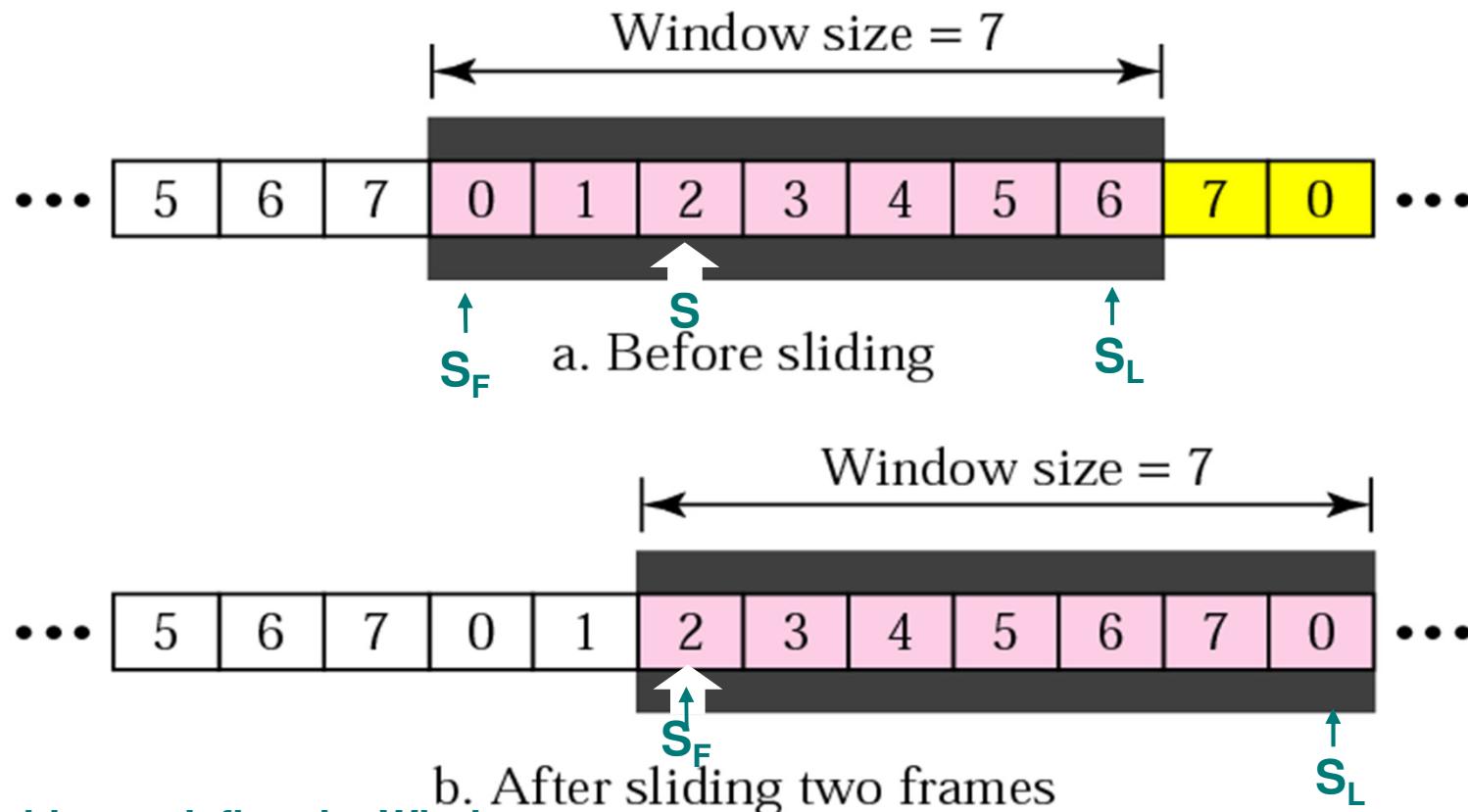
A. Sliding Window

- ❖ The sender can transmit several data frames before needing (receiving) an ACK
- ❖ A (logical) window is used to control the maximum no. of frames can be transmitted
- ❖ When the (variable) window size becomes zero, the sender stops transmissions and waits for an ACK
- ❖ A single ACK can be used to confirm the receipt of multiple data frames

Sender Sliding Window



Sender Sliding Window



Variables to define the Window

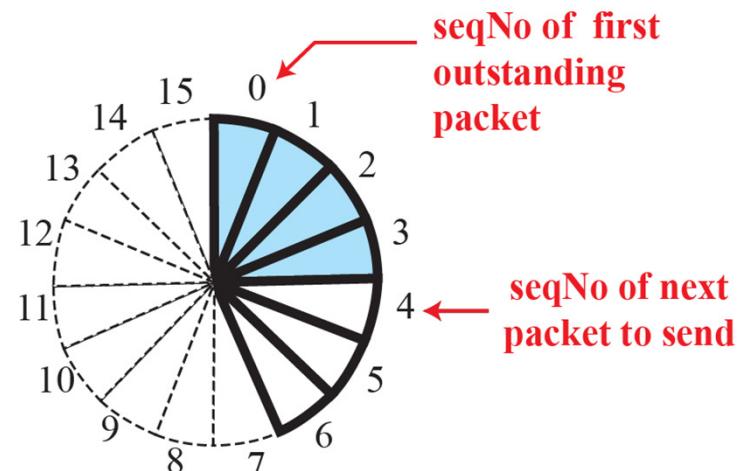
S_F : Seq. No. of 1st frame

S_L :Seq. No. of Last frame

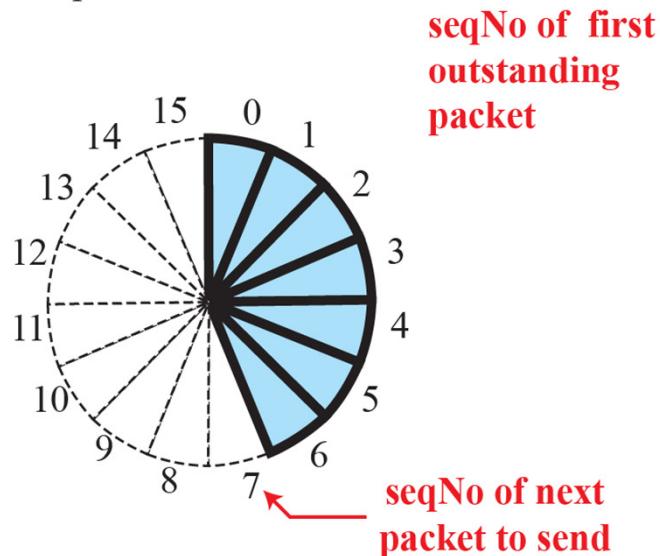
S : Seq. No. of next frame to be sent

Figure 23.12: Sliding window in circular format

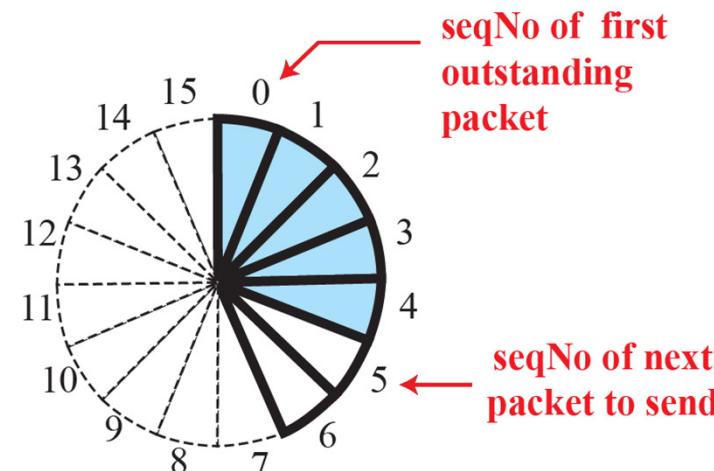
e.g. Window size = 7; Seq. no. has 4 bits (i.e. 0 - 15)



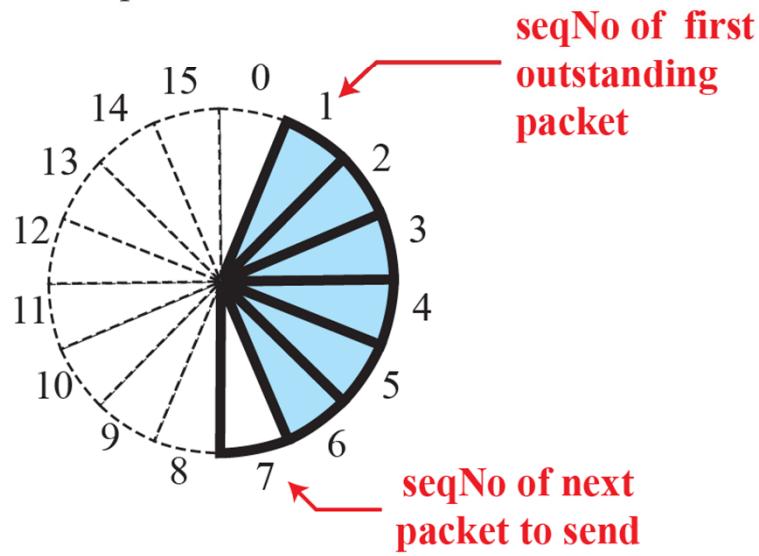
a. Four packets have been sent.



c. Seven packets have been sent;
window is full.



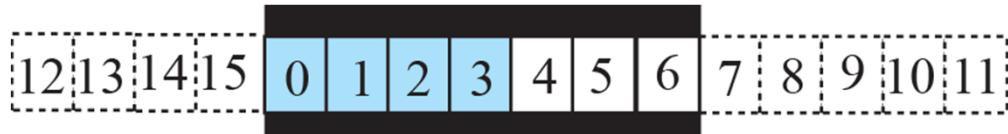
b. Five packets have been sent.



d. Packet 0 has been acknowledged;
window slides.

figure 20.10. Sliding window window

format e.g. Window size = 7 ; Seq. no. has 4 bits (i.e. 0 -15)



a. Four packets have been sent.



b. Five packets have been sent.

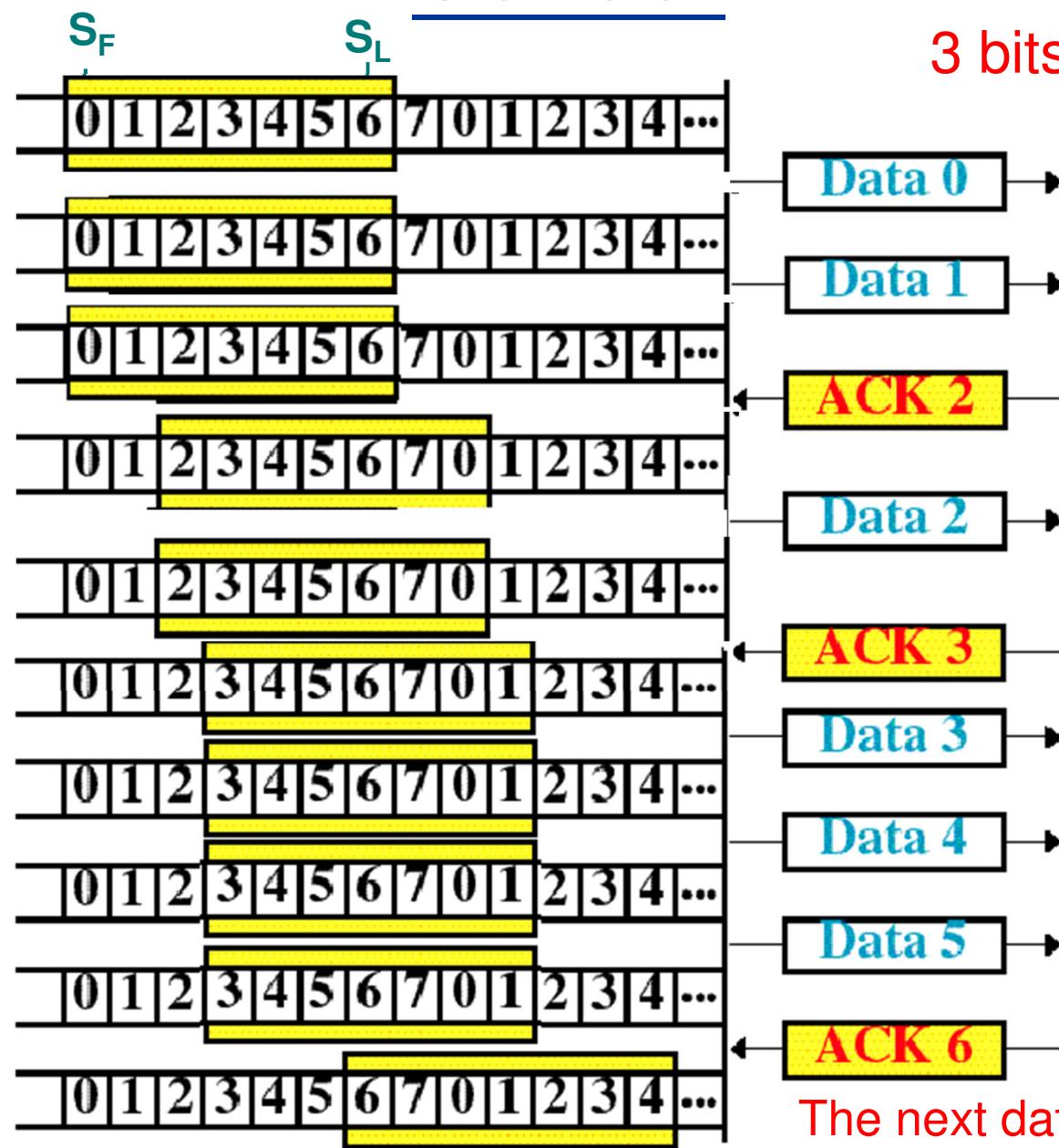


c. Seven packets have been sent;
window is full.



d. Packet 0 has been acknowledged;
window slides.

Sender



B. Stop-and-Wait ARQ

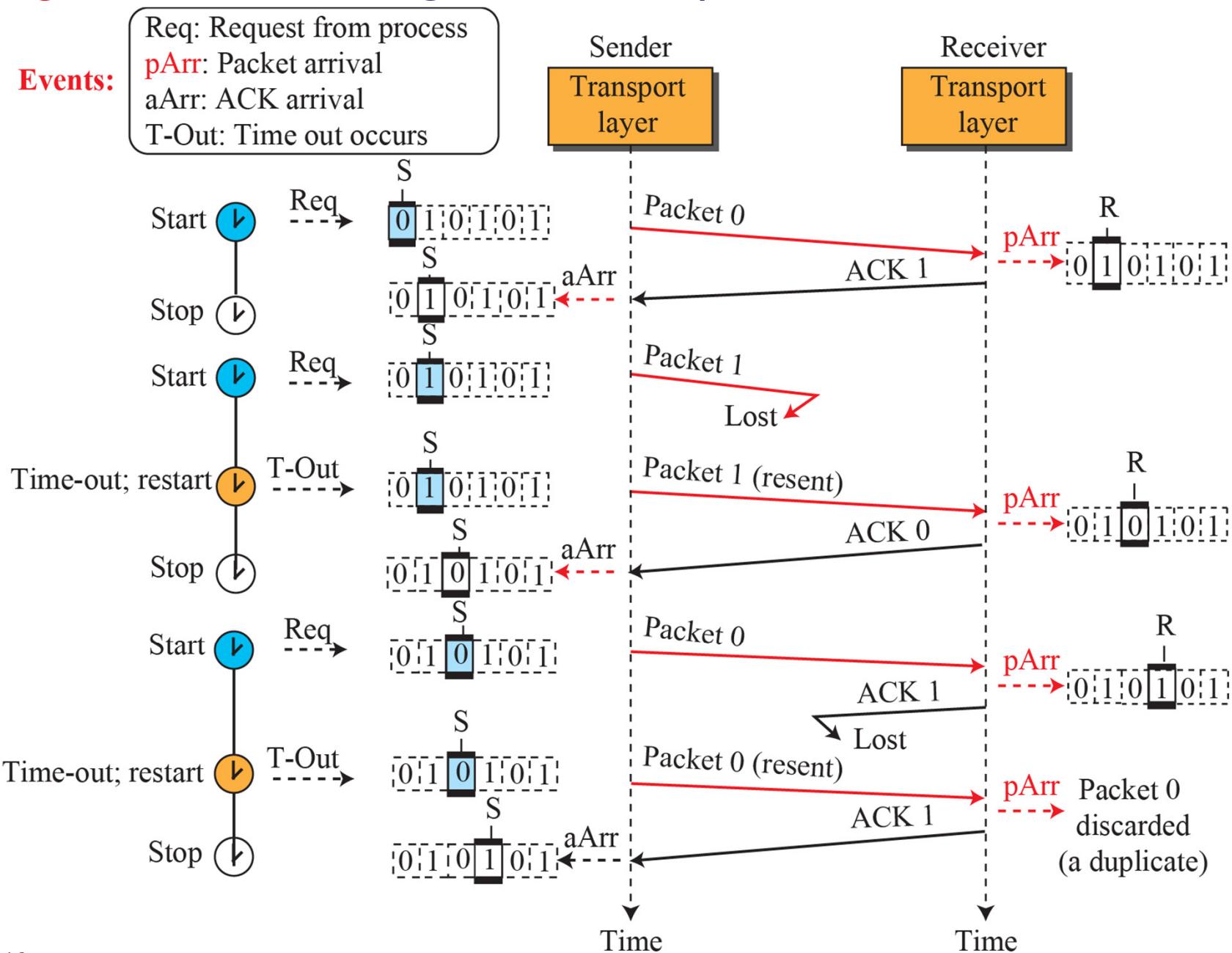
- ❖ The sender sends one frame and waits for an ACK before sending the next frame
- ❖ If no ACK is received after a period of time (timeout), the sender retransmits
- ❖ Use **1 bit sequence number**
- ❖ Both the sender and the receiver use a **sliding window of size 1**
- ❖ Advantage: Simple
- ❖ Disadvantage: Inefficient

Example 23.4

Figure 23.22 shows an **example of the Stop-and-Wait protocol.**

1. Packet 0 is sent and acknowledged.
2. Packet 1 is lost and resent after the time-out.
3. The resent packet 1 is acknowledged and the timer stops.
4. Packet 0 is sent and acknowledged, but the acknowledgment is lost. The sender has no idea if the packet or the acknowledgment is lost, so after the time-out, it resends packet 0, which is acknowledged.

Figure 23.22: Flow diagram for Example 3.4



C. Go-back-N (GBN) ARQ

- ❖ Use the concept of sliding window
 - ❖ Multiple frames are in transit while waiting acknowledgement
- ❖ At sender, the sliding window (buffer) holds the outstanding frames until they are acknowledged.
- ❖ Operation:
Sender sends multiple frames and set **a timer for each frame sent**. (The receiver has no timer)

S: Sequence no. of the recently sent frame

Go-Back-N ARQ, normal operation

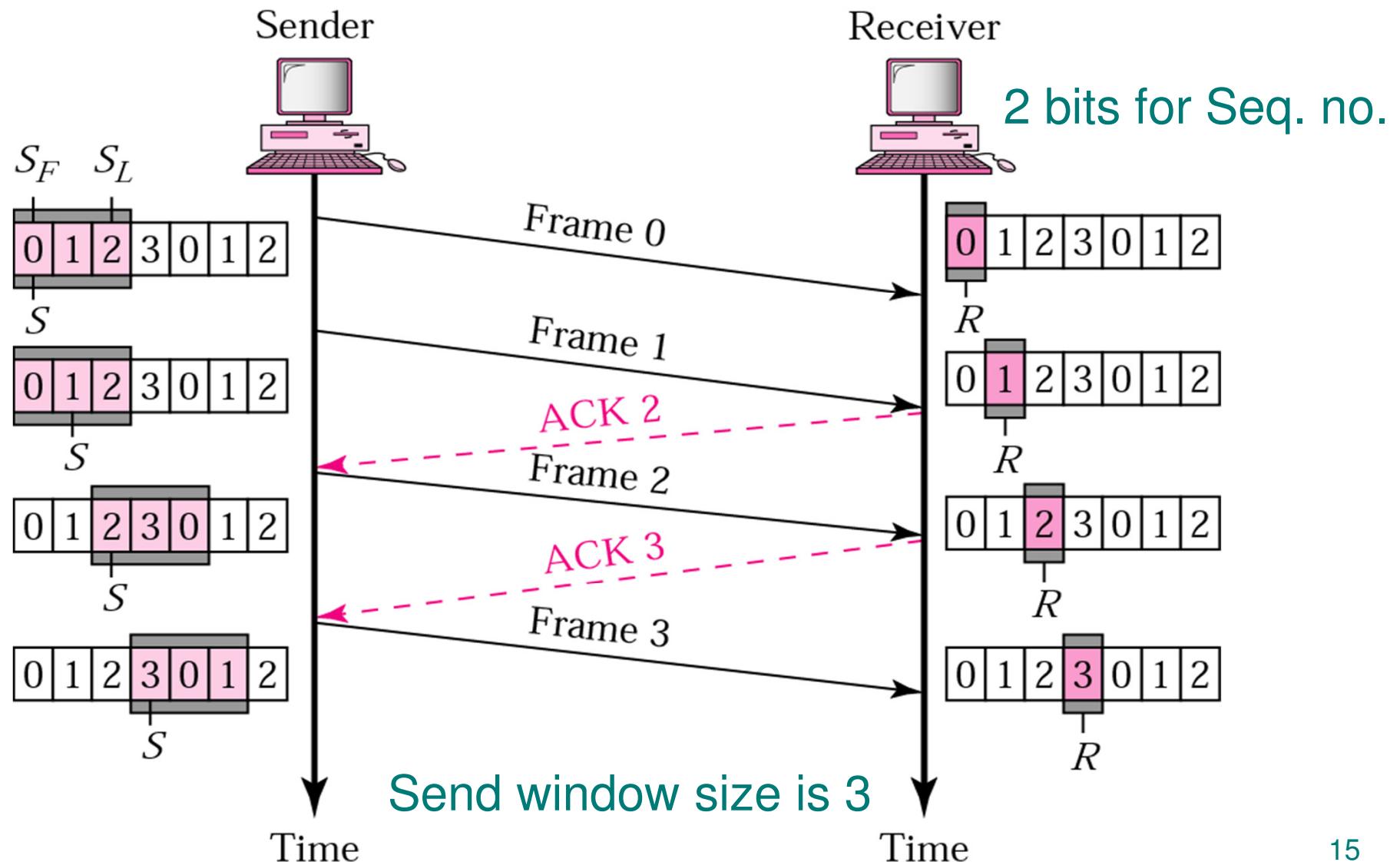
Receiver side

Case 1: frames arrived safe and in order

Receiver sends positive acknowledgements

R: Sequence number of the frame it ***expects*** to receive (R is contained in ACK packet)

Go-Back-N ARQ, normal operation



Go-Back-N ARQ, lost frame

Case 2: Frame is damaged or out of order

- ***Receiver is silent*** (send nothing) and discards all subsequent frames ***until*** it receives the one it is ***expecting***
- In sender, the timer for the unacknowledged frame expires
- The sender goes back and ***resends all frames, beginning from the one with the expired timer***

Go-Back-N ARQ, lost frame

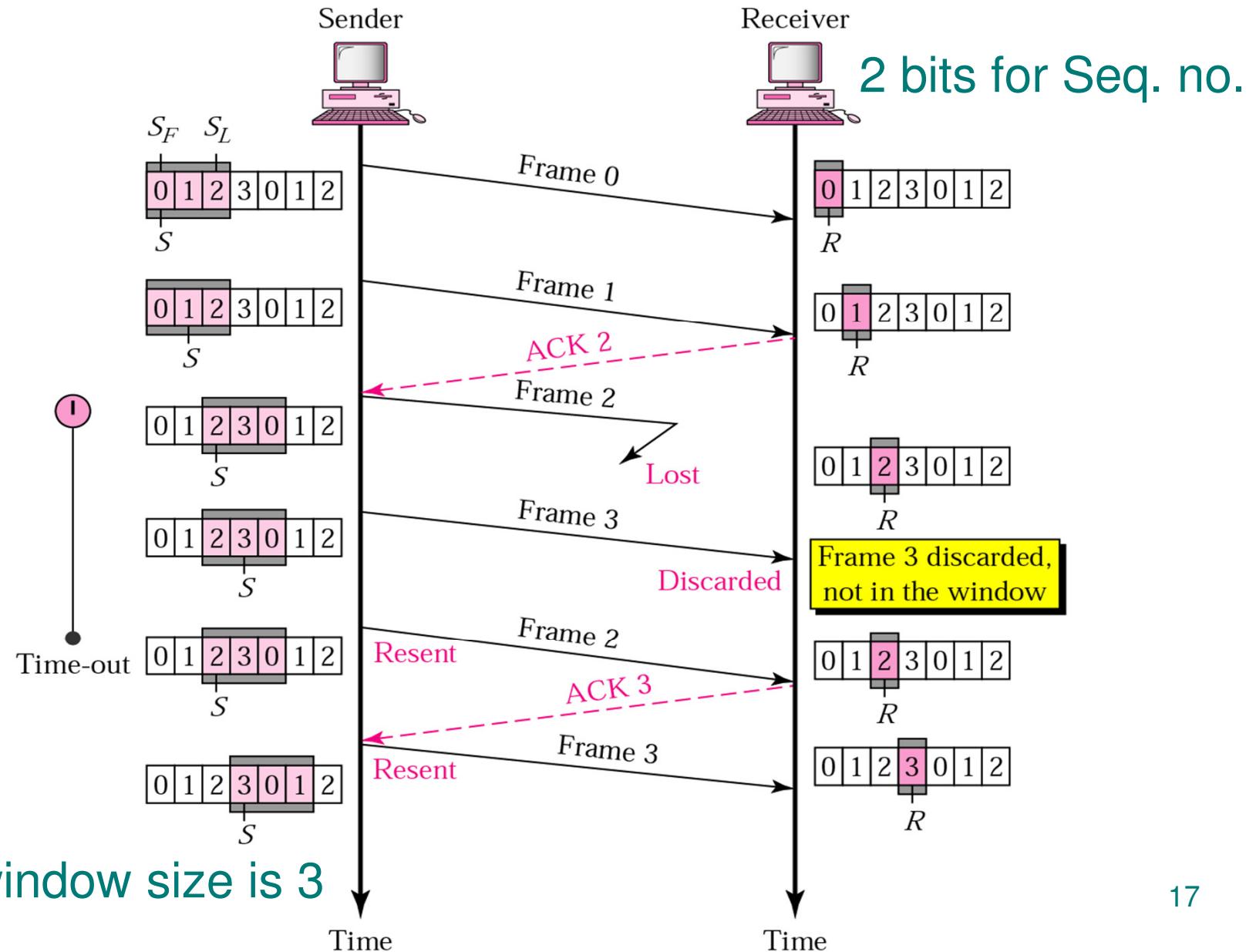


Figure 23.24: Send window for Go-Back-N

3 bits for Seq. no.

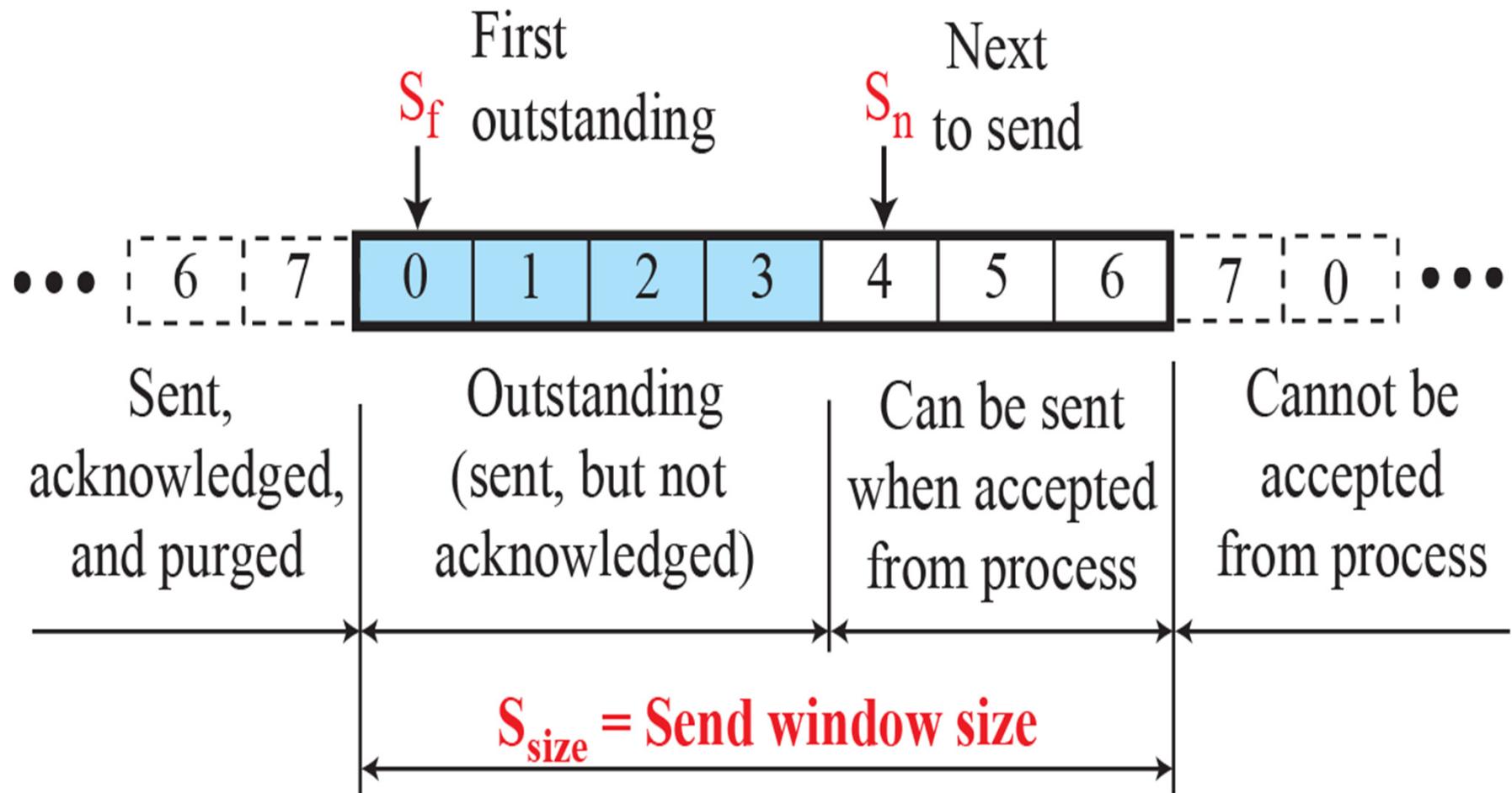
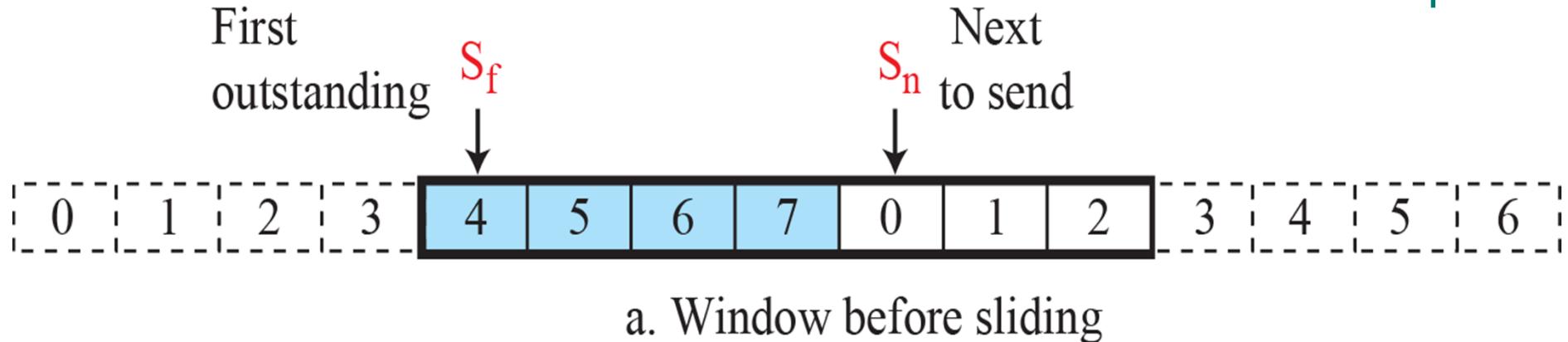


Figure 23.25: Sliding the send window

3 bits for Seq. no.



Send window size is 7

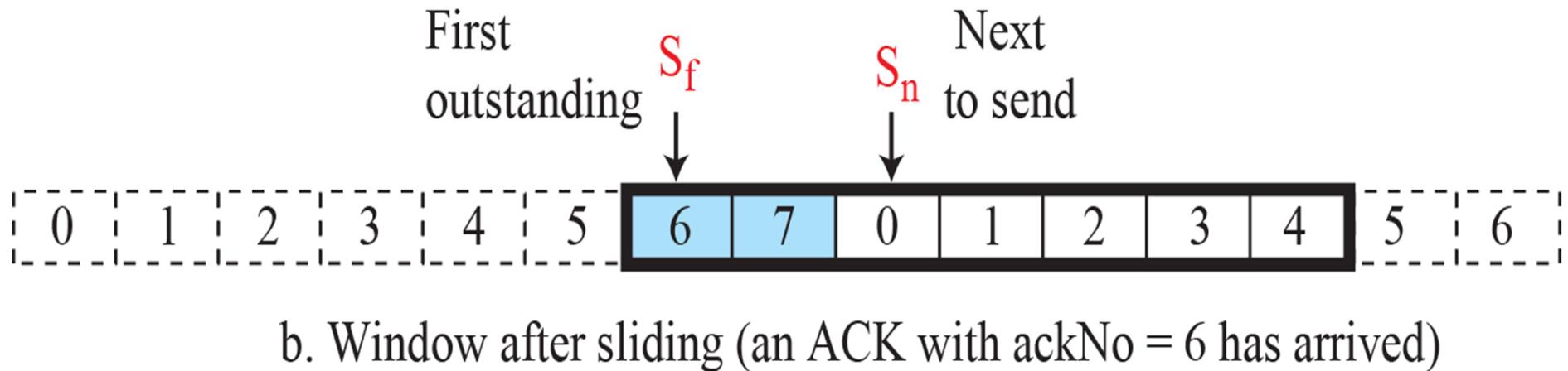
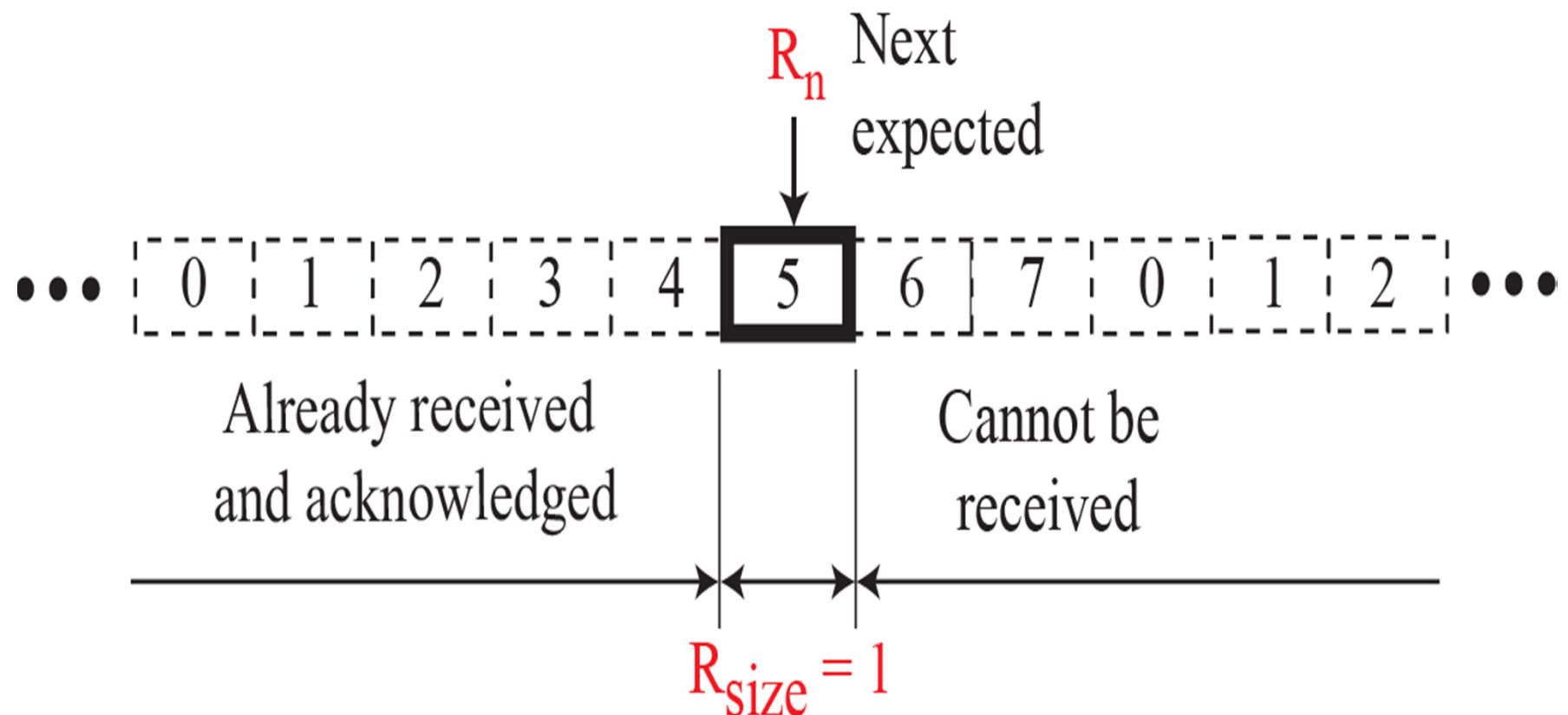
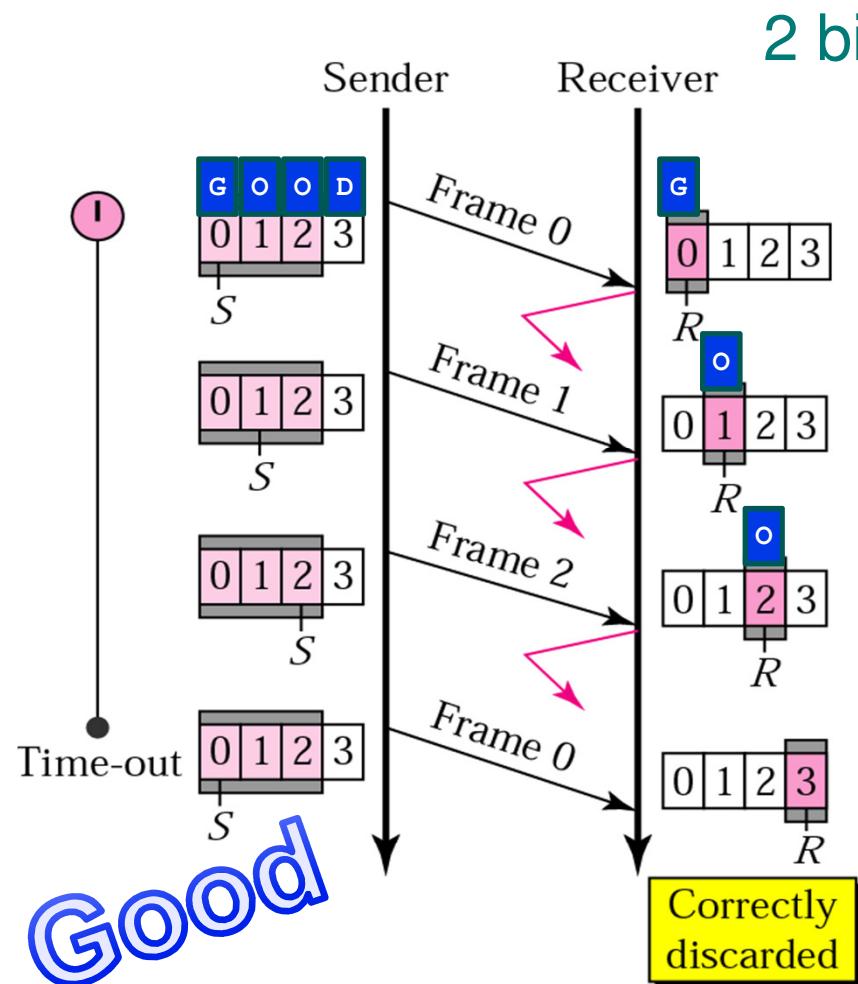


Figure 23.26: Receive window for Go-Back-N

3 bits for Seq. no.

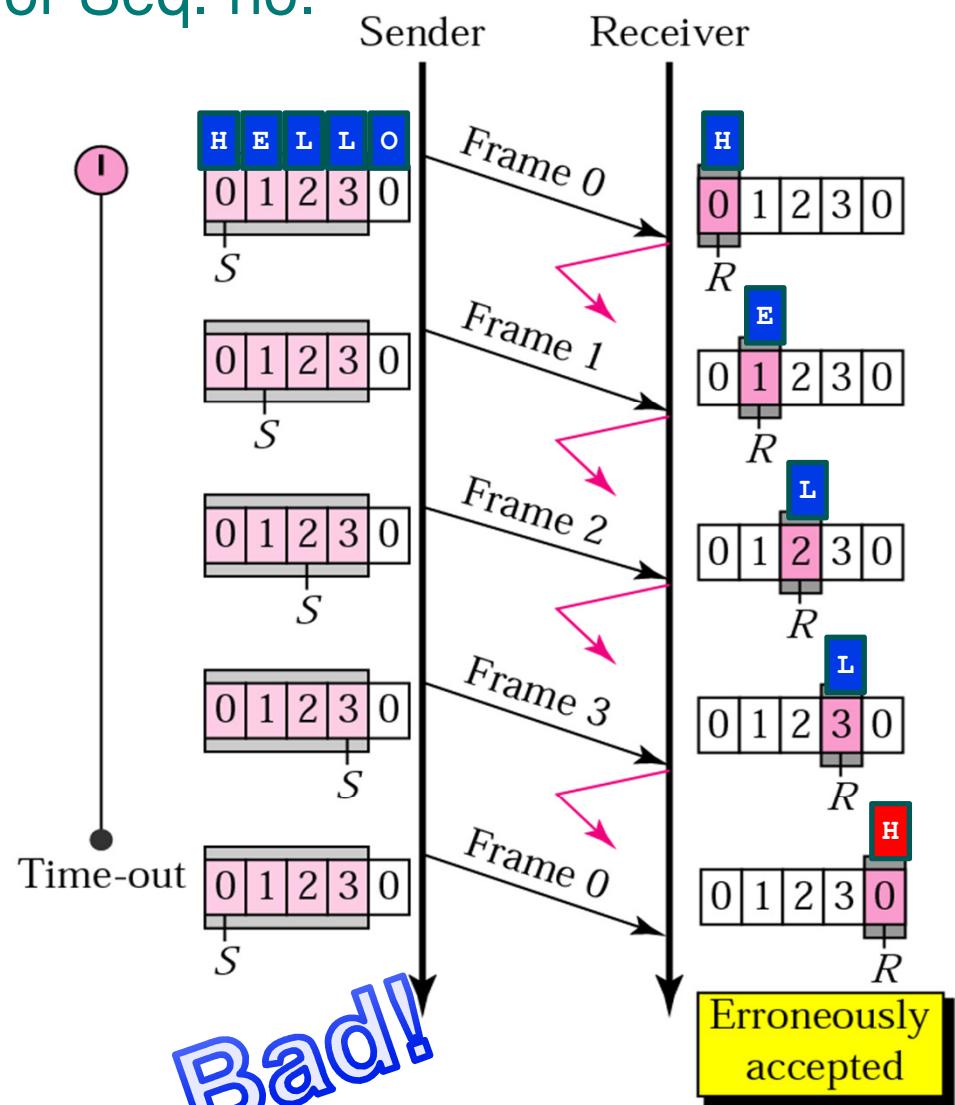


Go-Back-N ARQ: sender window size



m - no. of bits for Seq. no.

a. Window size $< 2^m$



b. Window size $= 2^m$

Sender Window Size



Note:

In Go-Back-N ARQ, the size of the sender window must be less than 2^m ;

the size of the receiver window is always 1.

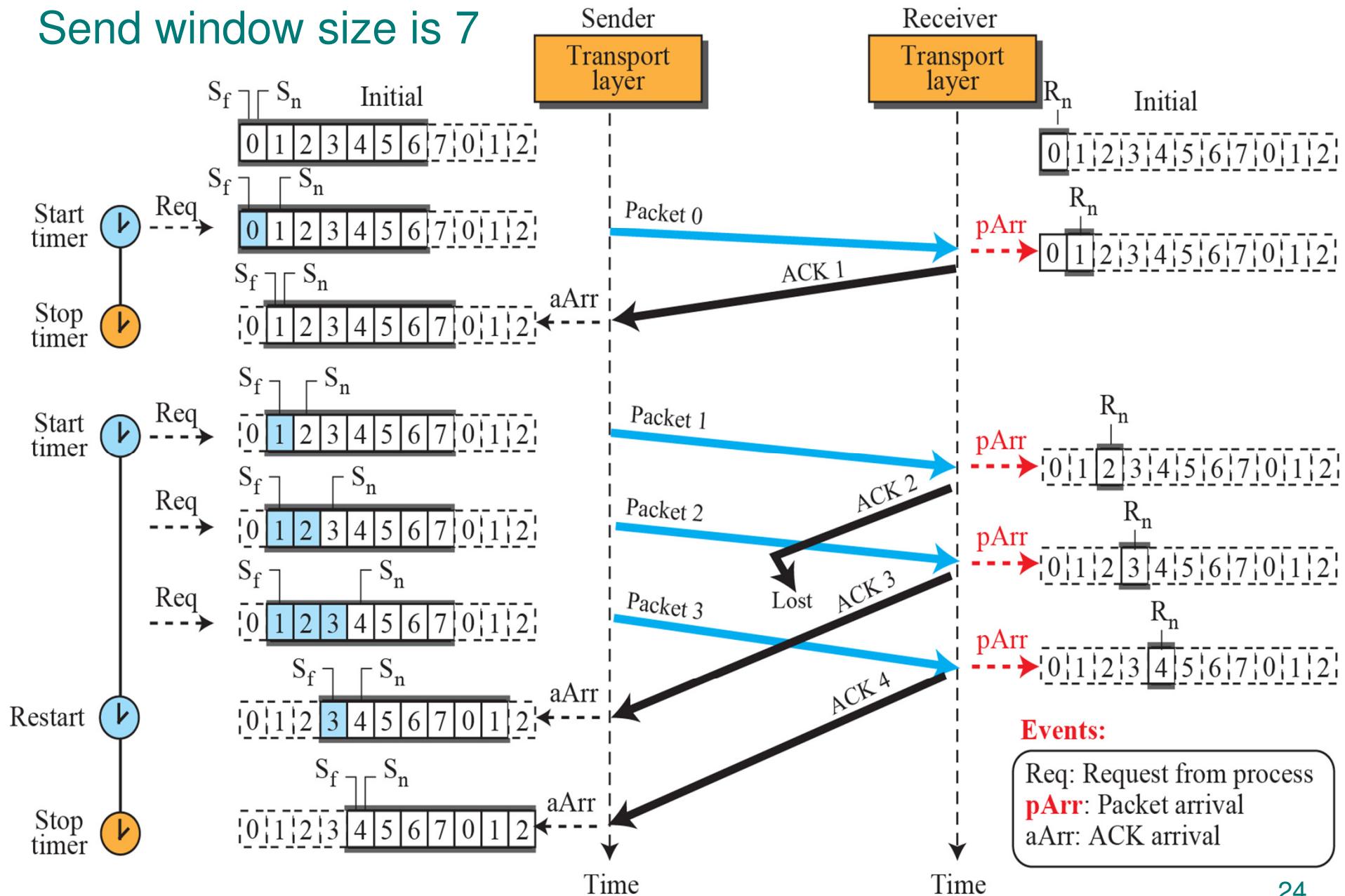
where m is the number of bits of the sequence number

Example 23.7

Figure 23.29 shows **an example of Go-Back-N**. This is an example of a case where the forward channel is reliable, but the reverse is not. No data packets are lost, but **some ACKs** are delayed and one is lost. The example also shows how **cumulative ACKs** can help if acknowledgments are delayed or lost.

Figure 23.29: Flow diagram for Example 3.7

Send window size is 7

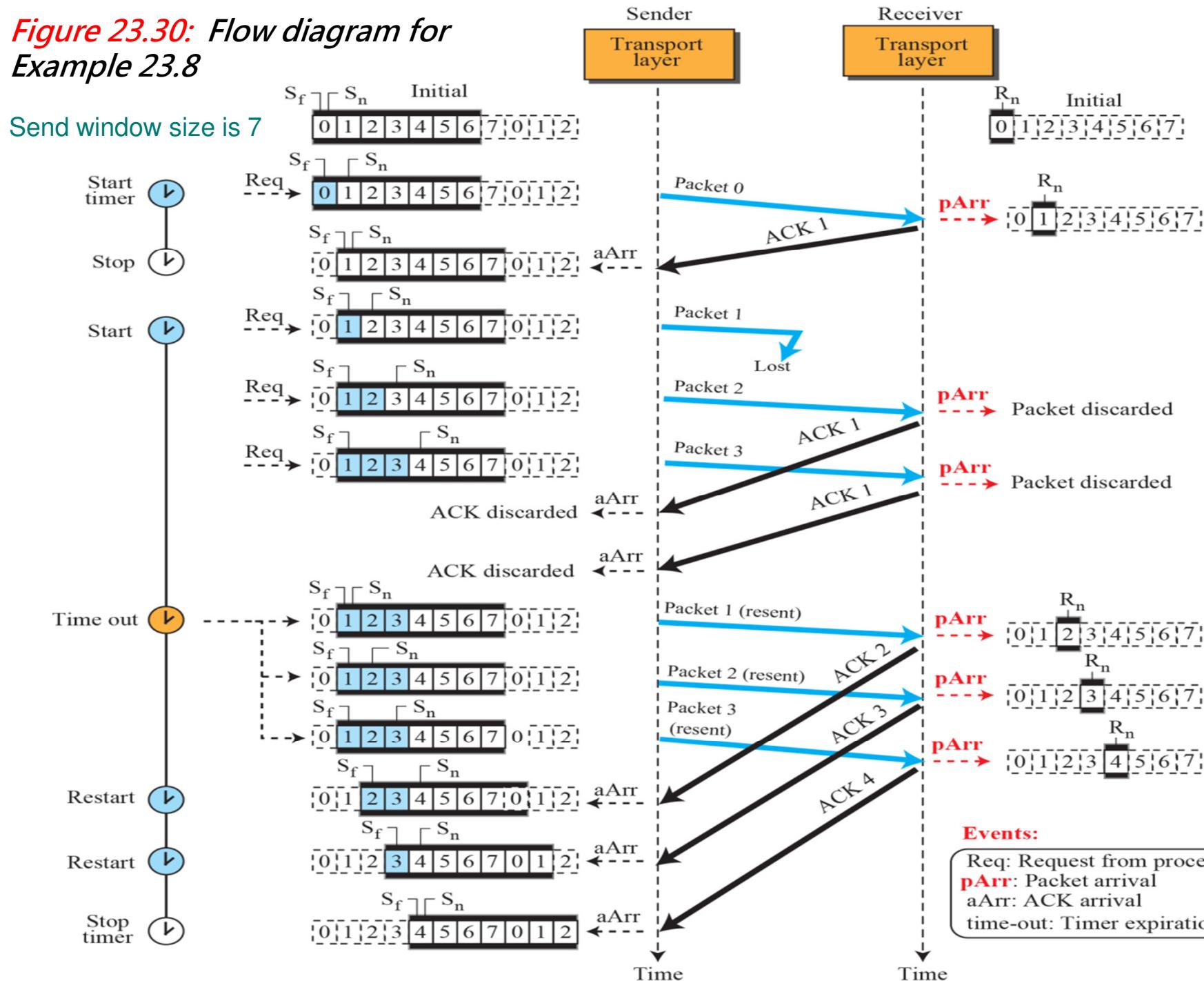


Example 23.8

Figure 23.30 shows what happens when a packet is lost. Packets 0, 1, 2, and 3 are sent.

- However, packet 1 is lost. The receiver receives packets 2 and 3, but they are **discarded because they are received out of order** (packet 1 is expected).
 - When the receiver receives packets 2 and 3, it sends ACK1 to show that it expects to receive packet 1.
 - However, these ACKs are not useful for the sender because the ackNo is equal to S_f , not greater than S_f . So the sender discards them.
- **When the time-out occurs, the sender resends** packets 1, 2, and 3, which will then be acknowledged.

Figure 23.30: Flow diagram for Example 23.8



Go-Back-N

Advantages & Disadvantages

- ❖ Maintain correct sequence
- ❖ Minimize the receiver buffer storage
- ❖ 1 storage unit is enough in the receiver buffer
- ❖ But need to retransmit some already correctly received frames
- ❖ Less efficient than selective-repeat

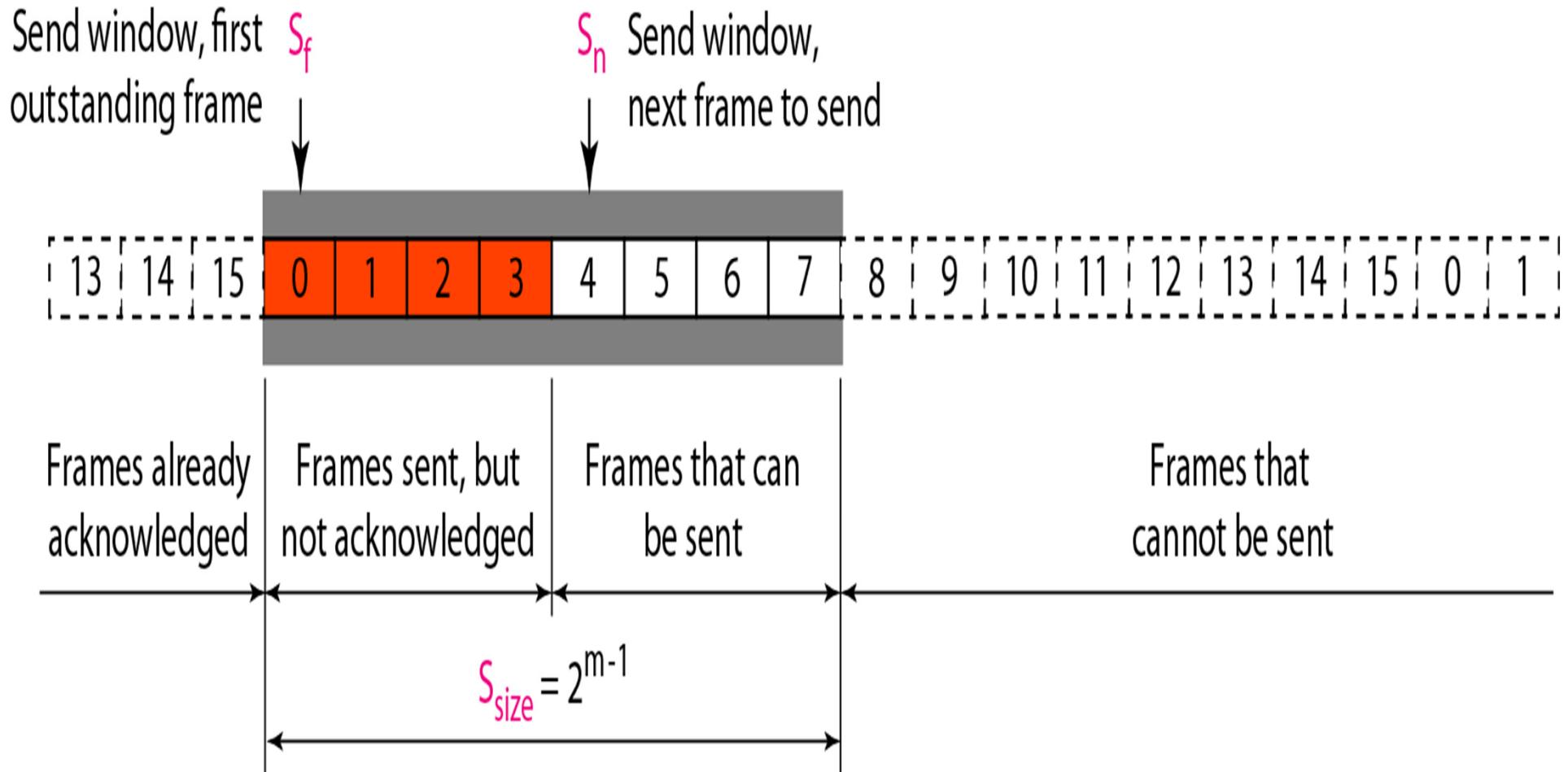
D. Selective Repeat ARQ

- ❖ In Go-back-N ARQ, the process at the receiver is simple
 - ❖ Receiver keeps track of only one variable
 - ❖ No need to buffer out-of-order frames
 - ❖ But multiple frames are resent when one frame is damaged
 - ❖ Use up bandwidth and slow down transmission
- ❖ Selective Repeat ARQ
 - ❖ Does not resend N frames when just one frame is damaged
 - ❖ Only the damaged frame is resent

Receiver in Selective Repeat ARQ

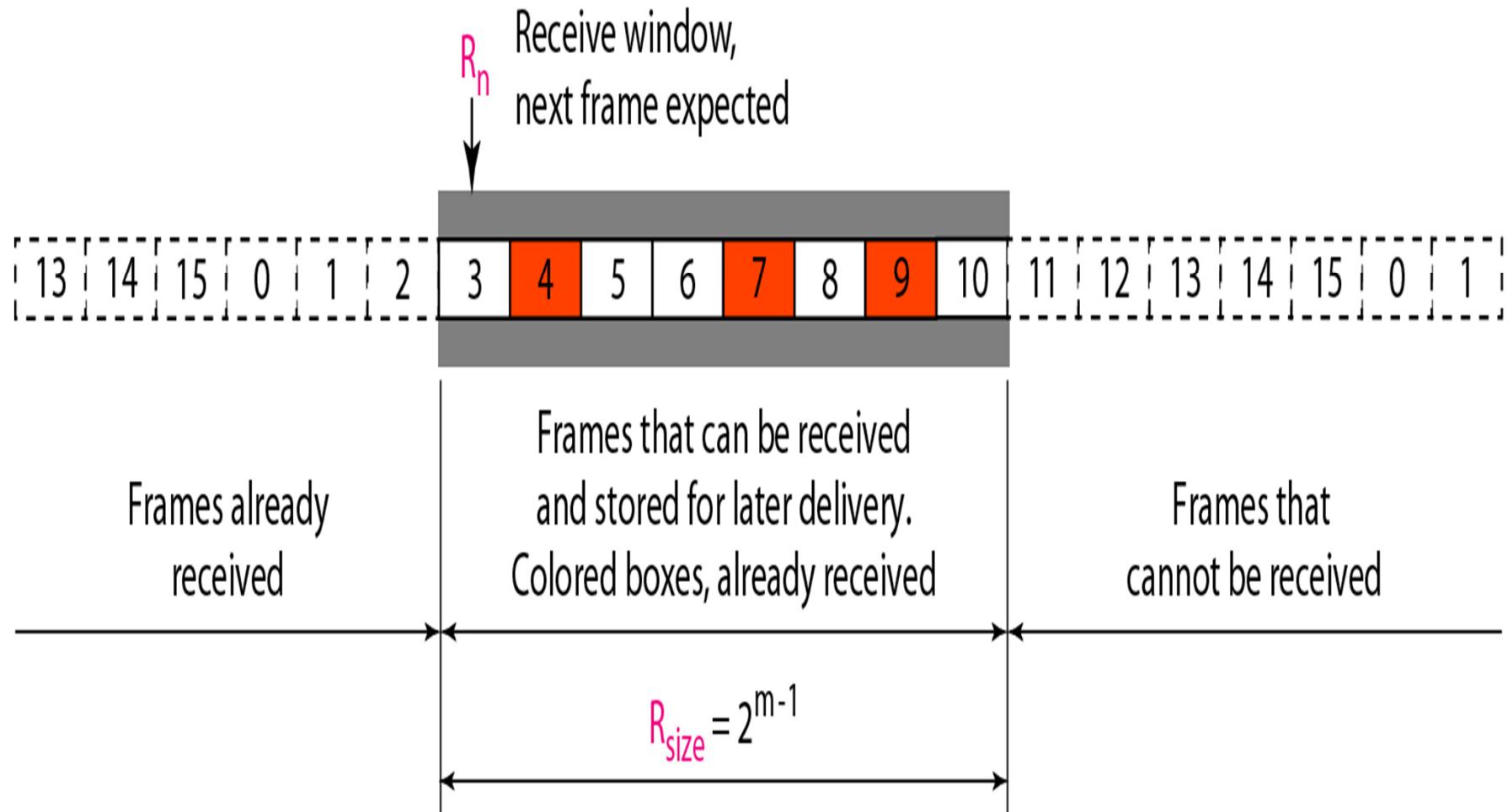
- ❖ Receiver detects those frames that are damaged
- ❖ Receiver retains out-of-sequence frames (without errors) in the **link receive list** until the next in-sequence frame is received; then a set of consecutive frames could be delivered to the upper layer
- ❖ ***Individual ACK***
 - ❖ **ACK(N)** *acknowledges only a single frame* with sequence number N
- ❖ In the sender, when a timer (waiting for ACK) expires, only the corresponding frame is resent

Send Window for Selective Repeat ARQ



Similar to Figure 23.32

Receive Window for Selective Repeat ARQ



Similar to Figure 23.33

Sender and Receiver Window Size

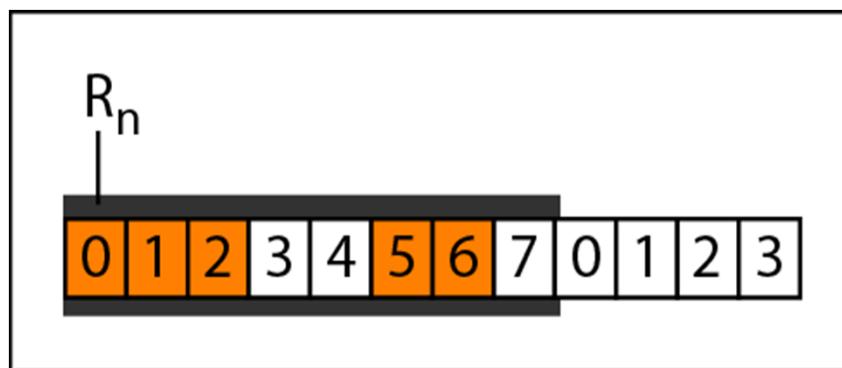
In Selective Repeat ARQ, the sender and receiver window have the same size and it must be

at most one-half of 2^m

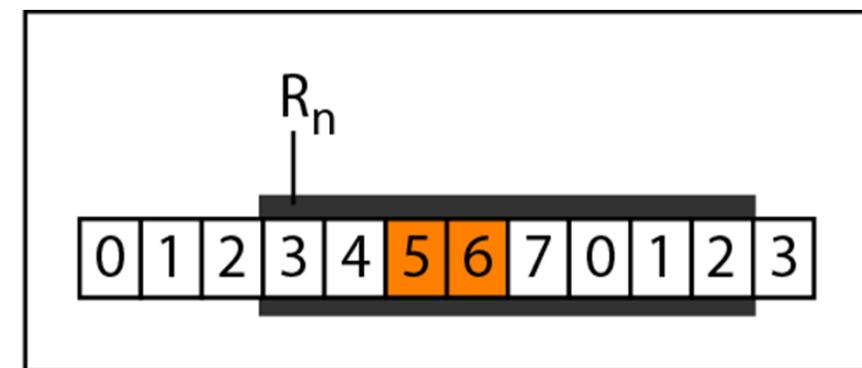
where m is the number of bits of the sequence number
(skip the explanation)

Window Slide and Change of R_n

- ❖ At the **receiver**, if the received frame is not damaged and the sequence no. is within the window,
- ❖ Receiver will store the frame and mark the slot (e.g. slot 5 and 6 on orange.)
- ❖ If contiguous frames, starting from R_n have been marked, data is delivered to the upper layer, and the window *slides*.



a. Before delivery
(Say just received data No. 0)



b. After delivery
(data No. 0, 1 and 2 are delivered)

Figure *Delivery of data in Selective Repeat ARQ*

Example 23.9

Assume a sender sends 6 packets: packets 0, 1, 2, 3, 4, and 5. The sender receives an ACK with ackNo = 3. What is the interpretation if the system is using GBN or SR?

Solution

If the system is using GBN, it means that packets 0, 1, and 2 have been received uncorrupted and the receiver is expecting packet 3.

If the system is using SR, it means that packet 3 has been received uncorrupted; the ACK does not say anything about other packets.

Example 23.10

Similar to Example 23.8 (Figure 23.30) in which packet 1 is lost but using **Selective-Repeat**.

At the sender, packet 0 is transmitted and acknowledged. Packet 1 is lost. Packets 2 and 3 arrive out of order and are acknowledged. When the timer times out, packet 1 (the only unacknowledged packet) is resent and is acknowledged. The send window then slides.

Timer

Theoretically, Select-Repeat uses one timer for each outstanding (unacknowledged) packet. When a timer expires, only the corresponding packet is resent. However, **implementation with a single timer** also works, as shown in this example.

Example 23.10 (continued)

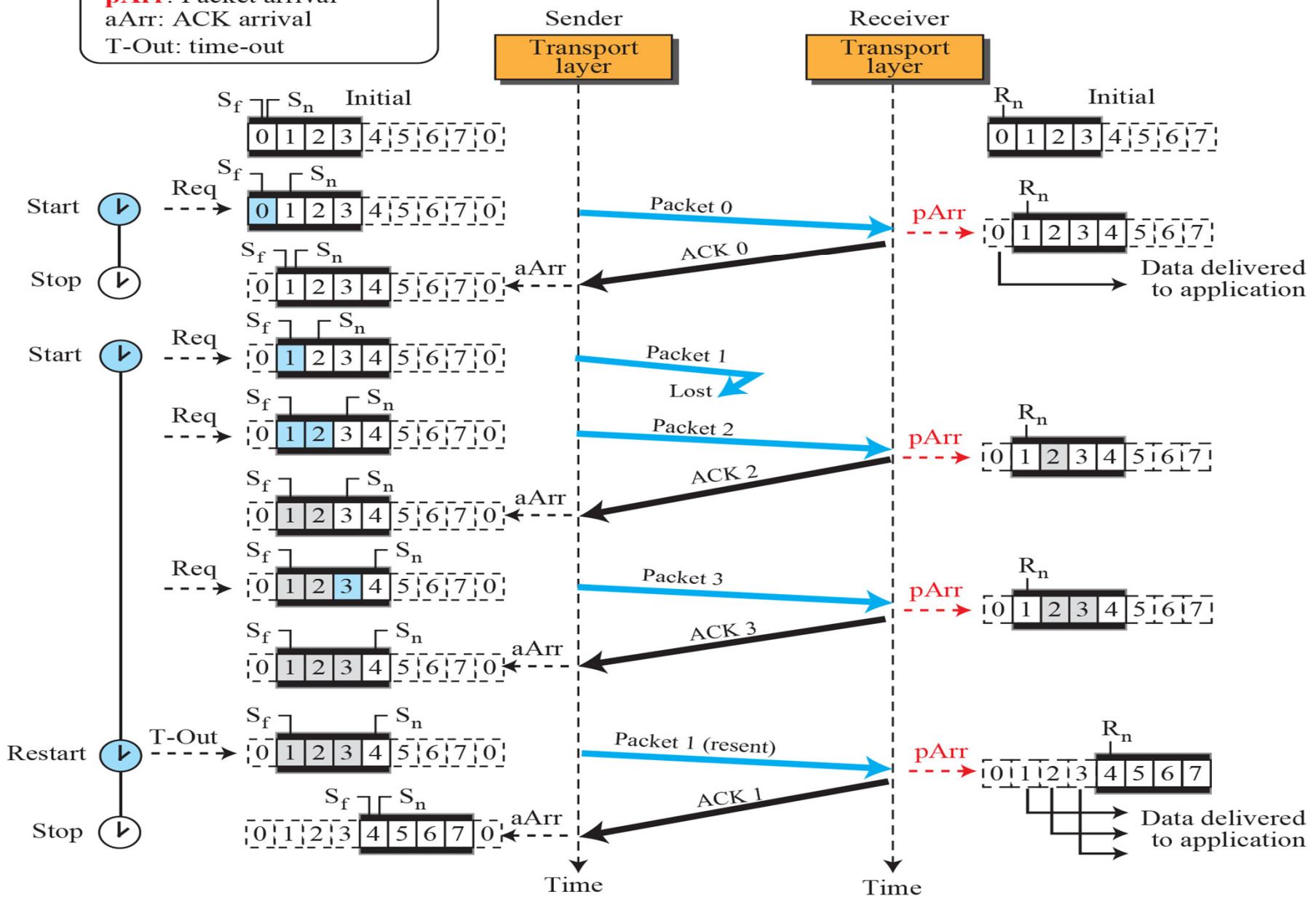
At the receiver site we need to distinguish between the acceptance of a packet and its delivery to the application layer.

- At the second arrival, packet 2 arrives and is stored and marked (shaded slot), but it cannot be delivered because packet 1 is missing.
- At the next arrival, packet 3 arrives and is marked and stored, but still none of the packets can be delivered.
- Only at the last arrival, when finally a copy of packet 1 arrives, can packets 1, 2, and 3 be delivered to the application layer.
- There are **two conditions for the delivery of packets to the application layer:**
 - ❖ *A set of consecutive packets must have arrived.*
 - ❖ *The set starts from the beginning of the window.*

Figure 23.35: Flow diagram for Example 3.10

Events:

Req: Request from process
pArr: Packet arrival
 aArr: ACK arrival
 T-Out: time-out



Selective Repeat Disadvantages

- ❖ Order of receiving data frames is not maintained
- ❖ Re-sequencing is required in Receiver
- ❖ Number of buffers can be large and non-deterministic
- ❖ (***Advantage***: but the channel is more efficient than Go-Back-N)

Summary on Sliding Window ARQ

- ❖ The link utilization is much improved at the expense of **larger buffer** storage requirements
- ❖ Sender sends data frame continuously without waiting for an ACK
- ❖ Sender retains a copy of each transmitted data frame in a ***retransmission list***
- ❖ Receiver returns an ACK for each correctly received data frame

Sliding Window ARQ (cont.)

- ❖ Each data frame contains a **unique identifier** (the sequence number)
- ❖ On receipt of an ACK the corresponding data frame is removed from the retransmission list by Sender
- ❖ Receiver retains a ***link receive list*** containing the correctly received data frames (but may be out-of-order for selective-repeat)
- ❖ **Retransmission strategies** when an error occurs
 - ☞ Go-Back-N
 - ☞ Selective Repeat

Summary

❖ **Stop-and-Wait**

- ❖ Simplest but least efficient
- ❖ Minimum buffer storage (only 1 in sender & receiver)

❖ **Go-Back-N**

- ❖ Maintain correct sequence
- ❖ Less demand on the buffer storage (1 in receiver)
- ❖ But need to retransmit some already correctly received frames
- ❖ Channel less efficient than selective-repeat

❖ **Selective Repeat**

- ❖ Re-sequencing (and more buffers) required in receiver
- ❖ Number of buffers can be large and non-deterministic
- ❖ Channel more efficient than Go-Back-N

References

- ❖ Go-Back-N Video

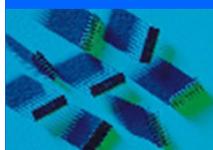
- ❖ <http://www.youtube.com/watch?v=yT8SkFyRRrl>

- ❖ Simulation on Go-Back-N and Selective Repeat

- ❖ http://www.ccs-labs.org/teaching/rn/animations/gbn_sr/

- ❖ Revision Quiz

- ❖ http://highered.mheducation.com/sites/0073376221/student_view0/chapter23/quizzes.html



Lecture 10 Introduction to Network Security

Textbook: Ch. 31

Main Topics

- A. **Security Goal (31.1)**
- B. **Cryptography (31.2)**
 - ❖ **Symmetric-Key Cryptography (31.2.1)**
 - ❖ **Monoalphabetic Substitution**
 - ❖ **Polyalphabetic Substitution**
 - ❖ **Transpositional Encryption**
 - ❖ **Asymmetric-key cryptography (31.2.2)**
 - ❖ **Requirements for Public Key**
 - ❖ **RSA**
- C. **Security Aspects (31.3)**
 - ❖ **Message Integrity (31.3.1)**
 - ❖ **Message Authentication (31.3.2)**
 - ❖ **Digital Signature (31.3.3)**

A. Security Goals

- ❖ Information needs to be secured from attacks.
- ❖ To be secured, information needs to be
 - ❖ hidden from unauthorized access (**confidentiality**),
 - ❖ protected from unauthorized change (**integrity**),
 - ❖ available to an authorized entity when it is needed (**availability**).

Attacks

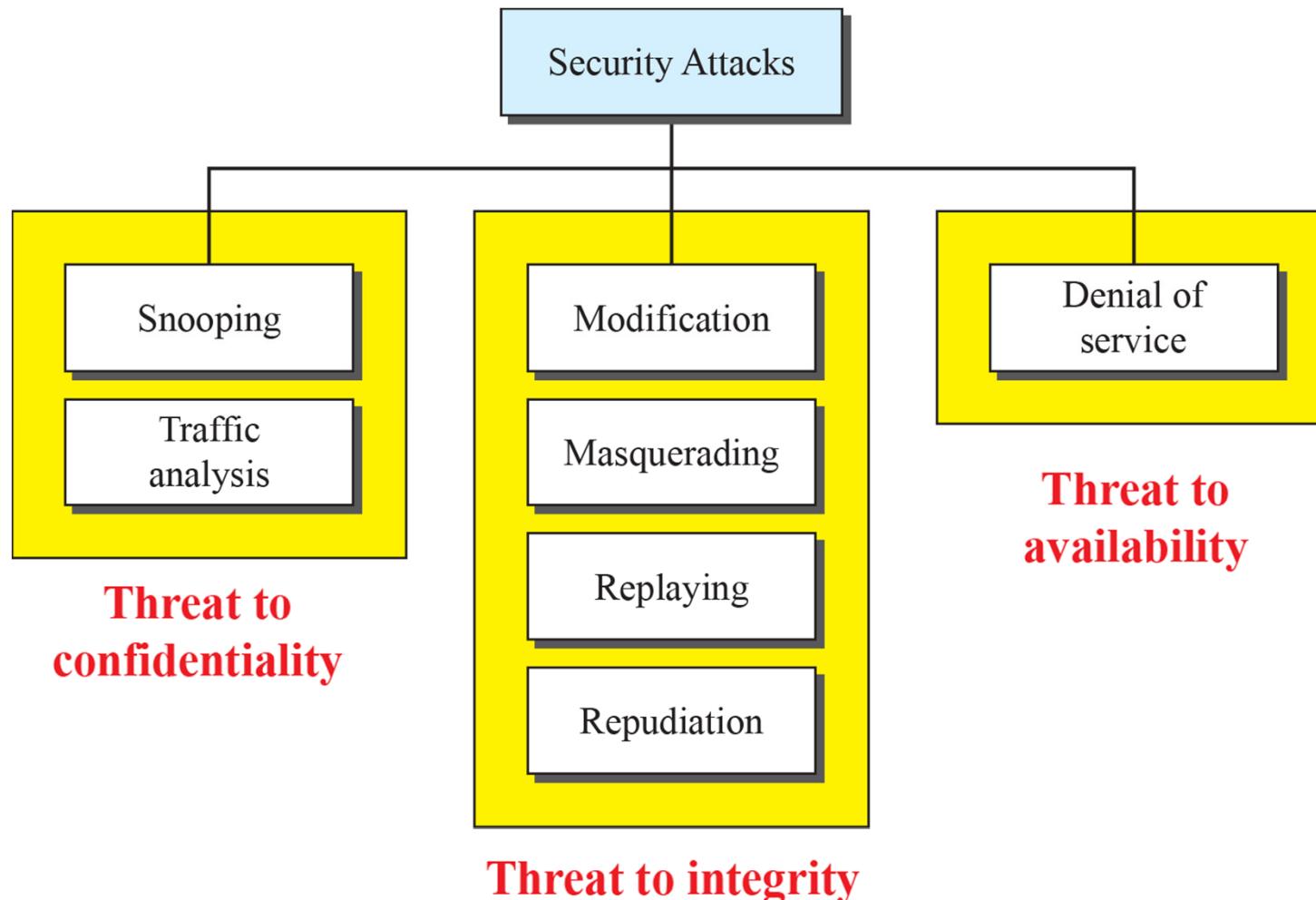
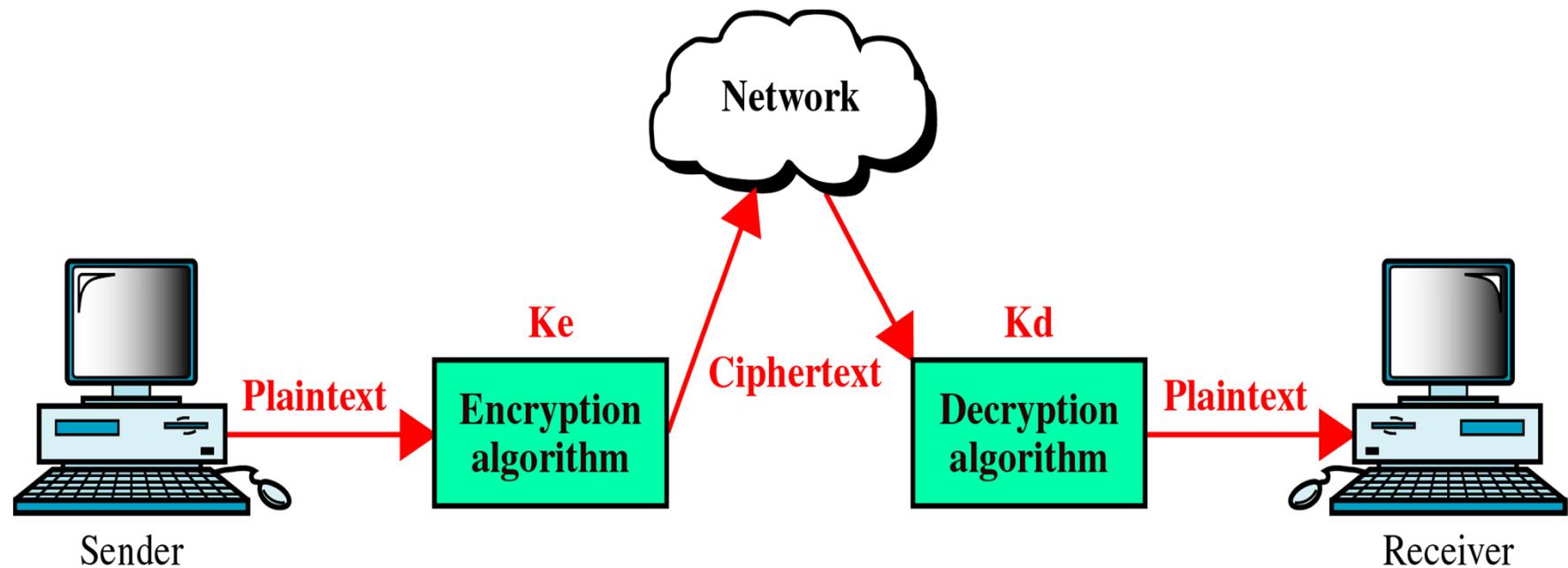


Figure 31.1: Taxonomy of attacks with relation to security goals

B. Cryptography

- ❖ Network security is mostly achieved through the use of cryptography.
 - ❖ Cryptography is the science of transforming messages to make them secure and immune to attack.
- ❖ Aim
 - ❖ Confidentiality
 - ❖ Integrity
 - ❖ Authentication

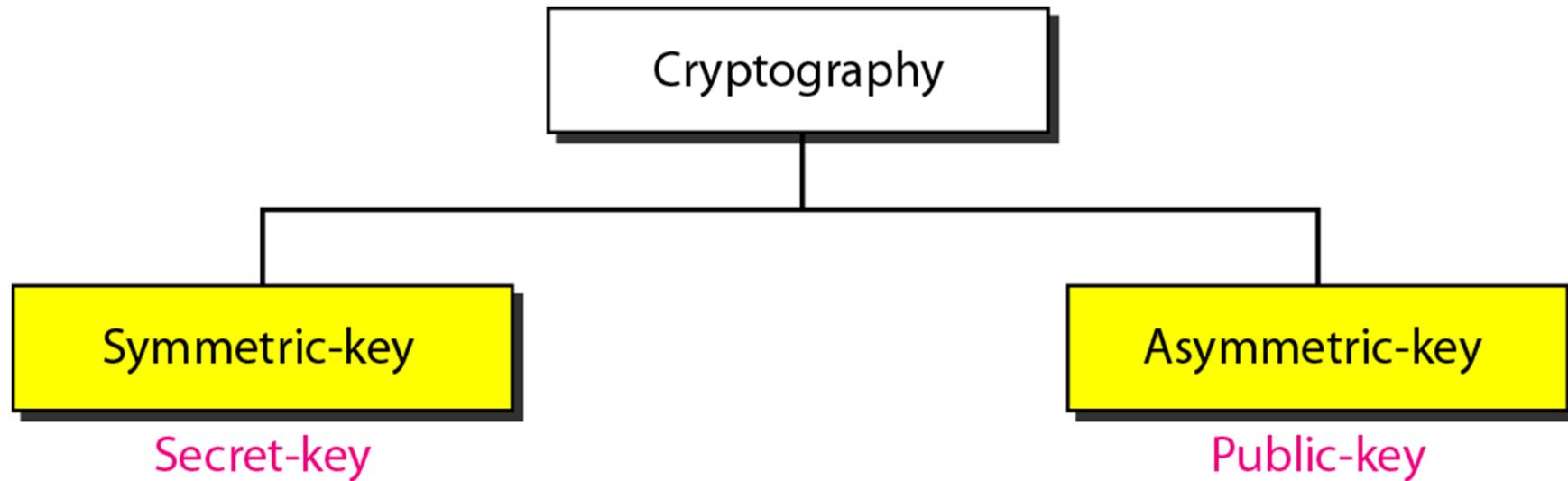
Concept of Encryption and Decryption



Ke is the encryption key

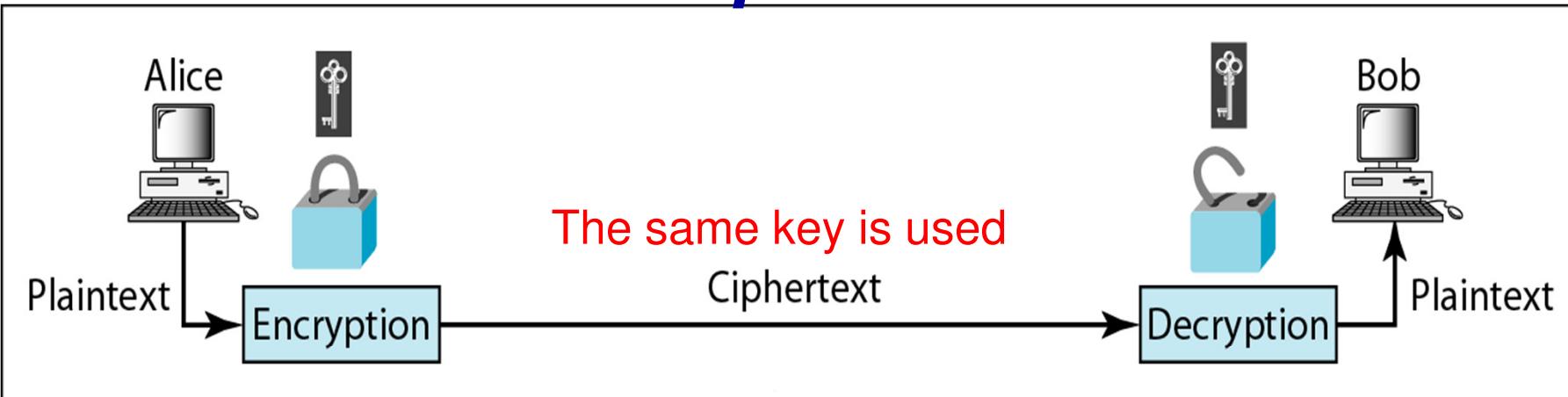
Kd is the decryption key

Encryption/Decryption Methods

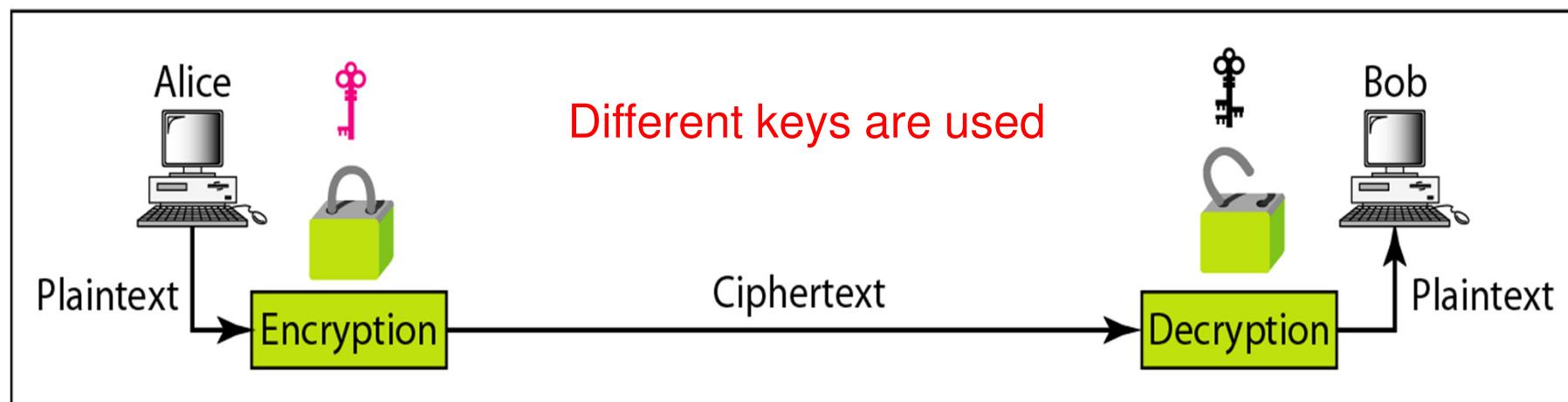


- In **traditional encryption (symmetric)**, the encrypting algorithm is known to everyone but the key is secret except to the sender and receiver
- In **public key encryption (asymmetric)**, both the encrypting algorithm and the encryption key are known to everyone but the decryption key is known only to the receiver

Comparison

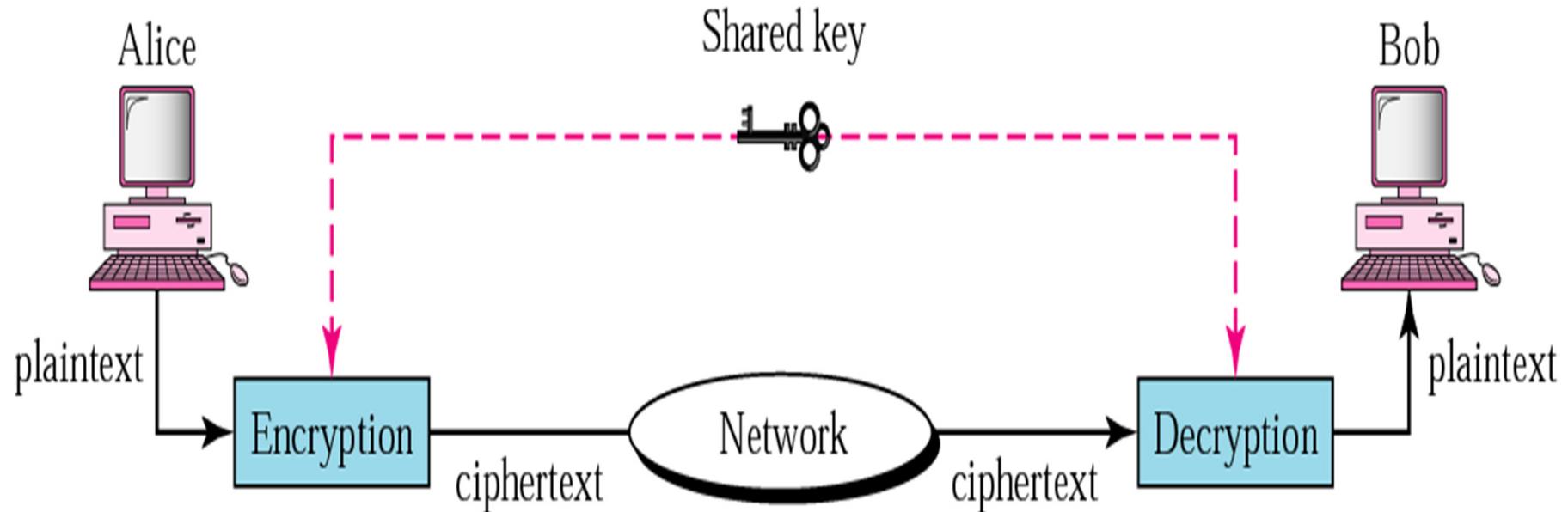


a. Symmetric-key cryptography



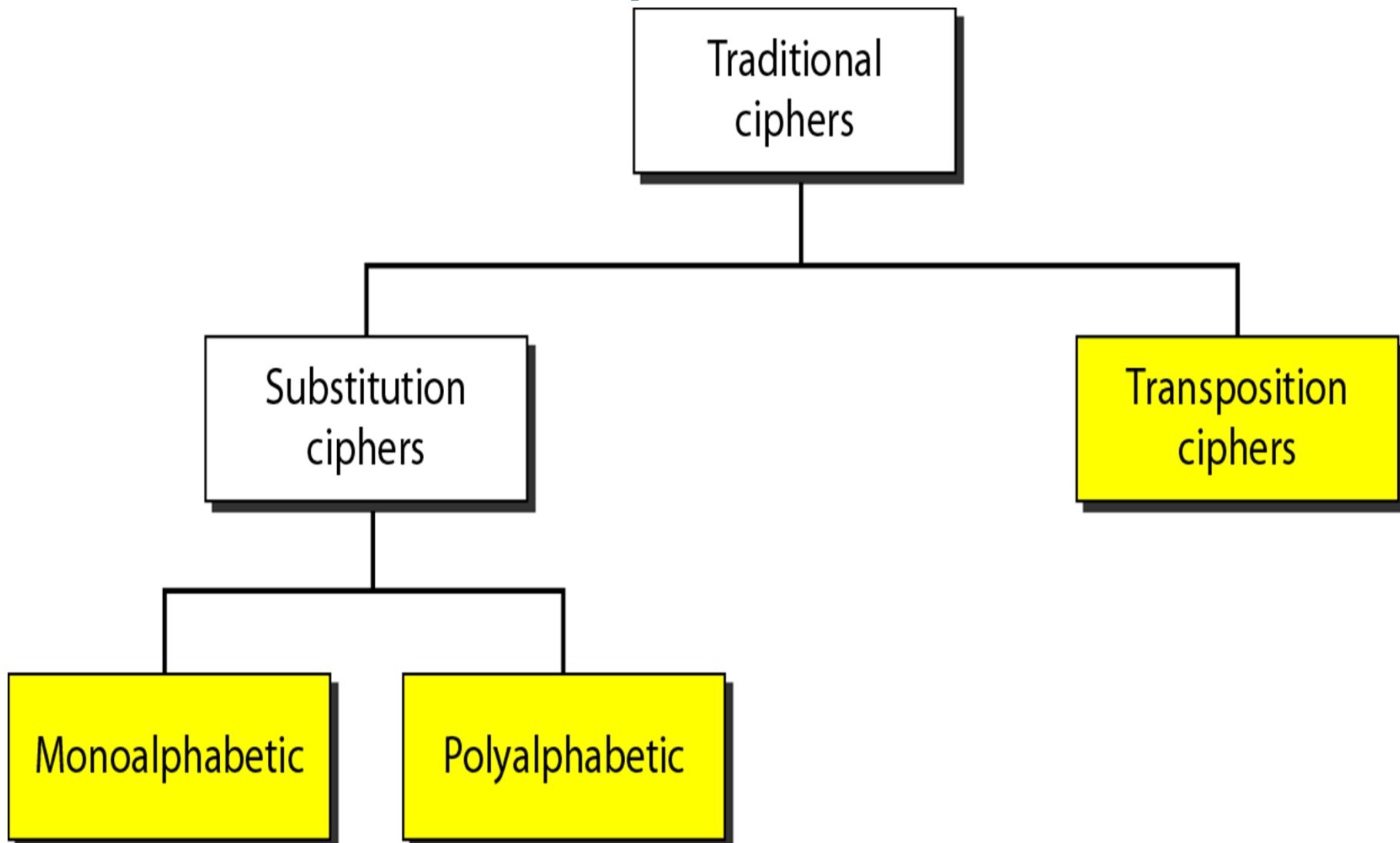
b. Asymmetric-key cryptography

I. Symmetric-Key Cryptography

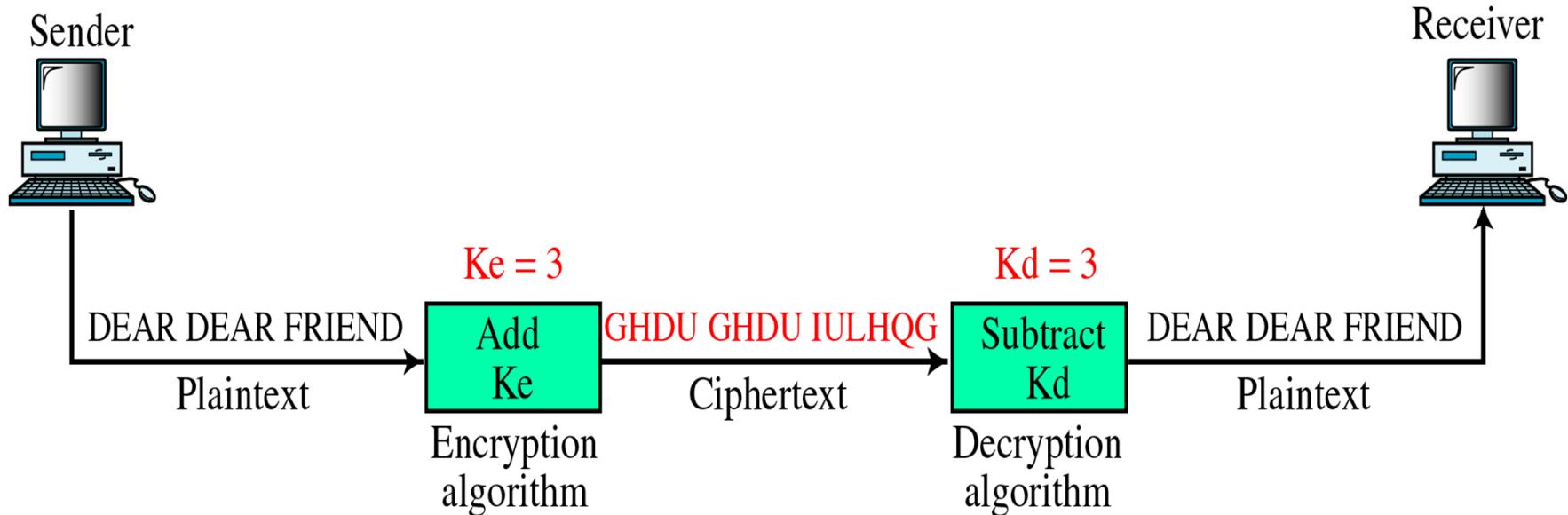


- The **same key** (called shared key) is used by the sender (for encryption) and the receiver (for decryption)
- e.g. the methods in the following slides
- **Each pair of users must have a unique symmetric key**

Traditional Ciphers (Symmetric-Key)



1. Monoalphabetic Substitution



- *Map* every alphabet to another (unique) alphabet. OR
- *Shift* the plaintext alphabet by n places (n is the key)
- In monoalphabetic substitution, the relationship between a character in the plaintext to the character in the ciphertext is always one-to-one.

Example of monoalphabetic substitution

Encryption algorithm

Substitute top row character
with bottom row character

Decryption algorithm

Substitute bottom row character
with top row character

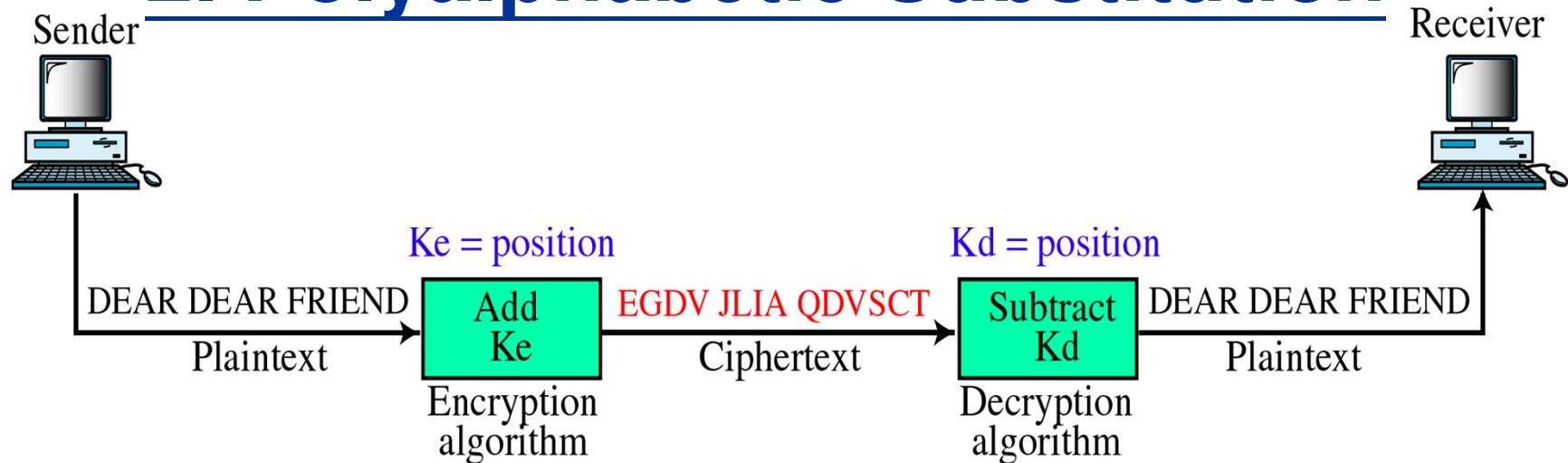
| | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| K | C | P | S | V | M | H | F | D | B | U | W | Q | N | R | Y | T | J | O | I | X | E | L | A | Z | G |

Key

Problem?

- can be attacked easily
- cannot hide natural frequencies of characters

2. Polyalphabetic Substitution



- ❖ Use different monoalphabetic substitutions as one proceeds through the plaintext message.
- ❖ e.g. use the position of the character in the text as the key (of substitution).
- ❖ e.g. define a table which maps every plaintext alphabet to a ciphertext alphabet.

Example

Character in plaintext

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | W | R | K | D | O | V | C | A | S | B | Y | Q | M | L | H | I | T | U | F | E | Z | N | G | J | P | X |
| 1 | H | Q | B | G | W | E | R | K | F | C | O | A | Z | J | M | S | L | V | N | I | P | U | D | T | X | Y |
| 2 | P | I | D | Z | X | V | S | T | O | C | M | J | N | L | B | Q | R | U | W | K | H | G | E | F | A | Y |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 25 | M | C | I | D | A | X | V | S | T | O | N | L | K | U | R | E | W | Z | H | F | P | G | Y | J | B | Q |

Key = (Position of character in the text) mod 26

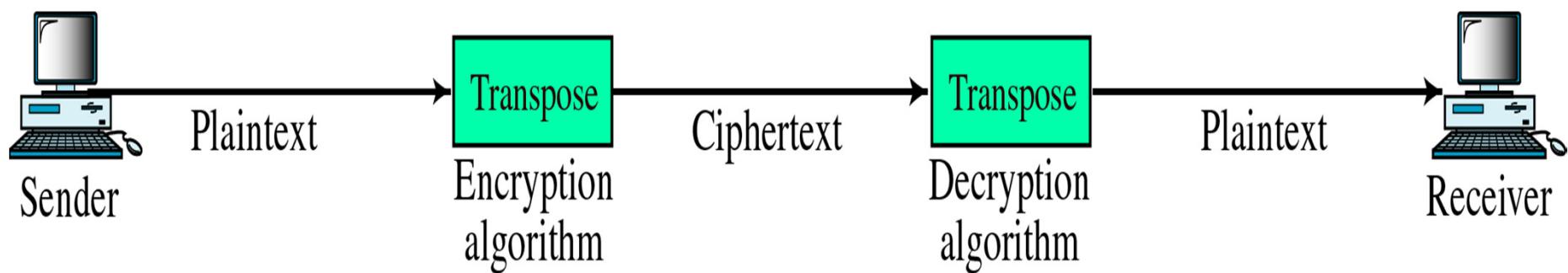
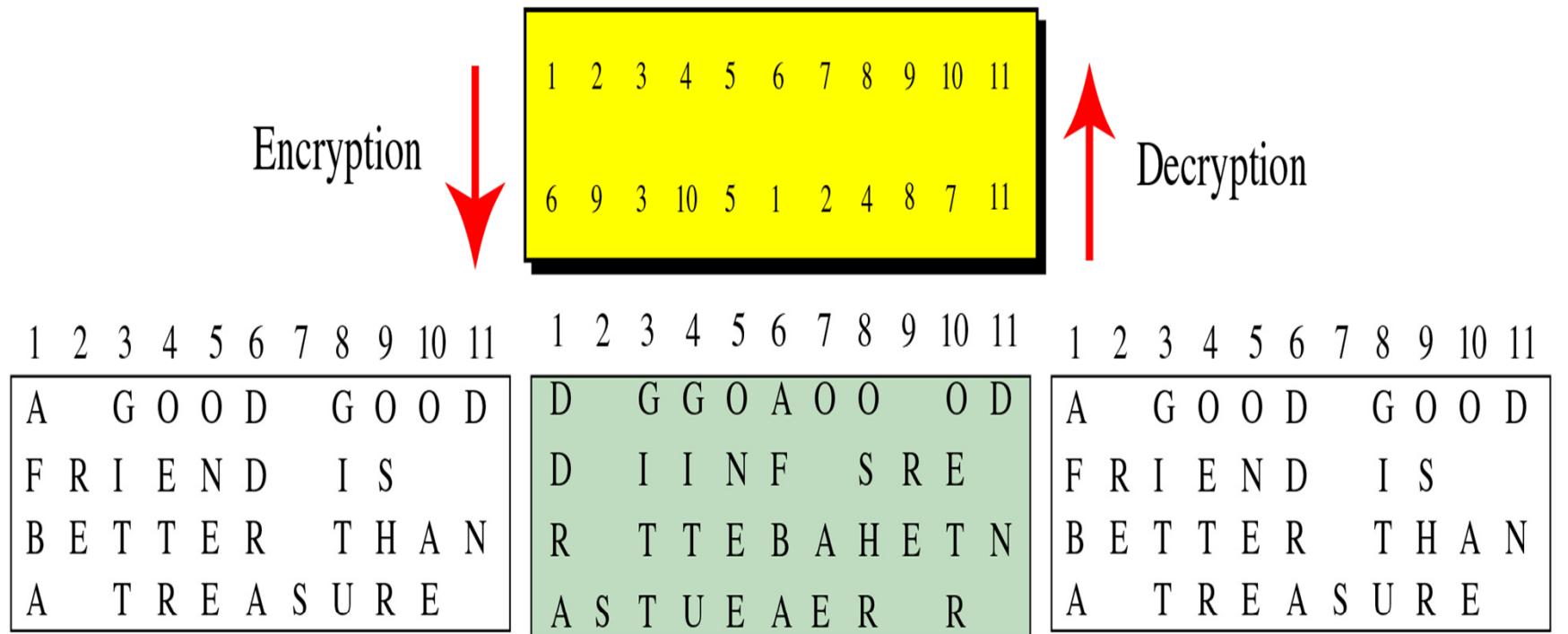
- ❖ According to this table, A is encrypted as W if it is in position 0 and as M if it is in position 25.

3. Transpositional Encryption

- ❖ Re-order the positions of the characters in the plaintext
- ❖ e.g. Organize the plaintext into a table of n columns (n is the key length)
 - ❖ The columns are interchanged according to the key, which is a series of numbers
 - ❖ After exchanging the columns, the “encrypted” data is outputted “row by row”
- ❖ e.g. The key in the following slide is
 - ❖ 6, 9, 3, 10, 5, 1, 2, 4, 8, 7, 11 (and the key length is 11)
- ❖ Means column 1 becomes column 6,
- ❖ column 2 becomes column 9 and so on

Transpositional Encryption

$$K_e = K_d$$



II. Asymmetric-key cryptography

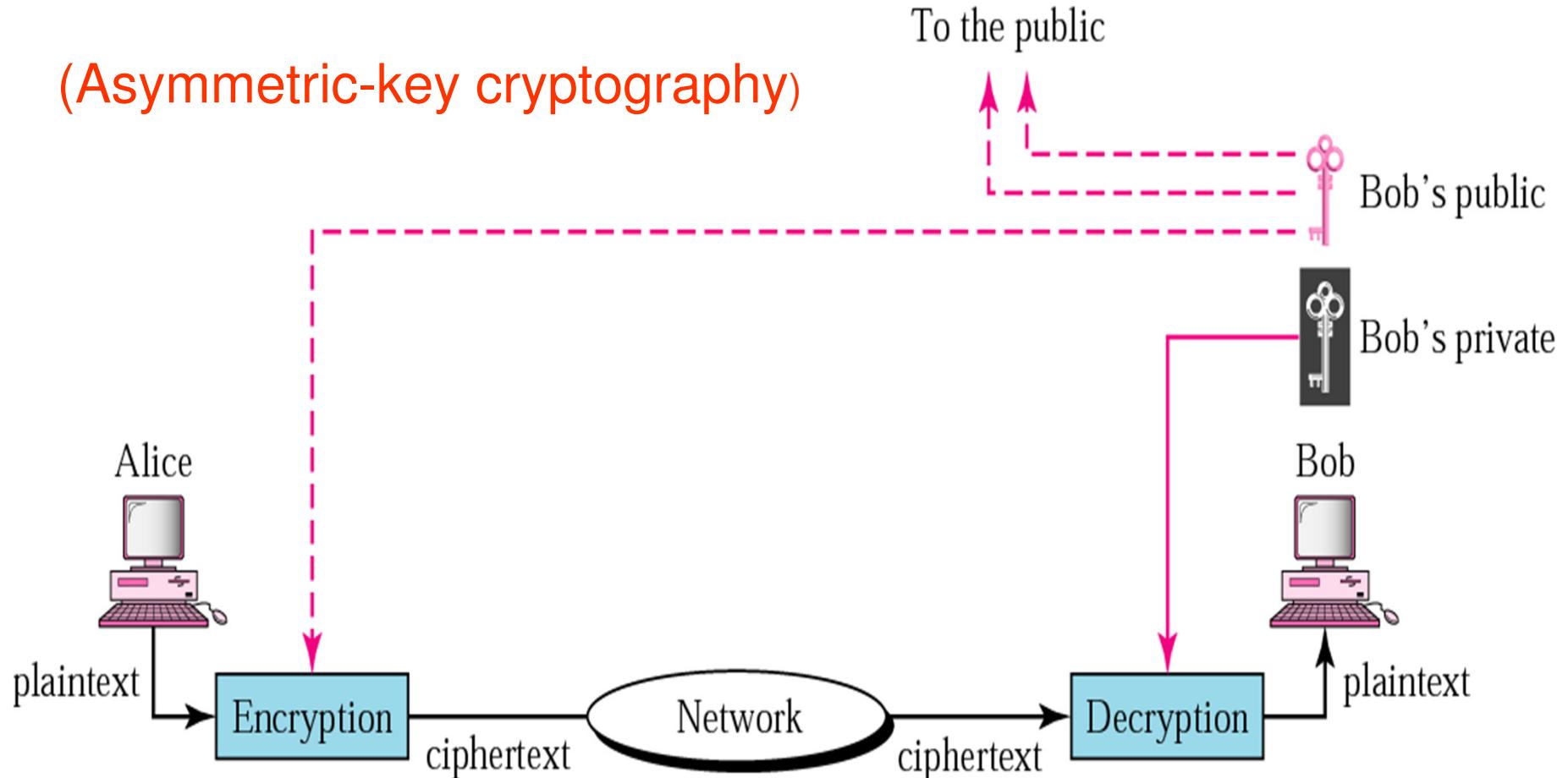
- ❖ It is also called **Public Key Cryptography**
- ❖ Encryption uses the key E called ***public key***, while decryption uses another key D called ***private key***
- ❖ i.e. encryption and decryption use different keys (this is an **asymmetric method**)
- ❖ (Here $E(P)$ represents the ciphertext formed by encrypting the plaintext P using the key E)

1. Requirements for Public Key

- ❖ 1) The encryption key (called public key) is made public, while the decryption key (called private key) is kept by the user securely
 - ❖ 2) $D(E(P)) = P$, i.e. using D to decrypt a ciphertext message which is encrypted by E can get back the original message P
 - ❖ 3) It is very, very difficult to deduce D from E
-
- ❖ e.g. The RSA method
 - ❖ Each user creates a pair of keys (E & D), which can be used to communicate with any other users

Public-key cryptography

(Asymmetric-key cryptography)

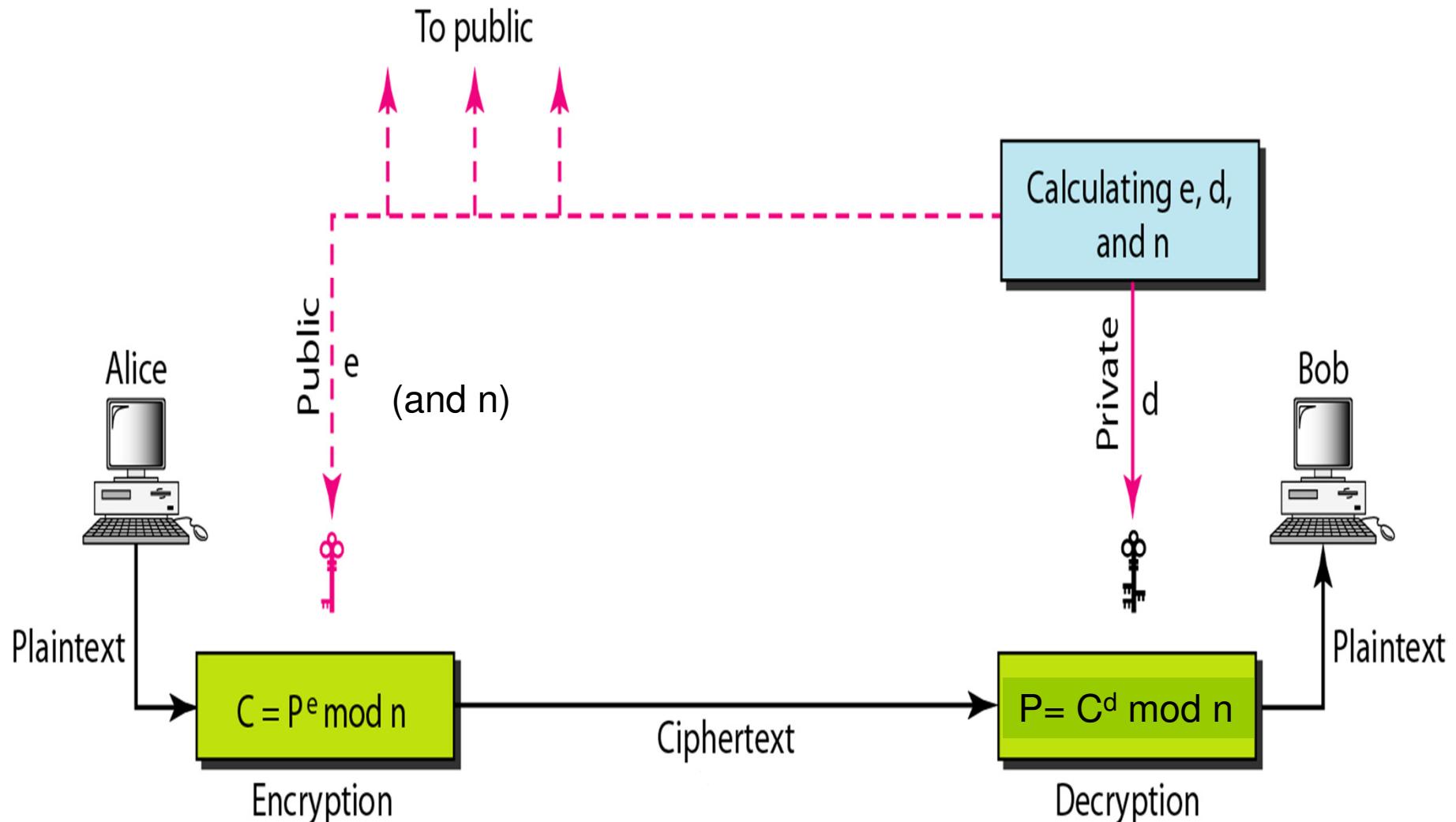


Sender uses the receiver's *public key* to encrypt the message

Receiver uses its own *private key* to decrypt the ciphertext

2. RSA Cryptosystem

RSA is named for its inventors Rivest, Shamir, and Adleman.



Selecting Key for RSA

- ❖ Bob uses the following steps to select the private and public keys:
 1. Chooses two very large prime numbers p and q .
 2. Get n and Φ by $n = p \times q$ and $\Phi = (p-1) \times (q-1)$
 3. Choose a random integer e and calculate d so that $d \times e \text{ mod } \Phi = 1$.
 4. **e and n are announced to the public; d and Φ are kept secret.**

In RSA, e and n are announced to the public; d and Φ are kept secret.

Encryption

$$C = P^e \pmod{n}$$

❖ Example 31.7

Bob chooses 7 and 11 as p and q and calculates $n = 7 \cdot 11 = 77$.

$$37 \times 13 \pmod{60} = 1$$

The value of $\Phi = (7 - 1)(11 - 1) = 60$.

Now he chooses two keys, e and d. If he chooses e to be 13, then d is 37.

Now imagine Alice sends the plaintext 5 to Bob.

She uses the public key 13 to encrypt 5.

Plaintext: 5

$$C = 5^{13} \pmod{77} = 26$$

Ciphertext: 26

Decryption

$$P = C^d \pmod{n}$$

- ❖ **Example 31.7 (continued)**

Bob receives the ciphertext 26 and uses the private key 37 to decipher the ciphertext:

Ciphertext: 26

$$P = 26^{37} \pmod{77} = 5$$

Plaintext: 5

The plaintext 5 sent by Alice is received as plaintext 5 by Bob.

How many keys are needed?

- ❖ N users in a network
 - a) Total number of keys?
 - b) Each user needs to know/store how many keys?
- ❖ **Symmetric-key System**
 - a) $N(N-1)/2$
 - b) $N-1$
 - Why?
- ❖ **Asymmetric-key System**
 - a) $2N$
 - b) $N+1$
 - Why?

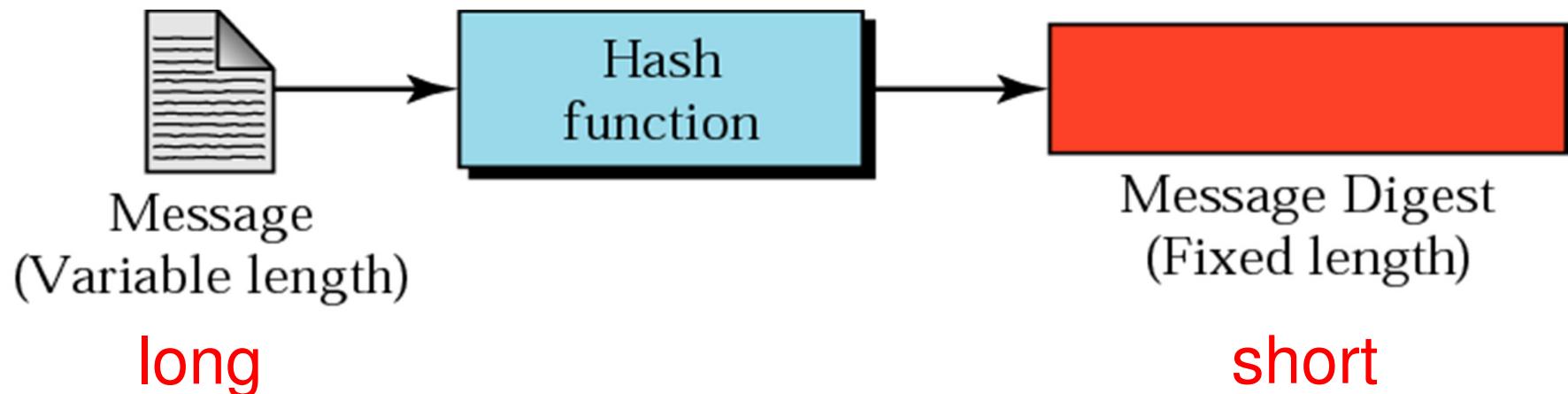
C. Security Aspects

1. Message Integrity

- ❖ There are occasions where we may not even need secrecy but instead must have integrity: the message should **remain unchanged**.
- ❖ For example, Alice may write a will to distribute her estate upon her death. The will does not need to be encrypted. After her death, anyone can examine the will.
- ❖ The integrity of the will, however, needs to be preserved. Alice does not want the contents of the will to be changed.

Message Digest

- ❖ A *miniature version (**digest**)* of the message (like a *fingerprint*)
- ❖ Created by a one-way hash function: the digest can only be created from the message, not vice versa
- ❖ Common hash functions: MD5 and SHA-1



Message and Digest for checking the Integrity

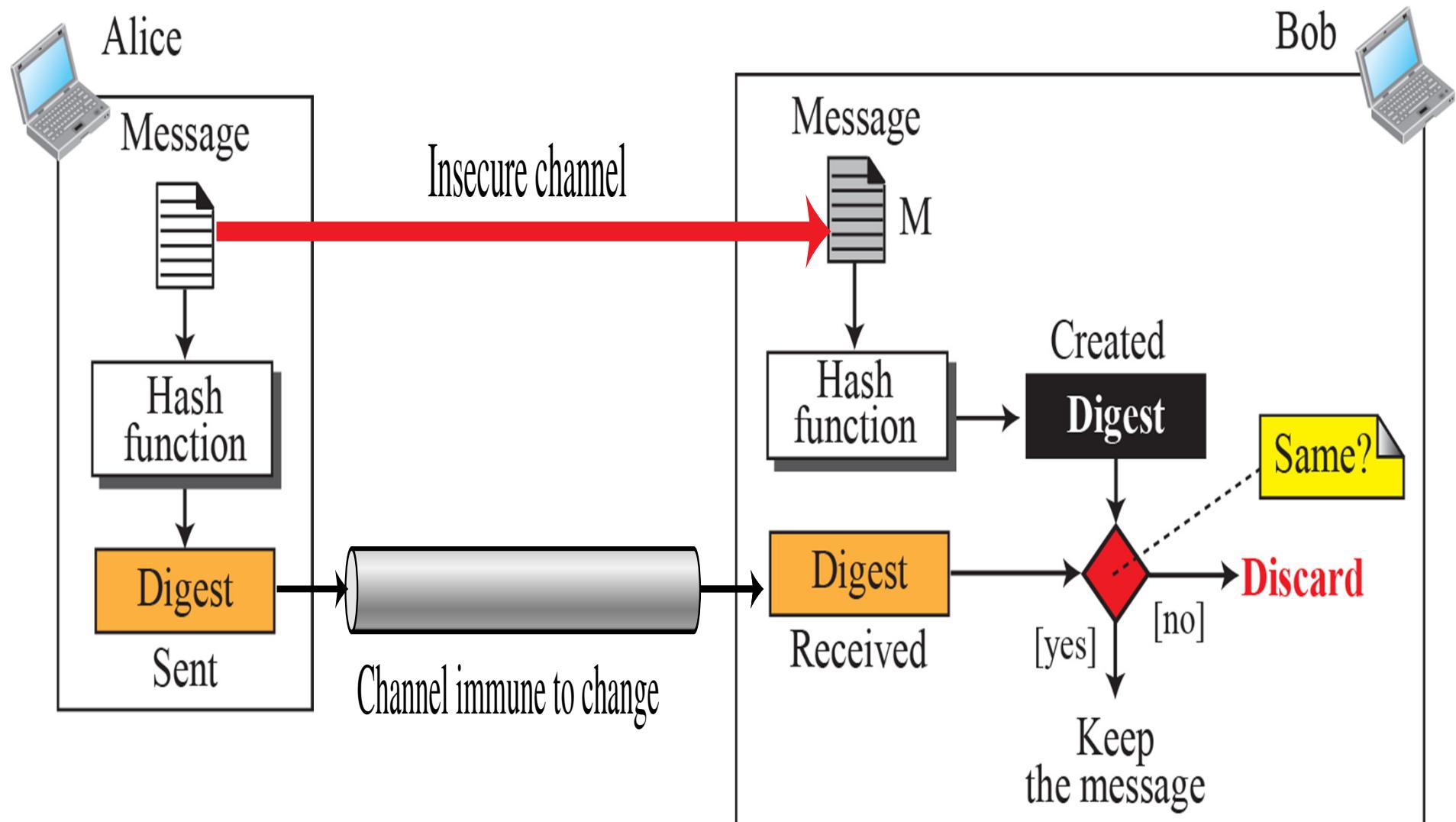
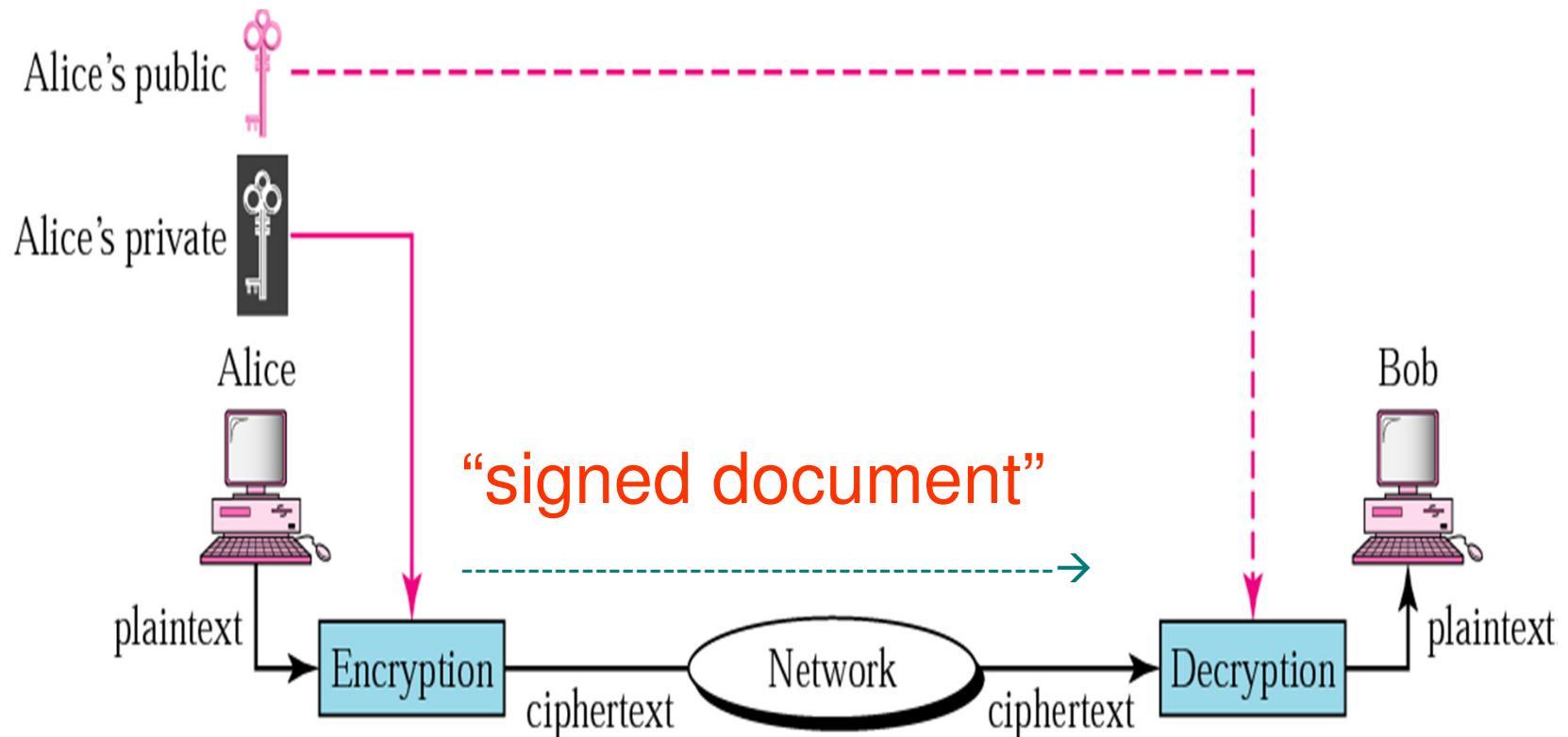


Figure 31.16

2. Message Authentication

- ❖ Means verifying the identity of a sender
- ❖ One method called ***digital signature*** is based on public key cryptography
- ❖ To ***prevent*** a user from ***repudiating*** the message that he has sent
- ❖ Additional Requirement: $E(D(P)) = P$
- ❖ (Both encryption and decryption are just transformation algorithms)

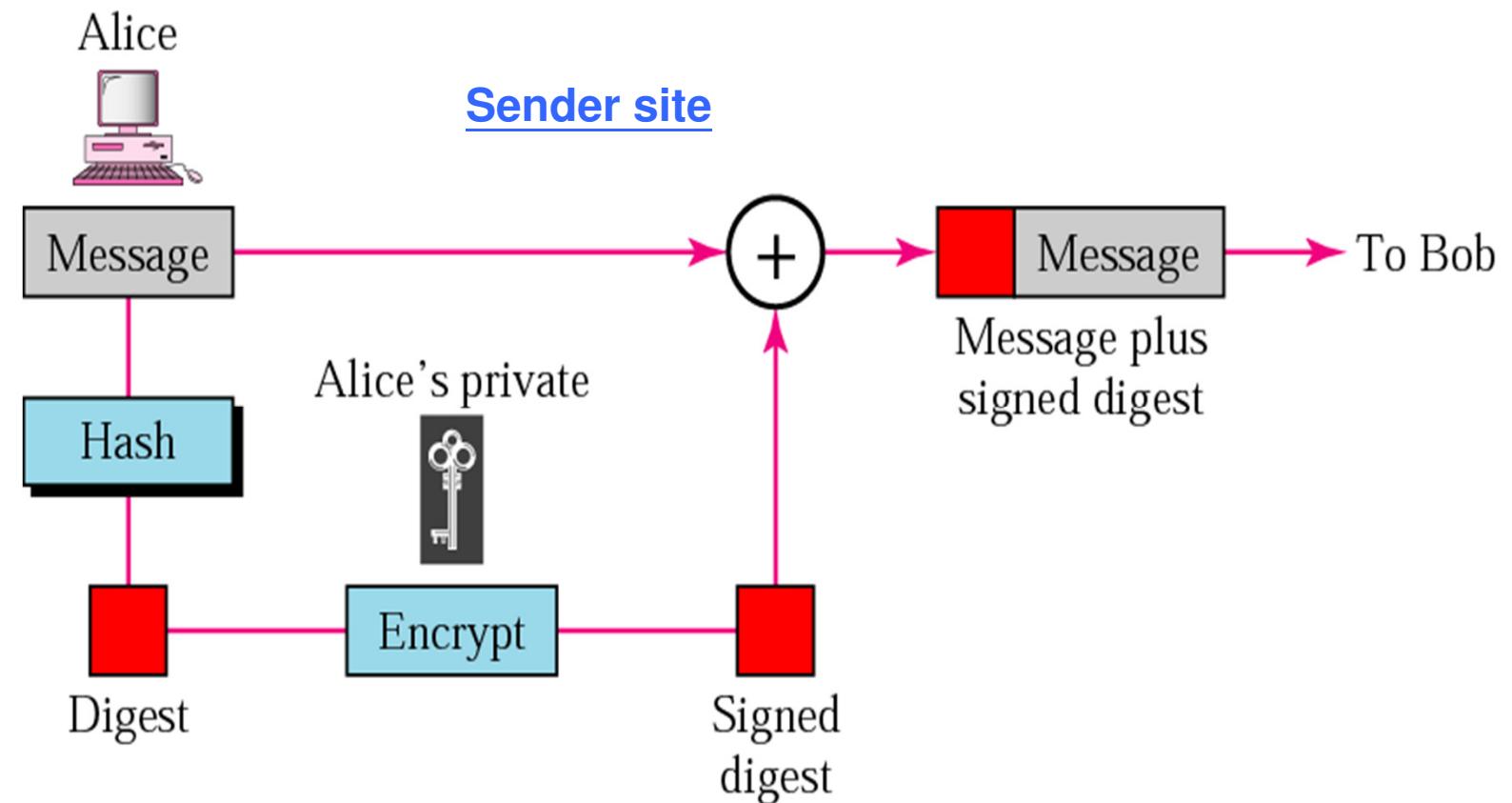
Signing the whole document



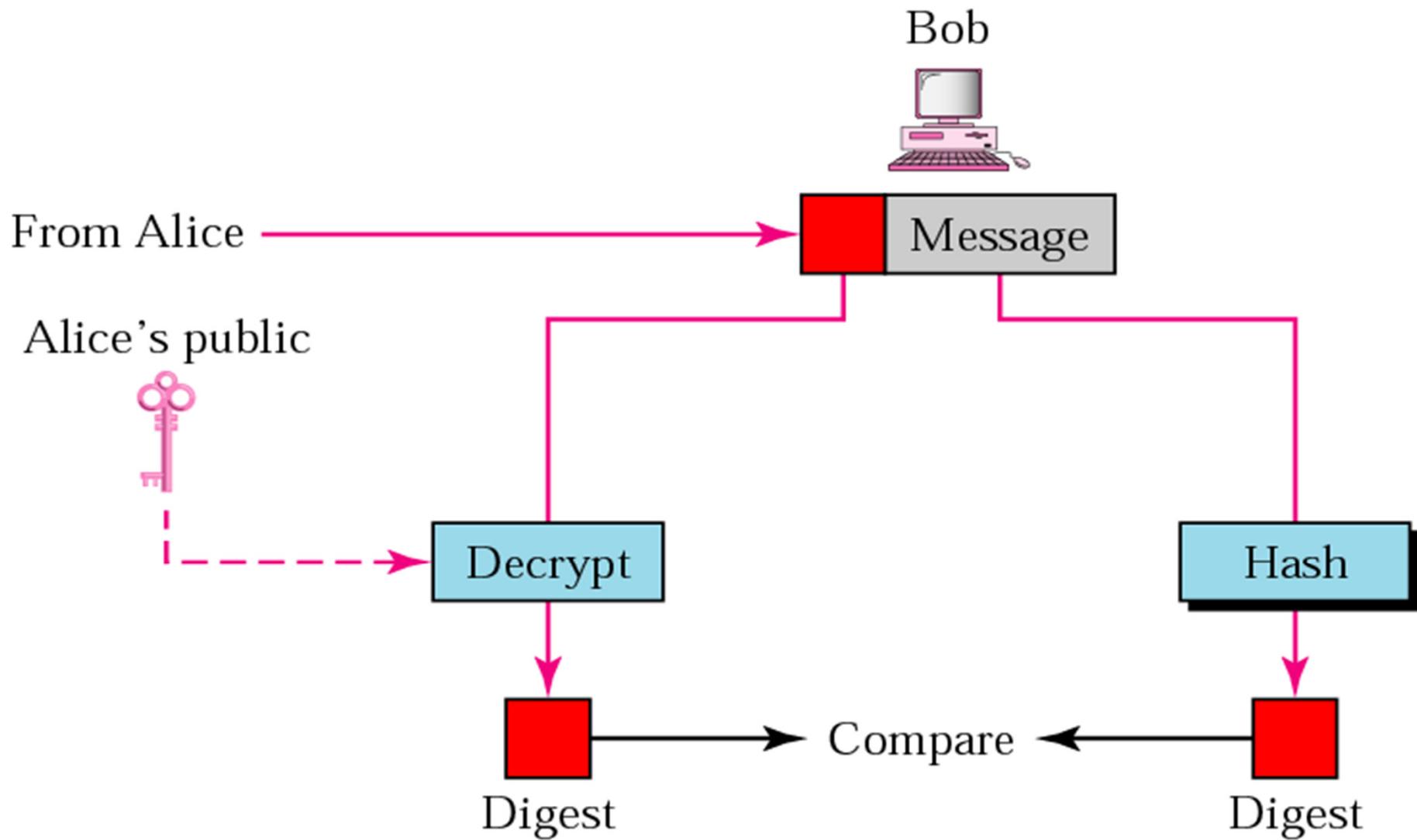
- ❖ Sender uses its own *private key* to **sign (/encrypt)**
- ❖ Receiver uses the sender's *public key* to **verify (/decrypt)**
- ❖ Digital signature does not provide privacy (i.e. secret of the message)

Signing the Digest

❖ Digital Signature - Signing the Digest Only



Receiver site (verify)



3. Digital Signature together with Encryption

- ❖ For user A, denote
 - ❖ E_A = public key
 - ❖ D_A = private key
 - ❖ $E_A(P)$ = encrypt message P using the key E_A
 - ❖ $D_A(P)$ = decrypt message P using the key D_A
- ❖ The encryption and decryption algorithms should have the property that
 - ❖ $D(E(P)) = P$
 - ❖ $E(D(P)) = P$

Digital Signature together with Encryption

- ❖ User A sends a message P to user B by transmitting $E_B(D_A(P))$
- ❖ B decrypts the ciphertext using its own private key:
☞ $D_B(E_B(D_A(P))) = D_A(P)$
- ❖ User B stores $D_A(P)$ in a safe place and then decrypts it (check A's signature) using the public key E_A of user A to get the original message P
- ❖ **Message Nonrepudiation**
- ❖ When A denies having sent the message P to B
 - ☞ User B can show both P and $D_A(P)$ as evidence
 - ☞ (since $D_A(P)$ can only be produced by user A)

Summary

- ❖ Cryptography
 - ❖ Symmetric-Key Cryptography
 - ❖ Asymmetric-key cryptography

- ❖ Security Aspects
 - ❖ Message Integrity
 - ❖ Message Authentication
 - ❖ Digital Signature

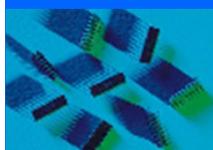
References

- ❖ Video on Distributed Denial of Service (DDOS) Attacks

- ☞ <http://www.youtube.com/watch?v=NogCN78XN2w>
 - ☞ <http://www.youtube.com/watch?v=SCcpauJp63c>

- ❖ Revision Quiz

- ☞ http://highered.mheducation.com/sites/0073376221/student_view0/chapter31/quizzes.html



Lecture 11

Process-to-process Delivery UDP and TCP

Textbook: Ch. 23, 24

Main Topics

A. Transport Layer

- ❑ Transport-Layer Services
- ❑ Port Number and Socket Address

B. Transport Layer Protocol

- ❑ Transport Layer in TCP/IP (24.1)

C. UDP (24.2)

- ❑ User Datagram Format
- ❑ UDP Applications and Examples

D. TCP (24.3)

- ❑ TCP Connection Establishment
- ❑ TCP Data Transfer
- ❑ Sequence Number and Acknowledgement
- ❑ Retransmission and Timeout

A. Transport Layer

- ❖ The transport layer is located between the application layer and the network layer.
- ❖ It provides a process-to-process communication between two application layers, one at the local host and the other at the remote host.
- ❖ Communication is provided using a logical connection.

Idea behind this logical connection.

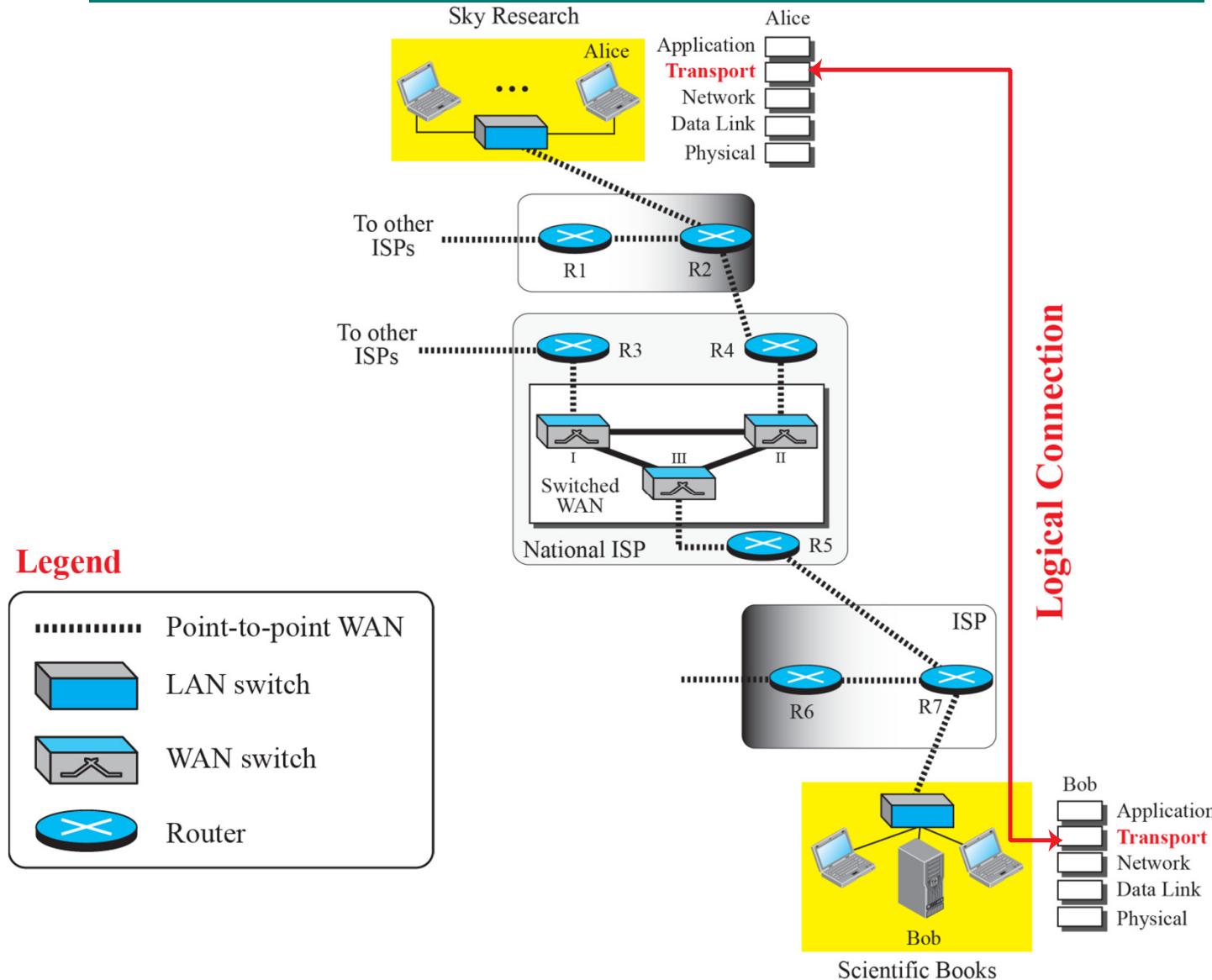
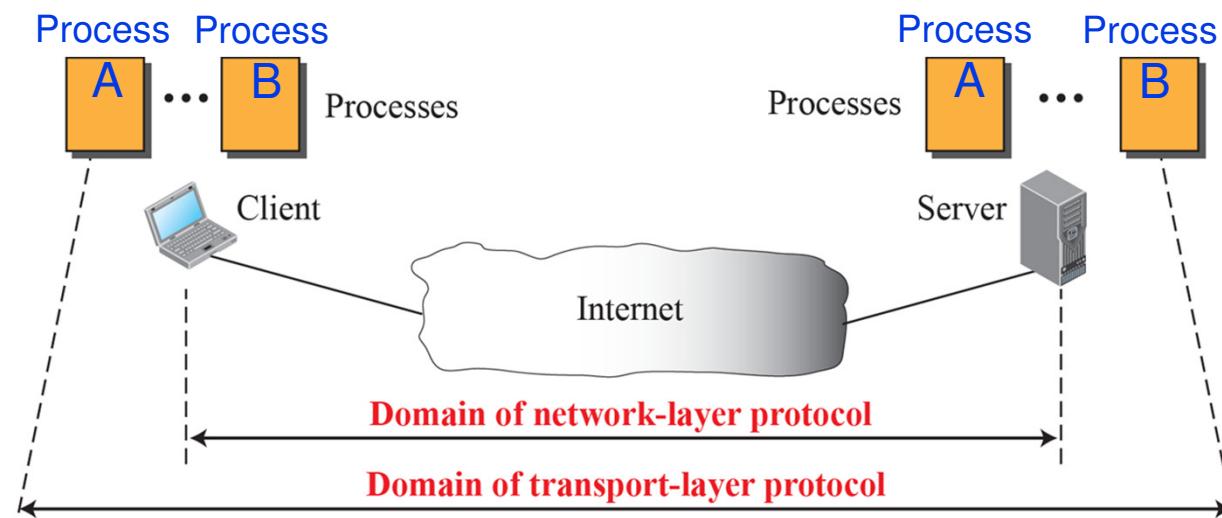


Figure 23.1: Logical connection at the transport layer

Transport-Layer Services

- ❖ The transport layer
 - ❖ is responsible for providing services to the *application layer*; and
 - ❖ receives services from the *network layer*.
- ❖ Process-to-process Communication

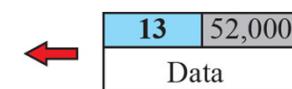


Addressing : Port Numbers

- ❖ A computer can run several server programs and/or several client programs at the same time.
- ❖ To identify a process, a **port number** is used.
 - ❖ E.g. A service called “Daytime” uses 13 as the port number.

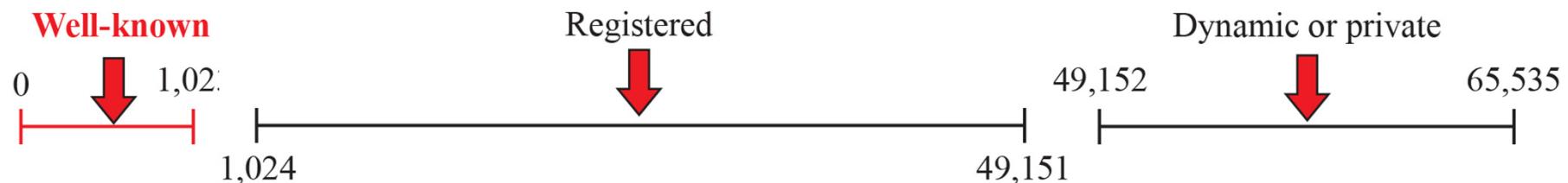


Figure 23.3: Port numbers



Port Number

- ❖ In TCP/IP protocol suite, the port numbers are integers between 0 and 65,535 (16 bits).
 - ❖ *Well-known ports*: 0 to 1023.
 - ❖ *E.g. 23: Telnet remote login, 80: HTTP*
 - ❖ *Registered ports*: 1024 to 49151.
 - ❖ *IANA maintains the official list.*
 - ❖ *Dynamic or private ports*: 49152 to 65535.
 - ❖ *One common use is for temporary ports.*



- ❖ The Full list can be found in Service Name and Transport Protocol Port Number Registry
 - ❖ <http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml>

Socket Address

- ❖ The combination of an IP address and a port number is called a socket address.

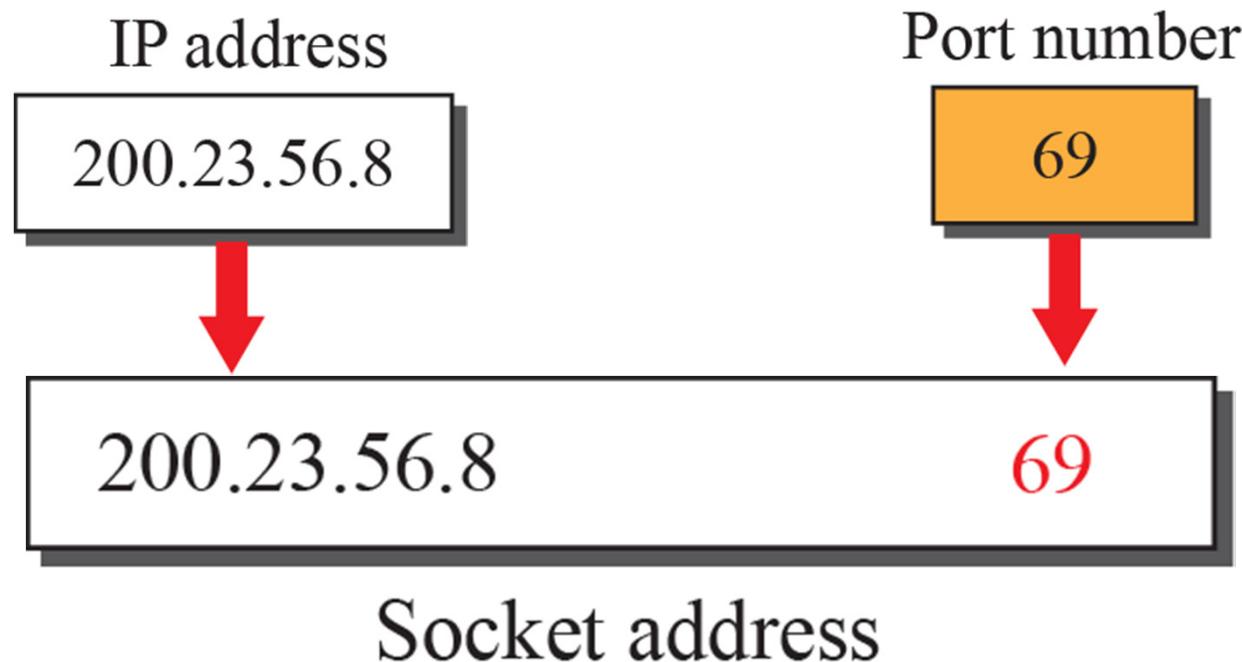


Figure 23.6: Socket address

B. Transport Layer in TCP/IP

- ❖ TCP/IP is a **set of protocols**, or protocol suite, that defines how all transmissions are exchanged across the Internet
- ❖ TCP/IP is a **five-layer** protocol: physical, data link, network, transport and application
- ❖ ***Transport layer***: (2 protocols/services)
 - ❖ Transmission Control Protocol (**TCP**) - data unit is called TCP segment
 - ❖ User Datagram Protocol (**UDP**)
 - ❖ ***Network layer***: Internet Protocol (**IP**)

Transport-layer protocols in the TCP/IP protocol suite

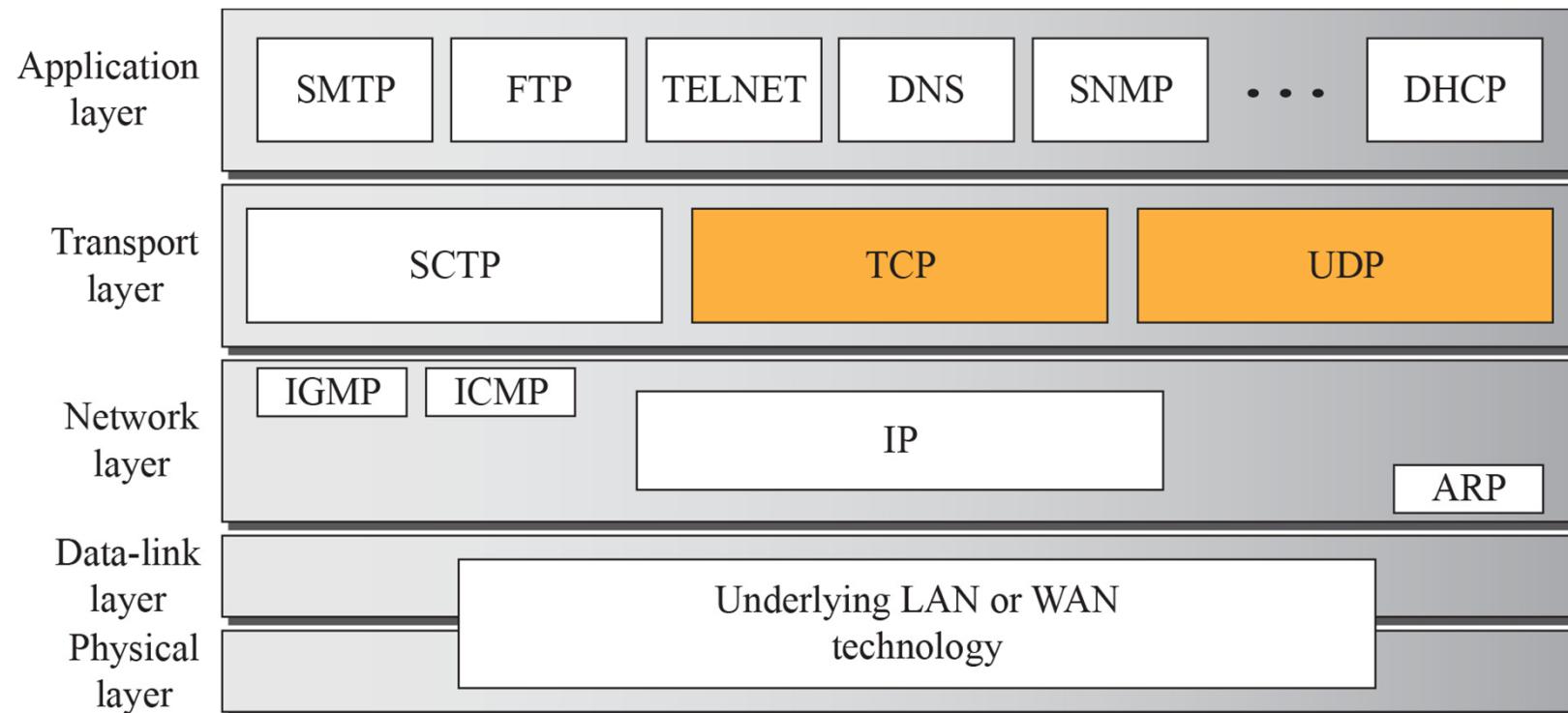


Figure 24.1: Position of transport-layer protocols in the TCP/IP protocol suite

Some well-known UDP and TCP Protocols

| <i>Port</i> | <i>Protocol</i> | <i>UDP</i> | <i>TCP</i> | <i>Description</i> |
|-------------|-----------------|------------|------------|--|
| 7 | Echo | √ | | Echoes back a received datagram |
| 9 | Discard | √ | | Discards any datagram that is received |
| 11 | Users | √ | √ | Active users |
| 13 | Daytime | √ | √ | Returns the date and the time |
| 17 | Quote | √ | √ | Returns a quote of the day |
| 19 | Chargen | √ | √ | Returns a string of characters |
| 20, 21 | FTP | | √ | File Transfer Protocol |
| 23 | TELNET | | √ | Terminal Network |
| 25 | SMTP | | √ | Simple Mail Transfer Protocol |
| 53 | DNS | √ | √ | Domain Name Service |
| 67 | DHCP | √ | √ | Dynamic Host Configuration Protocol |
| 69 | TFTP | √ | | Trivial File Transfer Protocol |
| 80 | HTTP | | √ | Hypertext Transfer Protocol |
| 111 | RPC | √ | √ | Remote Procedure Call |
| 123 | NTP | √ | √ | Network Time Protocol |
| 161, 162 | SNMP | | √ | Simple Network Management Protocol |

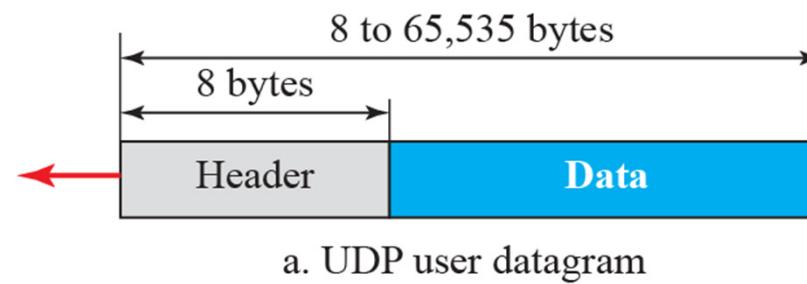
Table 24.1: Some well-known ports used with UDP and TCP

C. User Datagram Protocol (UDP)

- ❖ **The User Datagram Protocol (UDP) is a connectionless, unreliable transport protocol.**
- ❖ UDP packets, called user datagrams, have a fixed-size header of 8 bytes made of four fields, each of 2 bytes (16 bits).
- ❖ **If UDP is so powerless, why would a process want to use it?**
 - ❖ **UDP is a very simple protocol using a minimum of overhead.**

UDP Format

- ❖ The first two fields define the source and destination port numbers.
- ❖ The third field defines the total length of the user datagram, header plus data.
 - ❖ The 16 bits can define a total length of 0 to 65,535 bytes.
 - ❖ However, the total length needs to be less because a UDP user datagram is stored in an IP datagram with the total length of 65,535 bytes.



a. UDP user datagram

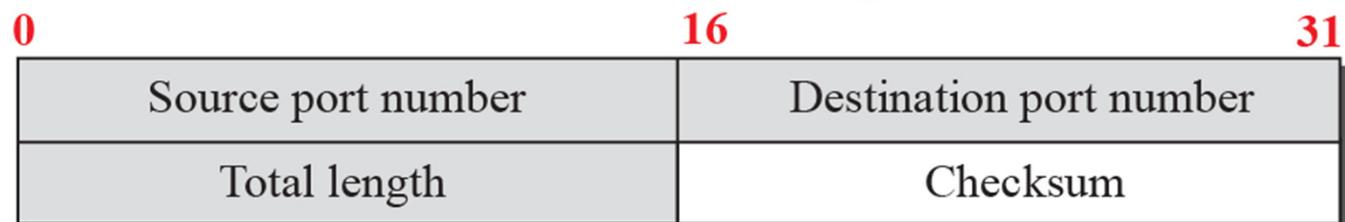


Figure 24.2: User datagram packet format

b. Header format

Example 24.1

The following is the contents of a UDP header in hexadecimal format.

CB84000D001C001C

- a.** What is the source port number?
- b.** What is the destination port number?
- c.** What is the total length of the user datagram?
- d.** What is the length of the data?
- e.** Is the packet directed from a client to a server or vice versa?
- f.** What is the client process?

Example 24.1

Solution

- a. The source port number is the first four hexadecimal digits $(CB84)_{16}$ or 52100
- b. The destination port number is the second four hexadecimal digits $(000D)_{16}$ or 13.
- c. The third four hexadecimal digits $(001C)_{16}$ define the length of the whole UDP packet as 28 bytes.
- d. The length of the data is the length of the whole packet minus the length of the header, or $28 - 8 = 20$ bytes.
- e. Since the destination port number is 13 (well-known port), the packet is from the client to the server.
- f. The client process is the Daytime (see Table 3.1).

Example 24. 3 - DNS

- ❖ **Domain Name Service (DNS)**
 - ❖ A client-server application
 - ❖ uses the services of UDP
 - ❖ Because a client needs to send a short request to a server and to receive a quick response from it.
 - ❖ The request and response can each fit in **one** user datagram.
- ❖ Quick reference for DNS:
 - ❖ <http://www.youtube.com/watch?v=ZBi8GCxk7NQ>
 - ❖ <http://www.youtube.com/watch?v=2ZUxoI7YNgs>

Real-time interactive application

- ❖ Audio and video are divided into frames and sent one after another (Such as Skype).
- ❖ If the transport layer is supposed to resend a corrupted or lost frame,
 - ❖ The synchronizing of the whole transmission may be lost.
 - ❖ The viewer suddenly sees a blank screen and needs to wait until the second transmission arrives.
 - ❖ This is not tolerable.
 - ❖ Each small part of the screen is sent using one single user datagram
 - ❖ The receiving UDP can easily ignore the corrupted or lost packet and deliver the rest to the application program.
 - ❖ That part of the screen is blank for a very short period of time, which most viewers do not even notice.

Very large text file?

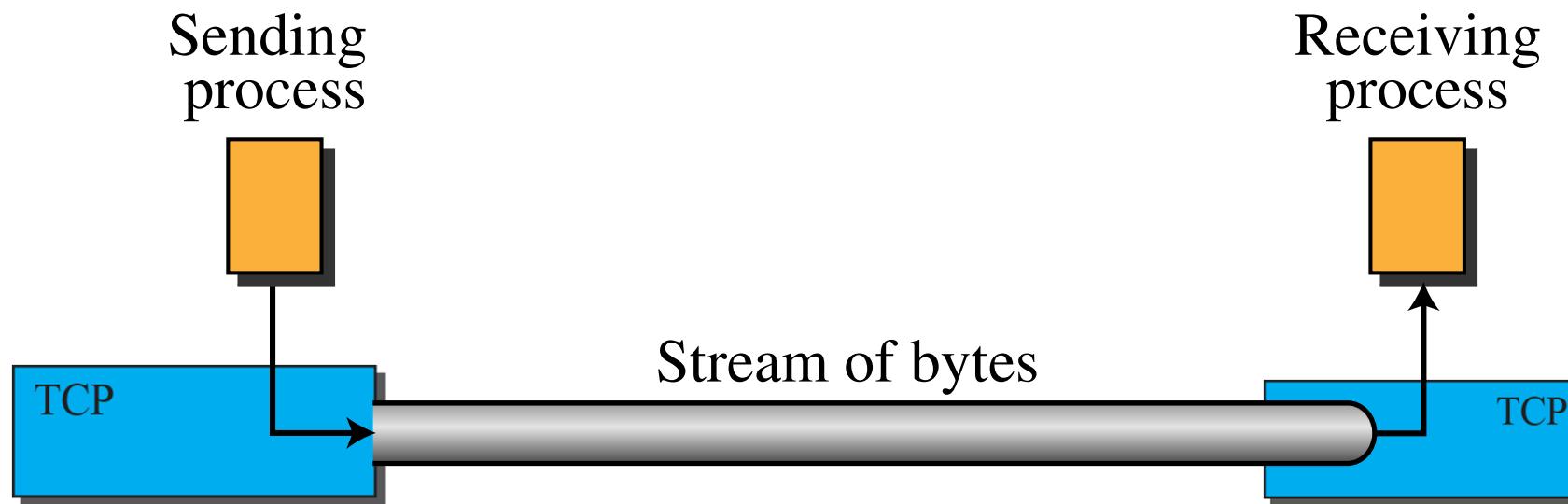
- ❖ How about downloading a very large text file from the Internet?
 - ❖ We definitely need to use a transport layer that provides reliable service.
 - ❖ We don't want part of the file to be missing or corrupted.
 - ❖ The delay created between the deliveries of the parts is not an overriding concern for us; we wait until the whole file is composed before looking at it.
 - ❖ In this case, UDP is not a suitable transport layer.
- ❖ Then, use TCP service.

D. Transmission Control Protocol (TCP)

- ❖ Transmission Control Protocol (TCP) is a connection-oriented, reliable protocol.
- ❖ TCP explicitly defines connection establishment, data transfer, and connection teardown phases to provide a connection-oriented service.
- ❖ TCP uses a combination of Go-back N (GBN) and Selective Repeat (SR) protocols to provide reliability.

TCP Services

- ❖ TCP provides process-to-process communication using port numbers.
- ❖ It is a stream-oriented protocol.
- ❖ TCP creates an environment in which the two processes seem to be connected by an “imaginary tube” that carries their bytes across the network.



TCP Segment

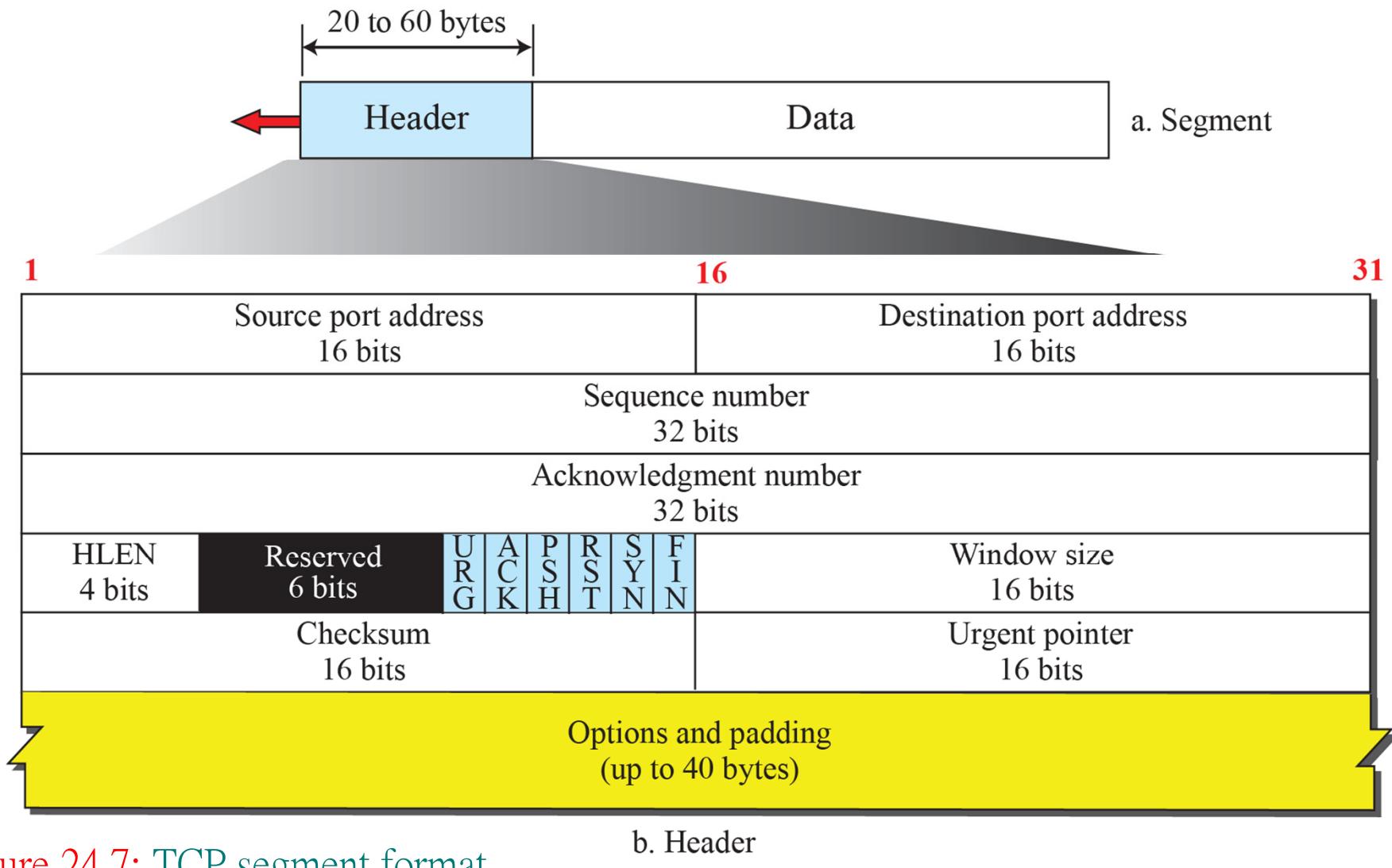


Figure 24.7: TCP segment format

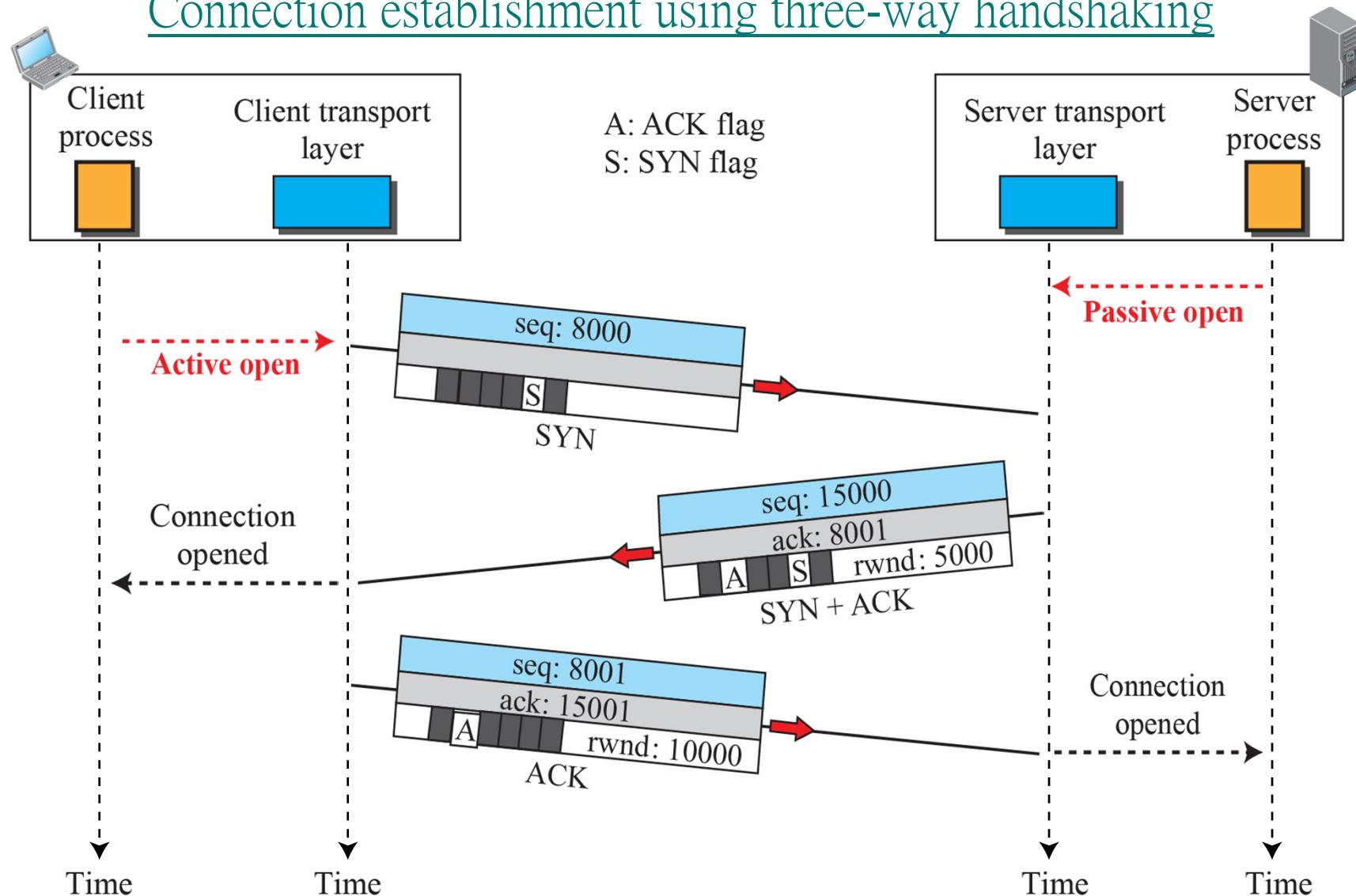
Header Length and Control Field

- ❖ Header Length (HLEN)
 - ❖ Indicates the number of 4-byte words in the TCP header.
 - ❖ Header length is between 20 to 60 bytes.
 - ❖ So, the field is 5 ($5 \times 4 = 20$) to 15.
- ❖ Control Field (6 bits)
 - ❖ ACK – indicate that the value carried in the acknowledgement field is valid
 - ❖ RST, SYN, FIN – for connection setup and teardown (skip details)
 - ❖ PSH – indicate that the receiver should pass the data to the upper layer immediately
 - ❖ URG – indicate that this segment contains urgent data; the location of the last byte is indicated by the *urgent data pointer field*

TCP Connection

- ❖ TCP is connection-oriented.
- ❖ All of the segments belonging to a message are then sent over this logical path.
 - ❖ Using a single logical pathway for the entire message facilitates the acknowledgment process as well as retransmission of damaged or lost frames.
- ❖ How TCP, which uses the services of IP, a connectionless protocol, can be connection-oriented?
 - ❖ The point is that a TCP connection is logical, not physical.
 - ❖ TCP operates at a higher level. TCP uses the services of IP to deliver individual segments to the receiver, but it controls the connection itself.

Connection establishment using three-way handshaking



1.24

Figure 24.10: Connection establishment using three-way handshaking

Connection Establishment

- ❖ The server program tells its TCP that it is ready to accept a connection. The request is called a **passive open**.
- ❖ When a client wants to connect with the server, it issues a request for **active open**, and TCP will start a **Three- Way Handshaking**:
 - ✉ Step One: Client sends a SYN segment (with Clients' sequence number).
 - ✉ Step Two: Server sends a SYN + ACK segment (with servers' sequence number).
 - ✉ Step Three: Client sends an ACK segment.

Data Transfer

- ❖ In the example:
 - ❖ After a connection is established, the client sends 2,000 bytes of data in two segments.
 - ❖ The server then sends 2,000 bytes in one segment.
 - ❖ The client sends one more segment.
 - ❖ The first three segments carry both data and acknowledgment, but the last segment carries only an acknowledgment because there is no more data to be sent.
 - ❖ PSH (push) flag set so that the server TCP knows to deliver data to the server process as soon as they are received.

TCP Data transfer

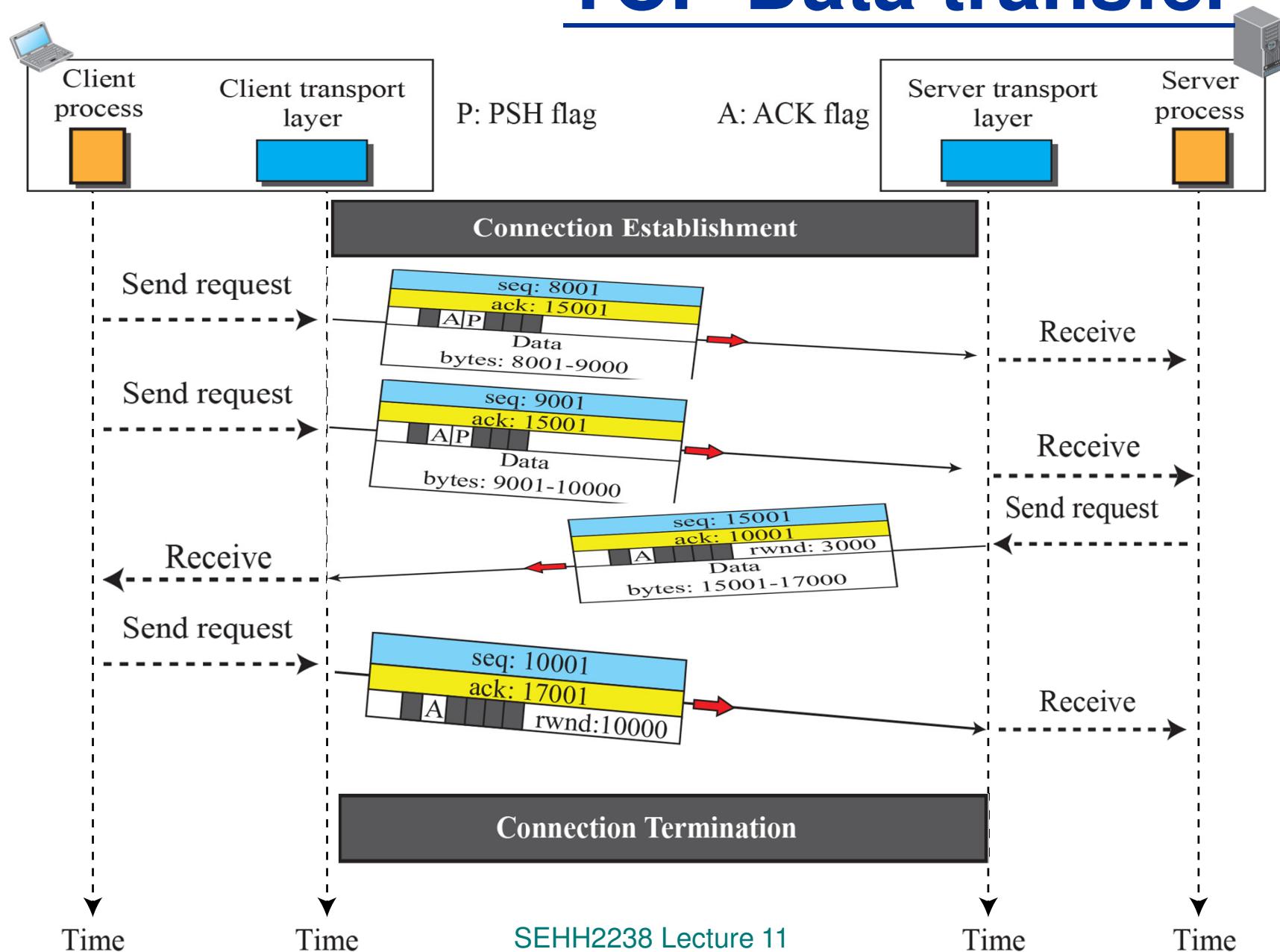


Figure 24.11: Data transfer

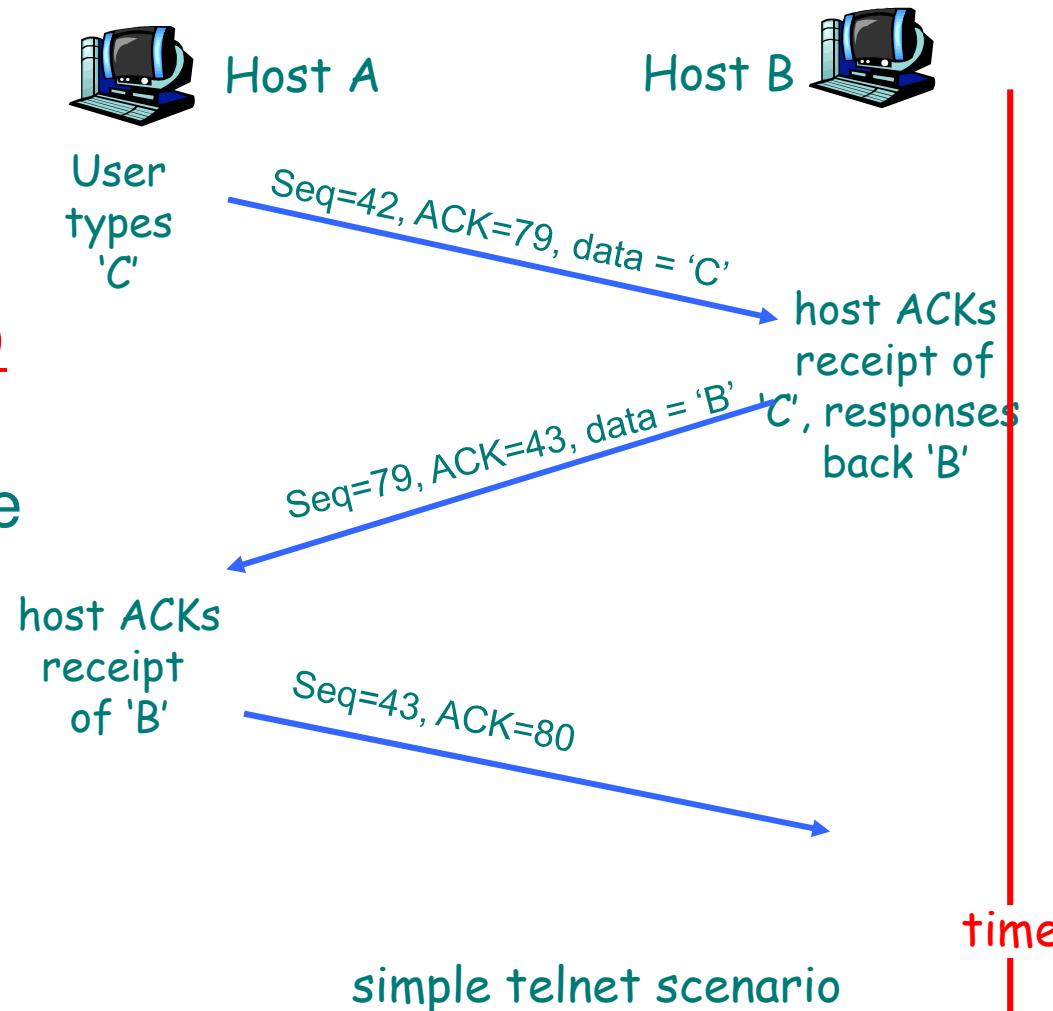
TCP seq. #'s and ACKs

Seq. #'s:

- ❖ byte stream “number” of first byte in segment’s data

ACKs: (TCP is full-duplex)

- ❖ seq # of *next byte* expected from other side
- ❖ *cumulative* ACK



TCP Round Trip Time and Timeout

Q: how to set TCP timeout value?

- ❖ *longer than RTT*
 - ❧ but RTT varies
- ❖ too short: premature timeout
 - ❧ unnecessary retransmissions
- ❖ too long: slow reaction to segment loss

Q: how to estimate RTT?

- ❖ **SampleRTT**: measured time from segment transmission until ACK receipt
 - ❧ ignore retransmissions
- ❖ **SampleRTT** will vary, want estimated RTT “smoother”
 - ❧ average several recent measurements, not just current **SampleRTT**

TCP Round Trip Time and Timeout

EstimatedRTT =

$$(1 - \alpha) * \text{EstimatedRTT} + \alpha * \text{SampleRTT}$$

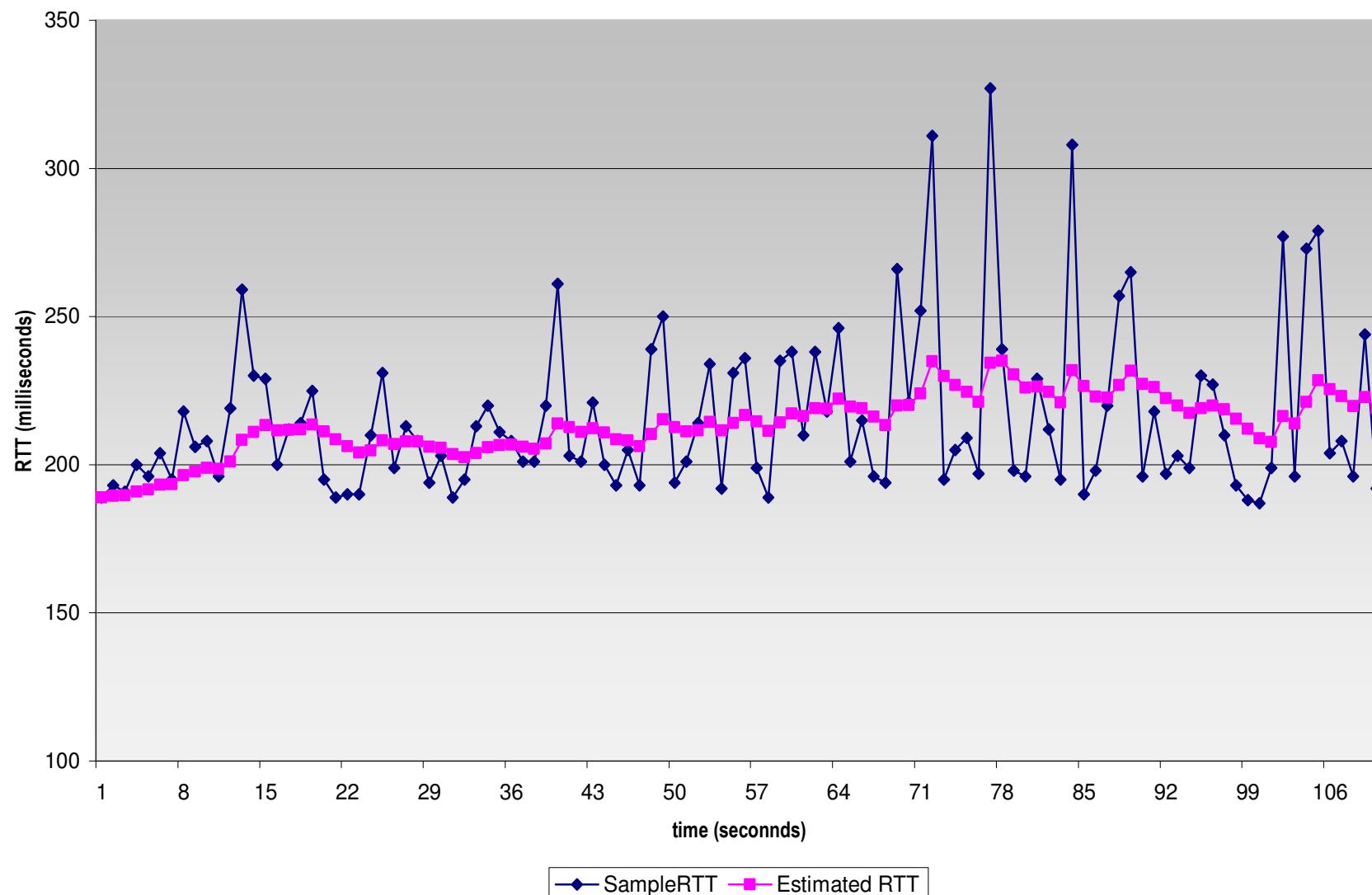
- ❖ Exponential weighted moving average
- ❖ influence of past sample decreases exponentially fast
- ❖ typical value: $\alpha = 0.125$

Example of Estimating RTT(E_RTT)

- ❖ T0 2:00 send data
 - ❖ 2:15 ACK back
 - ❖ SampleRTT=15
- ❖ T1 2:15 send data
 - ❖ 2:28 ACK back
 - ❖ SampleRTT=13
- ❖ T2 2:30 send data
 - ❖ 2:46 ACK back
 - ❖ SampleRTT=16
- ❖ T3 2:50 send data
 - ❖ 3:00 ACK back
 - ❖ SampleRTT=10
- ❖ T4 3:00 send data
 - ❖ Assume $\alpha = 0.1$ and
 - ❖ initial $E_{RTT0} = 10$ minutes
 - ❖ $EstimatedRTT = (1 - \alpha) * EstimatedRTT + \alpha * SampleRTT$
 - ❖ $E_{RTT1} = 0.9 \times 10 + 0.1 \times 15 = 10.5$
 - ❖ $E_{RTT2} = 0.9 \times 10.5 + 0.1 \times 13 = 10.75$
 - ❖ $E_{RTT3} = 0.9 \times 10.75 + 0.1 \times 16 = 11.275$
 - ❖ $E_{RTT4} = 0.9 \times 11.275 + 0.1 \times 10 = 11.1475$

Example RTT estimation:

RTT: gaia.cs.umass.edu to fantasia.eurecom.fr



TCP Round Trip Time and Timeout

Setting the timeout

- ❖ EstimatedRTT plus “safety margin”
 - ❖ large variation in EstimatedRTT
 - larger safety margin
- ❖ first estimate of how much SampleRTT deviates from EstimatedRTT:

$$\text{DevRTT} = (1-\beta) * \text{DevRTT} + \beta * |\text{SampleRTT} - \text{EstimatedRTT}|$$

(typically, $\beta = 0.25$)

Then set timeout interval:

$$\text{TimeoutInterval} = \text{EstimatedRTT} + 4 * \text{DevRTT}$$

Example of Setting Timeout(TO)

- ❖ T0 2:00 send data
 - ❖ 2:15 ACK back
 - ❖ SampleRTT=15
 - ❖ Assume $\beta = 0.2$ and
 - ❖ initial D_RTT0 = 1 minutes
 - ❖ $\text{DevRTT} = (1-\beta) * \text{DevRTT} + \beta * |\text{SampleRTT} - \text{EstimatedRTT}|$
 - ❖ $\text{TimeoutInterval} = \text{EstimatedRTT} + 4 * \text{DevRTT}$
 - ❖ $D_{RTT1} = 0.8 \times 1 + 0.2 \times |15 - 10| = 1.8$
 - ❖ $TO1 = 10.5 + 4 \times 1.8 = 17.7$
 - ❖ $D_{RTT2} = 0.8 \times 1.8 + 0.2 \times |13 - 10.5| = 1.94$
 - ❖ $TO2 = 10.75 + 4 \times 1.94 = 18.51$
 - ❖ $D_{RTT3} = 0.8 \times 1.94 + 0.2 \times |16 - 10.75| = 2.60$
 - ❖ $TO3 = 11.275 + 4 \times 2.6 = 21.675$
 - ❖ $D_{RTT4} = 0.8 \times 2.60 + 0.2 \times |10 - 11.275| = 2.335$
 - ❖ $TO4 = 11.1475 + 4 \times 2.335 = 20.4875$
- ❖ T1 2:15 send data
 - ❖ 2:28 ACK back
 - ❖ SampleRTT=13
- ❖ T2 2:30 send data
 - ❖ 2:46 ACK back
 - ❖ SampleRTT=16
- ❖ T3 2:50 send data
 - ❖ 3:00 ACK back
 - ❖ SampleRTT=10
- ❖ T4 3:00 send data

TCP reliable data transfer

- ❖ TCP creates reliable data transfer service on top of IP's unreliable service
- ❖ Pipelined segments
- ❖ Cumulative acks and Selective Repeat.
- ❖ TCP uses single retransmission timer (to reduce overhead)
- ❖ Retransmissions are triggered by:
 - ❖ Timeout events
 - ❖ Duplicate acks
- ❖ Initially consider simplified TCP sender:
 - ❖ ignore duplicate acks
 - ❖ ignore flow control, congestion control

Summary

- ❖ Transport Layer provides process-to-process logical connection
- ❖ Socket address = IP address + port number
- ❖ The User Datagram Protocol (UDP) is a connectionless, unreliable but simple.
- ❖ TCP creates reliable data transfer service
 - ❖ TCP Round Trip Time and Timeout

References

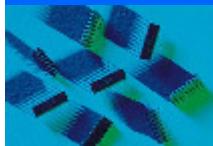
- ❖ Video on Comparing UDP and TCP

- ❖ <http://www.youtube.com/watch?v=Vdc8TCESIg8>

- ❖ Revision Quiz

- ❖ http://highered.mheducation.com/sites/0073376221/student_view0/chapter23/quizzes.html

- ❖ http://highered.mheducation.com/sites/0073376221/student_view0/chapter24/quizzes.html



Lecture 12 Network Applications and Sockets

Textbook: Ch. 26

Main Topics

A. WORLD WIDE WEB AND HTTP (26.1)

❖ **Browser**

❖ **HTTP**

B. FTP (26.2)

❖ **Connections**

C. Emails (26.3)

❖ **Sending and retrieving**

D. DNS (26.6)

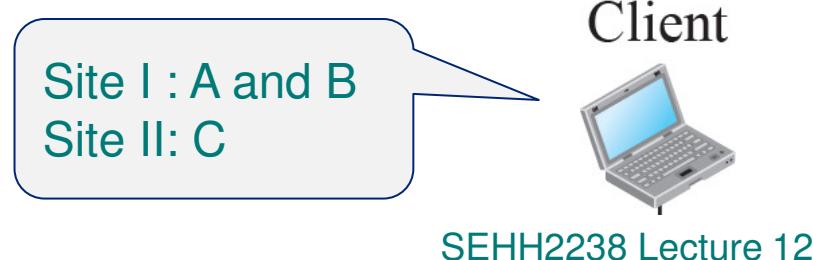
❖ **Name Spaces**

❖ **Resolution**

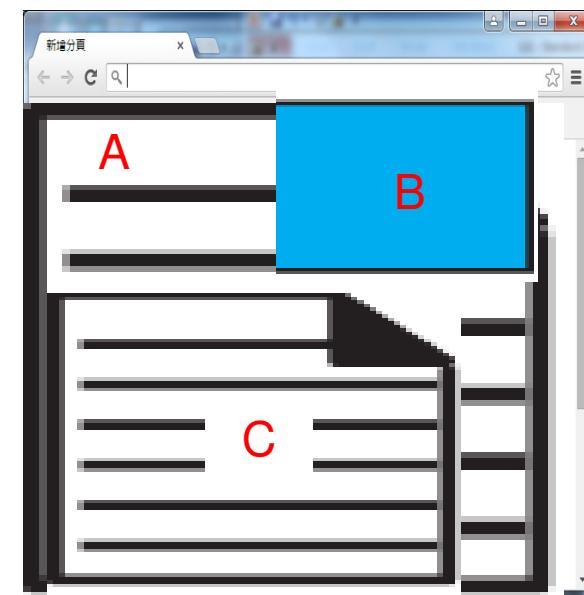
E. Sockets (25)

A. World Wide Web

- ❖ Consists of webpages with various items
 - ❖ Text, image, video, etc.
- ❖ Consider the main document (file A) of a webpage contains a reference to an image file (file B) and a reference to a text file (file C)
 - ❖ File A and file B are in Site I
 - ❖ File C is in Site II



SEHH2238 Lecture 12



Web pages linking to multiple sites

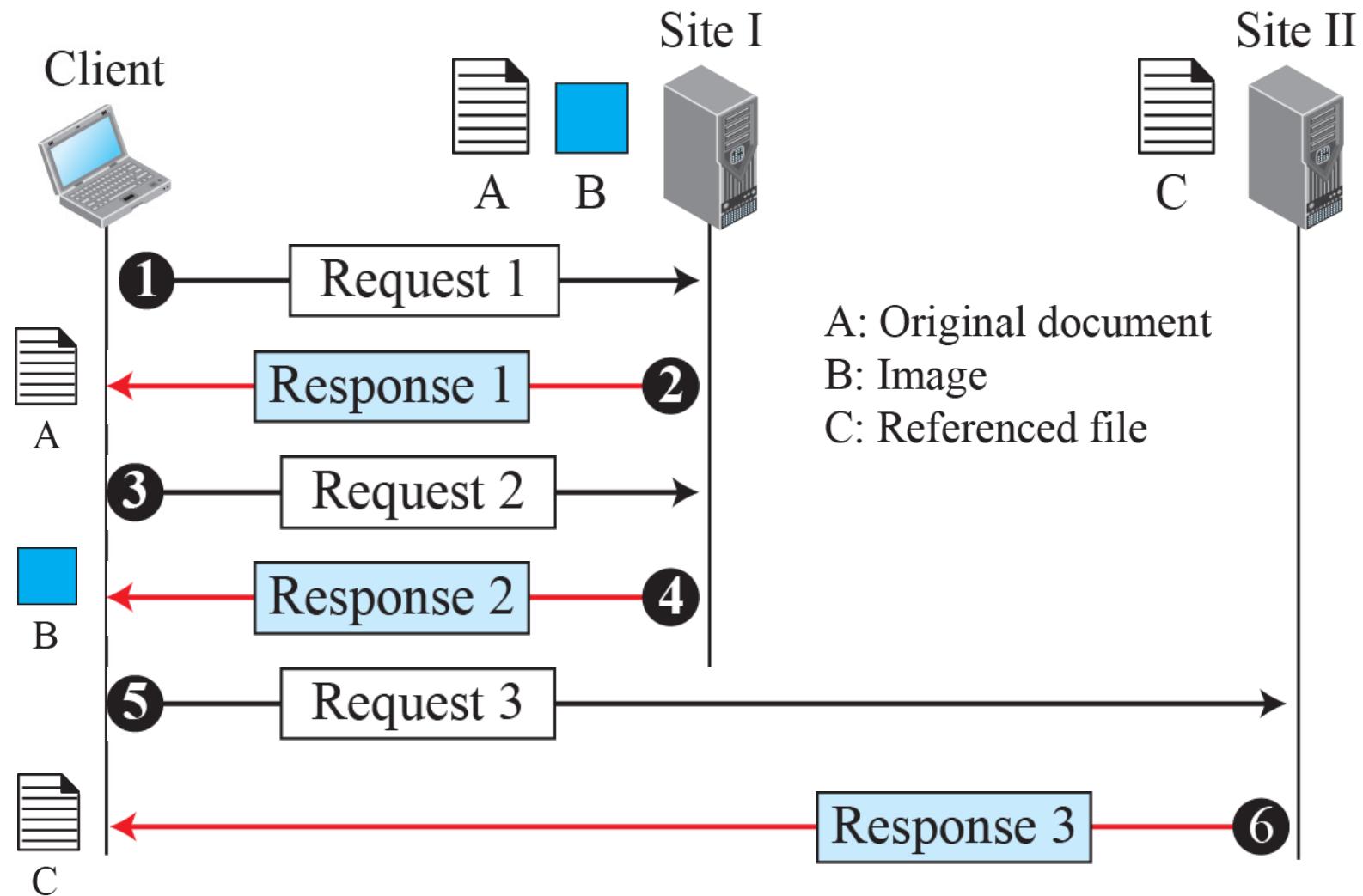


Figure 26.1: Example 26.1 SEHH2238 Lecture 12

Browser

- ❖ It allows them to view content via the web in its graphic form rather than HTML code, the primary language used by a website's designers to place the varying elements of a website

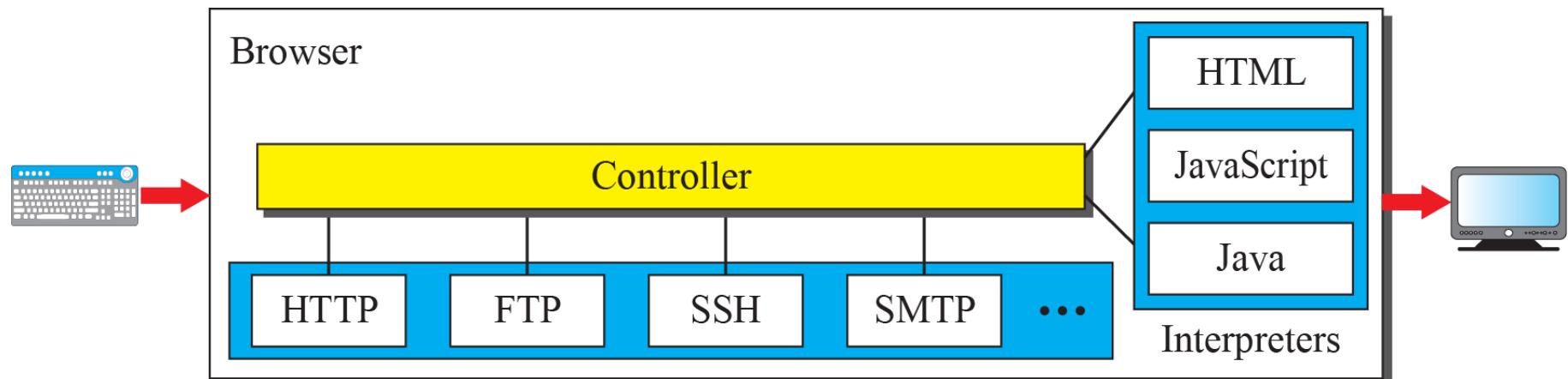


Figure 26.2: Browser

Uniform Resource Identifier (URL)

- ❖ URL format
 - ❖ protocol://host/path used most of the time
 - ❖ protocol://host:port/path used when port number is needed
- ❖ The URL <http://www.mhhe.com/compsci/forouzan/> defines the web page related to one of the computer in the McGraw-Hill company
 - ❖ The computer (host) name is www.mhhe.com
 - ❖ The three letters [www](http://www.mhhe.com) are part of the host name.
 - ❖ The path is [compsci/forouzan/](http://www.mhhe.com/compsci/forouzan/), which defines Forouzan's web page under the directory [compsci](http://www.mhhe.com/compsci).

HyperText Transfer Protocol

- ❖ The HyperText Transfer Protocol (HTTP) is used to define how the client-server programs can be written to retrieve web pages from the Web.
 - ❖ An HTTP client sends a request; an HTTP server returns a response.
 - ❖ The server uses the port number 80; the client uses a temporary port number.
 - ❖ HTTP uses the services of TCP.

Connection Persistence

❖ *Non-persistent* connection

❖ For example:

- ❖ The client needs to access a file that contains one link to an image. The text file and image are located on the same server. Here we need two connections.
- ❖ For each connection, TCP requires at least three handshake messages to establish the connection, but the request can be sent with the third one.
- ❖ After the connection is established, the object can be transferred.
- ❖ After receiving an object, another three handshake messages are needed to terminate the connection

❖ Used in HTTP 1.0

Non-persistent connection

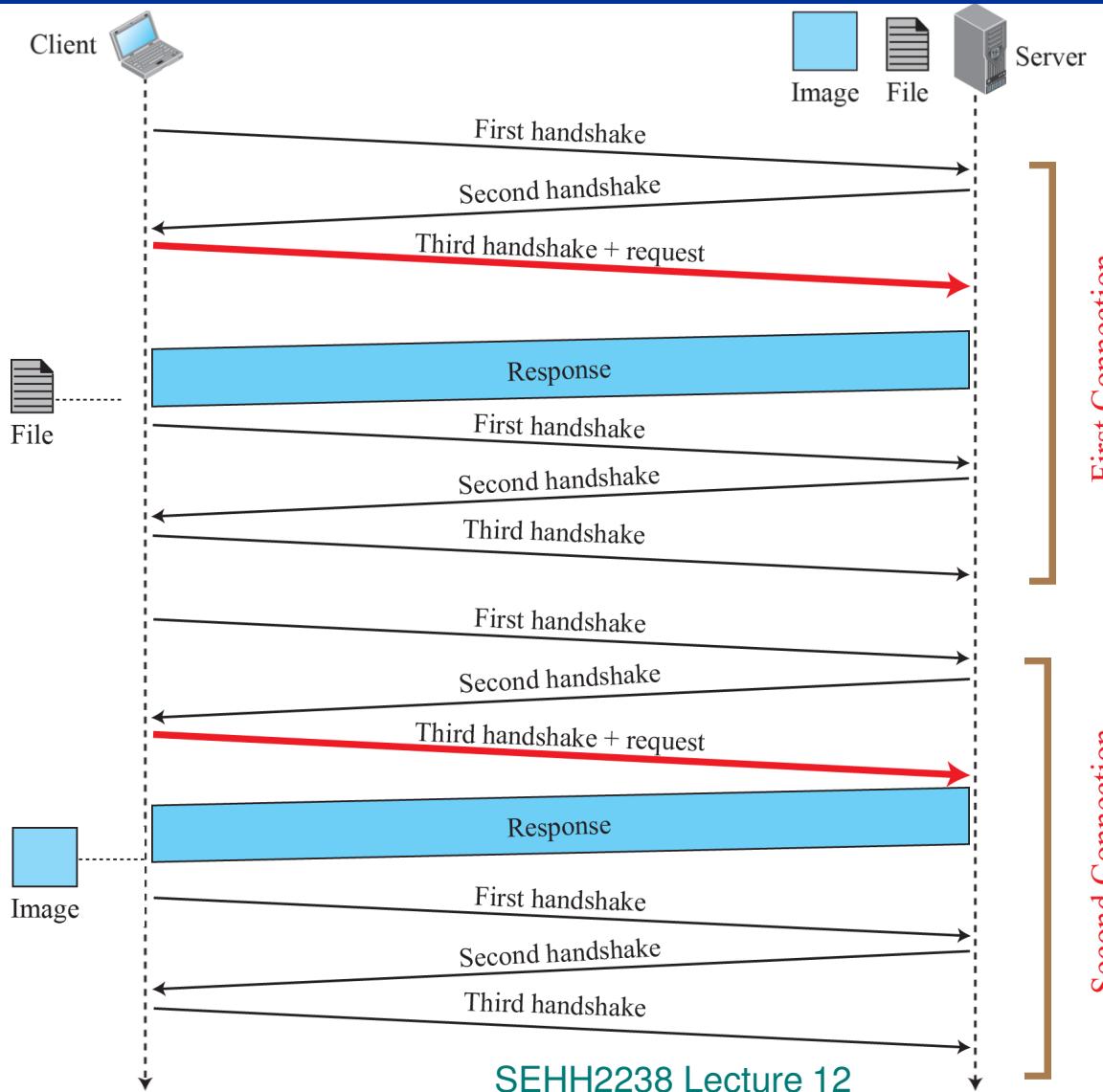


Figure 26.3:

Persistent connection

- ❖ Only one connection establishment and connection termination is used, but the request for the image is sent separately.
 - ❖ Used in HTTP 1.1

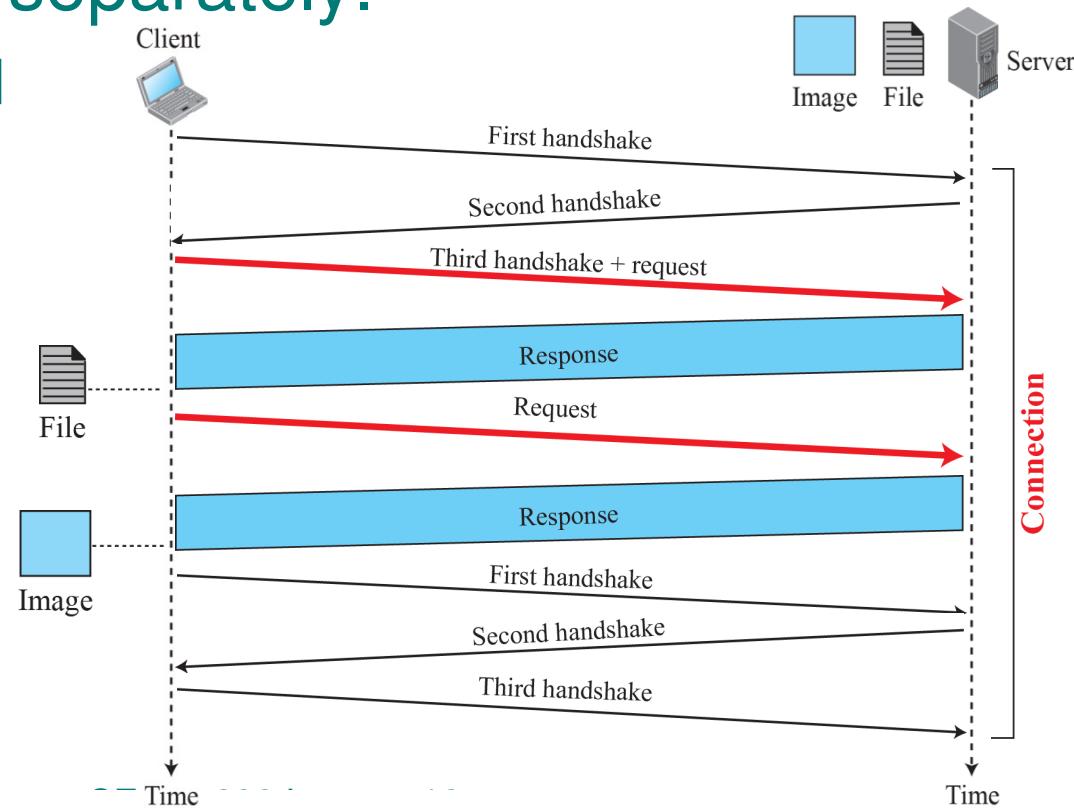


Figure 26.4:

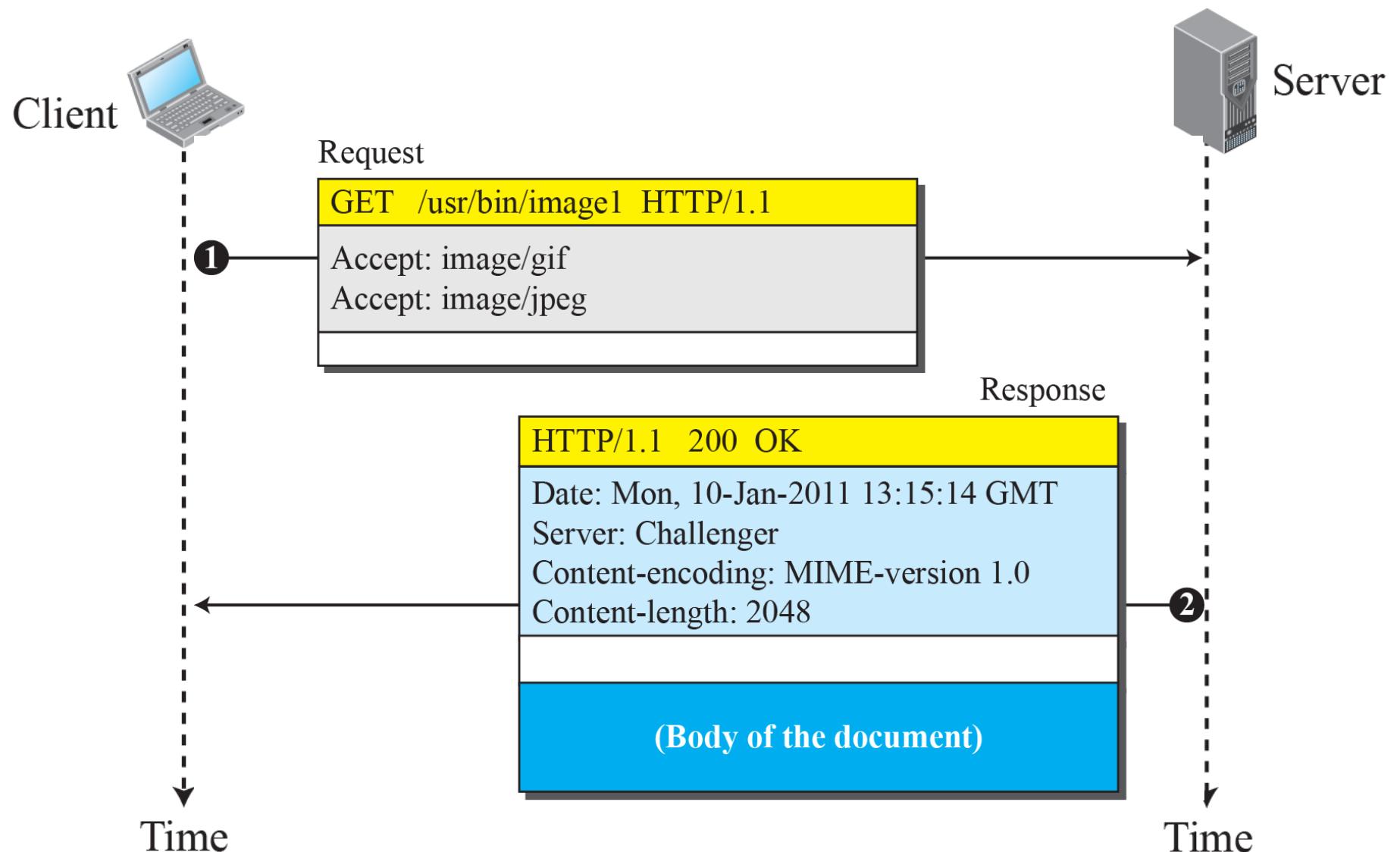


Figure 26.6

This example retrieves a document (see Figure 26.6). We use the GET method to retrieve an image with the path **/usr/bin/image1**.

- ❖ The request line shows the method (GET), the URL, and the HTTP version (1.1).
- ❖ The header has two lines that show that the client can accept images in the GIF or JPEG format.
- ❖ The request does not have a body.
- ❖ The response message contains the status line and four lines of header. The header lines define the date, server, content encoding (MIME version, which will be described in electronic mail), and length of the document.
- ❖ The body of the document follows the header.

Common HTTP Methods

| Method | Action |
|--------|---|
| GET | Requests a document from the server |
| HEAD | Requests information about a document but not the document itself |
| PUT | Sends a document from the client to the server |
| POST | Sends some information from the client to the server |
| DELETE | Removes a document from the server |

B. FTP

File Transfer Protocol (FTP) is the standard protocol provided by TCP/IP for copying a file from one host to another.

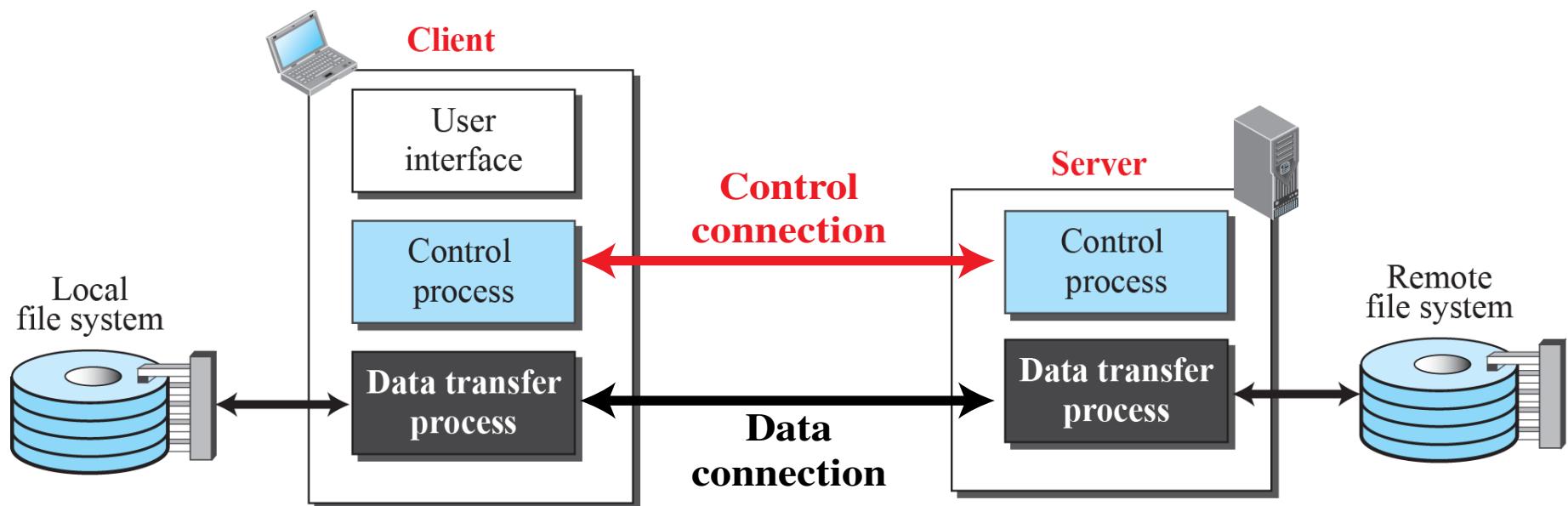


Figure 26.10: FTP

Two Connections

- ❖ Two connections in FTP
 - ❖ The control connection (port 21)
 - ❖ remains connected during the entire interactive FTP session.
 - ❖ The data connection (port 20)
 - ❖ is opened and then closed for each file transfer activity.
 - ❖ It opens each time commands that involve transferring files are used, and it closes when the file is transferred.

Data Connection

The data connection uses the well-known port 20 at the server site. However, the creation of a data connection is different from the control connection. The following shows the steps:

1. The client, not the server, issues a passive open using an ephemeral port.
2. Using the PORT command the client sends this port number to the server.
3. The server receives the port number and issues an active open using the well-known port 20 and the received ephemeral port number.

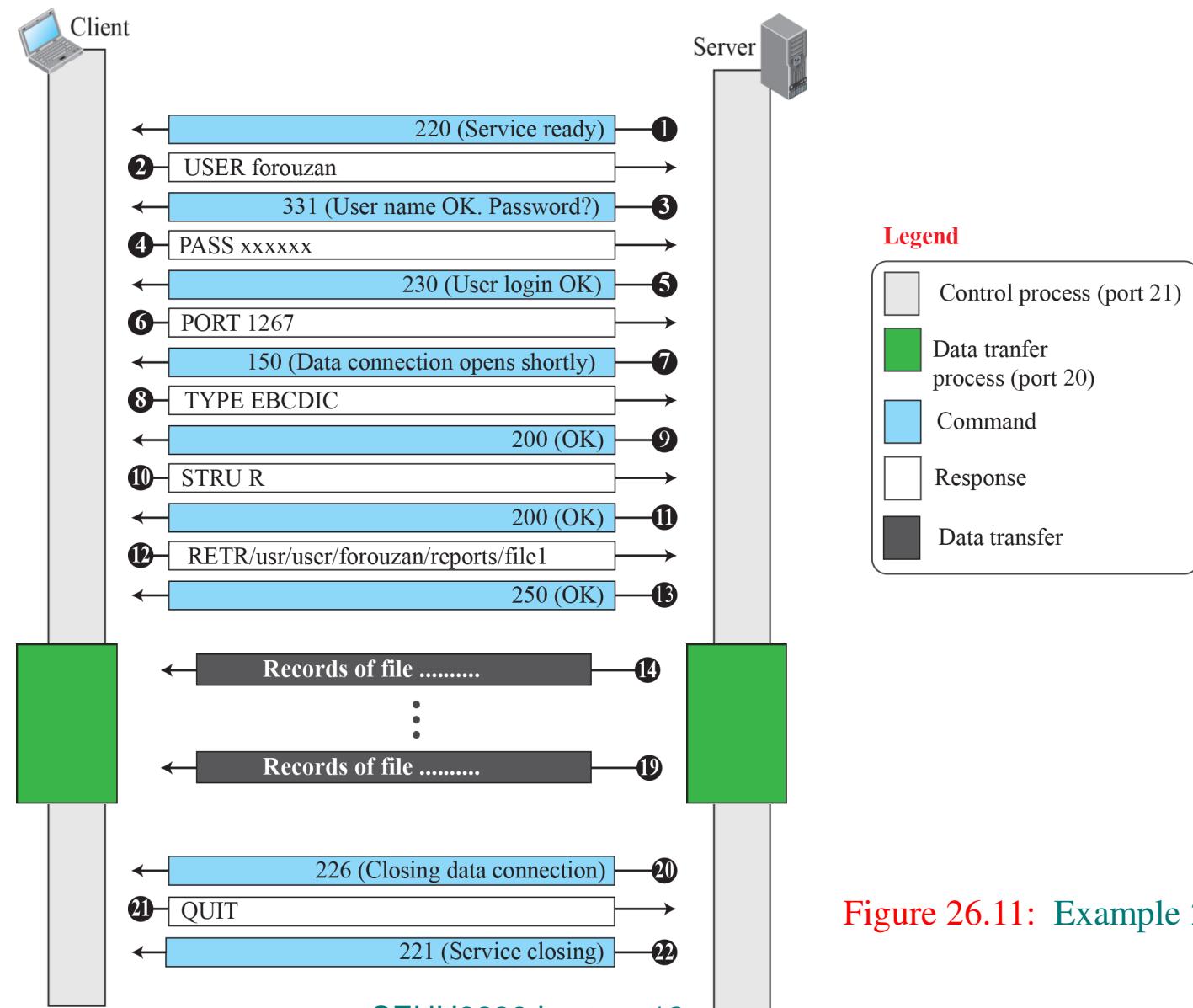


Figure 26.11: Example 26.12

Figure 26.11 shows an example of using FTP for retrieving only one file.

- ❖ The control connection remains open all the time, but the data connection is opened and closed repeatedly.
- ❖ After all records have been transferred, the server control process announces that the file transfer is done.
- ❖ Since the client control process has no file to retrieve, it issues the QUIT command, which causes the service connection to be closed.

Example 26.11

The following shows an actual FTP session that lists the directories.

```
$ ftp voyager.deanza.fhda.edu  
Connected to voyager.deanza.fhda.edu.  
220 (vsFTPd 1.2.1)  
530 Please login with USER and PASS.
```

Name (voyager.deanza.fhda.edu:forouzan): *forouzan*

```
331 Please specify the password.
```

Password:*****

```
230 Login successful.
```

Remote system type is UNIX.

Using binary mode to transfer files.

```
227 Entering Passive Mode (153,18,17,11,238,169)
```

```
150 Here comes the directory listing.
```

| | | | | | | | |
|-------------------|----------|-------------|------------|-------------|---------------|-------------|-----------------|
| drwxr-xr-x | 2 | 3027 | 411 | 4096 | Sep 24 | 2002 | business |
| drwxr-xr-x | 2 | 3027 | 411 | 4096 | Sep 24 | 2002 | personal |
| drwxr-xr-x | 2 | 3027 | 411 | 4096 | Sep 24 | 2002 | school |

```
226 Directory send OK.
```

ftp> *quit*

```
221 Goodbye.
```

C. ELECTRONIC MAIL

- ❖ ***Electronic mail (or e-mail) allows users to exchange messages.***
- ❖ **E-mail is a one-way transaction**
- ❖ The email system needs two user agents (UAs), two pairs of Mail Transfer Agent (MTAs) (client and server), and a pair of Mail Access Agent (MAAs) (client and server).

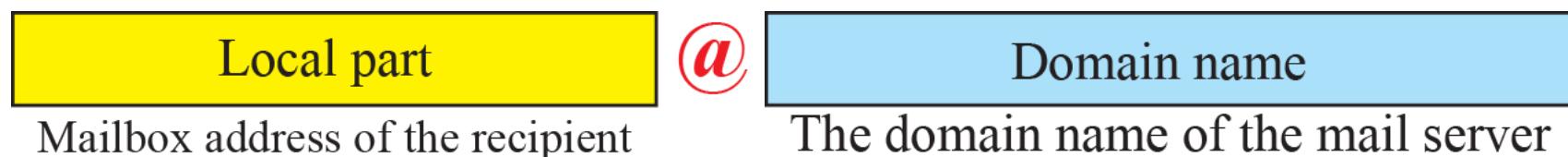


Figure 26.14: E-mail address

Email Architecture

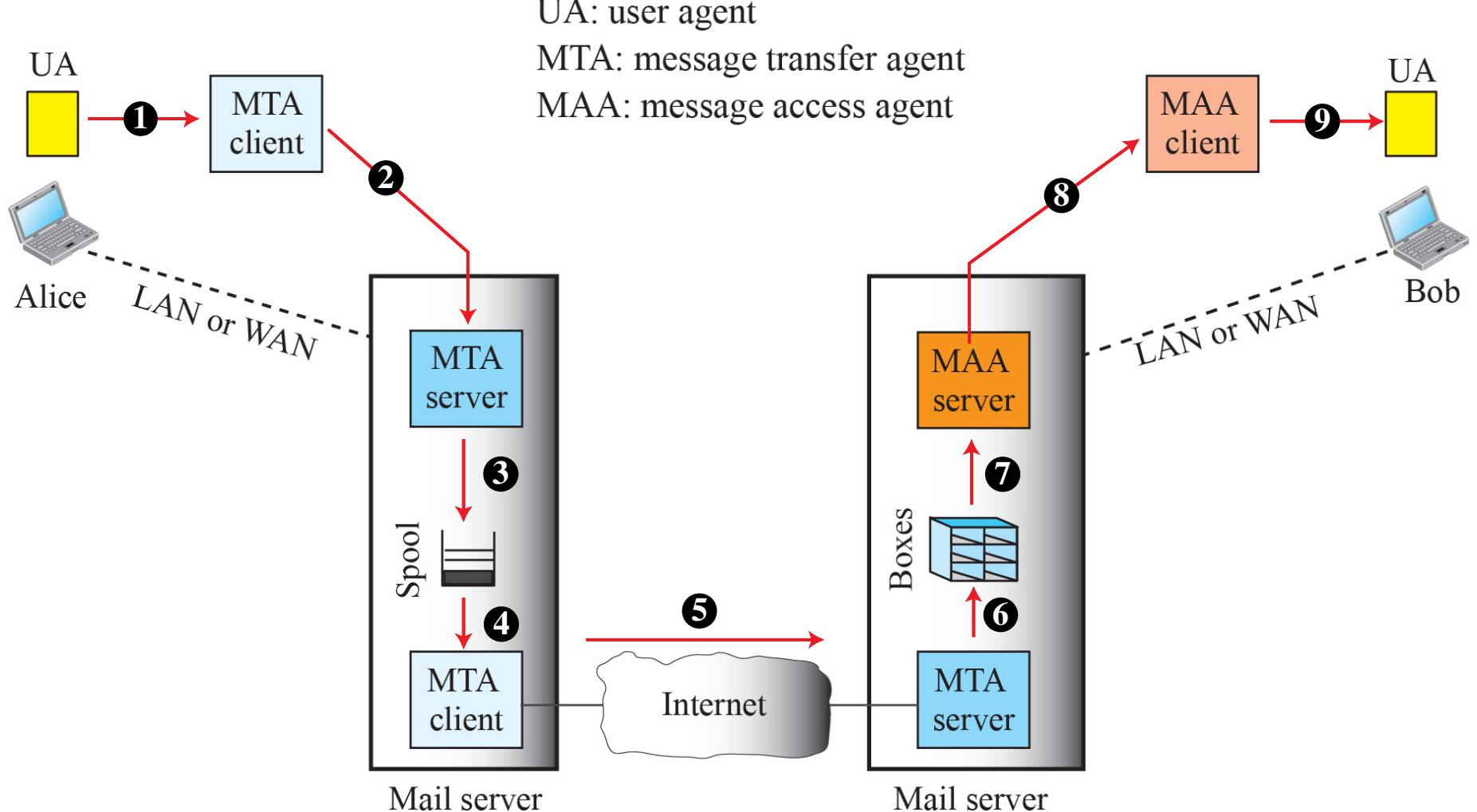


Figure 26.12: Common scenario

Email Sending

1. User Agent (UA) provides service to the user to create email.
2. UA sends the mail to the mail server by Simple mail transfer protocol (SMTP).
3. The mail server at sending site use a queue (spool) to store messages
4. The message is waiting to be sent.
5. Mail Transfer Agent (MTA) client in the sending site sends the mail to receiver's mail server, by SMTP.

Email Reading

6. MTA server in the receiving site store the mail in the reserving users' mailbox.
7. Receiving user connects to the mail server (login)
8. Then, the mail is delivered by POP or IMAP protocol by Mail Access Agent (MAA).

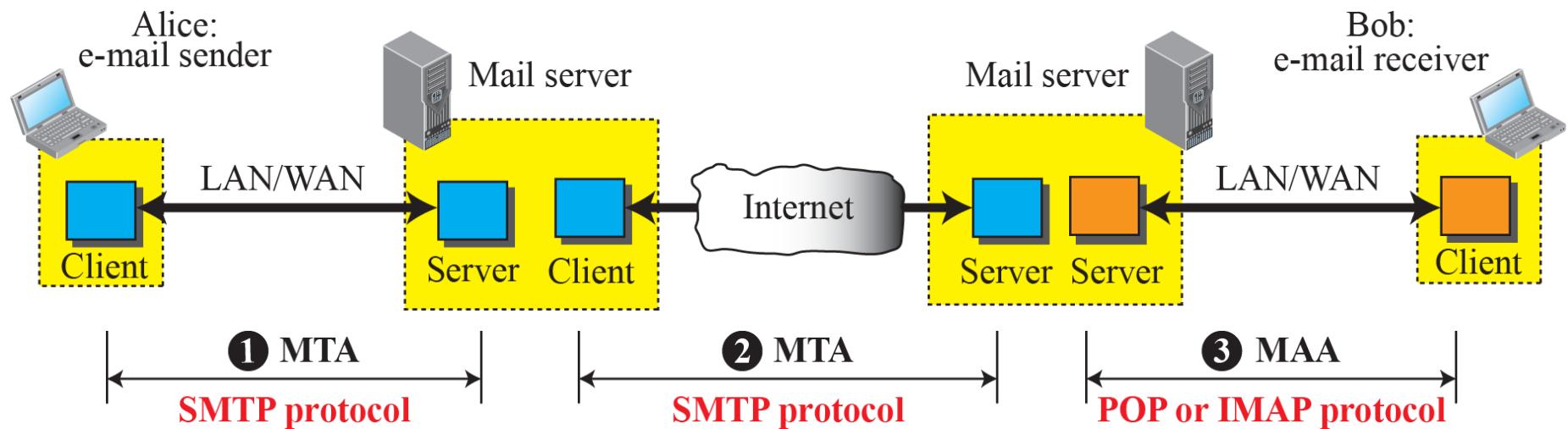


Figure 26.15: Protocols used in electronic mail

D. DOMAIN NAME SYSTEM (DNS)

- ❖ The Internet needs to have a directory system that can map a name to an address.
- ❖ The names must be unique because the addresses are unique.
 - ❖ A **name space** that maps each address to a unique name.

Flow of Domain Name to IP

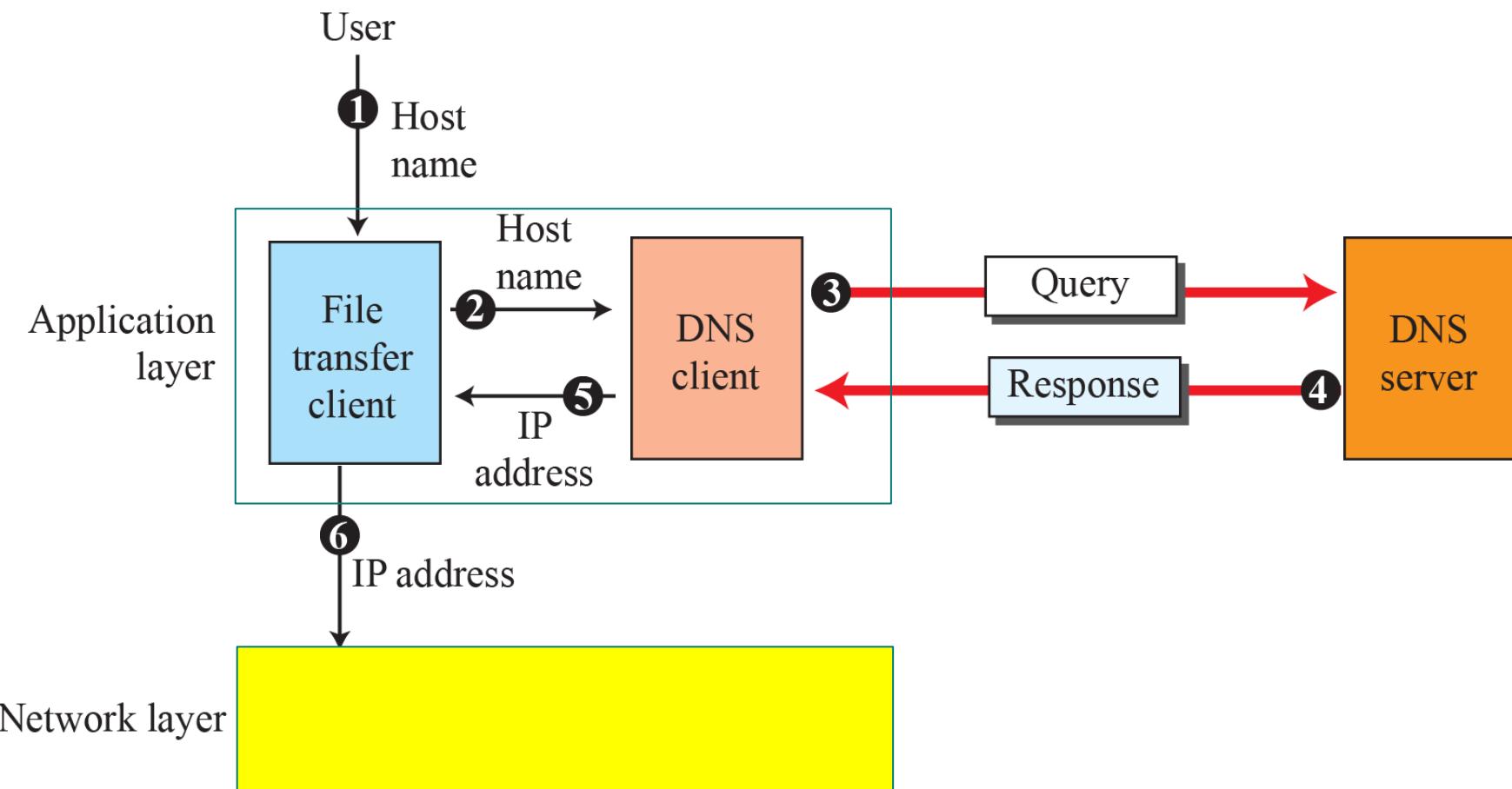


Figure 26.28: Purpose of DNS

Domain names and labels

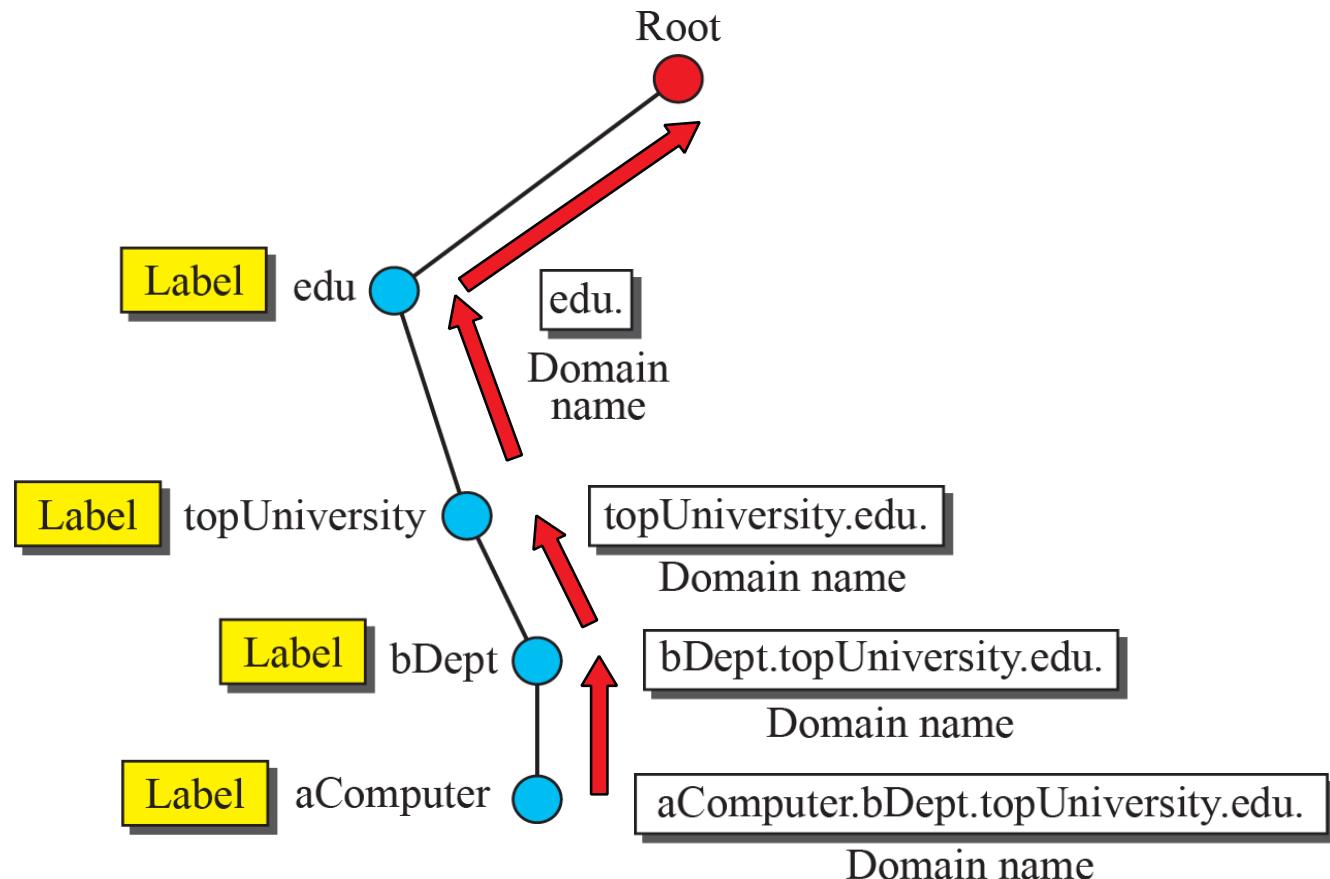


Figure 26.30: Domain names and labels

Resolution

- ❖ Mapping a name to an address is called name-address resolution.
- ❖ DNS is designed as a client-server application.
- ❖ The resolver accesses the closest DNS server with a mapping request.
 - ❖ If the server has the information, it satisfies the resolver;
 - ❖ otherwise, it either refers the resolver to other servers or asks other servers to provide the information.

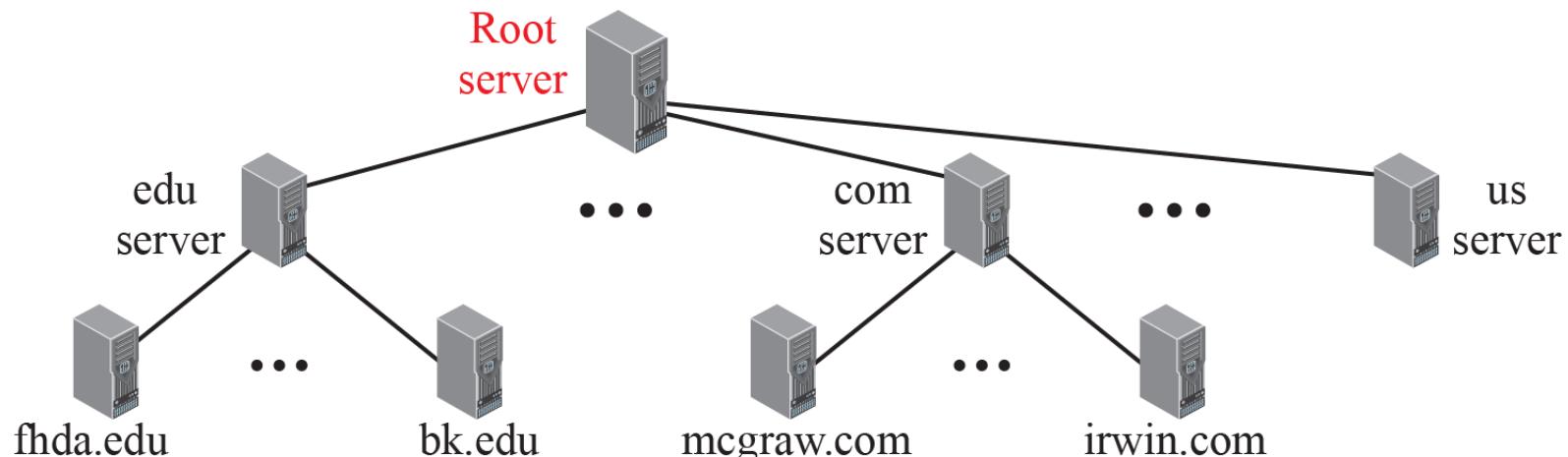


Figure 26.32: Hierarchy of name servers

Recursive resolution

Query for the IP address of
engineering.mcgraw-hill.com

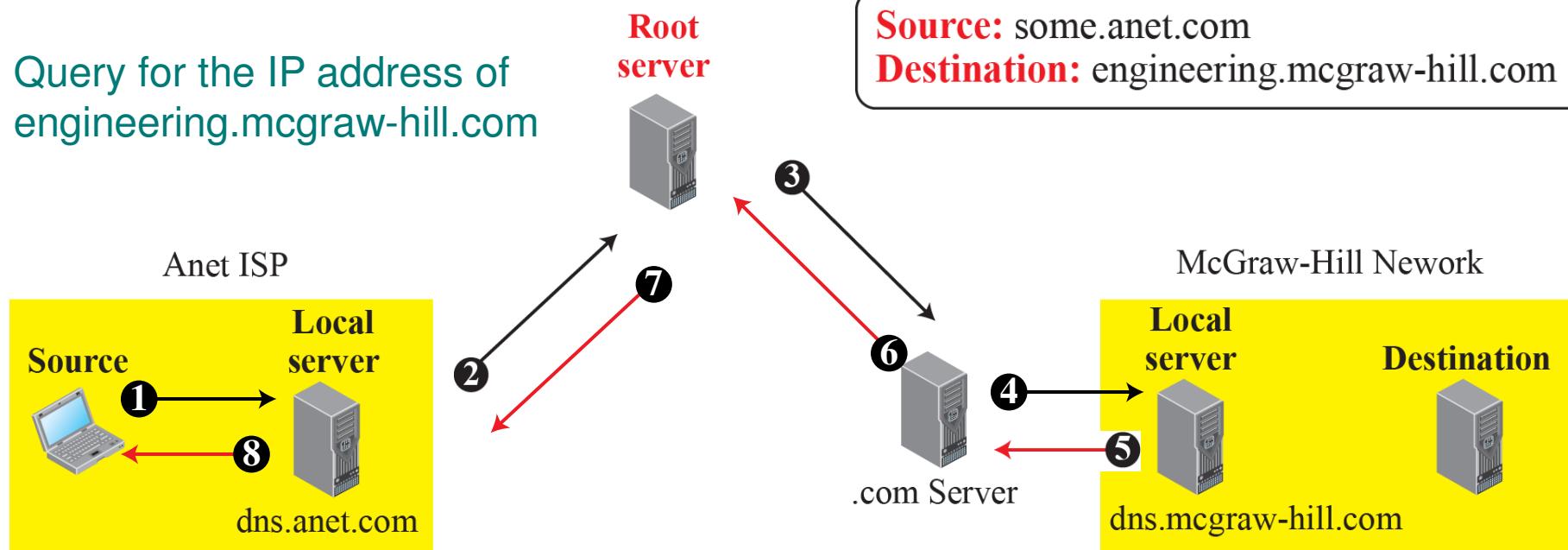


Figure 26.36: Recursive resolution

Iterative resolution

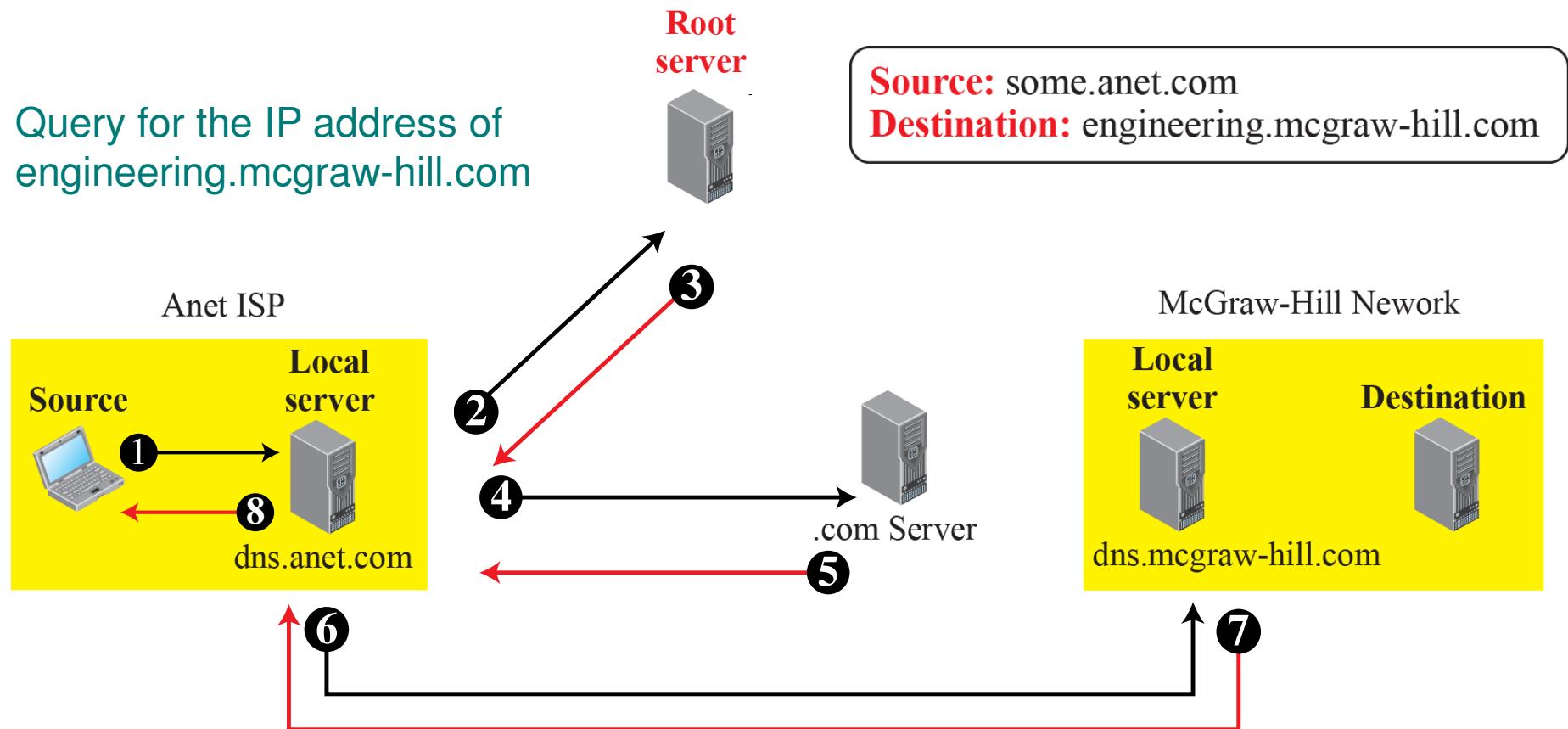
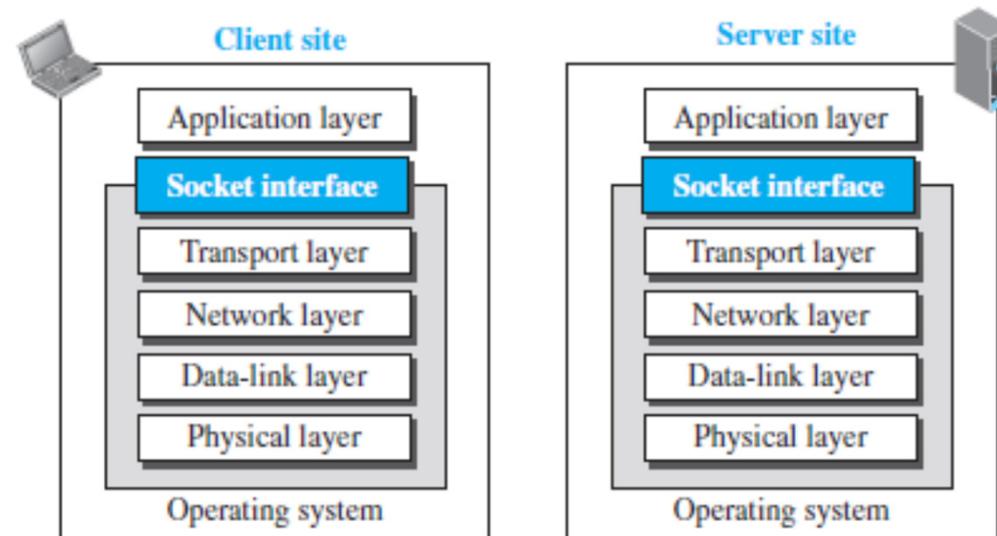


Figure 26.37: Iterative resolution

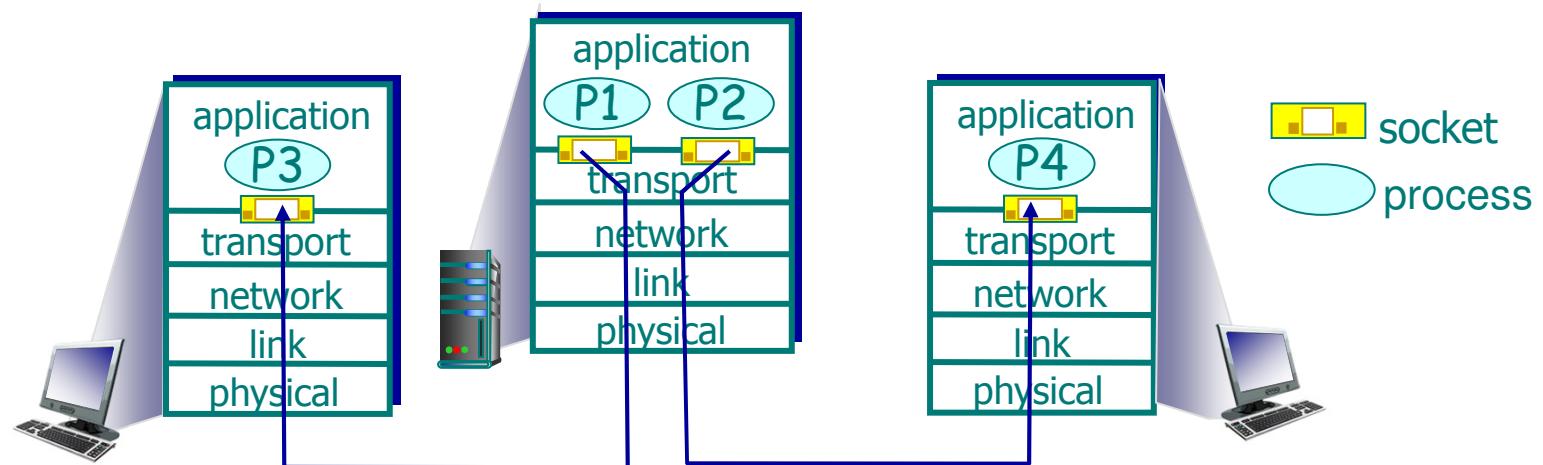
E. Sockets

- ❖ There are different types of application programs
- ❖ Using a common set of lower layer protocols in TCP/IP protocol suite, usually provided by OS
- ❖ Need a socket for such interaction
 - ❖ Analogy: Different electric appliances use the same electricity supply network via electric socket

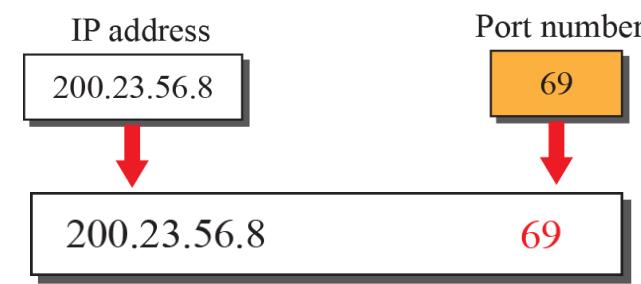


Socket Address

- ❖ A host can run multiple applications, each has a socket for communication.

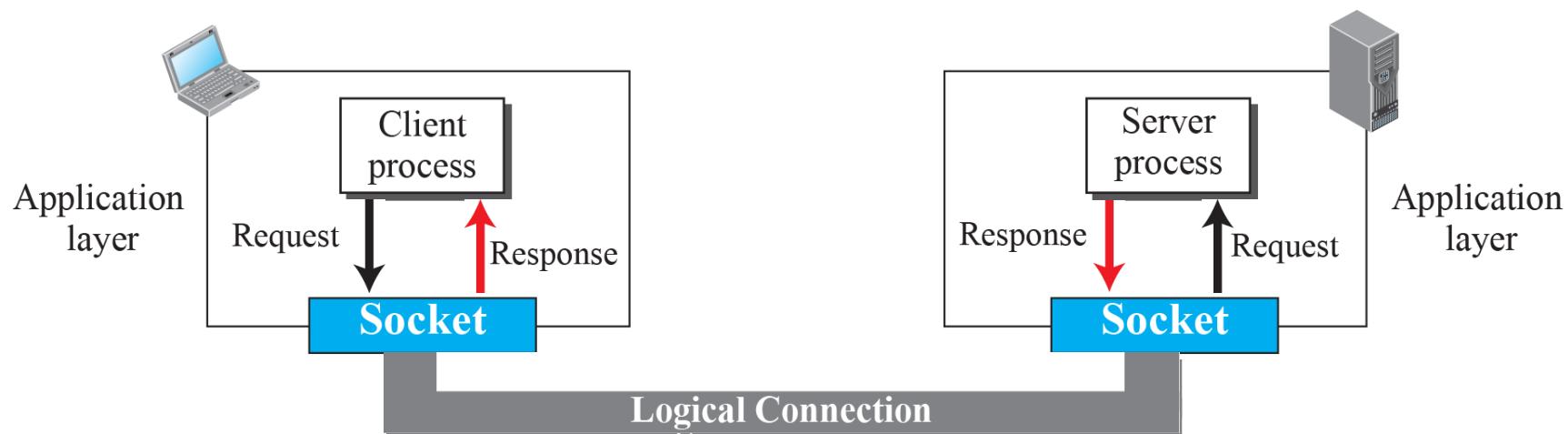


- ❖ The combination of an IP address and a port number is called a socket address.



Application programming interface (API)

- ❖ How can a client process communicate with a server process?
 - ❖ We need a set of instructions to tell the lowest four layers of the TCP/IP suite to
 - ❖ open the connection,
 - ❖ send and receive data from the other end, and
 - ❖ close the connection.
- ❖ A set of instructions of this kind is normally referred to as an application programming interface (API).



Socket Programming

- ❖ A component of application design that involves network connection.
- ❖ Write programs to... (*refer to appendix for details*)
 - ❖ create socket depends on TCP or UDP is used
 - ❖ read and write data from the socket
 - ❖ perform necessary processing on the data
- ❖ Client initiates data transmission to a server.
- ❖ For server, the socket is always waiting (listening) for incoming data.

Summary

- ❖ Web: HTTP
 - ❖ Non-persistent (1.0) vs. persistent (1.1)
- ❖ File Transfer Protocol: FTP
 - ❖ Control and Data connection separated
- ❖ Email: SMTP, POP, IMAP
- ❖ Domain Name: DNS
 - ❖ Recursive vs. iterative resolution
- ❖ Sockets: Address, API

References

❖ Video on DNS

☞ <http://www.youtube.com/watch?v=ZBi8GCxk7NQ>

❖ Video on IMAP vs POP

☞ <http://www.youtube.com/watch?v=BK4ng6Gcits>

❖ Revision Quiz

☞ http://highered.mheducation.com/sites/0073376221/student_view0/chapter26/quizzes.html