7 Application

6 Presentation

5 Session

4 Transport

3 Network

2 Data Link

1 Physical



Middle Technical University
Electrical & Engineering technical college
Computer Engineering Dept

Computer Network protocols

4th Grad

2021-2022

BY

Asst.Prof.Dr.Eng: Mohamed I. Shujaa

7 Application

6 Presentation

5 Session

4 Transport

3 Network

2 Data Link

1 Physical

Source & References

1. Computer Networks, Andrew S. Tanenbaum
2. top to down-computer networking, James Kurose.

3. Network communication ,Bahrus

Course content

OSI Model

TCP/IP Model

Protocols at each layer

7 Application

6 Presentation

5 Session

4 Transport

3 Network

2 Data Link

1 Physical

Lecture I
INTRODUCTION

7 Application

6 Presentation

5 Session

4 Transport

3 Network

2 Data Link

1 Physical

NETWORK ARCHITECTURES

A set of layers and protocols is called the network architecture.

Protocol Hierarchies

Networks are organized as layers to reduce design complexity. Each layer offers *services* to the higher layers. Between adjacent layers is an *interface*.

Services – connection oriented and connectionless.

Interface – defines which *primitives* and services the lower layer will offer to the upper layer.

Primitives – operations such as request, indicate, response, confirm.

7 Application

6 Presentation

5 Session

4 Transport

3 Network

2 Data Link

1 Physical

NETWORK ARCHITECTURES

Design Issues for the Layers(functions)

- Mechanism for connection establishment
- Rules for data transfer
- Error control
- Fast sender swamping a slow receiver
- Inability of processes to accept long messages
- Routing in the case of multiple paths

Network Protocols

- Protocol: *is a format order of messages sent and received among the net entities and action taken on messages transmission receipt.*

Protocol process :

- *The format or structure of the message*
- *The process by which networking devices share information about pathways with other networks*
- *How and when error and system messages are passed between devices*
- *The setup and termination of data transfer sessions*

Layering In Networked Computing

- OSI Model (open system interconnection)
- TCP/IP Model
- Protocols at each layer

Learning outcomes

- Understand the need of layering in Networked computing
- Understand the OSI model and the TCP/IP model
  - Understand the function protocols and their role at each layer.
    - TCP protocol
    - UDP protocol
- Understand the role of header in communication between layers
- Understand how data sent from one host arrive to the target host.

Why a layered model?

- Easier to teach communication process.
- Speeds development, changes in one layer does not affect how the other levels works.
- Allows different hardware and software to work together.
- Reduces complexity
- Breaks down communication into smaller, simpler parts.

What is “THE MODEL?”

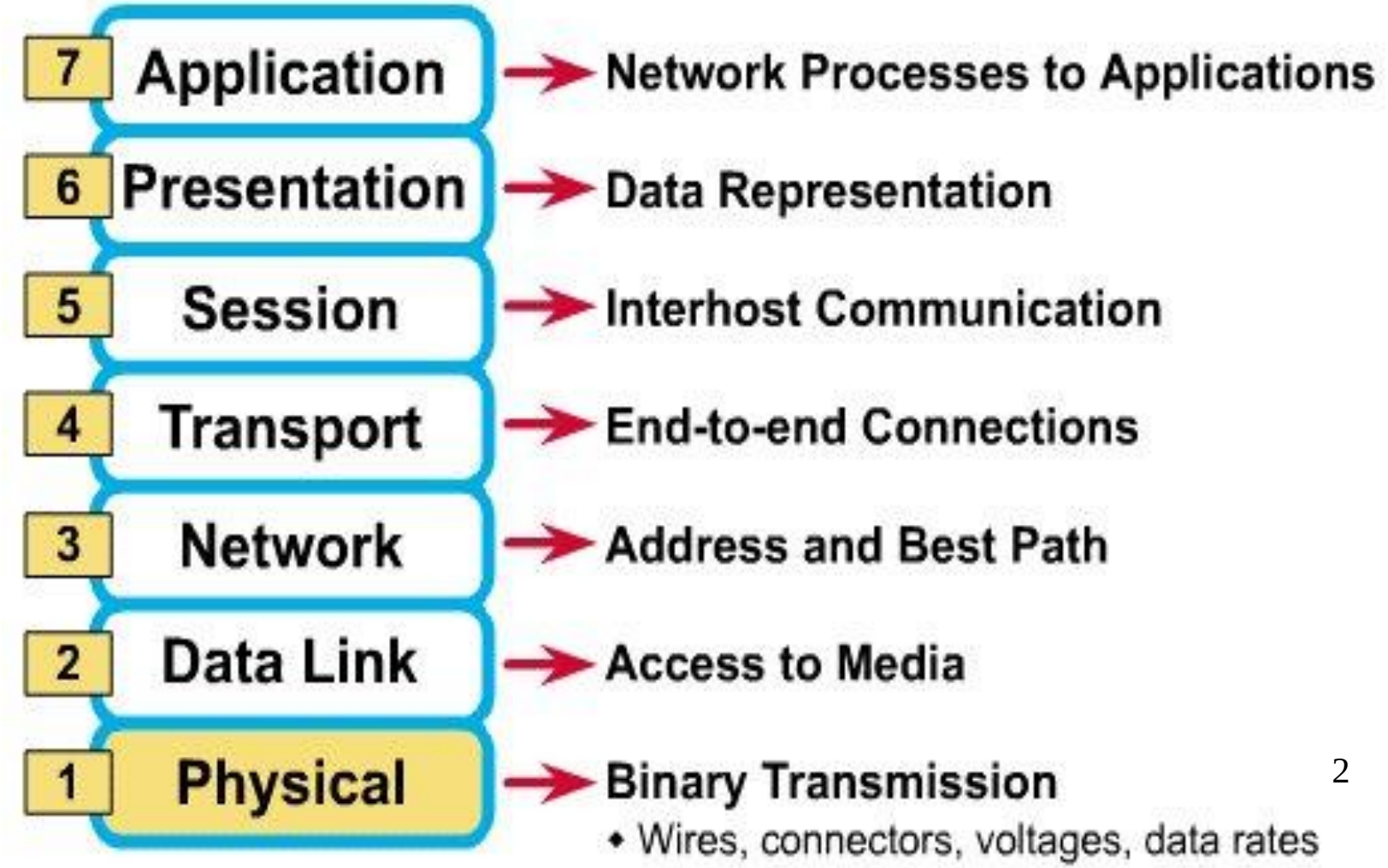
- the OSI reference model.
- The OSI model
  - is a theoretical blueprint that helps us understand how data gets from one user’s computer to another.
  - It is also a model that helps develop standards so that all of our hardware and software talks nicely to each other.

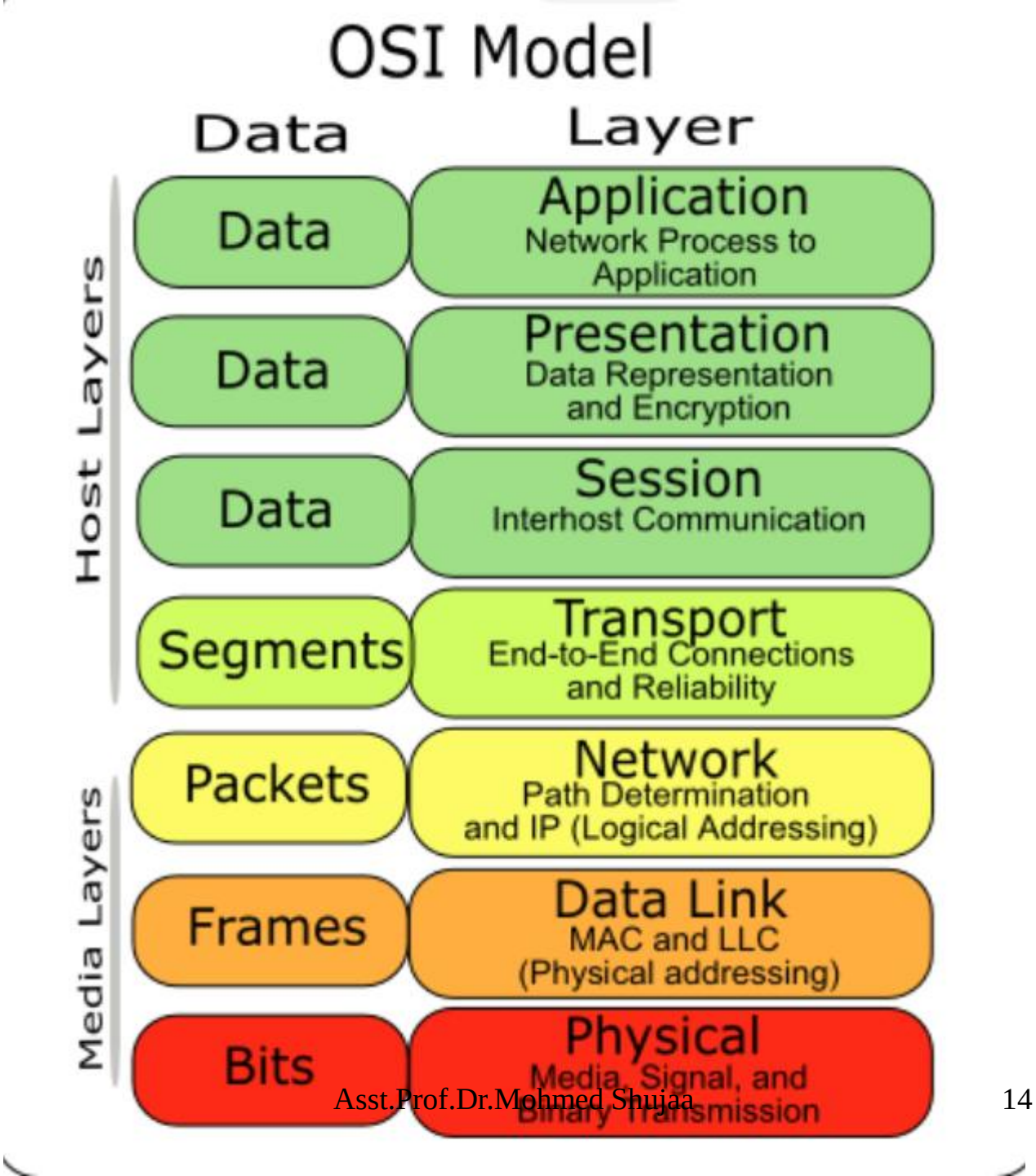
7 Layer OSI Model

- Q. ***Why use a reference model***?
  - Serves as an outline of rules for how protocols can be used to allow communication between computers.
  - Each layer has its own function and provides support to other layers.

- Q. ***What are Other reference models are in use***.
  - Most well known is the TCP/IP reference model.
  - We will compare OSI and TCP/IP models

What Each Layer Does

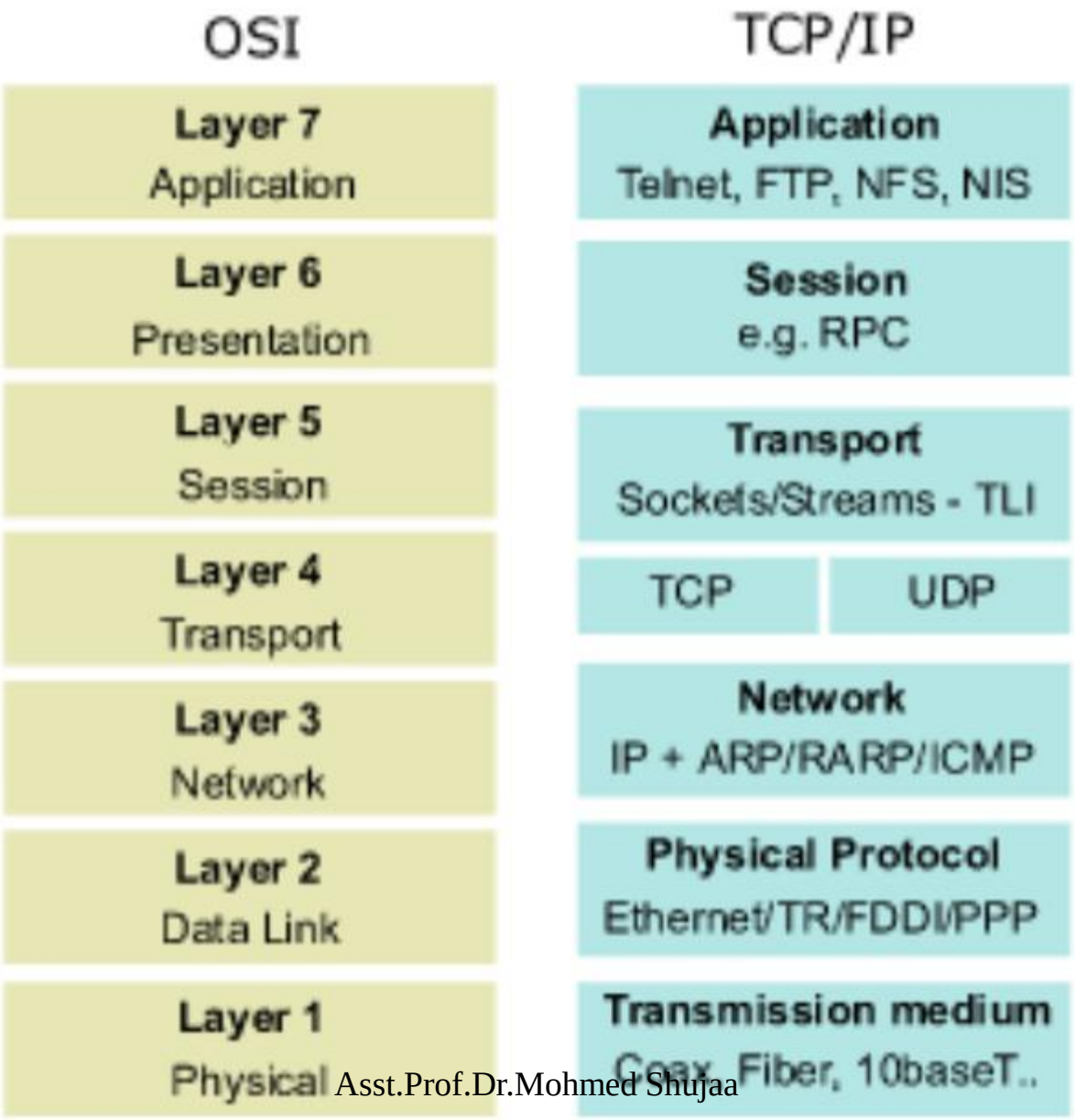



OSI Model
Data
Layer
Host Layers
Media Layers
Data
Application
Network Process to Application
Data
Presentation
Data Representation and Encryption
Data
Session
Interhost Communication
Segments
Transport
End-to-End Connections and Reliability
Packets
Network
Path Determination and IP (Logical Addressing)
Frames
Data Link
MAC and LLC (Physical addressing)
Bits
Physical
Media, Signal, and Binary Transmission

OSI model

Layer	Name	Example protocols
7	Application Layer	HTTP, FTP, DNS, SNMP, Telnet
6	Presentation Layer	SSL, TLS
5	Session Layer	NetBIOS, PPTP
4	Transport Layer	TCP, UDP
3	Network Layer	IP, ARP, ICMP, IPSec
2	Data Link Layer	PPP, ATM, Ethernet
1	Physical Layer	Ethernet, USB, Bluetooth, IEEE802.11

The OSI Layer Model



7 Application

6 Presentation

5 Session

4 Transport

3 Network

2 Data Link

1 Physical

OSI REFERENCE MODEL

1. Physical Layer

a) Convert the logical 1's and 0's coming from layer 2 into electrical signals.

b) Transmission of the electrical signals over a communication channel.

Main topics:

- Transmission mediums
- Encoding
- Modulation
- RS232 and RS422 standards
- Repeaters
- Hubs (multi-port repeater)

7 Application

6 Presentation

5 Session

4 Transport

3 Network

2 Data Link

1 Physical

OSI REFERENCE MODEL

2. Data Link Layer

a) Error control to compensate for the imperfections of the physical layer.

b) Flow control to keep a fast sender from swamping a slow receiver.

Main topics:

- Framing methods
- Error detection and correction methods
- Flow control
- Frame format
- IEEE LAN standards
- Bridges
- Switches (multi-port bridges)

7 Application

6 Presentation

5 Session

4 Transport

3 Network

2 Data Link

1 Physical

OSI REFERENCE MODEL

3. Network Layer

a) Controls the operation of the subnet.

b) Routing packets from source to destination.

c) Logical addressing.

Main topics:

- Internetworking
- Routing algorithms
- Internet Protocol (IP) addressing
- Routers

7 Application

6 Presentation

5 Session

4 Transport

3 Network

2 Data Link

1 Physical

OSI REFERENCE MODEL

4. Transport Layer

a) Provides additional Quality of Service.

b) Heart of the OSI model.

Main topics:

- Connection-oriented and connectionless services
- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)
- ACK

7 Application

6 Presentation

5 Session

4 Transport

3 Network

2 Data Link

1 Physical

OSI REFERENCE MODEL

5. Session Layer

a) Allows users on different machines to establish *sessions (dialogue)* between them.

b) One of the services is managing dialogue control.

c) Token management.

d) Synchronization.

7 Application

6 Presentation

5 Session

4 Transport

3 Network

2 Data Link

1 Physical

OSI REFERENCE MODEL

6. Presentation Layer

a) Concerned with the syntax and semantics of the information.

b) Preserves the meaning of the information.

c) Data compression.

d) Data encryption.

7 Application

6 Presentation

5 Session

4 Transport

3 Network

2 Data Link

1 Physical

OSI REFERENCE MODEL

7. Application Layer

a) Provides protocols that are commonly needed.

Main topics:

- File Transfer Protocol (FTP)
- HyperText Transfer Protocol (HTTP)
- Simple Mail Transfer Protocol (SMTP)
- Simple Network Management Protocol (SNMP)
- Network File System (NFS)
- Telnet

7 Application

6 Presentation

5 Session

4 Transport

3 Network

2 Data Link

1 Physical

SERVICES OF OSI

1. Connection-Oriented and Connectionless

Connection-Oriented – before data is sent, the service from the sending computer must establish a connection with the receiving computer.

Connectionless – data can be sent at any time by the service from the sending computer.

Q: Is downloading a music file from the Internet connection-oriented or connectionless?

Q: Is email connection-oriented or connectionless?

7 Application

6 Presentation

5 Session

4 Transport

3 Network

2 Data Link

1 Physical

SERVICES OF OSI

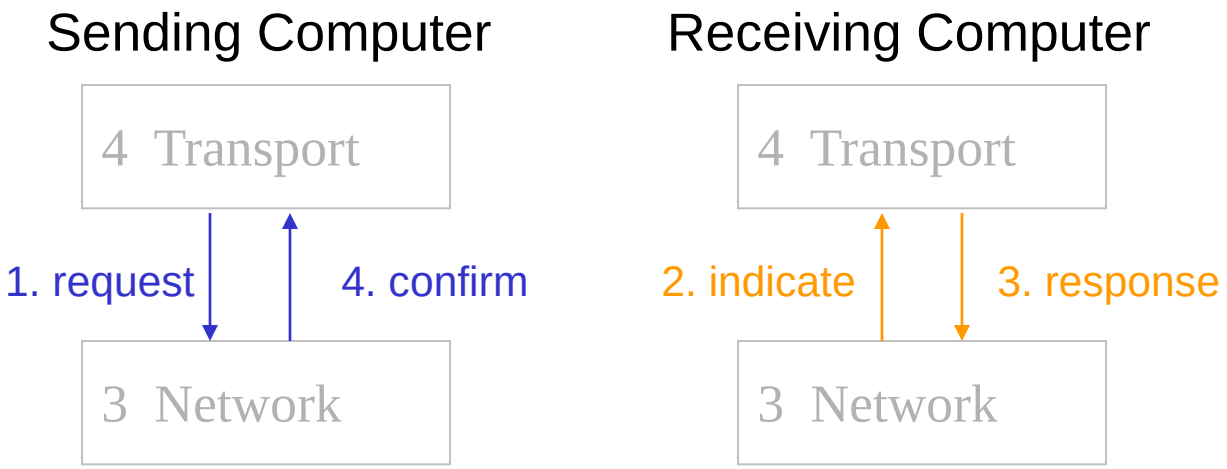
2. Service Primitives

Request – entity wants the service to do some work

Indicate – entity is to be informed about an event

Response – entity responds to an event

Confirm – entity is to be informed about its request



7 Application

6 Presentation

5 Session

4 Transport

3 Network

2 Data Link

1 Physical

Lecture II

1. PHYSICAL LAYER

Binary transmission physical addressing

OSI model

Layer	Name	Example protocols
7	Application Layer	HTTP, FTP, DNS, SNMP, Telnet
6	Presentation Layer	SSL, TLS
5	Session Layer	NetBIOS, PPTP
4	Transport Layer	TCP, UDP
3	Network Layer	IP, ARP, ICMP, IPSec
2	Data Link Layer	PPP, ATM, Ethernet
1	Physical Layer	Ethernet, USB, Bluetooth, IEEE802.11

7 Application

6 Presentation

5 Session

4 Transport

3 Network

2 Data Link

1 Physical

OVERVIEW

1. Signals
2. Transmission Media
3. Analog Transmission
4. Digital Transmission
5. Transmission and Switching

TRANSMISSION MEDIA

1. Guided

Data is sent via a wire or optical cable.

Twisted Pair

Two copper wires are twisted together to reduce the effect of crosstalk noise. (e.g. Cat5, UTP, STP)

- **Unshielded Twisted-Pair (UTP) cables**
- **Shielded Twisted-Pair (STP) cables**

Baseband Coaxial Cable

A 50-ohm cable used for digital transmission. Used in 10Base2 and 10Base5.

Broadband Coaxial Cable

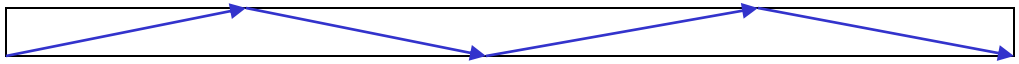
A 75-ohm cable used for analog transmission such as Cable TV.

TRANSMISSION MEDIA

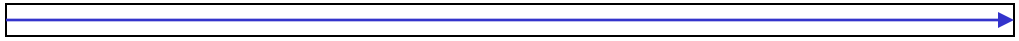
Fiber Optic Cables

Two general types are multimode and single mode.

In multimode, light is reflected internally. Light source is an LED.



In single mode, the light propagates in a straight line. Light source come from expensive laser diodes. Faster and longer distances as compared to multimode.



TRANSMISSION MEDIA

2. Unguided

Data is sent through the air.

Line-of-sight
Transmitter and receiver must “see” each other, such as microwave system.

Communication Satellites
A big microwave repeater in the sky. Data is broadcasted, and can be “pirated.”

Radio
Term used to include all frequency bands, such as FM, UHF, and VHF television.

ANALOG TRANSMISSION

1. Modulation

Modulating a sine wave carrier to convey data.

Amplitude Modulation (AM)

Amplitude is increased/decreased while frequency remains constant.

Frequency Modulation (FM)

Frequency is increased/decreased while amplitude remains constant.

Phase Modulation(PM)

Wave is shifted, while amplitude and frequency remains constant.

7 Application

6 Presentation

5 Session

4 Transport

3 Network

2 Data Link

1 Physical

Physical Devices& Transmission

1. Modems

A device that accepts digital signals and outputs a modulated carrier wave, and vice versa.

It is used to interconnect the digital computer to the analog telephone network.

2. RS-232 and RS-422

well known physical layer standards.

3. Encoding Schemes

4. Repeaters and Hubs

.

SONET\SDH Networks

- SONET\SDH, is used as a transport network to remove loads from other WANs.
- Synchronous Optical Network (**SONET**) is a standard developed by ANSI for fiber-optic networks.

SONET\SDH

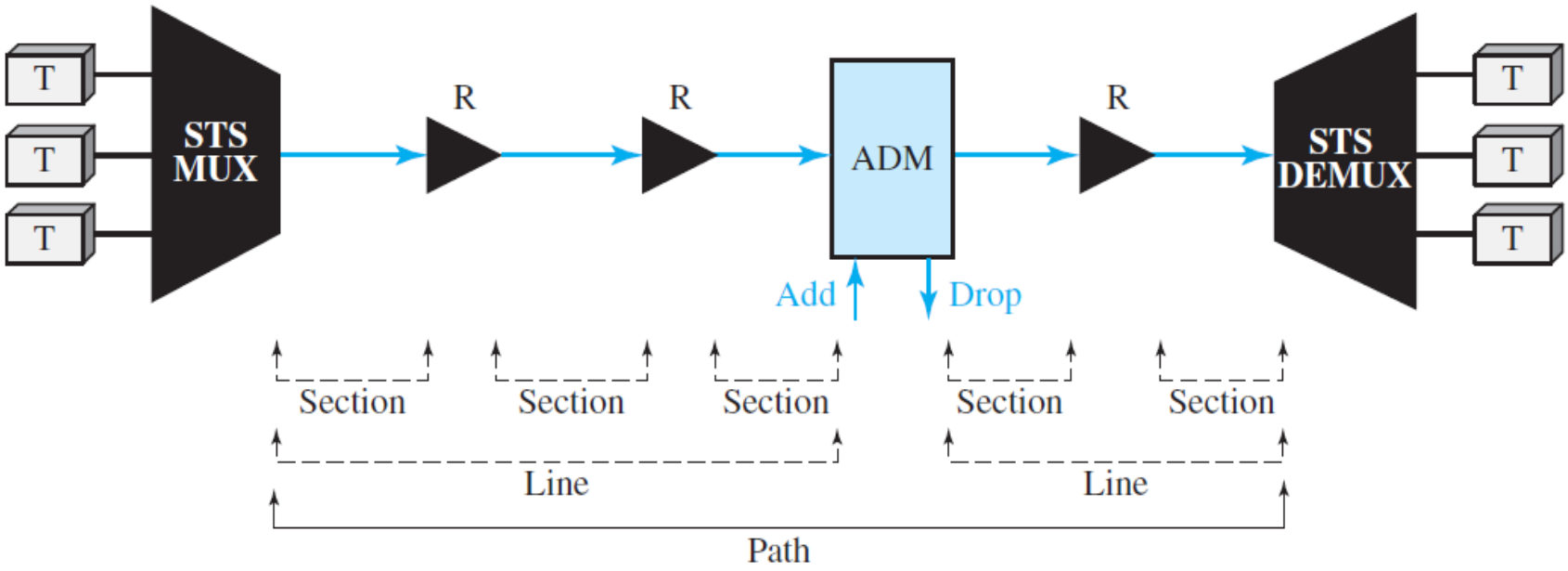
- SONET is defined as a hierarchy of signals called synchronous transport signals (**STSs**).
- SDH is defined as a similar hierarchy of signals called synchronous transfer modules (**STMs**).
- Each STS level (STS-1 to STS-192) supports a certain data rate, specified in megabits per second

SONET\SDH

- SDH is European standard network.
- SONET is American standard network.

SONET network

ADM: Add/drop multiplexer
STS MUX: Synchronous transport signal multiplexer
STS DEMUX: Synchronous transport signal demultiplexer
R: Regenerator
T: Terminal



BLUETOOTH

- Bluetooth was originally started as a project by the **Ericsson Company**.
- **Bluetooth is a wireless LAN technology designed to connect devices of different functions** such as telephones, notebooks, computers (desktop and laptop), cameras, printers, and even coffee makers when they are at a short distance from each other.

Architecture

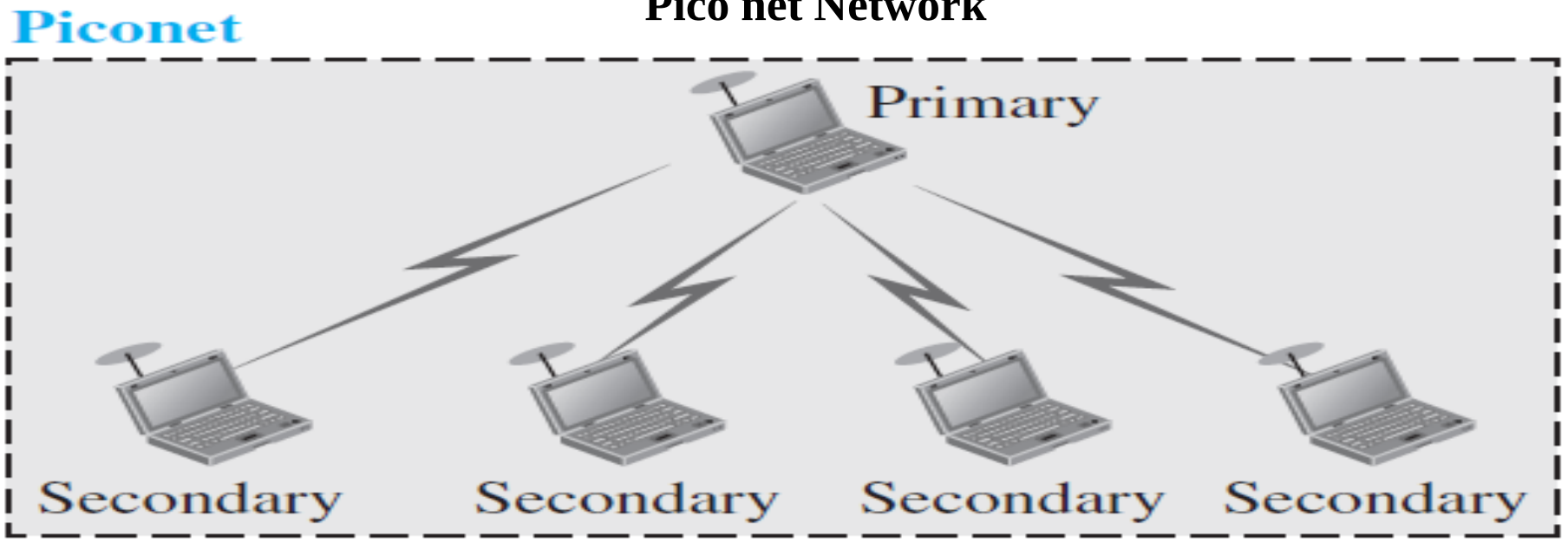
- Bluetooth defines two types of networks:

1.Pico net .

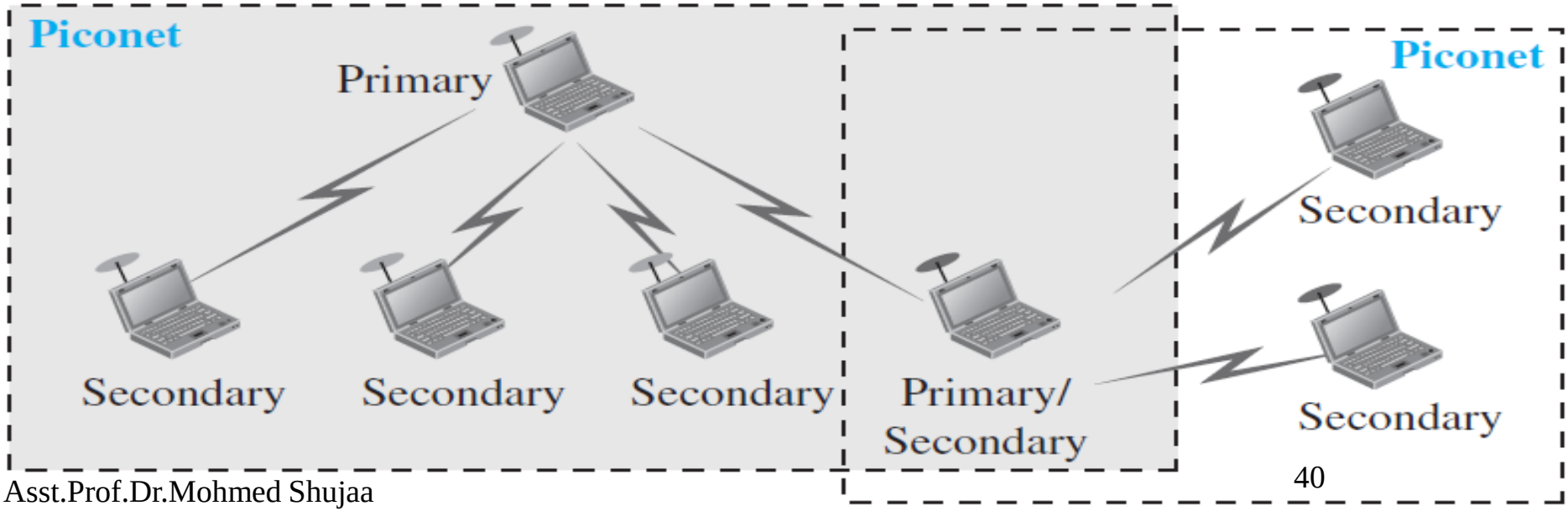
2. scatter net.

- A Bluetooth network is called a ***Pico net, or a small net. A Pico net can have up to eight*** stations, one of which is called the *primary; the rest are called secondary's*.

Pico net Network



Scatter net Network

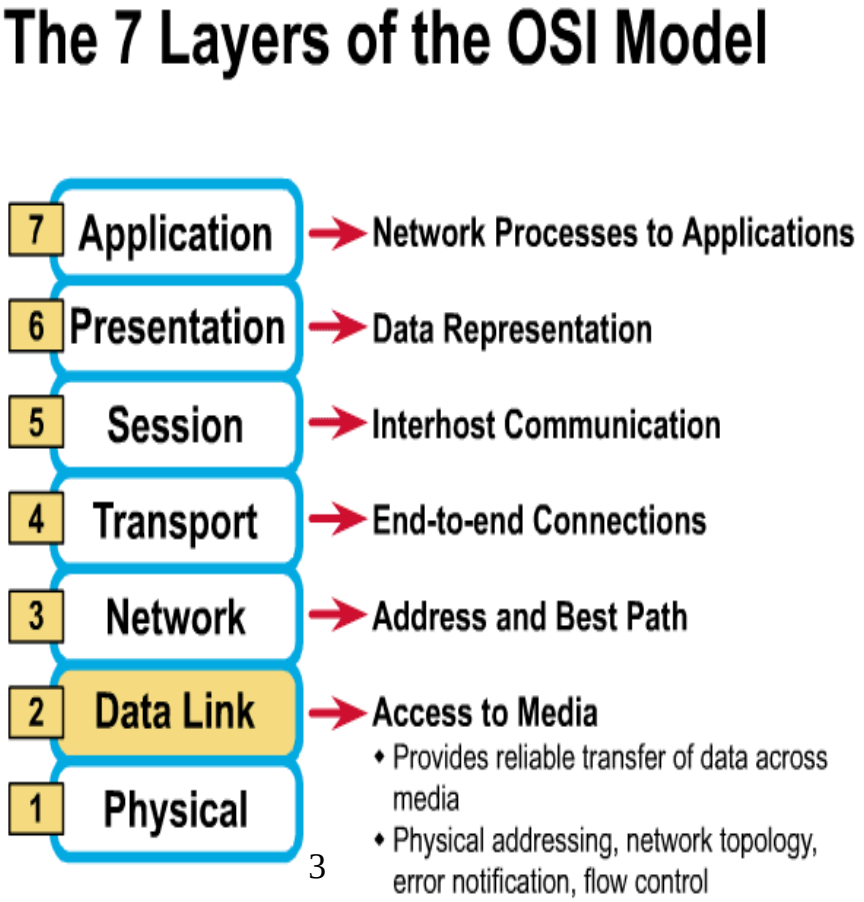


Lecture III

2. The Data Link Layer

Data Link Layer

- Places data and retrieves it from the physical layer and provides error detection capabilities



3

Data Link Layer Design Issues

data-link layer has responsibility of transferring datagram from one node to physically adjacent node over a link

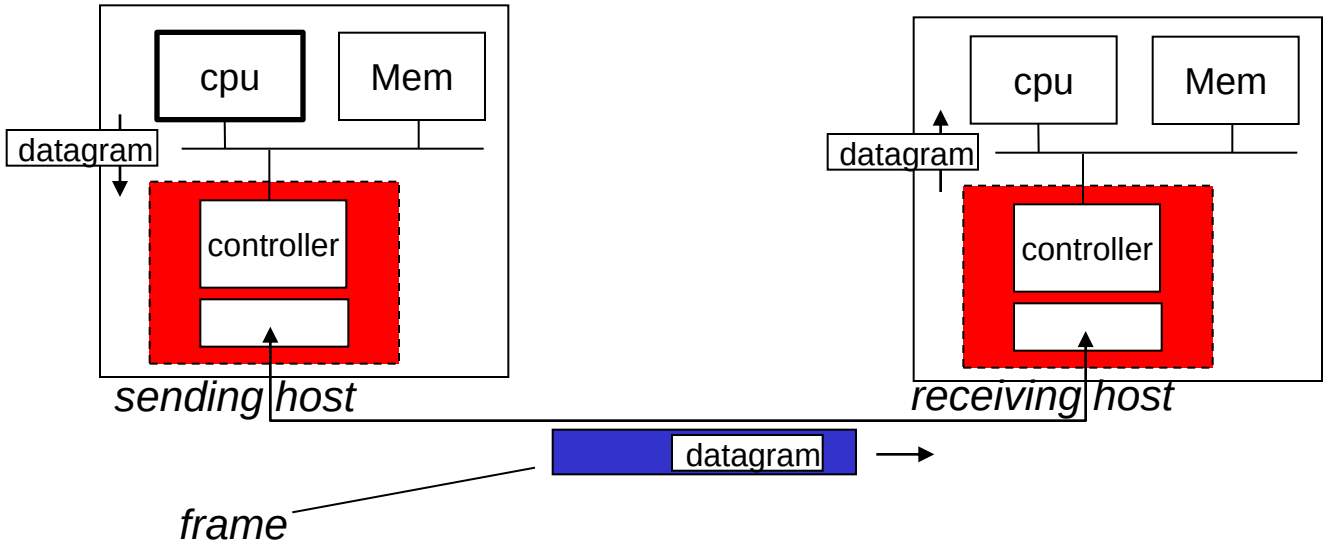
Services Provided:

- Framing
- Error Control
- Flow Control

Functions of the Data Link Layer

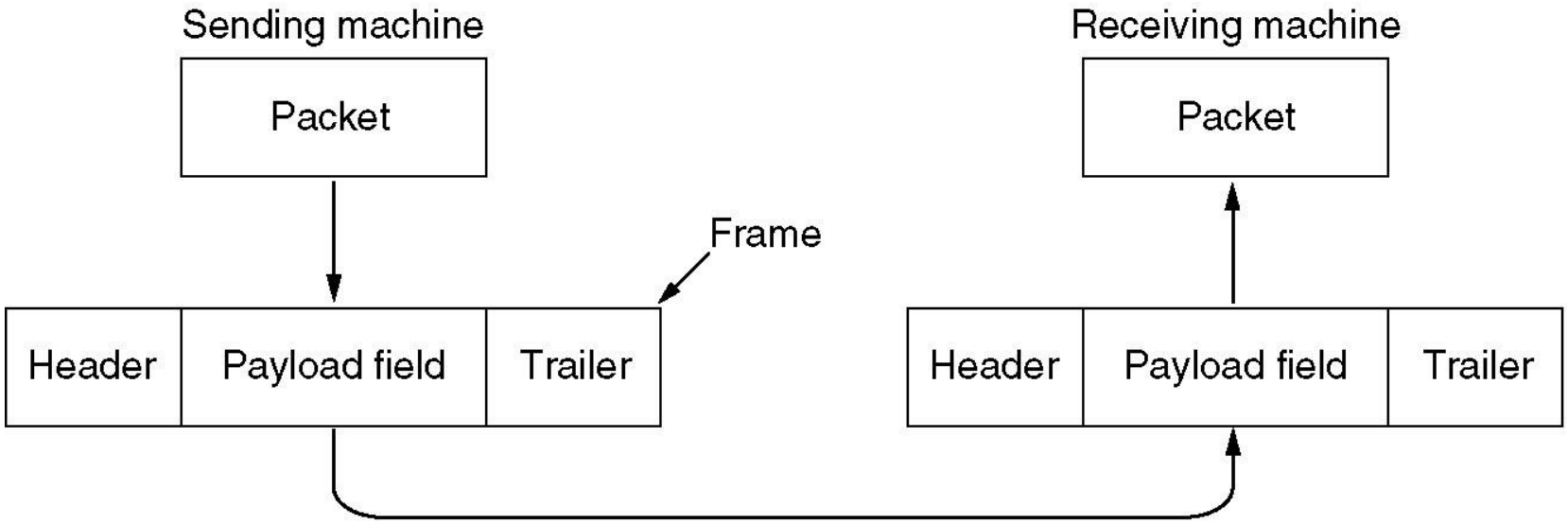
- Provide service interface to the network layer
- Dealing with transmission errors
- Regulating data flow
  - Slow receivers not swamped by fast senders

Adaptors communicating

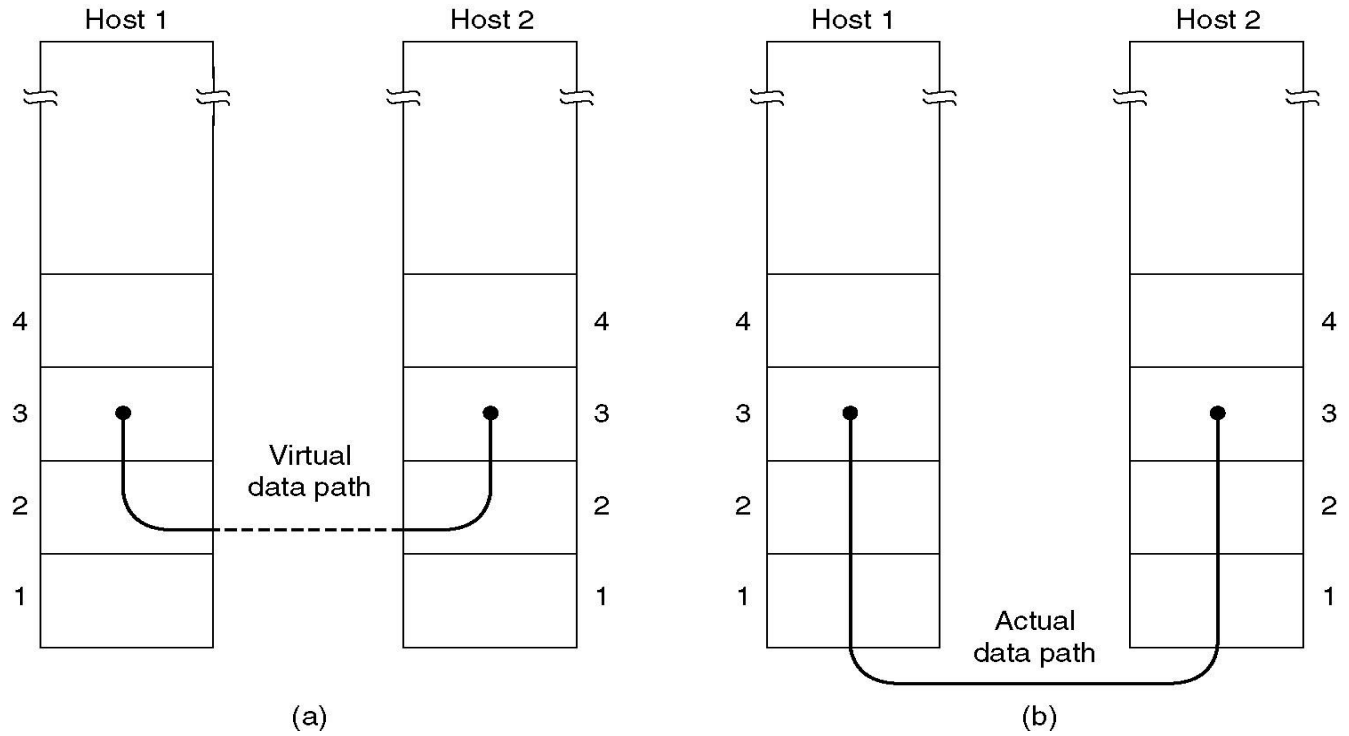


- sending side:
  - encapsulates datagram in frame
  - adds error checking bits, rdt, flow control, etc.
- receiving side
  - looks for errors, rdt, flow control, etc
  - extracts datagram, passes to upper layer at receiving side

Functions of the Data Link Layer



Services Provided to Network Layer



(a) Virtual communication.

(b) Actual communication.

Error Detection and Correction

- Error-Correcting Codes
- Error-Detecting Codes

Error (noise ,cable)……..bit error

frame= Bit stream (checksum), parity check

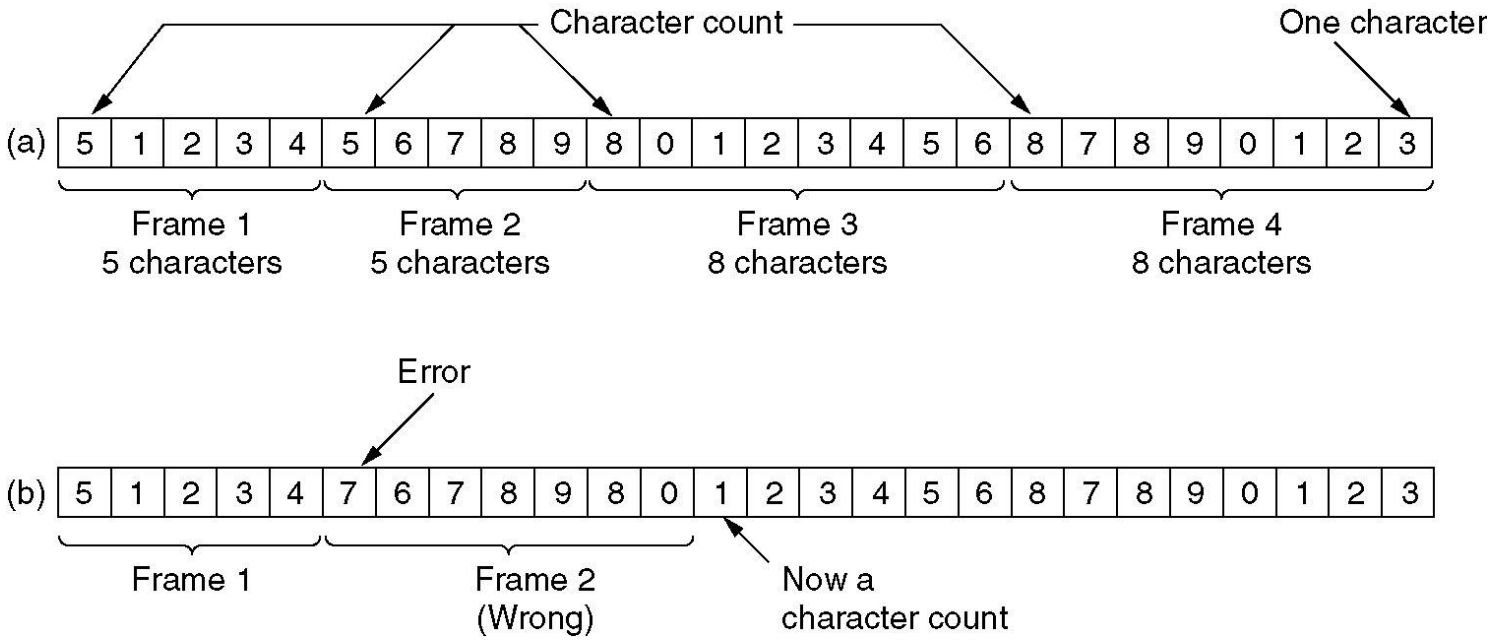
CRC field is used to detect errors.

functions:

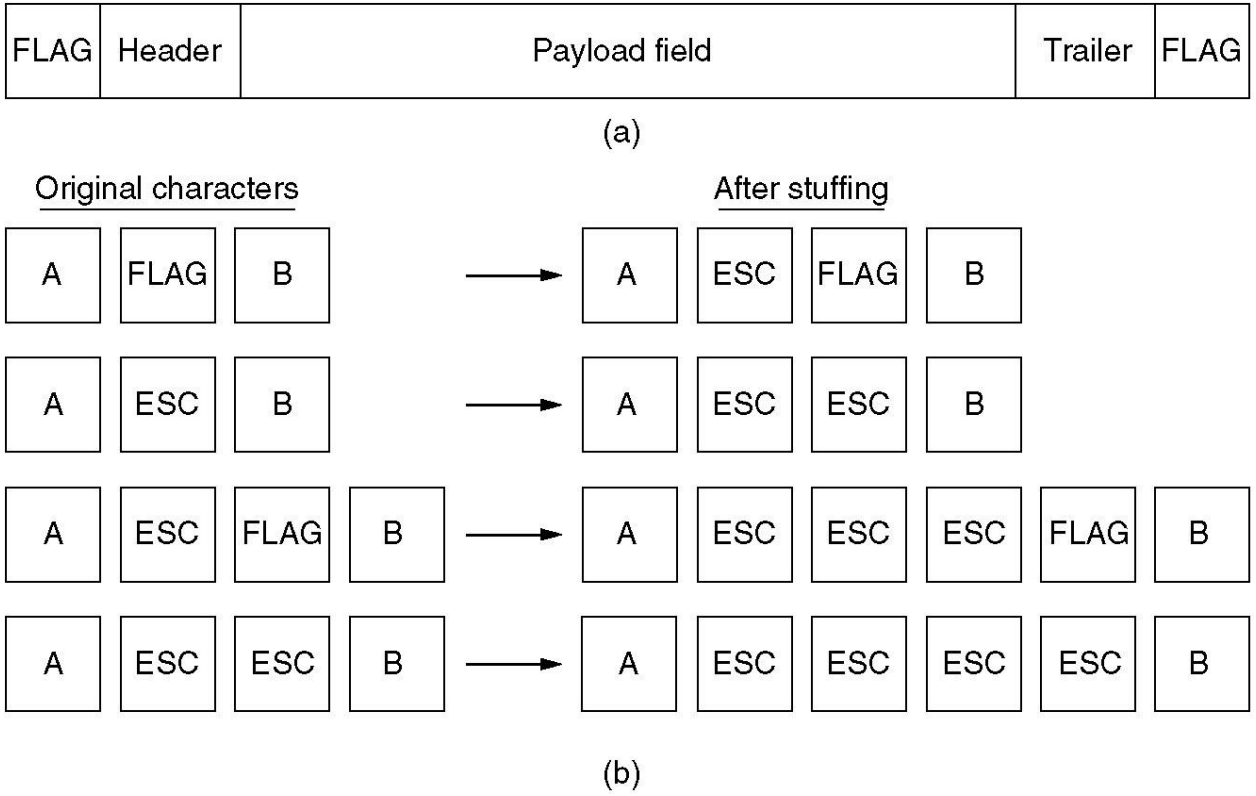
1. Flag byte
2. stuffing

Framing

A character stream. (a) Without errors.
(b) With one error.



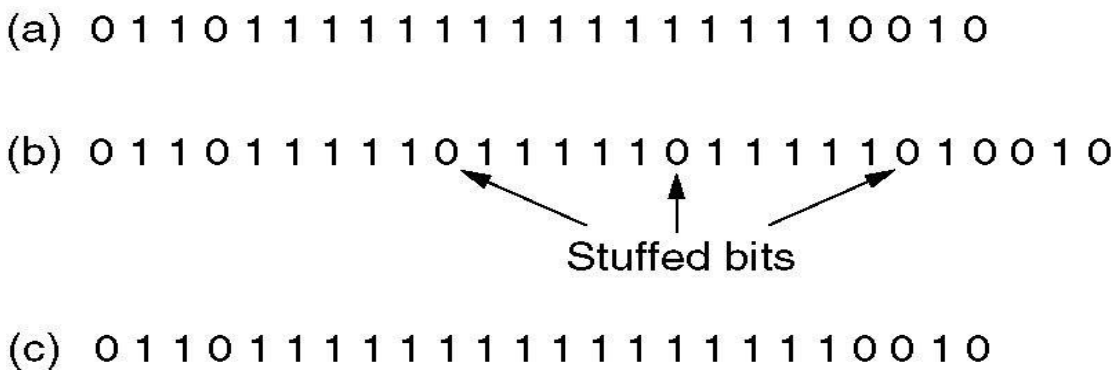
Framing (flag byte)



(a) A frame delimited by flag bytes.

(b) Four examples of byte sequences before and after stuffing.

Framing (stuffing)



Bit stuffing

(a) The original data.

(b) The data as they appear on the line.

(c) The data as they are stored in receiver's memory after de stuffing

(ppp) protocol is responsible for stuffing operation.

Lecture IV

Data link protocoles

Example Data Link Protocols

- HDLC – High-Level Data Link Control
- EDC= Error Detection and Correction bits (redundancy)
- D = Data protected by error checking, may include header fields

- Error detection not 100% reliable!
  - protocol may miss some errors, but rarely
  - larger EDC field yields better detection and correction

1.Elementary Data Link Protocols

- An Unrestricted Simplex Protocol (SP)
  (one direction transmitted data)
- A Simplex Stop-and-Wait Protocol(SSWP)
  (flooding control)
- A Simplex Protocol for a Noisy Channel(SPN)
  (limit send and receive between sender and receiver)
  Capacities are limited

2. Sliding Window Protocols

- A One-Bit Sliding Window Protocol(OSWP)

  (1- assign variable 2- define frame 3- accept frame)

- A Protocol Using Go Back N protocol

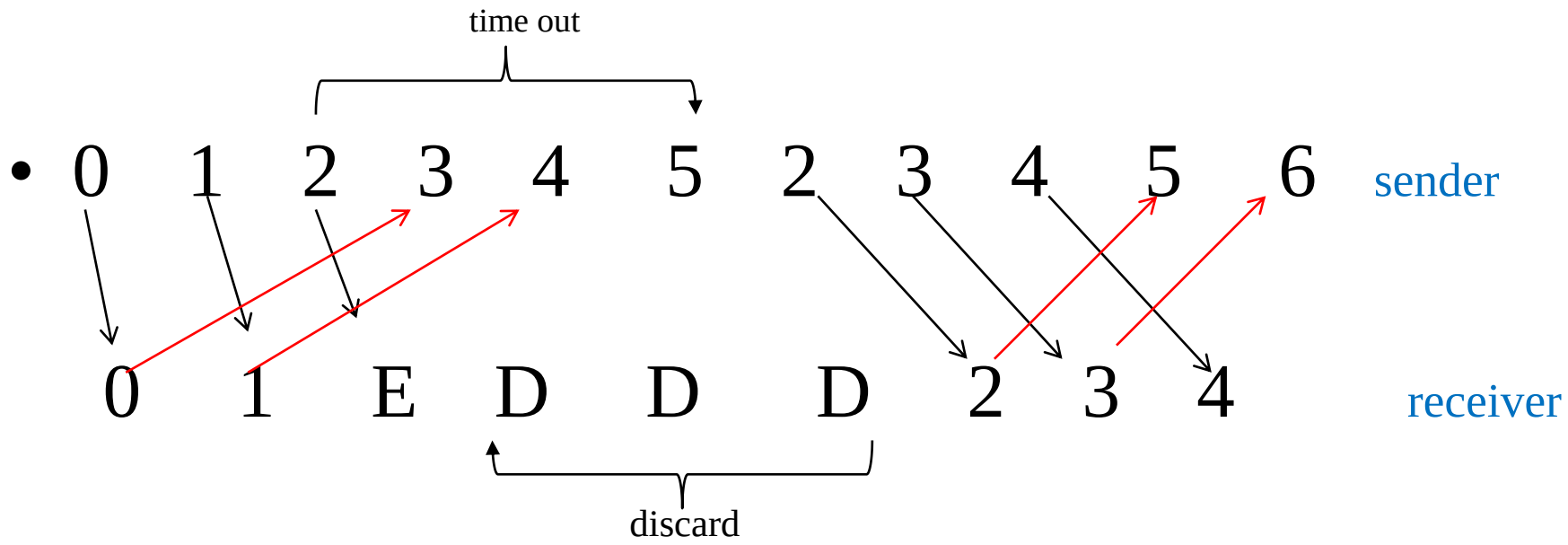
  (Discarding &Buffering)

- A Protocol Using Selective Repeat (SRP)

  (accept and buffer delay and effected frame)

  without ACK

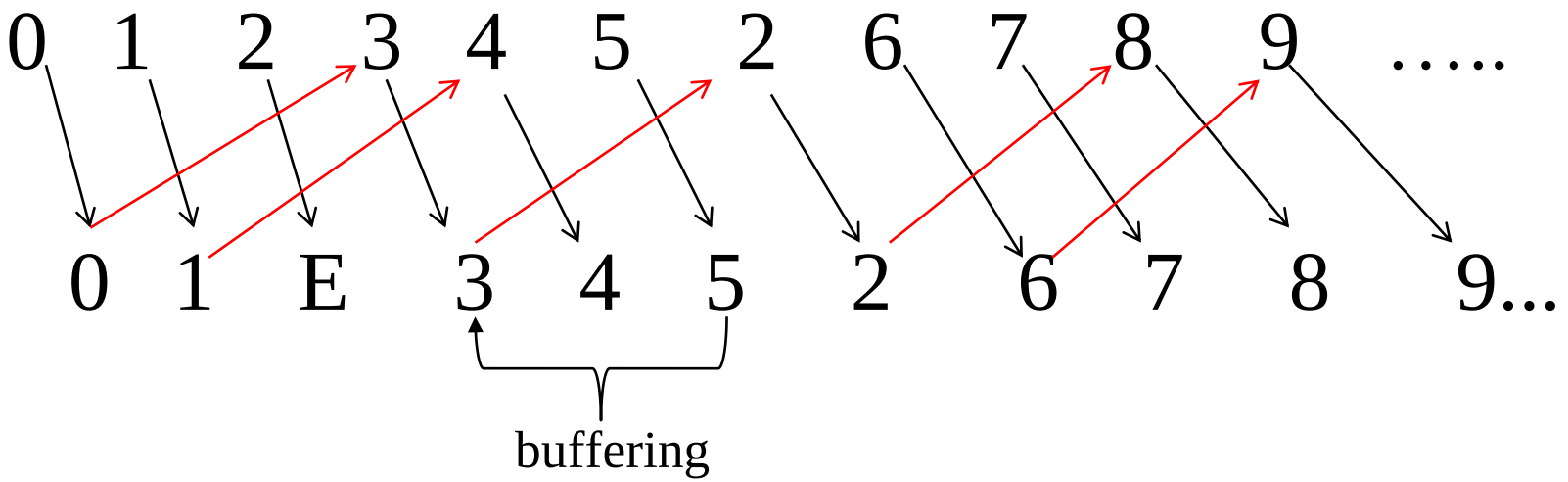
Go Back N Protocol



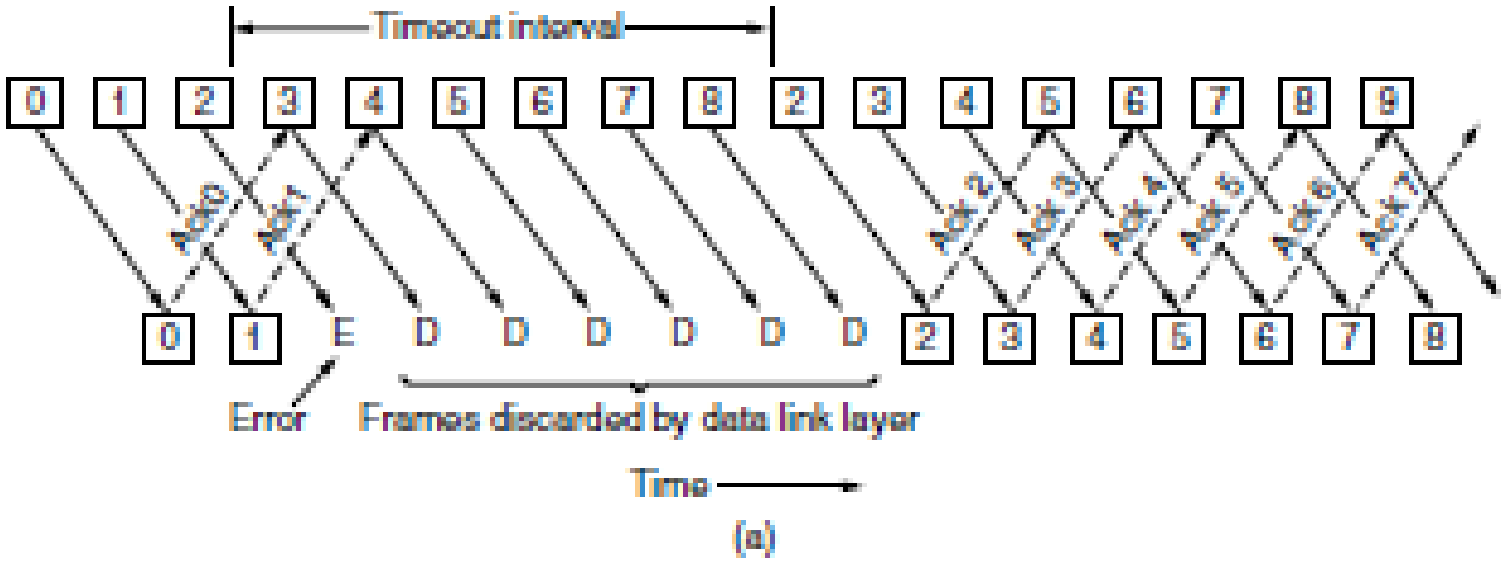
Discarding operation

Selective repeat Protocol SRP

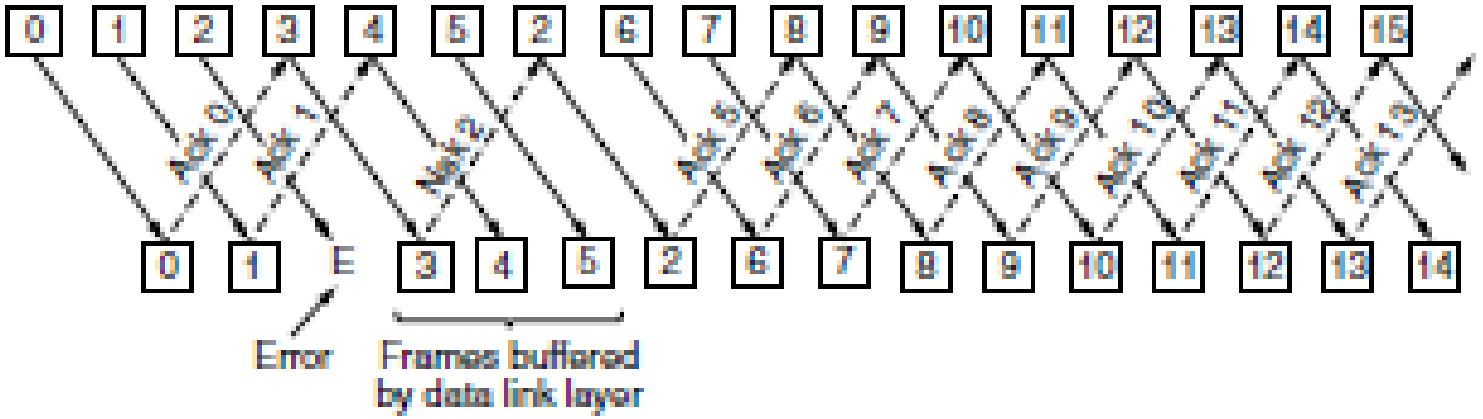
- Buffering



Sender may transmit up to max. without waiting for ACK


Timeout interval
0 1 2 3 4 5 6 7 8 2 3 4 5 6 7 8 9
Ack 0
Ack 1
Ack 2
Ack 3
Ack 4
Ack 5
Ack 6
Ack 7
0 1 E D D D D D D 2 3 4 5 6 7 8
Error
Frames discarded by data link layer
Time

(a)


0 1 2 3 4 5 2 6 7 8 9 10 11 12 13 14 15
Ack 0
Ack 1
Nak 2
Ack 5
Ack 6
Ack 7
Ack 8
Ack 9
Ack 10
Ack 11
Ack 12
Ack 13
0 1 E 3 4 5 2 6 7 8 9 10 11 12 13 14
Error
Frames buffered
by data link layer

(b)

PPP – Point ***to*** Point Protocol

- The PPP full frame format
- PPP **does not provide flow control.**
- Ability to de multiplex upward
- Error detection (no correction)(stuffing)
- Connection life, signal link, negotiator

Lecture V

MAC protocols:

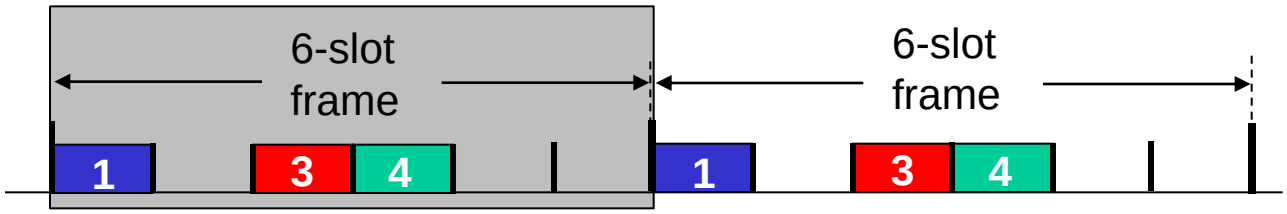
Three classes:

- *1. channel partitioning*
  - divide channel into smaller “pieces” (time slots, frequency, code)
  - allocate node for exclusive use
- *2. random access*
  - channel not divided, allow collisions
  - “recover” from collisions
- *3. “taking turns”*
  - nodes take turns, but nodes with more to send can take longer turns

Channel partitioning MAC protocols: TDMA

1.TDMA: time division multiple access

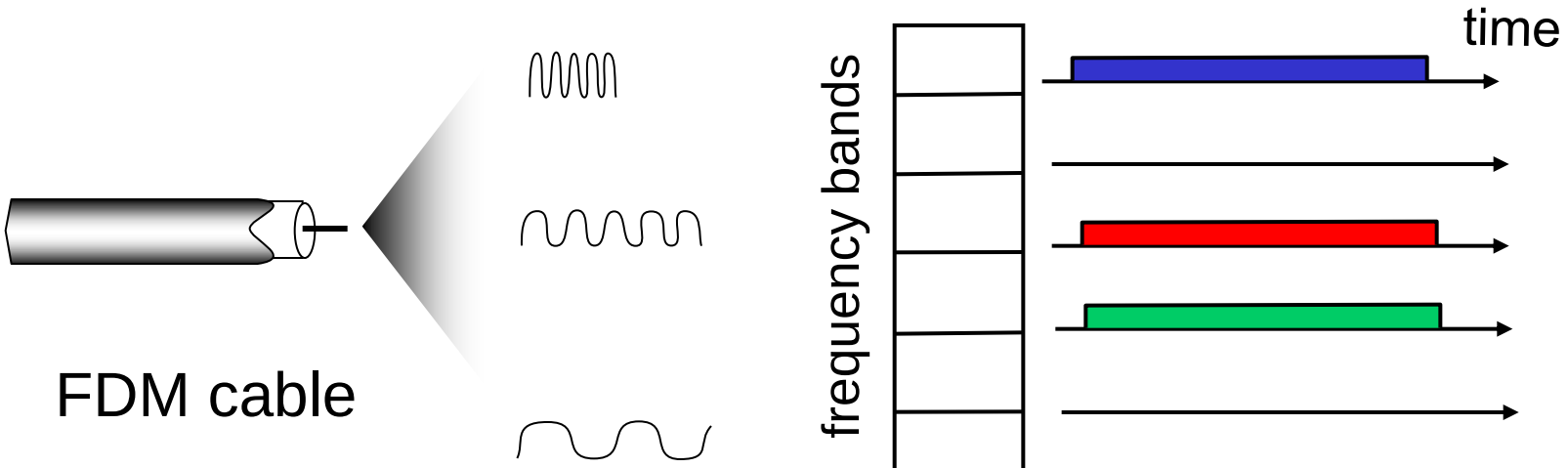
- access to channel in "rounds"
- each station gets fixed length slot (length = pkt trans time) in each round
- unused slots go idle
- example: 6-station LAN, 1,3,4 have pkt, slots 2,5,6 idle



Channel partitioning MAC protocols: FDMA

2. FDMA: frequency division multiple access

- channel spectrum divided into frequency bands
- each station assigned fixed frequency band
- unused transmission time in frequency bands go idle
- example: 6-station LAN, 1,3,4 have pkt, frequency bands 2,5,6 idle



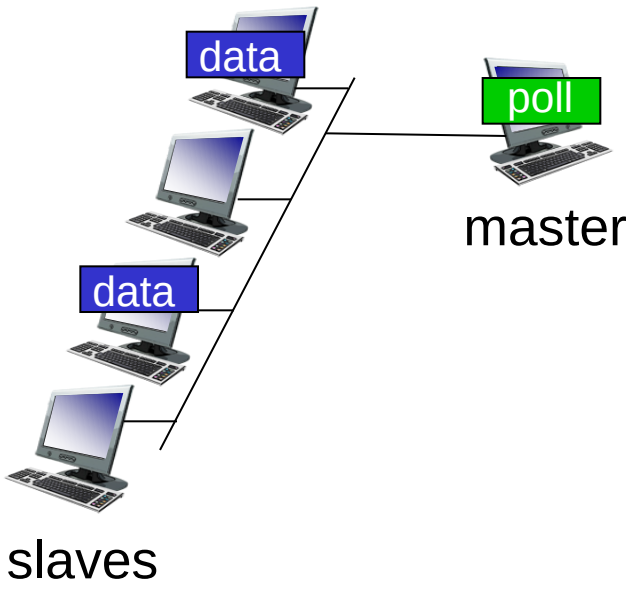
Random access protocols

- Two or more transmitting nodes ➜ “collision”,
- Random access MAC protocol specifies:
  - how to detect collisions
  - how to recover from collisions (e.g., via delayed retransmissions)
- examples of random access MAC protocols:
  1. slotted ALOHA
  2. ALOHA
  3. CSMA, CSMA/CD, CSMA/CA

“Taking turns” MAC protocols

1.polling:

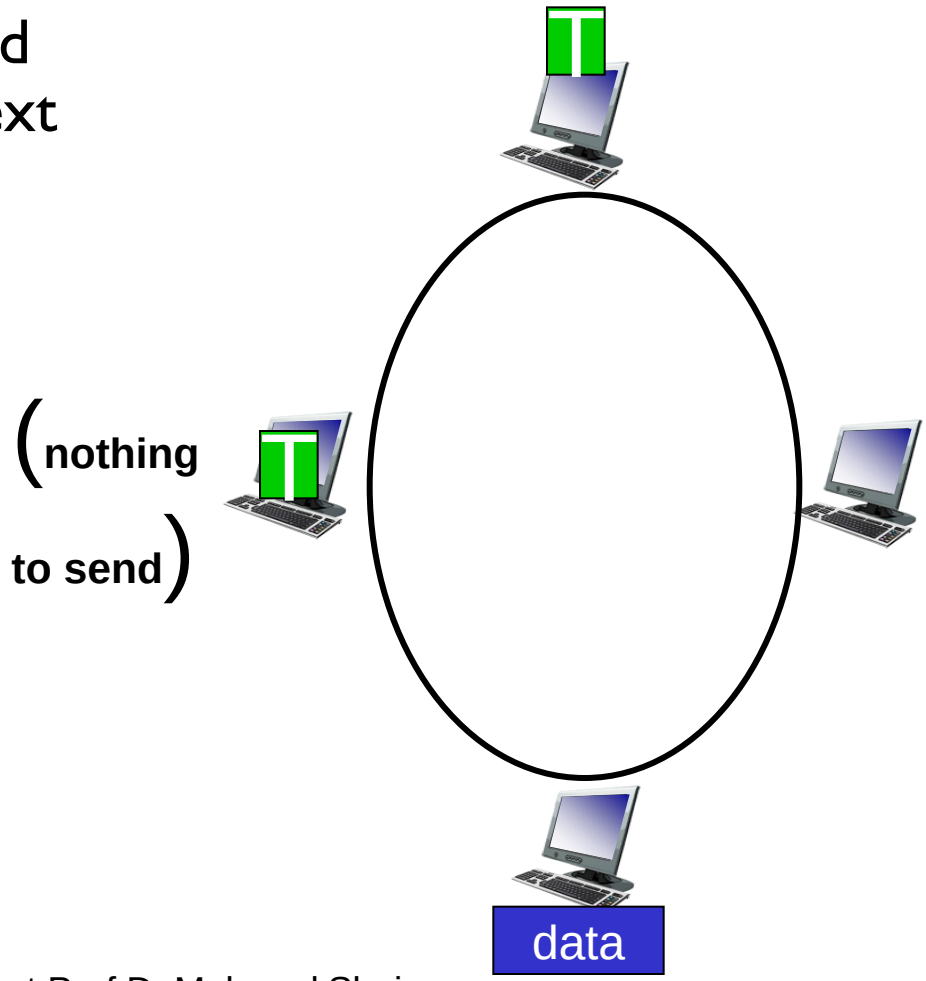
- master node “invites” slave nodes to transmit in turn
- typically used with “dumb” slave devices



"Taking turns" MAC protocols

2. Token passing:

- control token passed from one node to next sequentially.
- token message



CSMA (carrier sense multiple access)

CSMA: listen before transmit:

if channel sensed idle: transmit entire frame

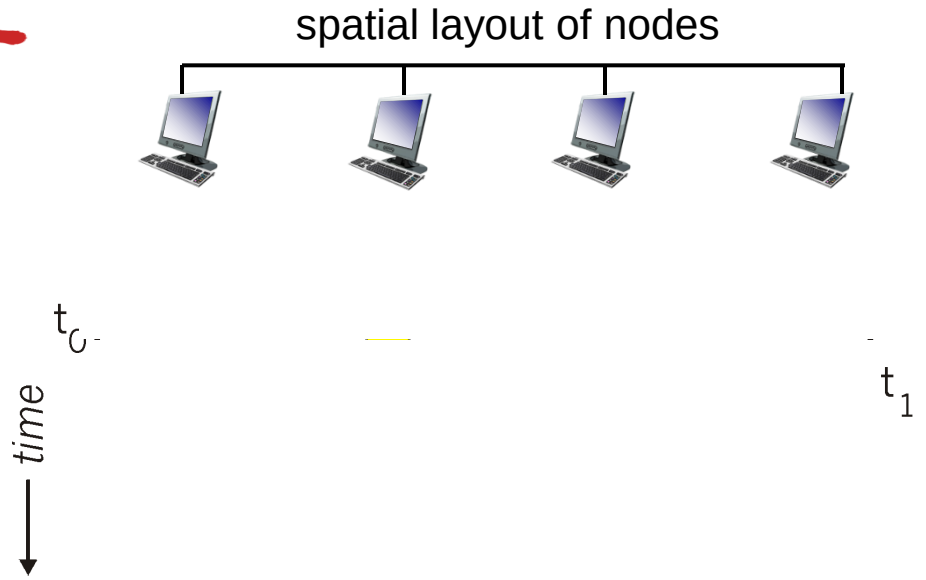
- if channel sensed busy, defer transmission

- human analogy: don't interrupt others!

CSMA collisions

- Collisions *can* still occur: propagation delay means two nodes may not hear each other's transmission

- Collision: entire packet transmission time wasted



CSMA/CD (collision detection)

CSMA/CD: carrier sensing, deferral as in CSMA

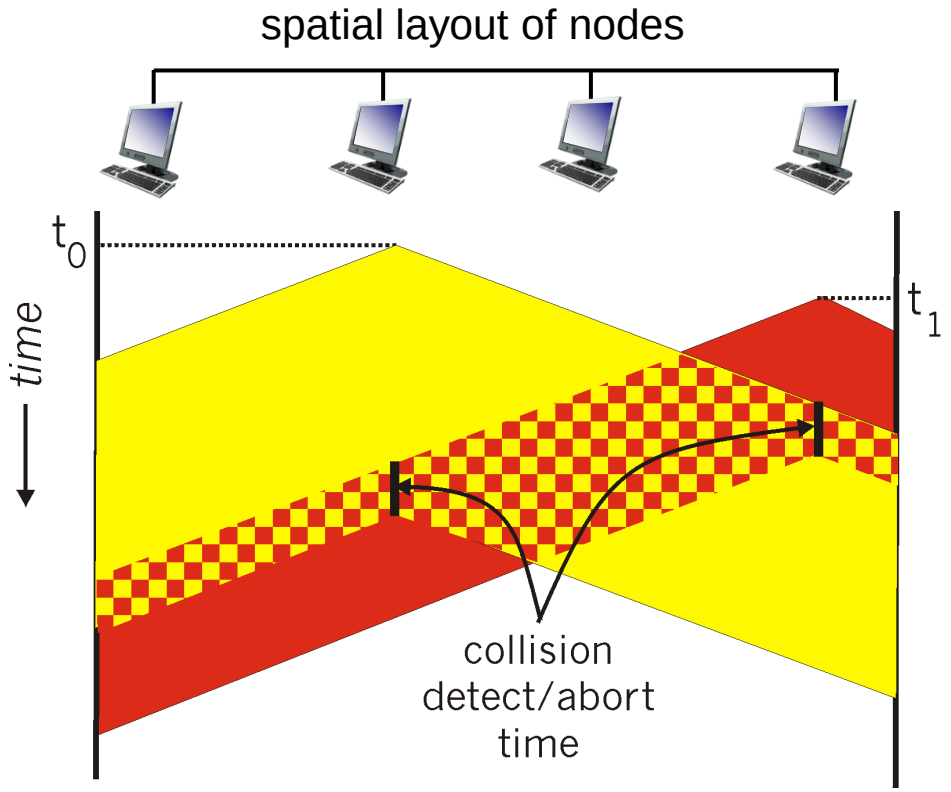
- collisions *detected* within short time
- colliding transmissions aborted, reducing channel wastage

❖ **Collision detection:**

- easy in wired LANs: measure signal strengths, compare transmitted, received signals
- Difficult in wireless LANs: received signal strength overwhelmed by local transmission strength

❖ **human analogy**: the polite conversationalist

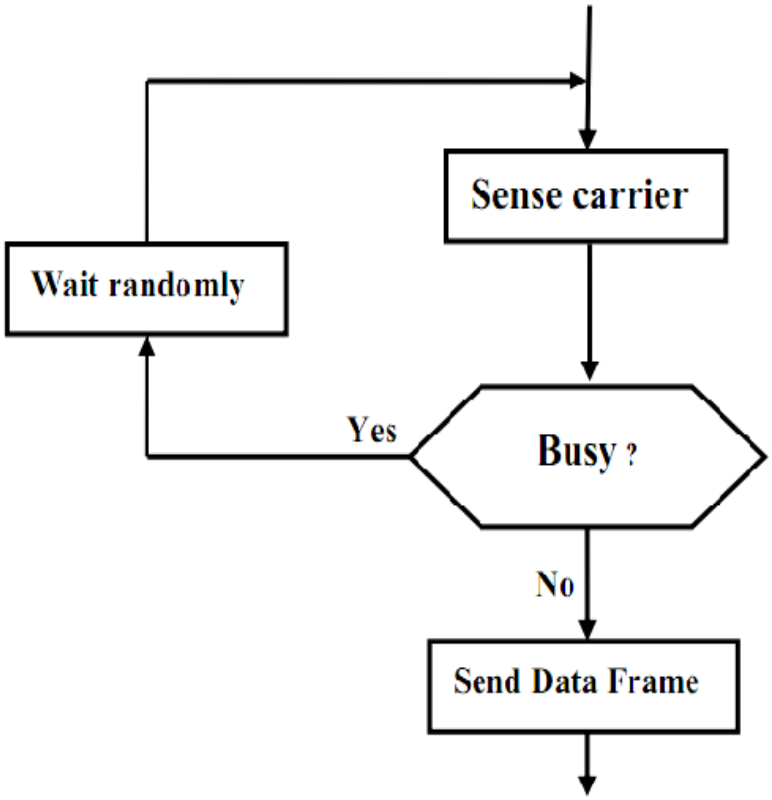
CSMA/CD (collision detection)



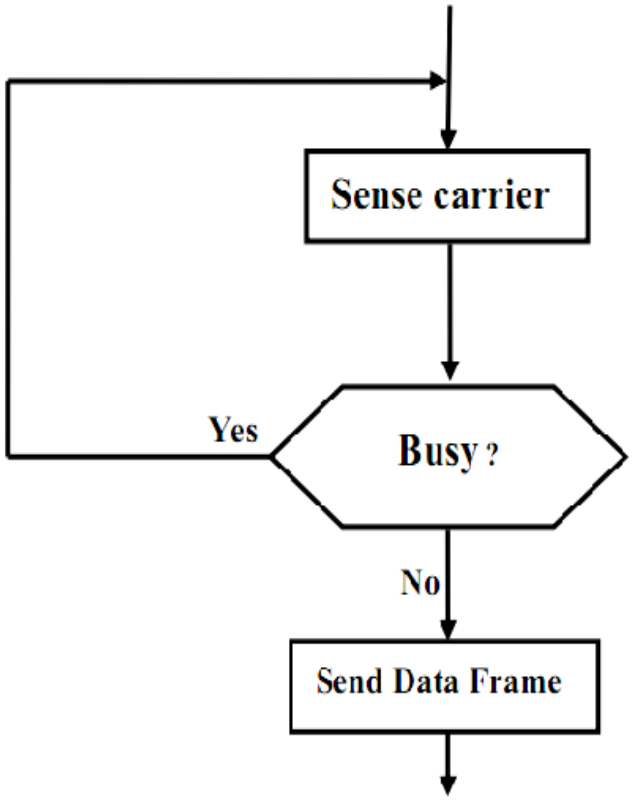
Ethernet CSMA/CD algorithm

1. NIC receives datagram from network layer, creates frame
2. If NIC senses channel idle, starts frame transmission. If NIC senses channel busy, waits until channel idle, then transmits.
3. If NIC transmits entire frame without detecting another transmission, NIC is done with frame !
4. If NIC detects another transmission while transmitting, aborts and sends jam signal
5. After aborting, NIC enters *binary (exponential) backoff:*

CSMA/CD

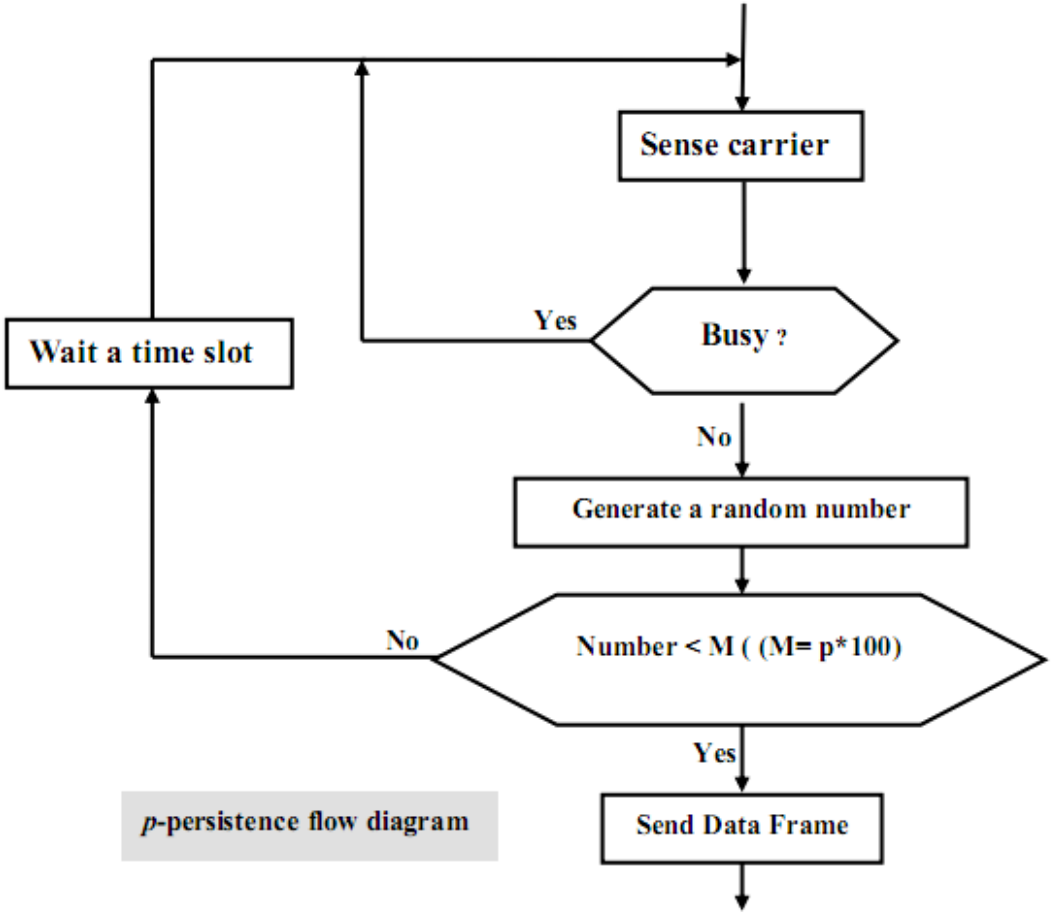


Non-persistence flow



1-persistence flow diagram

CSMA/CD



p-persistence flow diagram

CSMA/CD efficiency

- T_{prop} = max prop delay between 2 nodes in LAN
- t_{trans} = time to transmit max-size frame

$$efficiency = \frac{1}{1 + 5t_{prop}/t_{trans}}$$

- efficiency goes to = 1
  - as t_{prop} goes to 0
  - as t_{trans} goes to infinity

Lecture VI

Types of Errors
Detection
Correction

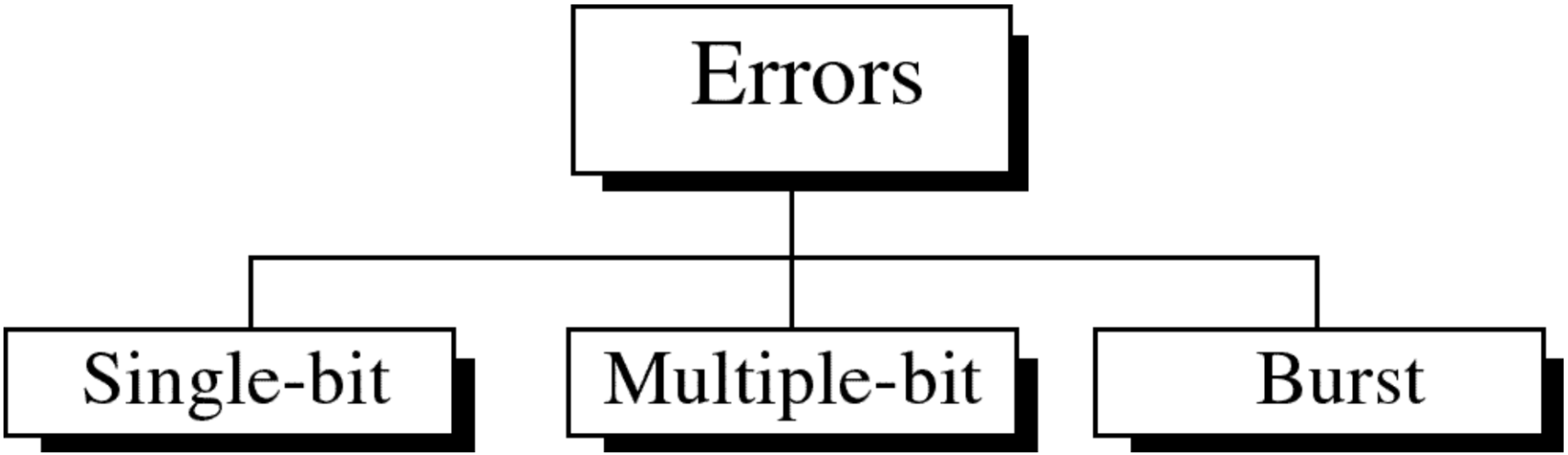
Error Detection

- Data transmission can contain errors
  - Single-bit
  - Burst errors of length n
    (n: distance between the first and last errors in data block)
- How to detect errors
  - If only data is transmitted, errors cannot be detected
    → Send more information with data that satisfies a special relationship
    → Add redundancy

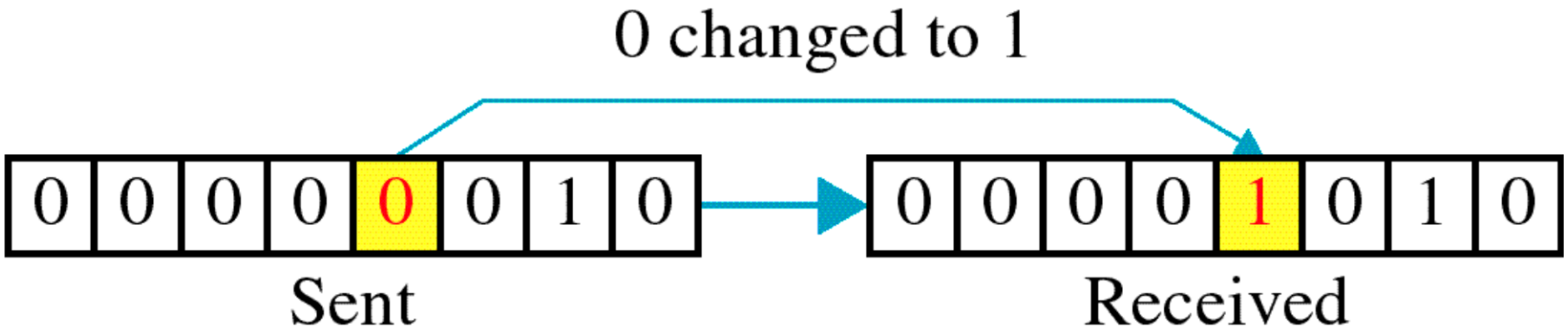
Basic concepts

- ★ Networks must be able to transfer data from one device to another with complete accuracy.
- ★ Data can be corrupted during transmission.
- ★ For reliable communication, errors must be detected and corrected.
- ★ **Error detection and correction** are implemented either at the **data link layer** or the **transport layer** of the OSI model.

Types of Errors



Single-bit error



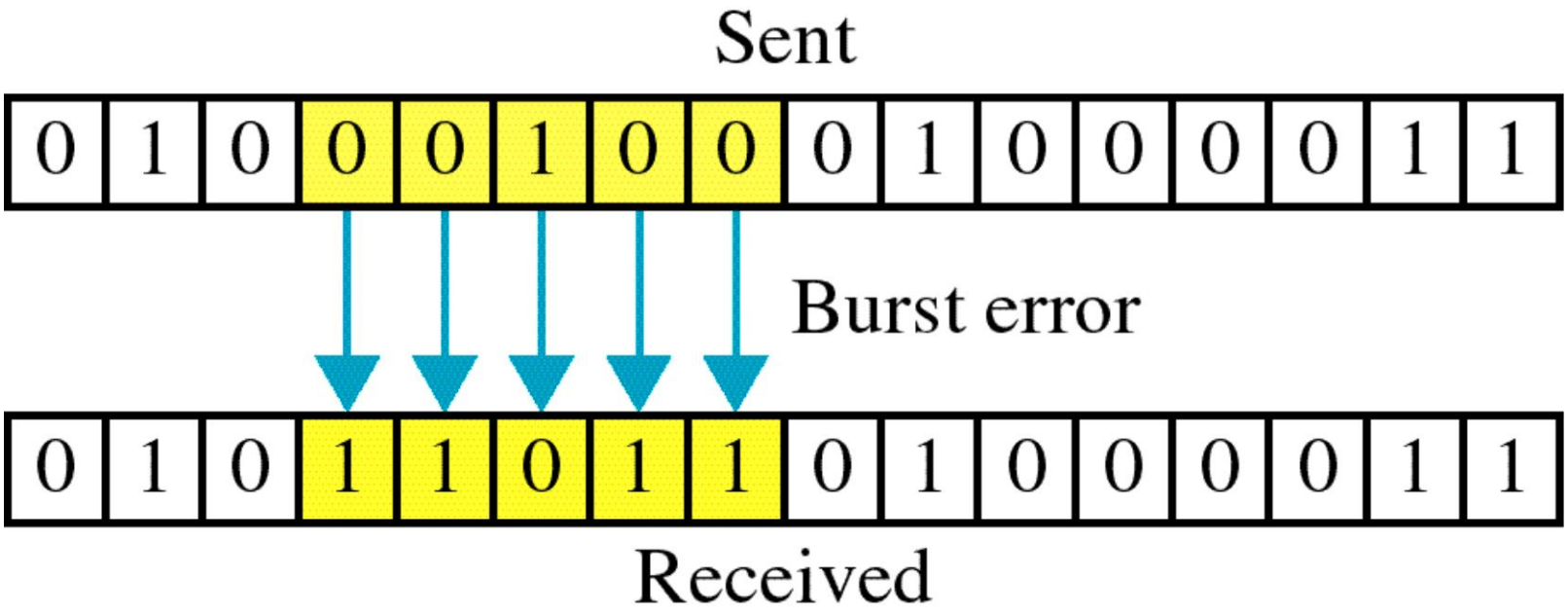
Single bit errors type of errors in **serial data transmission** because the noise must have a very short duration which is very rare. **Errors can happen in parallel transmission.**

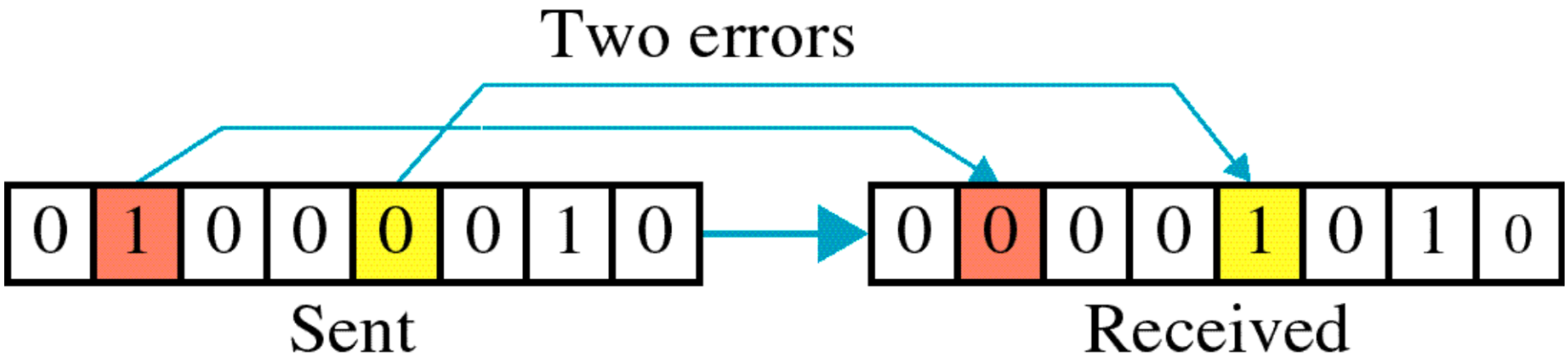
Example:

★If data is sent at 1Mbps then each bit lasts only 1/1,000,000 sec. or 1 μs.

★For a single-bit error to occur, the noise must have a duration of only 1 μs, which is very rare.

Burst error


Sent
0 1 0 0 0 1 0 0 0 1 0 0 0 0 1 1
Burst error
0 1 0 1 1 0 1 1 0 1 0 0 0 0 1 1
Received


Two errors
0 1 0 0 0 0 1 0
Sent
0 0 0 0 1 0 1 0
Received

burst error means that two or more bits in the data unit have changed from 1 to 0 or from 0 to 1.

Burst errors does not necessarily **mean that the errors occur in consecutive bits**, the length of the burst is measured from the first corrupted bit to the last corrupted bit. Some bits in between may not have been corrupted.

★**Burst error happens in serial transmission** since the duration of noise is normally longer than the duration of a bit.

★The number of bits affected depends on the data rate and duration of noise.

Example:

➔If data is sent at rate = 1Kbps then a noise of 1/100 sec can affect 10 bits.(1/100*1000)

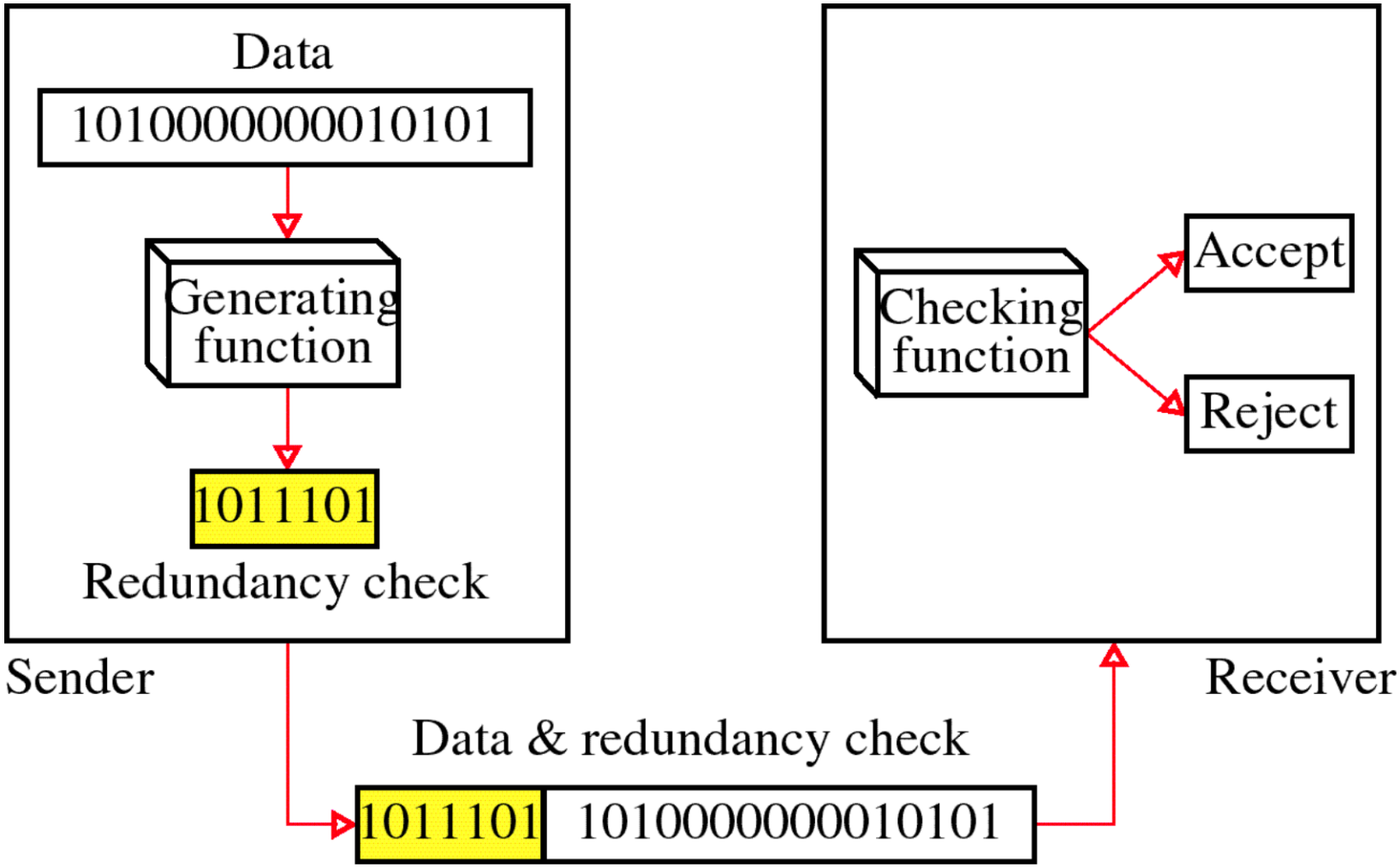
➔If same data is sent at rate = 1Mbps then a noise of 1/100 sec can affect 10,000 bits.($1/100*10^6$)

Error detection

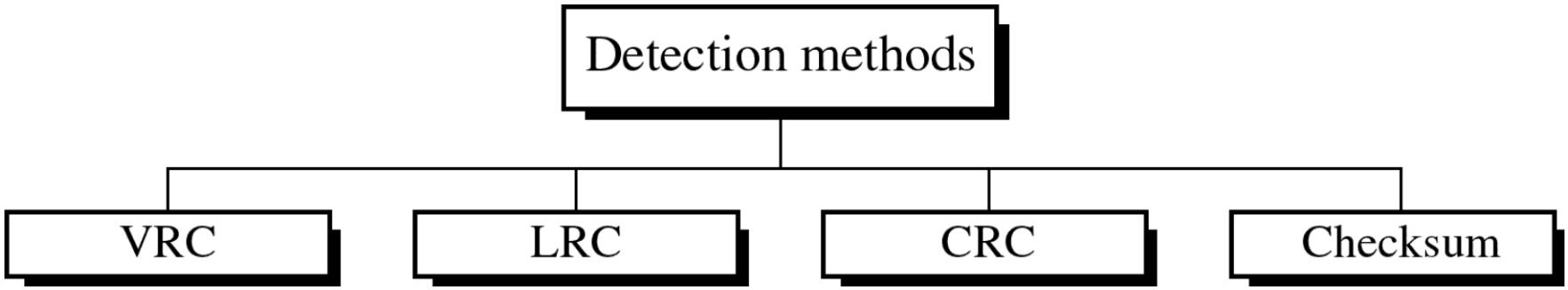
Error detection means to decide whether the received data is correct or not without having a copy of the original message.

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

Redundancy


Data
1010000000010101
Generating function
1011101
Redundancy check
Sender
Checking function
Accept
Reject
Receiver
Data & redundancy check
1011101
1010000000010101

Four types of redundancy checks are used in data communications



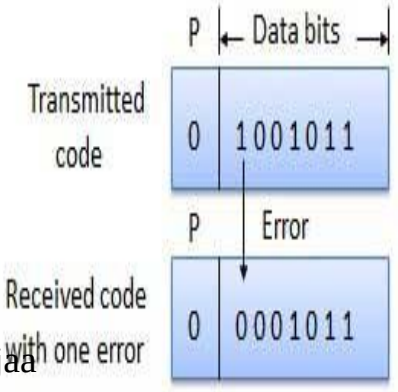
Error detection and correction

Error detection and correction has techniques allow the receiver to detect and possibly correct packet bit errors. For now, we need only know that these techniques require that extra bits (beyond the bits of original data to be transferred) be sent from the sender to the receiver.

There is three techniques for detecting error:

1. Parity checks
2. Checksum
3. Cyclic redundancy checks

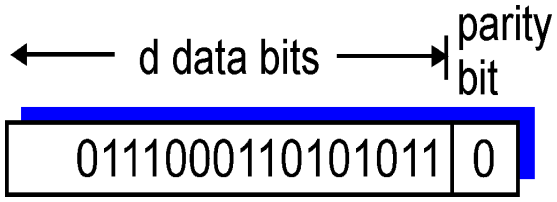
Parity checks: Parity checking at the receiver can detect the presence of an error if the parity of the receiver signal is different from t he expected parity.



Parity checking (error detection)

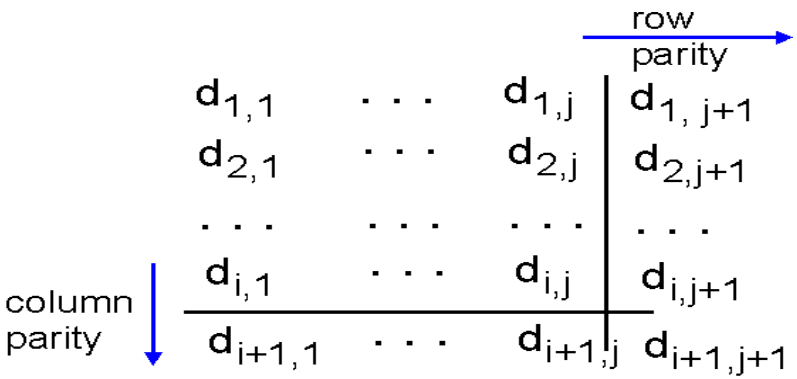
single bit parity:

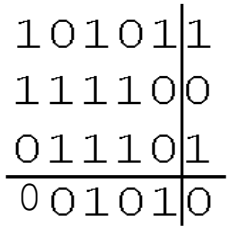
- *d*etect single bit errors



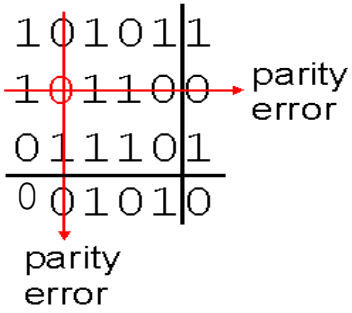
two-dimensional bit parity:

- detect and correct single bit errors





no errors



correctable single bit error

- **One-dimension Parity Check:** Blocks of data from the source are subjected to a check bit or Parity bit generator form, where a parity of 1 is added to the block if it contains an odd number of 1's (ON bits) and 0 is added if it contains an even number of 1's. At the receiving end the parity bit is computed from the received data bits and compared with the received parity bit, This scheme makes the total number of 1's even, that is why it is called even parity checking. Considering a 4-bit word, different combinations of the data words and the corresponding code words are given in Table below.

Decimal value	Data Block	Parity bit	Code word
0	0000	0	00000
1	0001	1	00011
2	0010	1	00101
3	0011	0	00110
4	0100	1	01001
5	0101	0	01010
6	0110	0	01100
7	0111	1	01111
8	1000	1	10001
9	1001	0	10010
10	1010	0	10100
11	1011	1	10111
12	1100	0	11000
13	1101	1	11011
14	1110	1	11101
15	1111	0	11110

Example

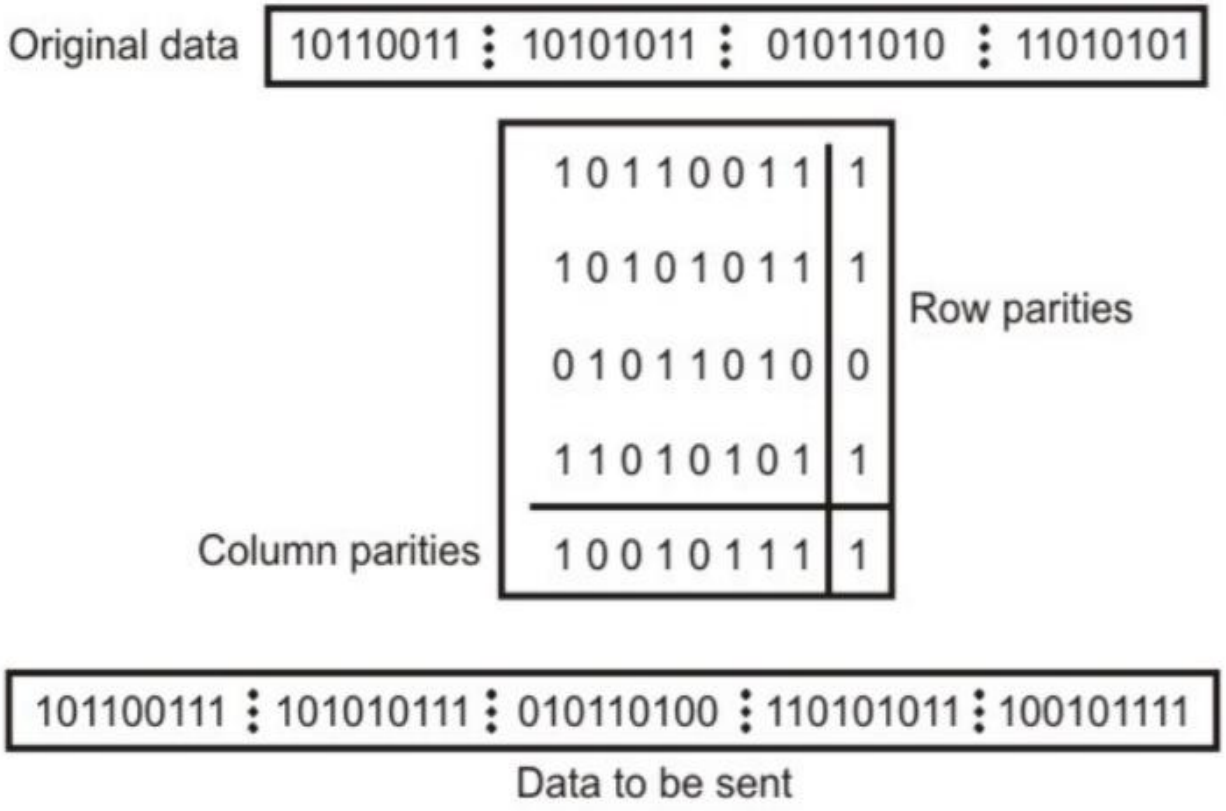
Suppose the sender wants to send the word world in ASCII the five characters are coded as

(1110111 1101111 1110010 1101100 1100100)

The following shows the actual bits sent(by use even parity check)

(11101110 11011110 11100100 11011000 11001001)

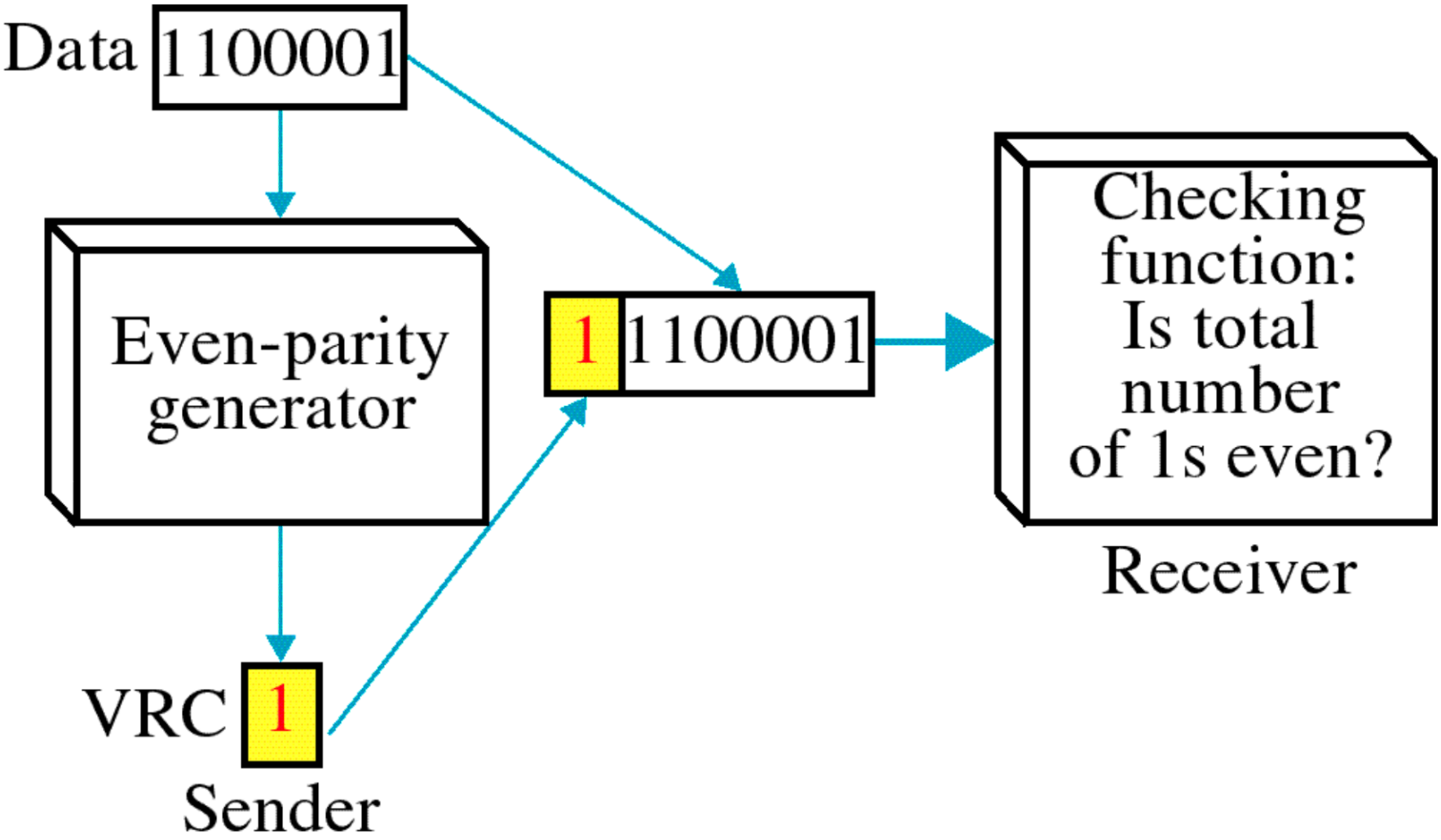
- **Two-dimension Parity Check:** Performance can be improved by using two-dimensional parity check, which organizes the block of bits in the form of a table. Parity check bits are calculated for each row, which is equivalent to a simple parity check bit. Parity check bits are also calculated for all columns then both are sent along with the data. At the receiving end these are compared with the parity bits calculated on the received data. This is illustrated in Figure below



Error Detection Methods

- **Vertical Redundancy Check (VRC)**
  - Append a single bit at the end of data block such that the number of ones is even
    → Even Parity (odd parity is similar)
    0110011 → 01100110
    0110001 → 01100011
  - VRC is also known as **Parity Check**
  - Performance:
    - Detects all odd-number errors in a data block

Vertical Redundancy Check
VRC

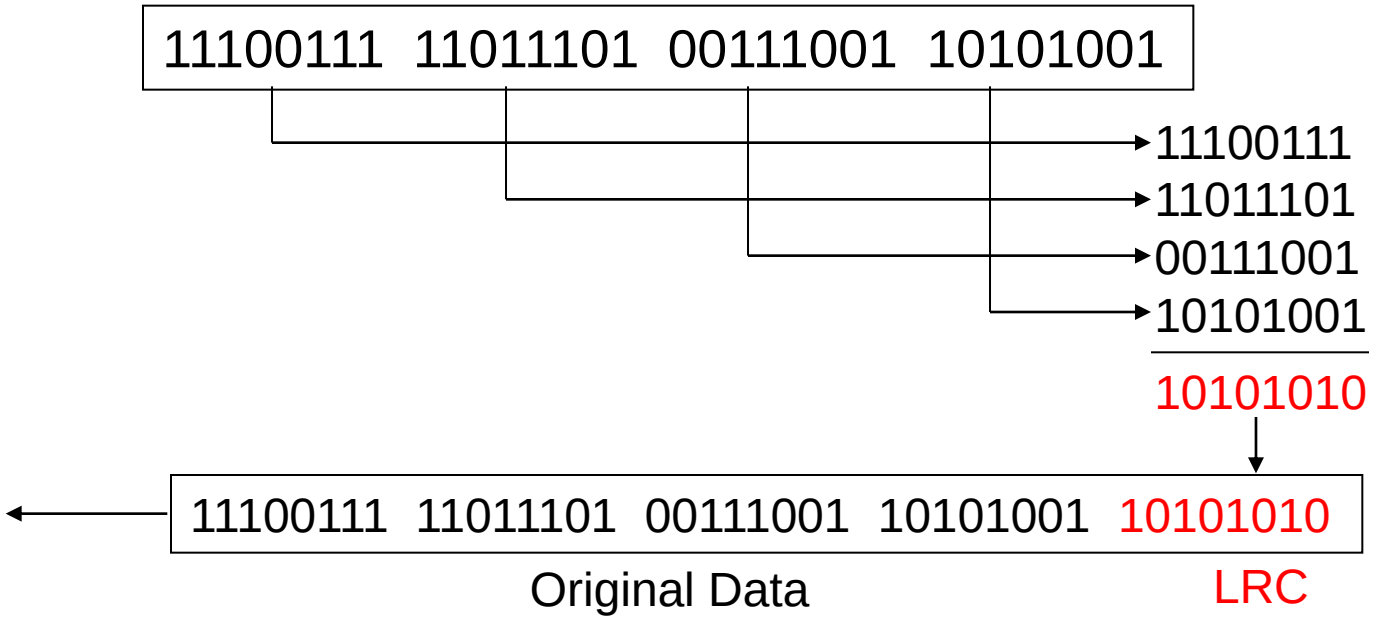


Performance

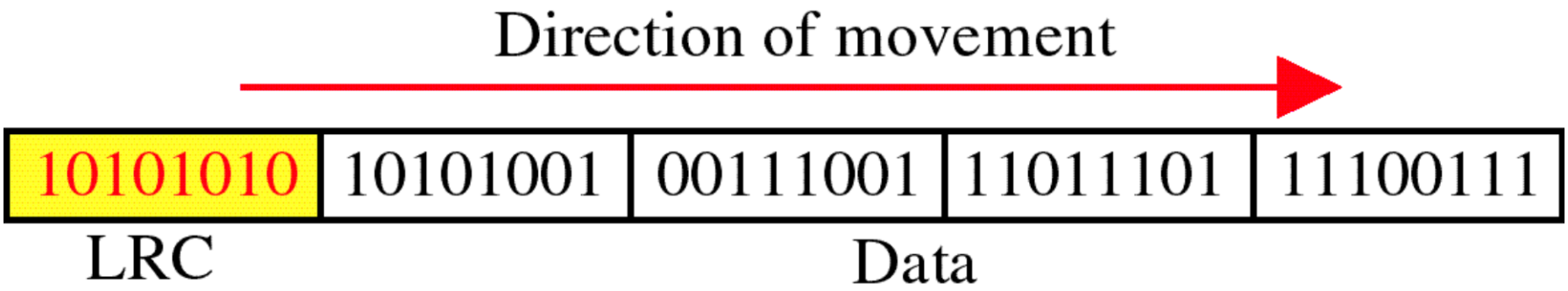
- It can detect single bit error
- It can detect burst errors only if the total number of errors is odd.
- Parity check

Error Detection Methods

- **Longitudinal Redundancy Check (LRC)**
  - Organize data into a table and create a parity for each column



Longitudinal Redundancy Check
LRC



Performance

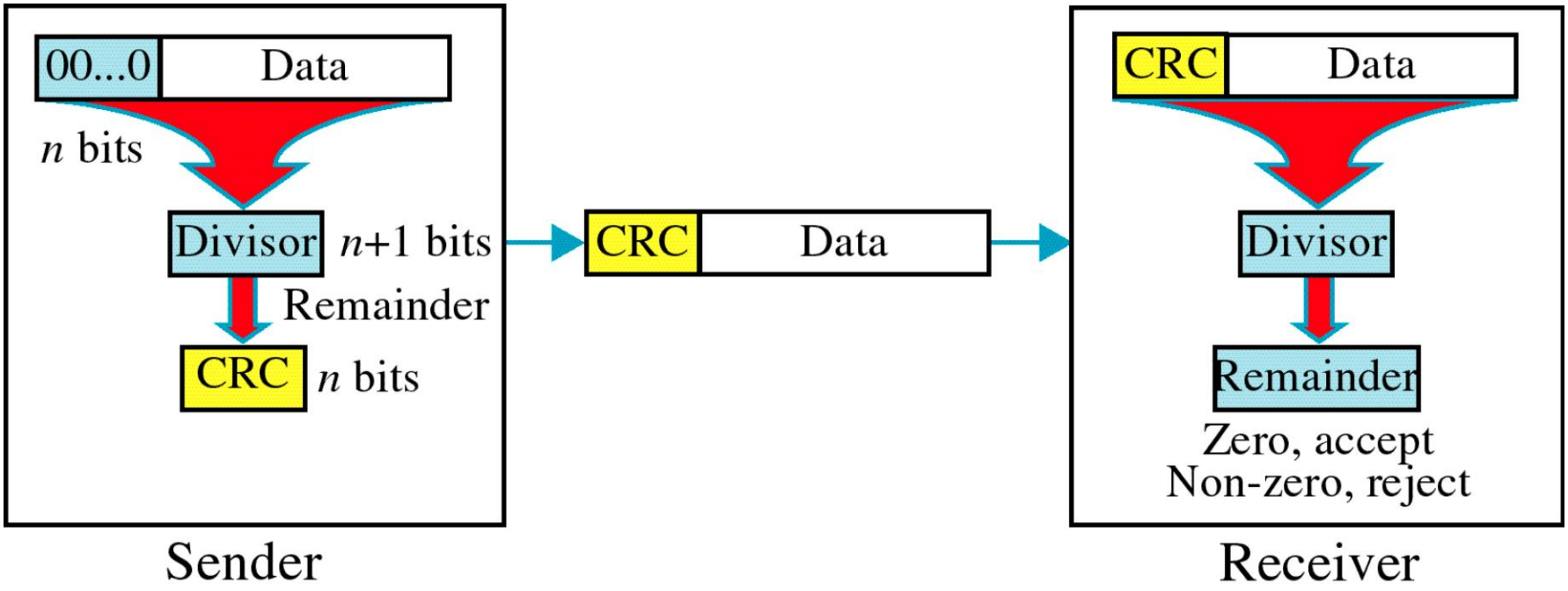
➔If two bits in one data units are damaged and two bits in exactly the same positions in another data unit are also damaged, the LRC checker will not detect an error.

Lecture VII

Types of Error

Cyclic Redundancy Check
CRC

- **Most power redundancy checker**
- **Based on binary addition and division**



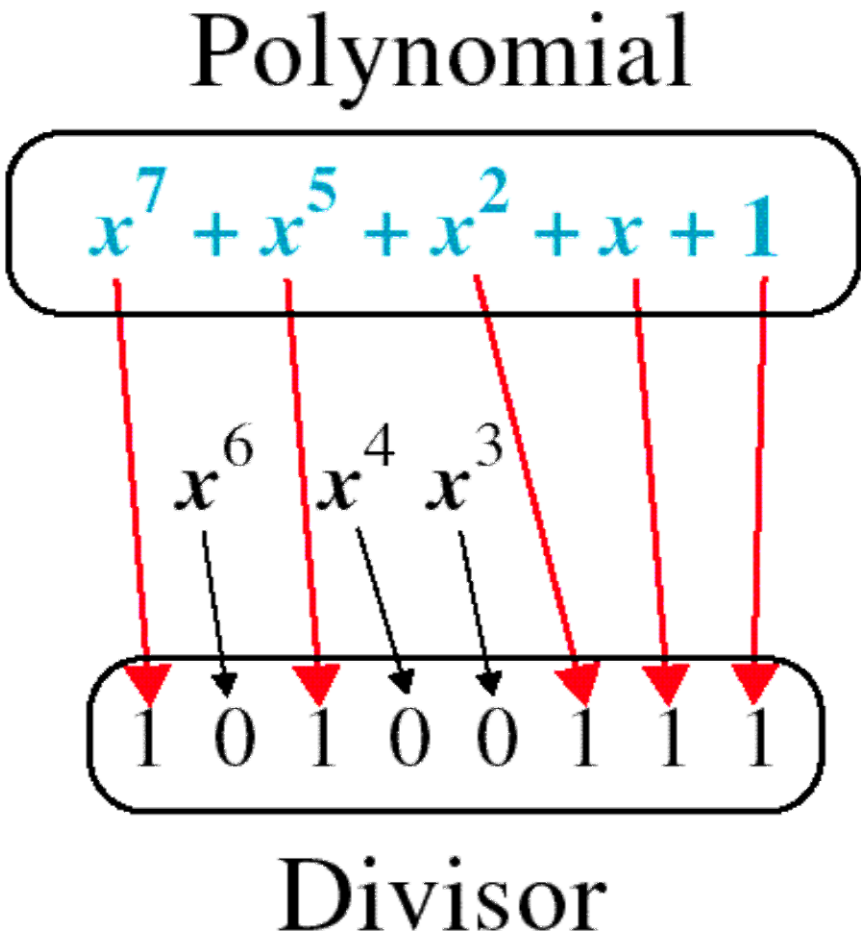
Cyclic Redundancy Check

- CRC is error detector for a message.
- Consist of : G(x),P(x) Generator & M(x) Message
- (x^3+x^2+1) polynomial mode
- (11011) binary or Divisor mode

Polynomial

$$x^7 + x^5 + x^2 + x + 1$$

Polynomial and Divisor



Example

- Send
  - M(x) = 110011 → x^5+x^4+x+1 (6 bits k=6)
  - P(x) = 11001 → x^4+x^3+1 (5 bits, n = 4)
    → 4 bits of redundancy (n-1)
  - M(x) → 110011 0000 ie (x^4)
  - Divide M(x) by P(x) to find C(x)
  - XOR operation

```
          100001
11001 ) 1100110000
        11001
          10000
          11001
           1001  = C(x)
```

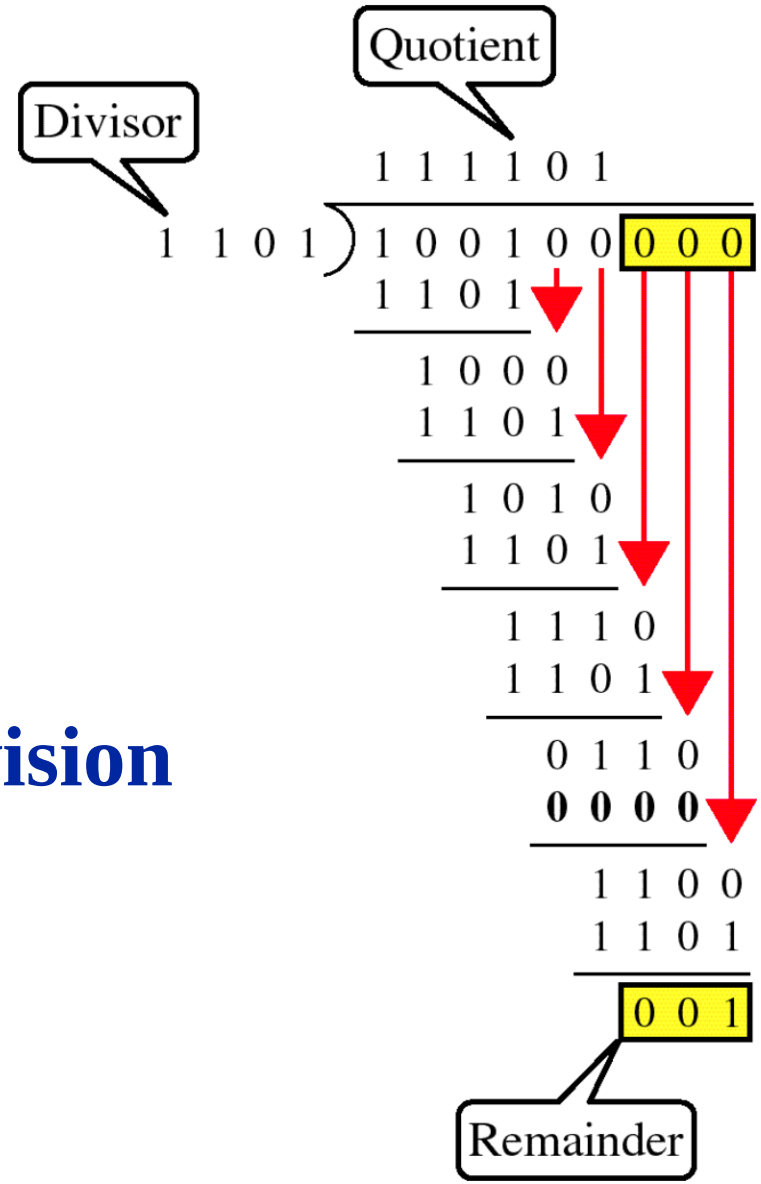
Send the block 110011 1001

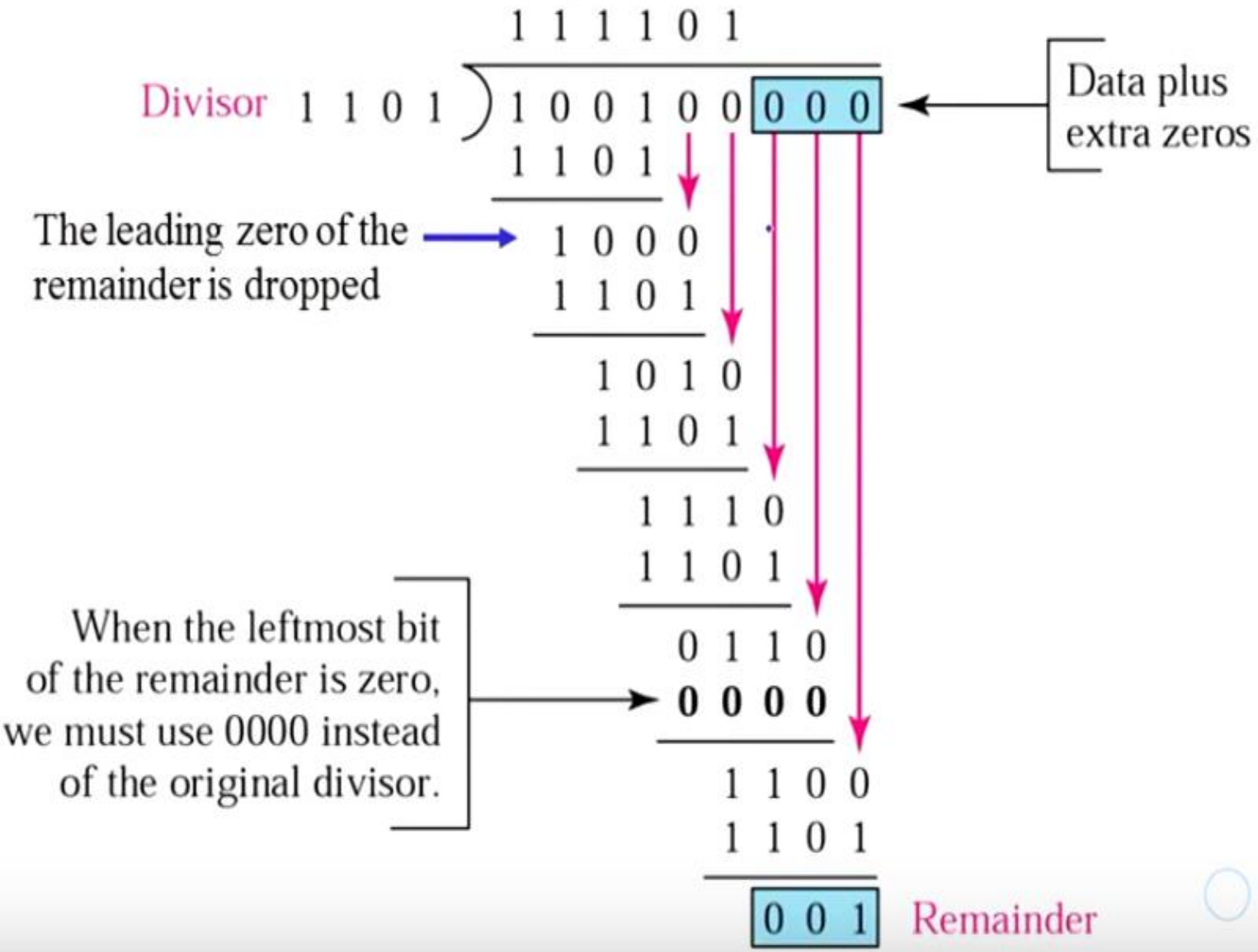
- Receive

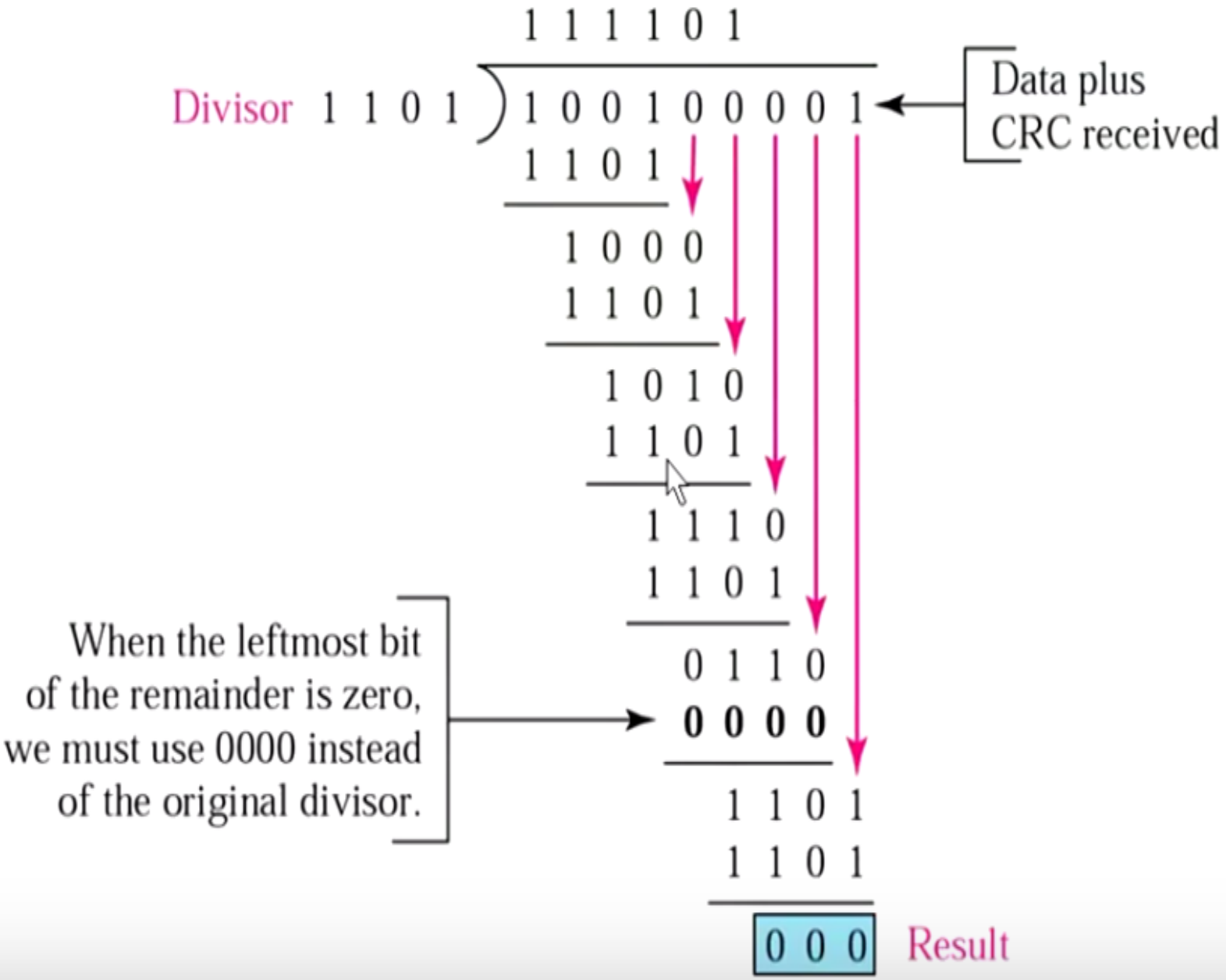
```
11001 ) 1100111001
        11001
          11001
          11001
          00000
            ↓
```

No remainder
→ Accept

The test
Binary Division




1 1 1 1 0 1
Divisor 1 1 0 1
1 0 0 1 0 0 0 0 0
Data plus extra zeros
1 1 0 1
The leading zero of the remainder is dropped
1 0 0 0
1 1 0 1
1 0 1 0
1 1 0 1
1 1 1 0
1 1 0 1
When the leftmost bit of the remainder is zero, we must use 0000 instead of the original divisor.
0 1 1 0
0 0 0 0
1 1 0 0
1 1 0 1
0 0 1
Remainder


1 1 1 1 0 1
Divisor 1 1 0 1
1 0 0 1 0 0 0 0 1
Data plus
CRC received
1 1 0 1
1 0 0 0
1 1 0 1
1 0 1 0
1 1 0 1
1 1 1 0
1 1 0 1
0 1 1 0
0 0 0 0
1 1 0 1
1 1 0 1
When the leftmost bit
of the remainder is zero,
we must use 0000 instead
of the original divisor.
0 0 0
Result

Example 1 :

FIND THE MESSAGE TO BE TRANSMITTED

- Message To be Transmitted M(x) = 1011011 k = 7
  P(x) = 1101 = $x^3+x^2+x^0$ n = 3

```
        1100101
    ------------
1101 ) 1011011000
       1101
       ----
        1100
        1101
        ----
         0011
         0000
         ----
          0111
          0000
          ----
           1110
           1101
           ----
            0110
            0000
            ----
             1100
             1101
             ----
              001
```

AT THE SENDER

- Step #1: At sender side, a string of n 0s is appended to the data unit to be transmitted where n is less than the number of bits in the CRC generator.
- Step #2: The newly formed data unit is divided by the divisor using binary division and reminder is obtained this reminder is called CRC, and when we divide we use XOR Gate.
- Step #3: String on n 0s appended to the data unit earlier is replaced by the CRC reminder –which us also n bit.
- Step #4: (Data unit + CRC) is transmitted to the receiver.

Example 2 :

FIND THE RECEIVED MESSAGE AND CHECKS ERROR

- Message Received M(x) = 1011011001 k = 7
  P(x) = 1101 = $x^3+x^2+x^0$ n = 3

```
          1100101
      ____________
1101 ) 1011011001
       1101
        1100
        1101
          0011
          0000
           0111
           0000
            1110
            1101
             0110
             0000
              1101
              1101
               000
```

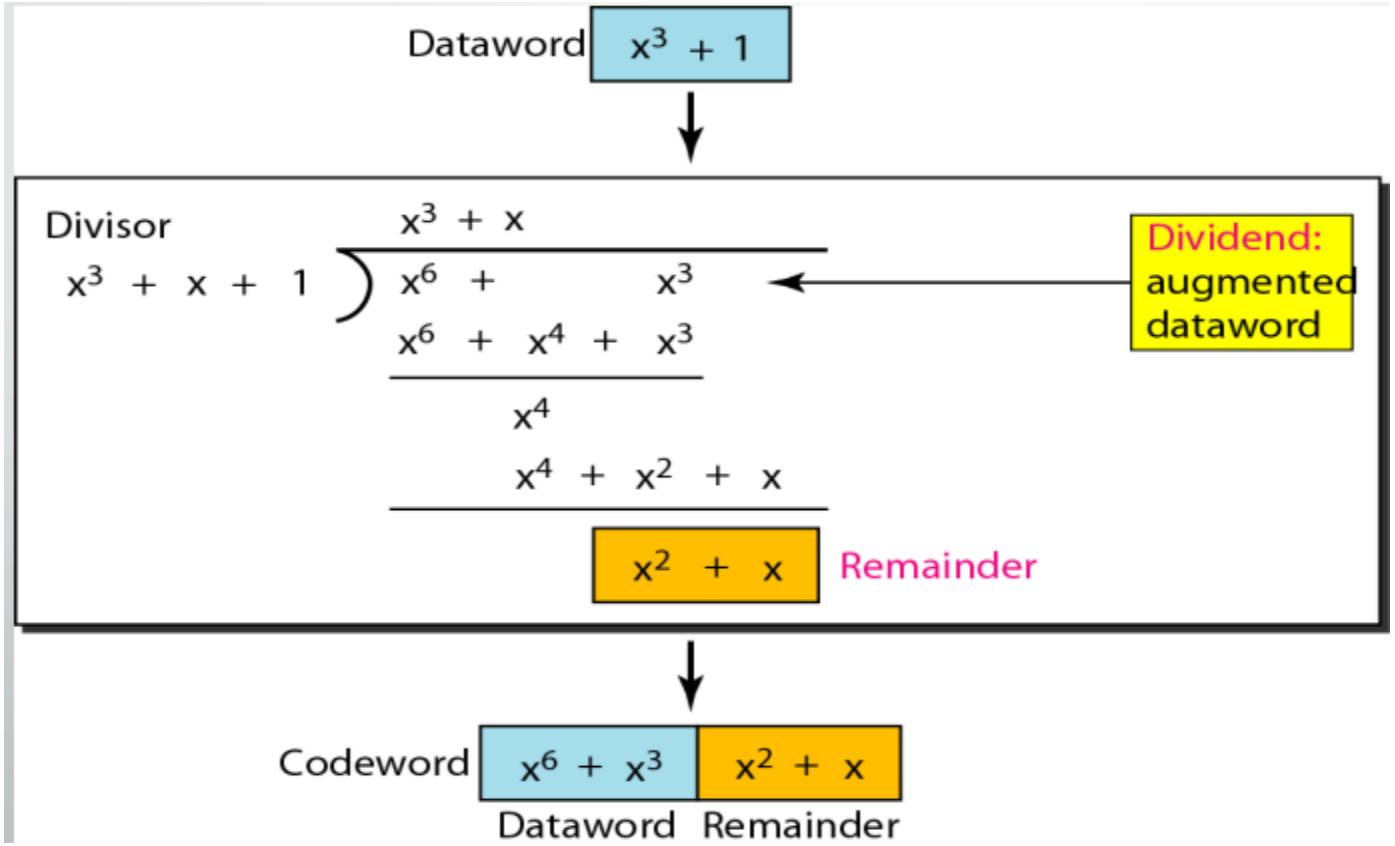
AT THE RECEIVER

- Step #1: The receiver on receiving the codeword, divides (Data unit + CRC) by the same division and checks the reminder.
- Step #2: The receiver checks if the reminder is zero.
- If Yes, Receiver assume that there is no error in data and and therefore accept it.
- If No, There is an error in data and therefore reject it.

USING POLYNOMIAL

- We can use a polynomial to represent a binary word.
- Each bit from right to left is mapped onto a power term.
- The rightmost bit represents the “0” power term. The bit next to it the “1” power term, etc.
- If the bit is of value zero, the power term is deleted from the expression

CRC DIVISION USING POLYNOMIALS



Standard Polynomials

CRC-12

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

CRC-16

$$x^{16}+x^{15}+x^{2}+1$$

CRC-ITU

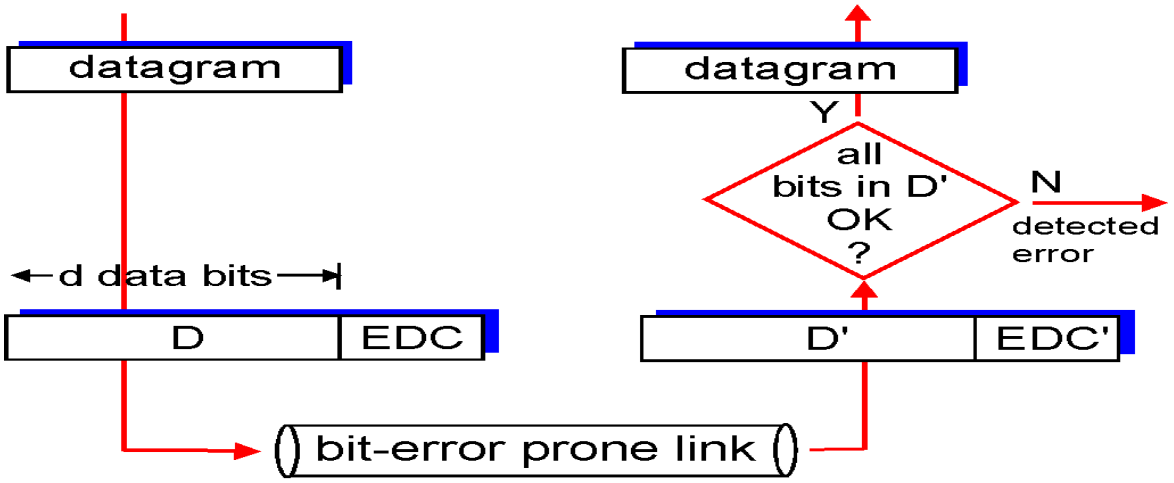
$$x^{16}+x^{12}+x^{5}+1$$

CRC-32

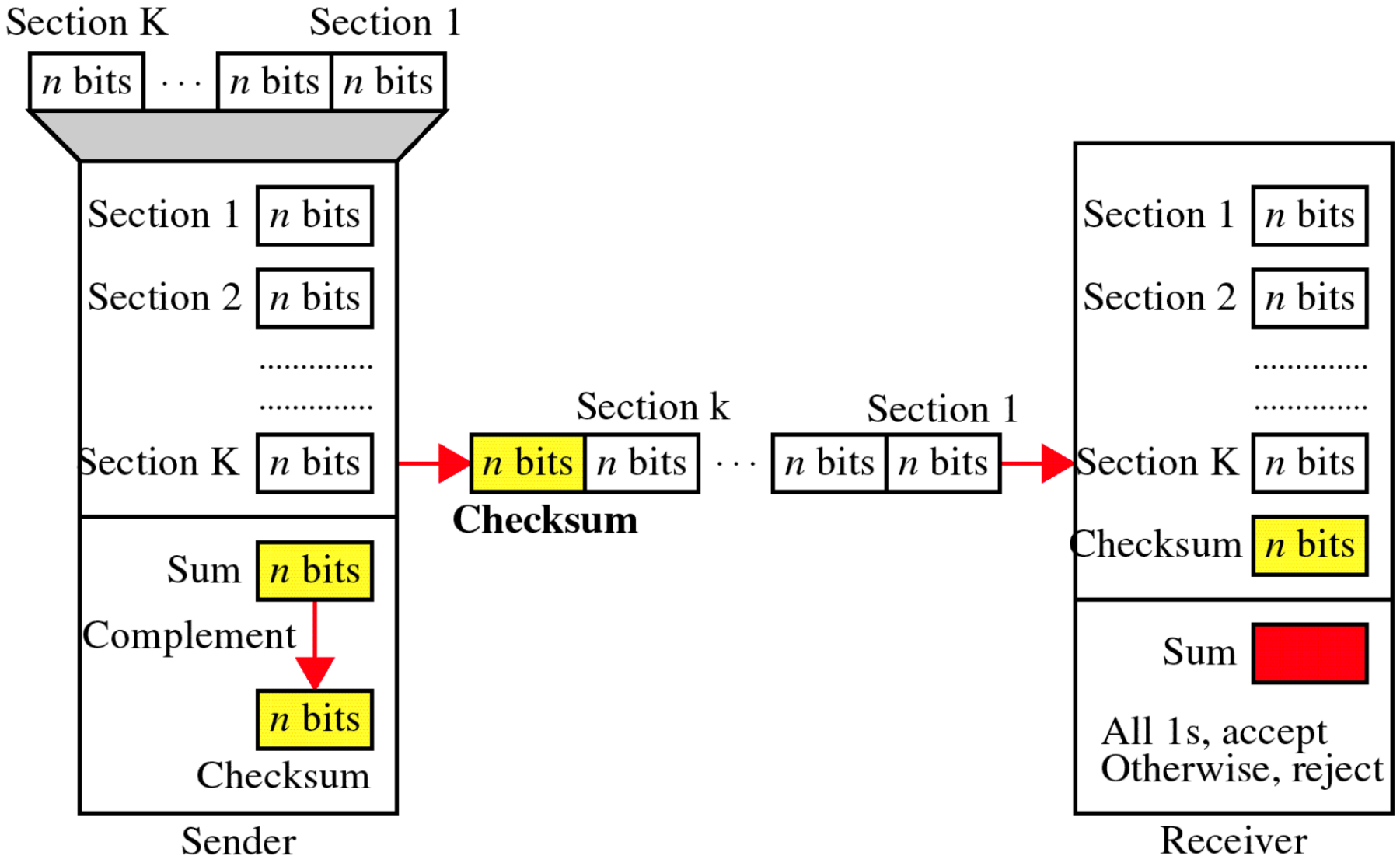
$$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^{8}+x^{7}+x^{5}+x^{4}+x^{2}+x+1$$

checksum

- Detected all errors involving odd numbers of bits and even number of bits



Checksum


Section K
Section 1
n bits
n bits
n bits
Section 1
Section 2
Section K
Sum
Complement
Checksum
Sender
Section k
Section 1
Checksum
Section 1
Section 2
Section K
Checksum
Sum
All 1s, accept
Otherwise, reject
Receiver

The sender follows these steps:

1. The unit is divided into k sections, each of n bits.
2. All sections are added using one's complement to get the sum.
3. The sum is complemented and becomes the checksum.
4. The checksum is sent with the data.

The receiver follows these steps:

1. The unit is divided into k sections, each of n bits.
2. All sections are added using one's complement to get the sum.
3. The sum is complemented.
4. If the result is zero, the data are accepted: otherwise, rejected.

Suppose the following block of 16 bits is to be sent using a checksum of 8 bits.

10101001 00111001

The numbers are added using one's complement

```
  10101001
  00111001
  ----------
Sum  11100010
```

Checksum **00011101**

The pattern sent is 10101001 00111001 **00011101**

there is no error.

10101001 00111001 00011101

When the receiver adds the three sections, it will get all 1s, which, after complementing, is all 0s and shows that there is no error.

	10101001
	00111001
	00011101
Sum	11111111
Complement	**00000000 means that the pattern is OK.**

Example 9

Now suppose there is a burst error of length 5 that affects 4 bits.

10101111 11111001 00011101

When the receiver adds the three sections, it gets

	10101111
	11111001
	00011101
Partial Sum	1 11000101
Carry	1
Sum	11000110
Complement	00111001 **the pattern is corrupted.**

Lecture VIII

ARP protocol

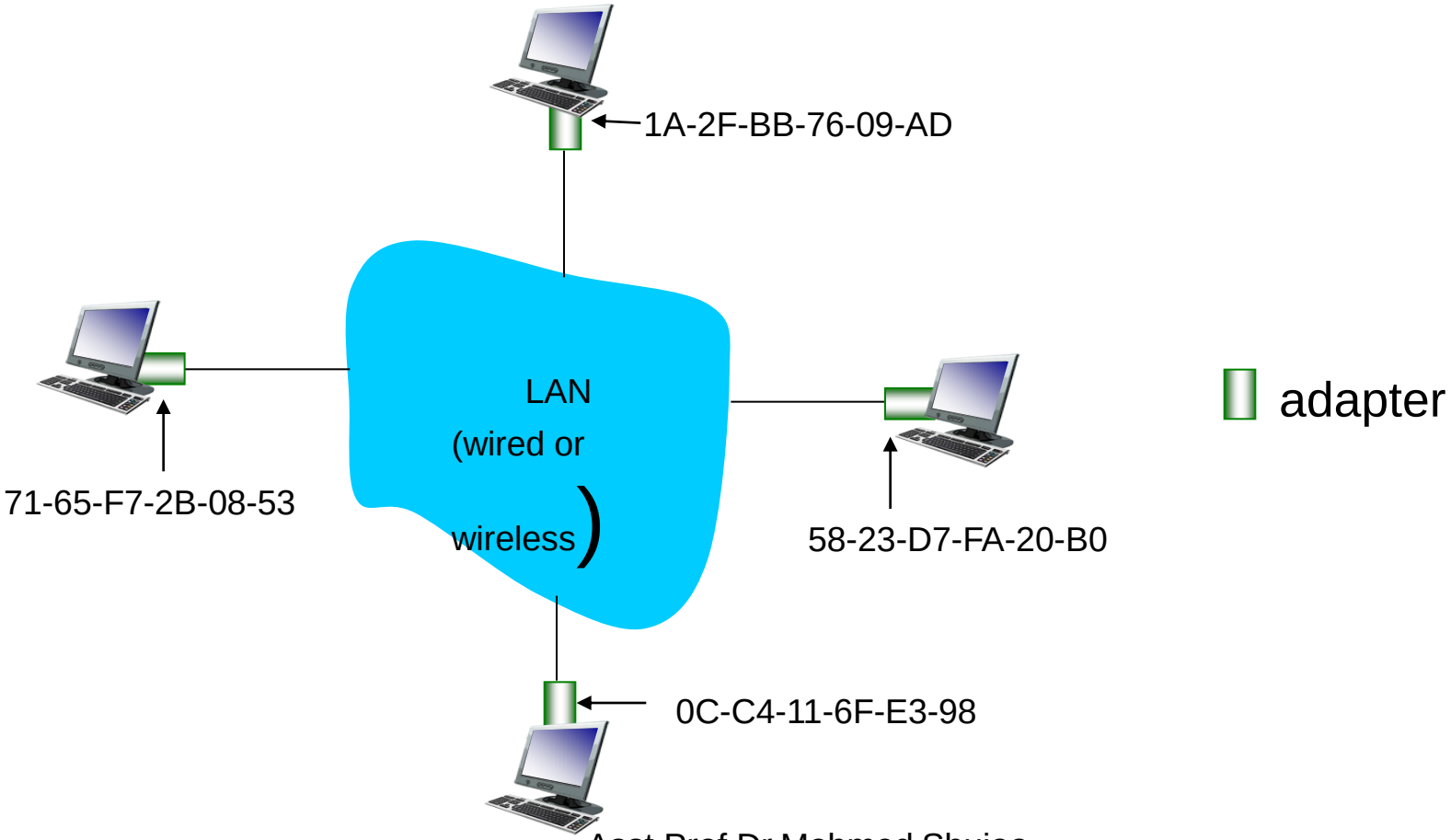
MAC addresses and ARP

- 32-bit IP address:
  - *network-layer* address for interface
  - used for layer 3 (network layer) forwarding
- MAC (or LAN or physical or Ethernet) address:
  - function: *used 'locally" to get frame from one interface to another physically-connected interface (same network, in IP-addressing sense)*
  - 48 bit MAC address (for most LANs) burned in NIC ROM, also sometimes software settable
  - e.g.: **1**A-2F-BB-76-09-AD

Hexadecimal (each "number" represents 4 bits)

LAN addresses -ARP(Address Resolution Protocol)

each adapter on LAN has unique *LAN* address



LAN addresses

- MAC address: like Social Security Number
- IP address: like postal address

❖ MAC flat address ➜ portability

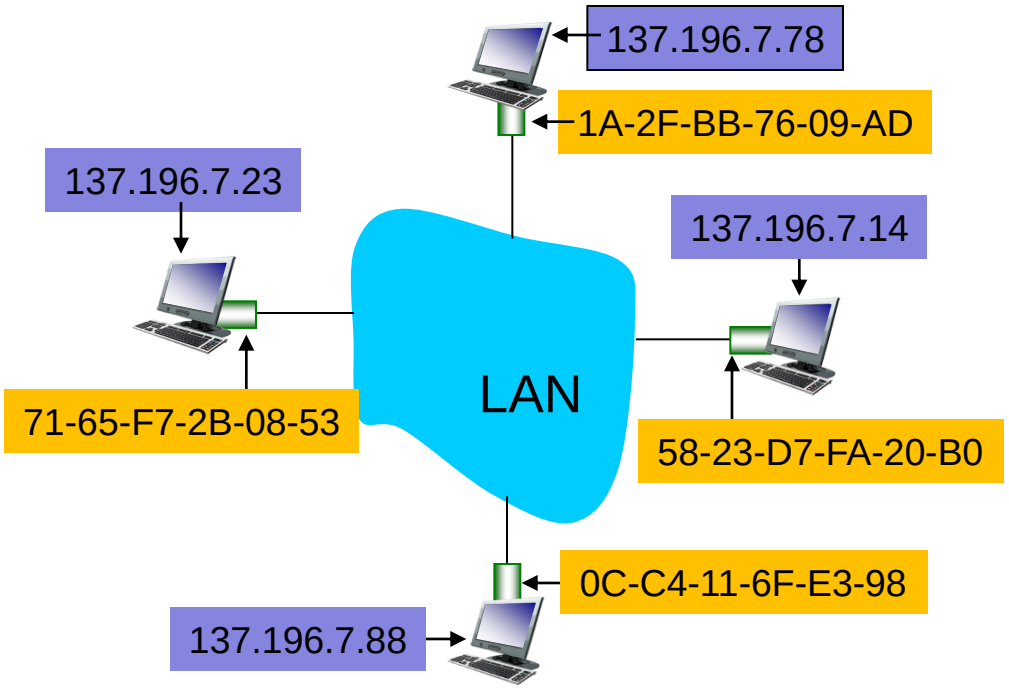
- can move LAN card from one LAN to another

❖ IP hierarchical address *not* portable

- address depends on IP subnet to which node is attached

ARP: Address Resolution Protocol

Question: how to determine interface's MAC address, knowing its IP address?



ARP table: each IP node (host, router) on LAN has 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: same LAN

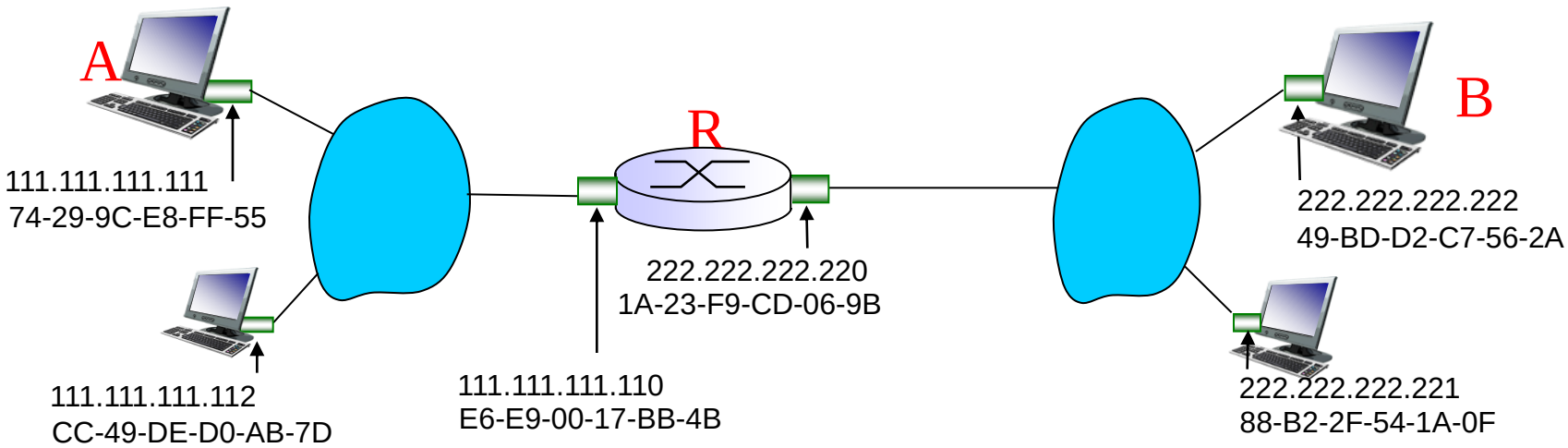
- A wants to send datagram to B
  - B's MAC address not in A's ARP table.
- A broadcasts ARP query packet, containing B's IP address
  - dest MAC address = FF-FF-FF-FF-FF-FF
  - all nodes on LAN receive ARP query
- B receives ARP packet, replies to A with its (B's) MAC address
  - frame sent to A's MAC address (unicast)

- A caches (saves) IP-to-MAC address pair in its ARP table until information becomes old (times out) TTL
  - soft state: information that times out (goes away) unless refreshed
- ARP is "plug-and-play":
  - nodes create their ARP tables *without intervention from net administrator*

Addressing: routing to another LAN

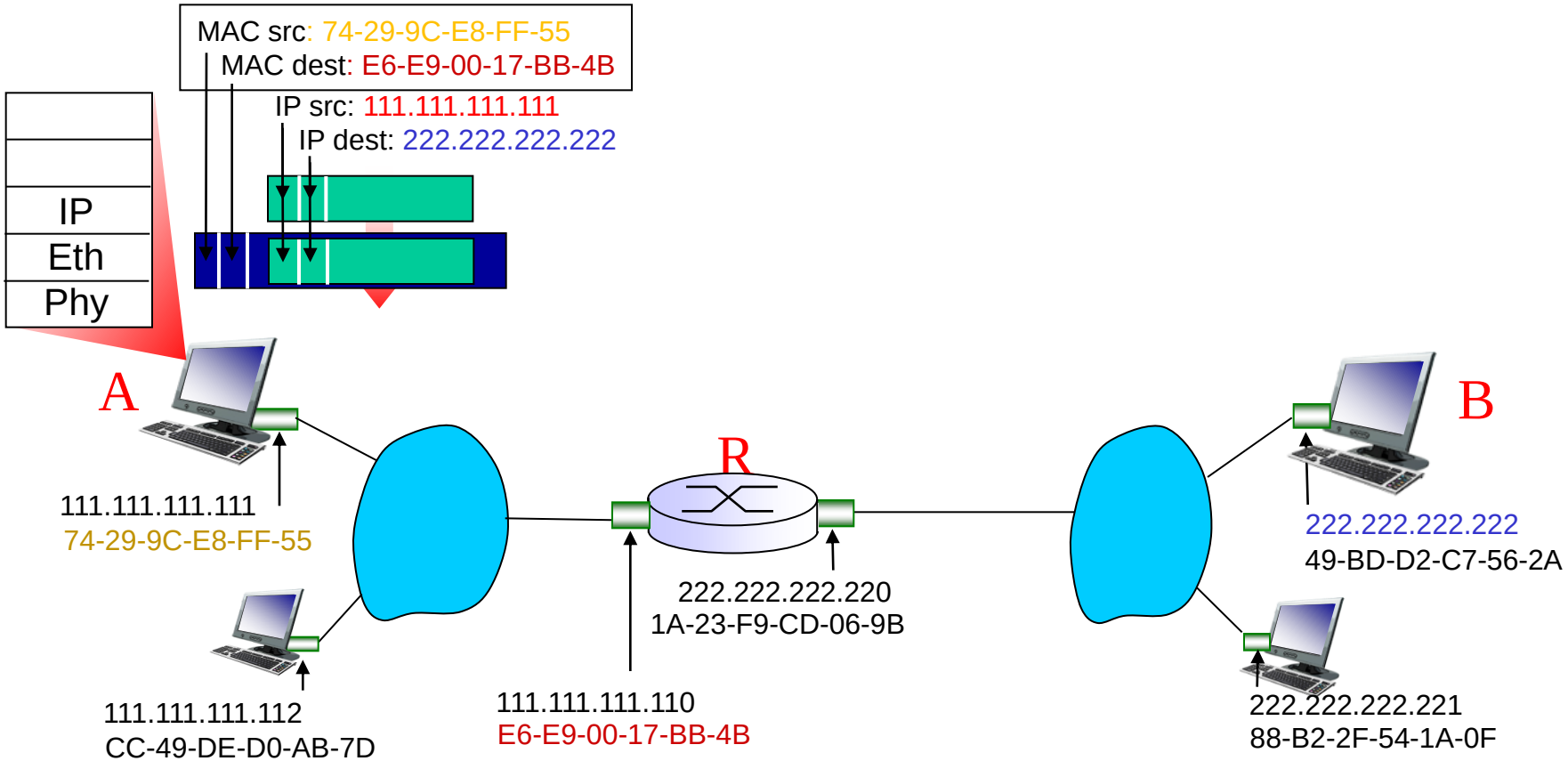
walkthrough: send datagram from A to B via R

- focus on addressing – at IP (datagram) and MAC layer (frame)
- assume A knows B's IP address
- assume A knows IP address of first hop router, R (how?)
- assume A knows R's MAC address (how?)



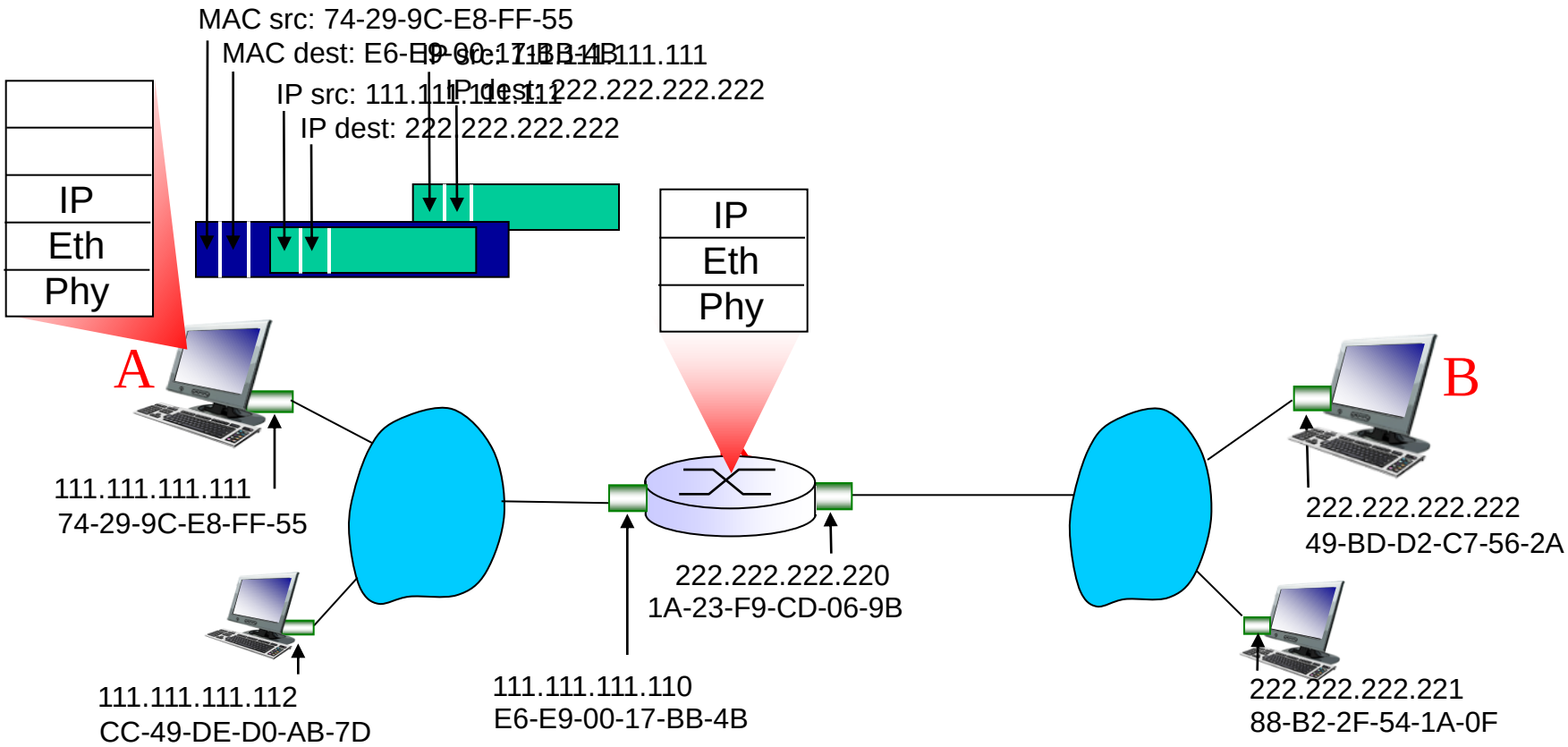
Addressing: routing to another LAN

- A creates IP datagram with IP source A, destination B
- A creates link-layer frame with R's MAC address as dest, frame contains A-to-B IP datagram



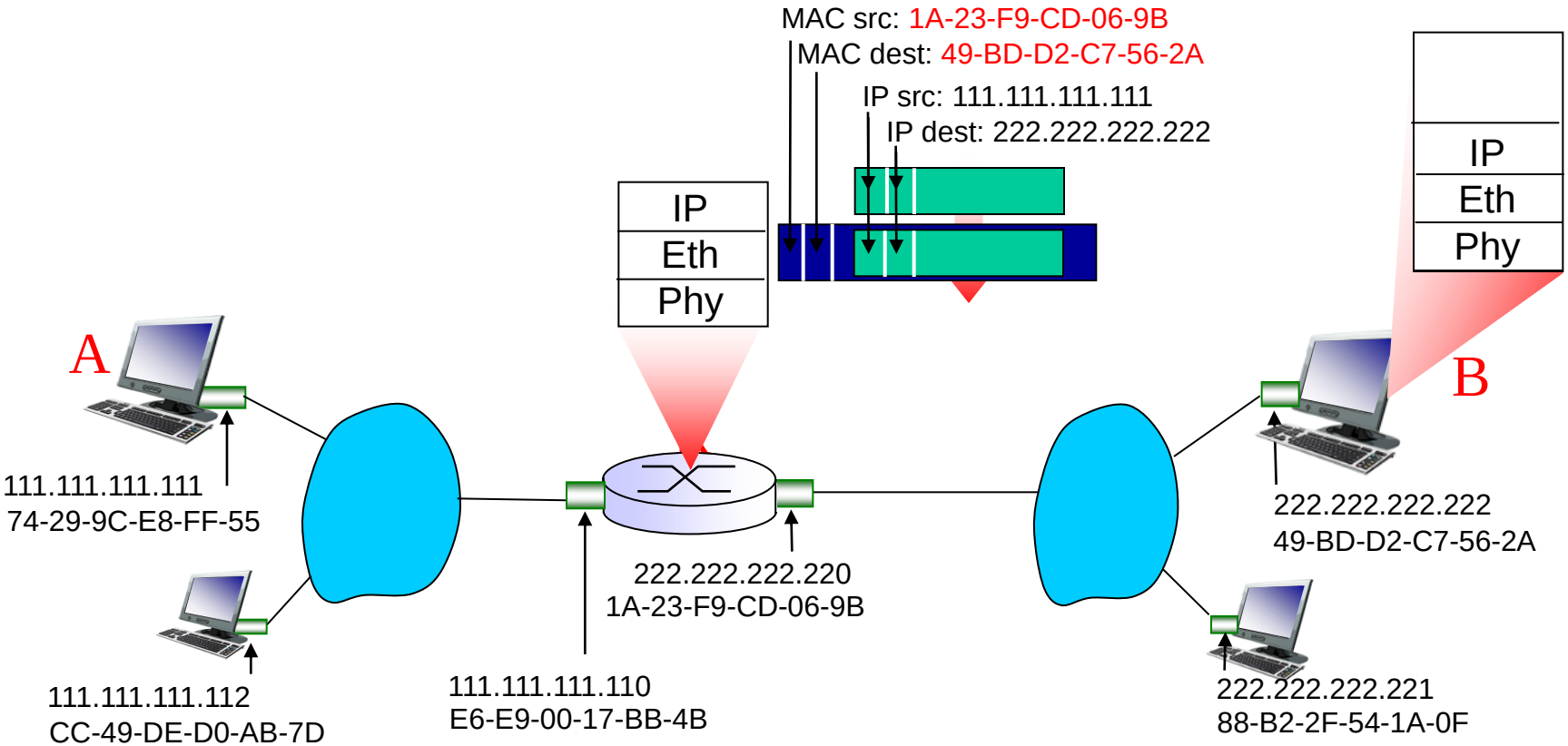
Addressing: routing to another LAN

- frame sent from A to R
- frame received at R, datagram removed, passed up to IP



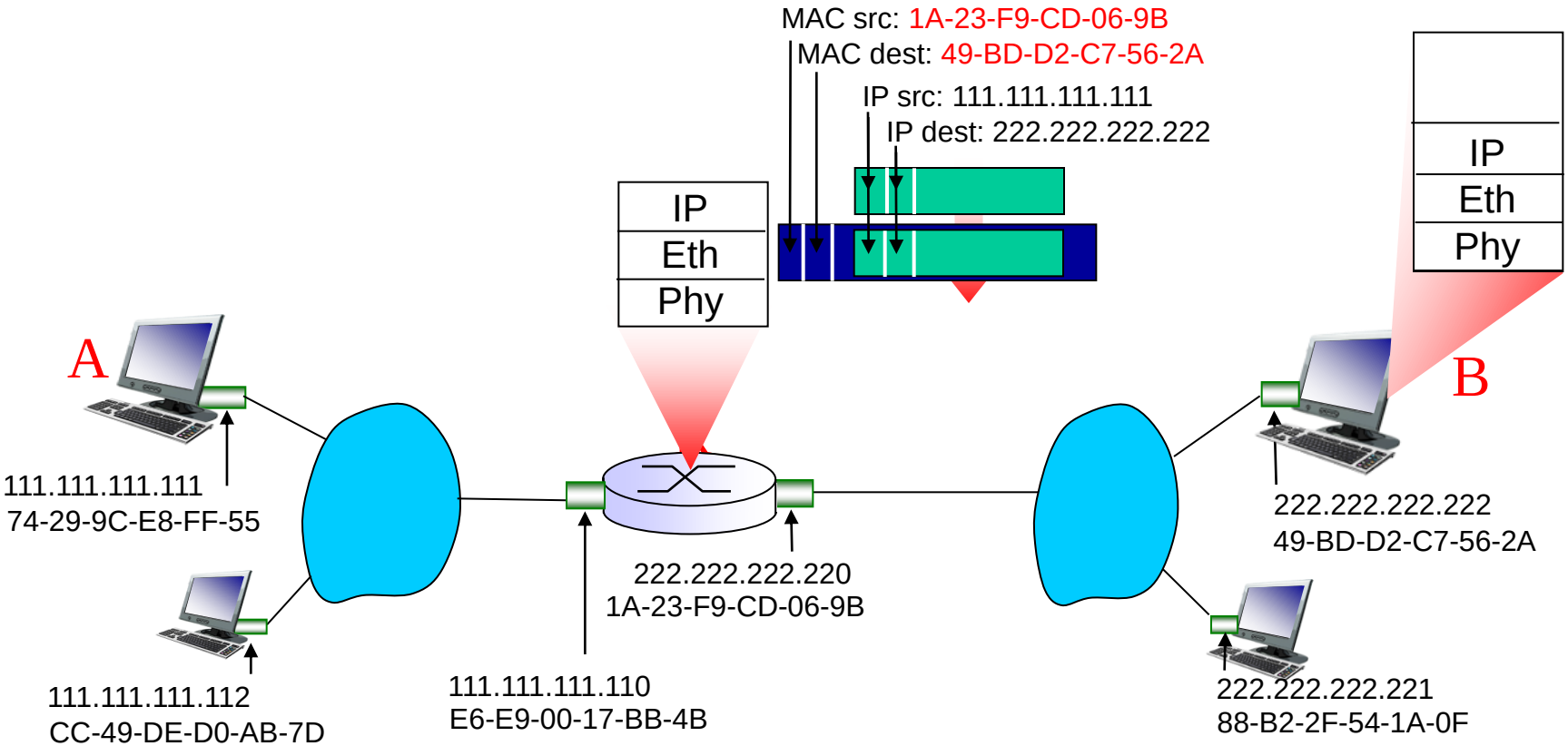
Addressing: routing to another LAN

- R forwards datagram with IP source A, destination B
- R creates link-layer frame with B's MAC address as dest, frame contains A-to-B IP datagram



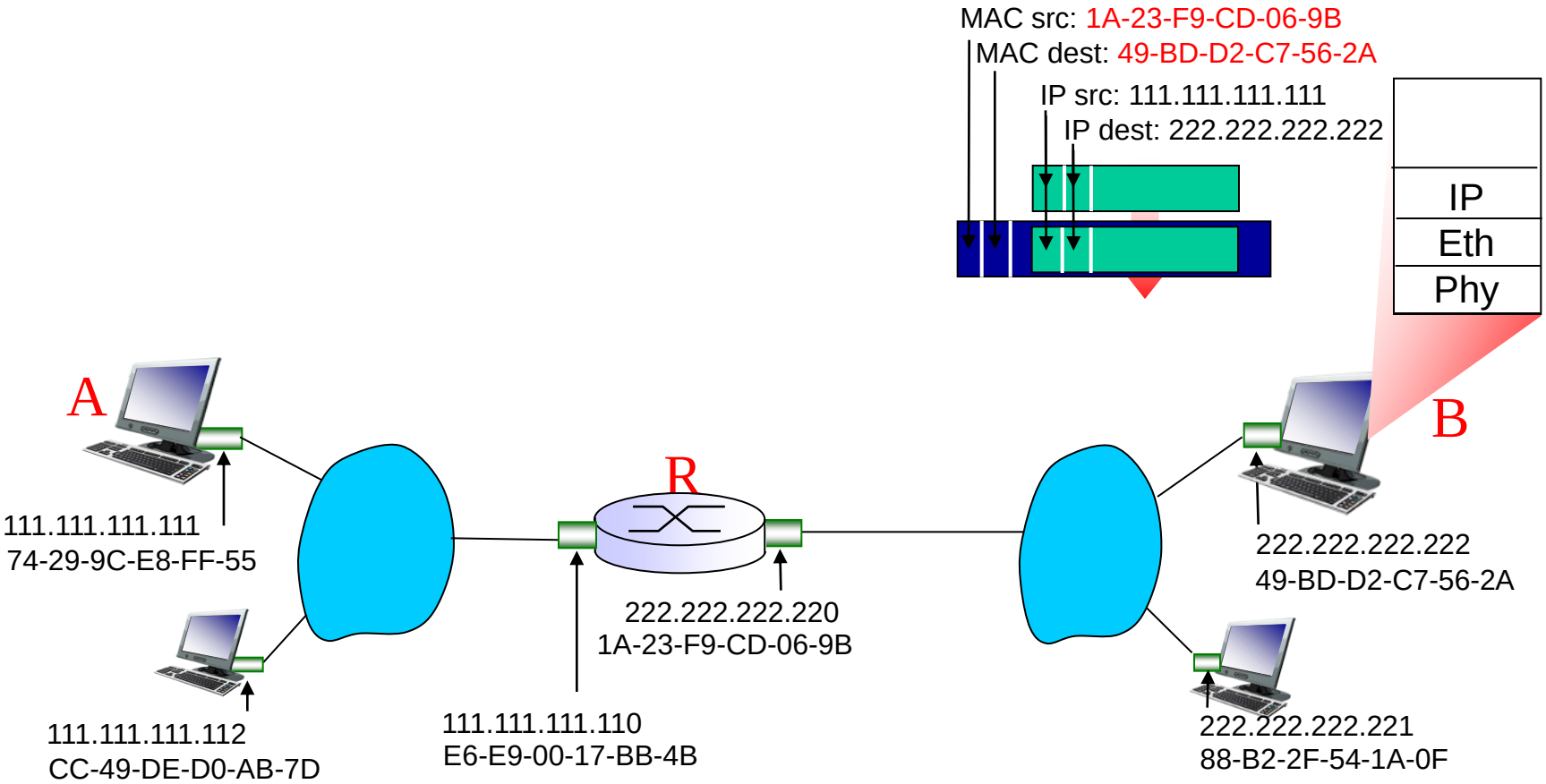
Addressing: routing to another LAN

- R forwards datagram with IP source A, destination B
- R creates link-layer frame with B's MAC address as dest, frame contains A-to-B IP datagram



Addressing: routing to another LAN

- R forwards datagram with IP source A, destination B
- R creates link-layer frame with B's MAC address as dest, frame contains A-to-B IP datagram

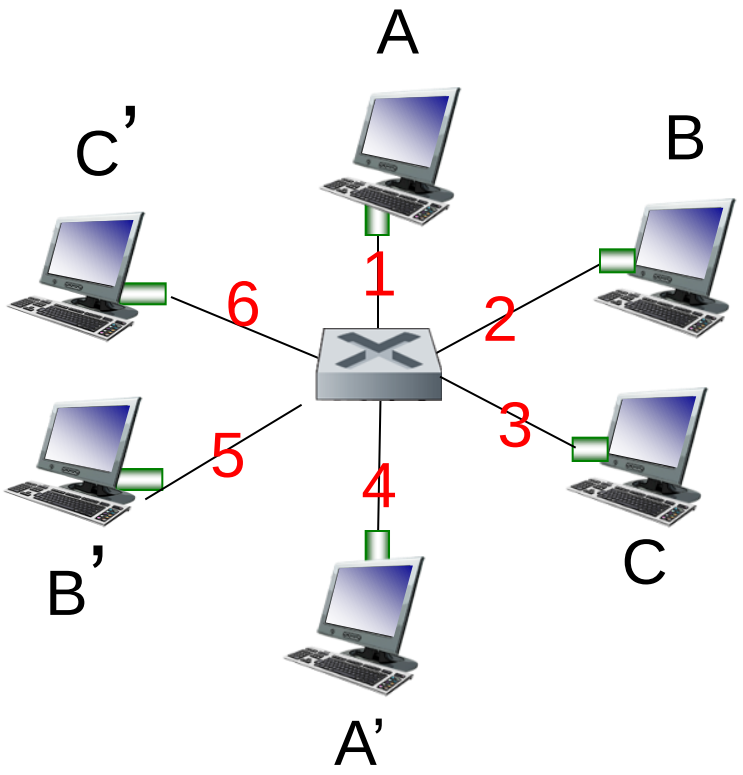


Ethernet switch

- link-layer device: takes an *active* role
  - store, forward Ethernet frames
  - examine incoming frame's MAC address, selectively forward frame to one-or-more outgoing links when frame is to be forwarded on segment, uses CSMA/CD to access segment
- *Transparent*
  - hosts are unaware of presence of switches
- *plug-and-play, self-learning*
  - switches do not need to be configured

Switch: *multiple* simultaneous transmissions

- Hosts have dedicated, direct connection to switch
- Switches buffer packets
- Ethernet protocol used on *each* incoming link, but no collisions; full duplex
  - each link is its own collision domain
- *switching:* A-to-A' and B-to-B' can transmit simultaneously, without collisions



switch with six interfaces (1,2,3,4,5,6)

Switch forwarding table

Q: how does switch know A' reachable via interface 4, B' reachable via interface 5?

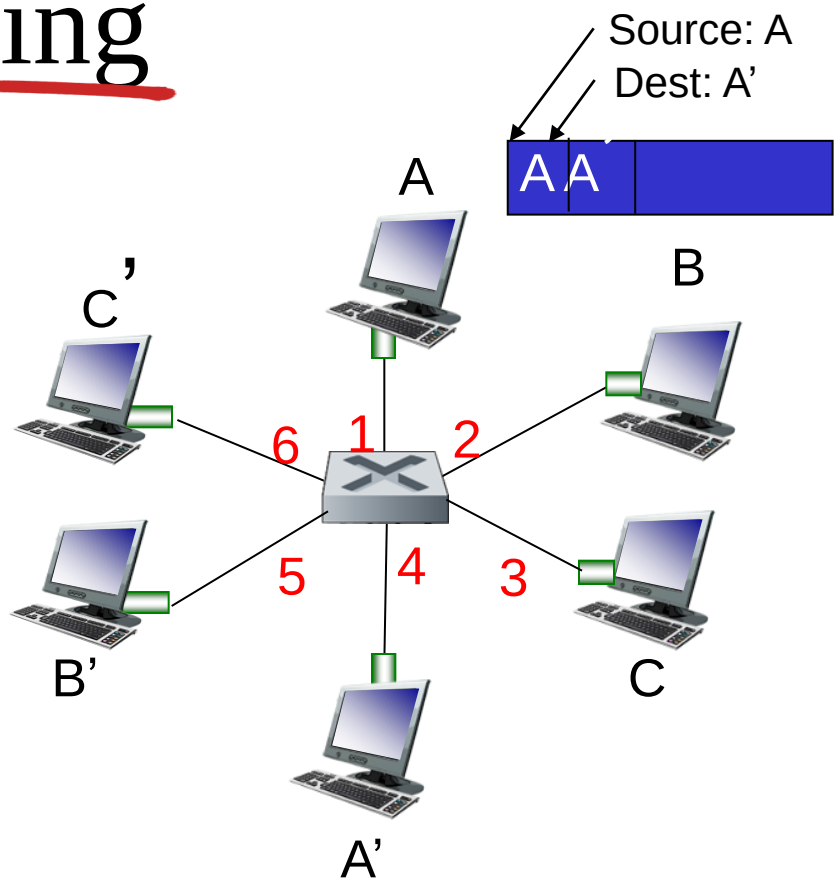
- A: each switch has a switch table, each entry:
  - (MAC address of host, interface to reach host, time stamp)
  - looks like a routing table!

Q: how are entries created, maintained in switch table?

- something like a routing protocol?

Switch: self-learning

- switch *learns* which hosts can be reached through which interfaces
  - when frame received, switch "learns" location of sender: incoming LAN segment
  - records sender/location pair in switch table



MAC addr	interface	TTL
A	1	60

Switch table
(initially empty)

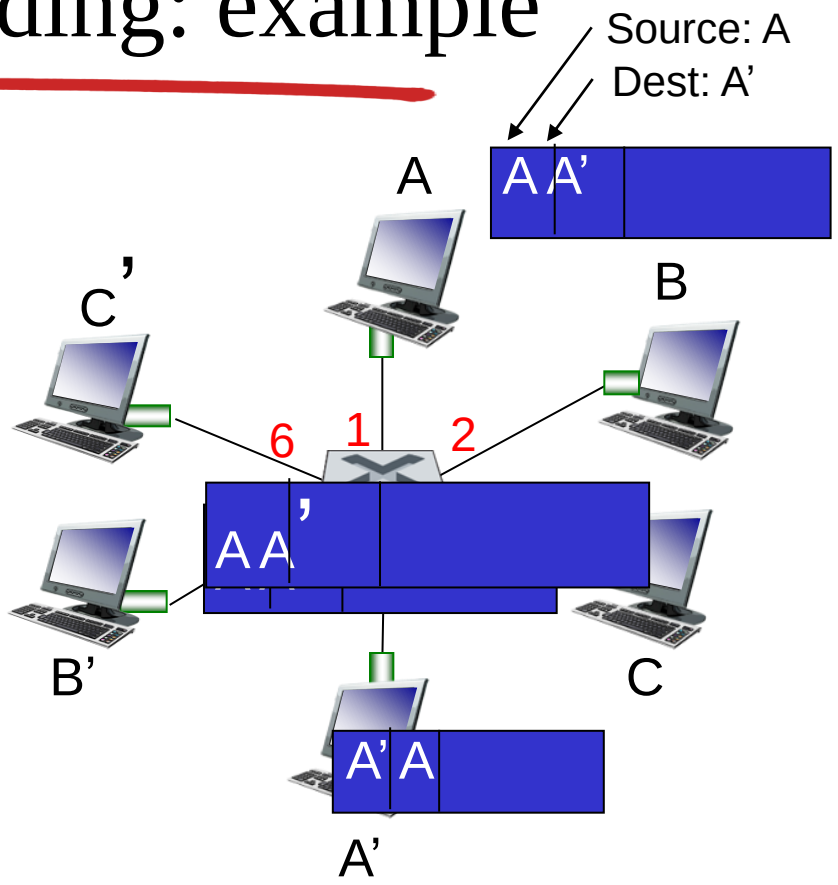
Switch: frame filtering/forwarding

when frame received at switch:

1. record incoming link, MAC address of sending host
2. index switch table using MAC destination address
3. if entry found for destination
   then {
   if destination on segment from which frame arrived
   then drop frame
   else forward frame on interface indicated by entry
   }
   else flood /* forward on all interfaces except arriving
   interface */

Self-learning, forwarding: example

- frame destination, A’, location unknown: flood
- destination A location known: selectively send on just one link

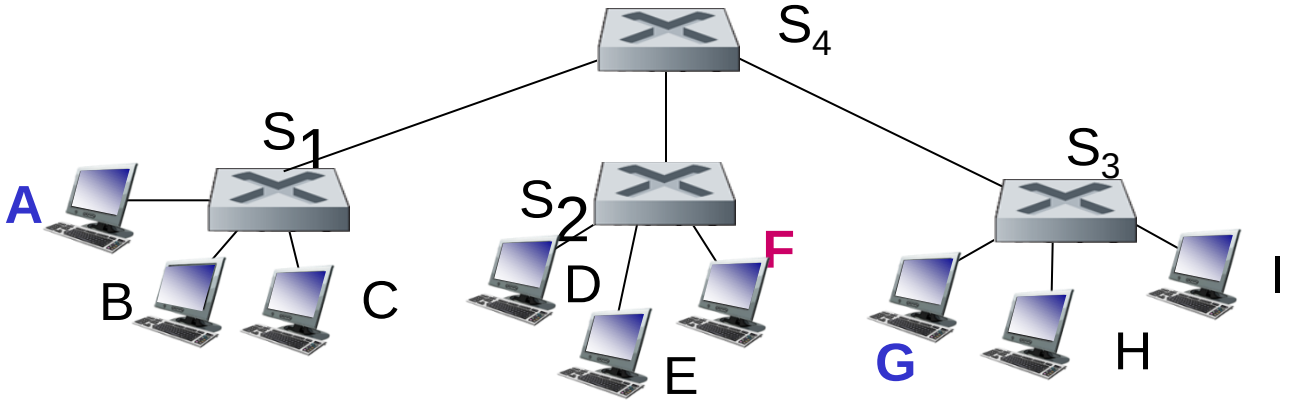


MAC addr	interface	TTL
A	1	60
A’	4	60

switch table (initially empty)

Interconnecting switches

❖switches can be connected together

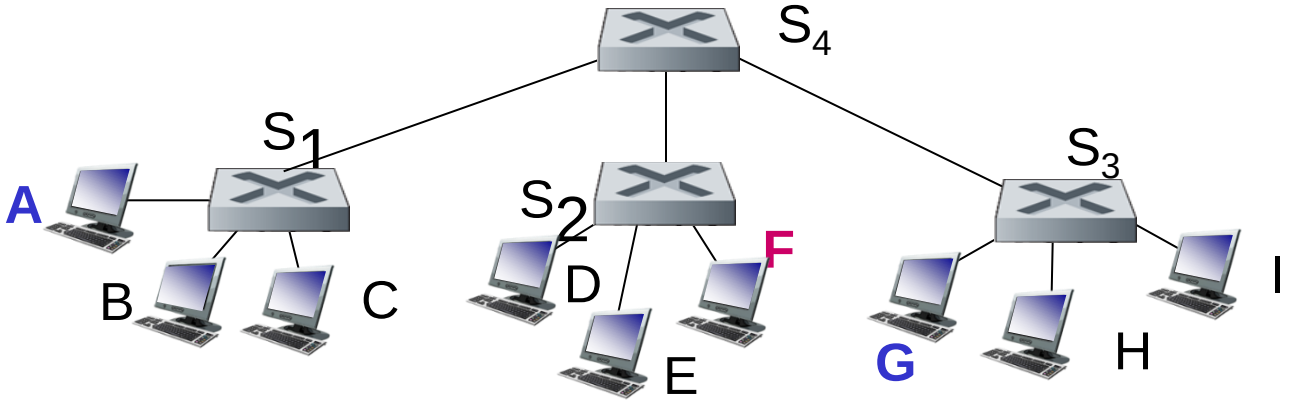


Q: sending from A to G - how does S_1 know to forward frame destined to F via S_4 and S_3?

- A: self learning! (works exactly the same as in single-switch case!)

Self-learning multi-switch example

Suppose C sends frame to I, I responds to C

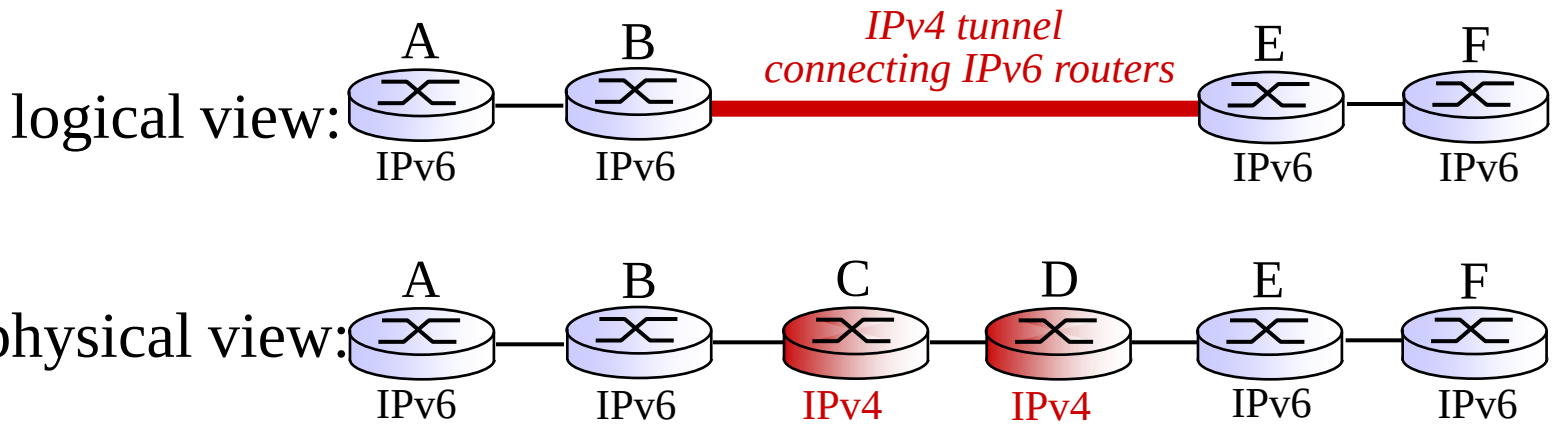


Q: show switch tables and packet forwarding in S_1, S_2, S_3, S_4

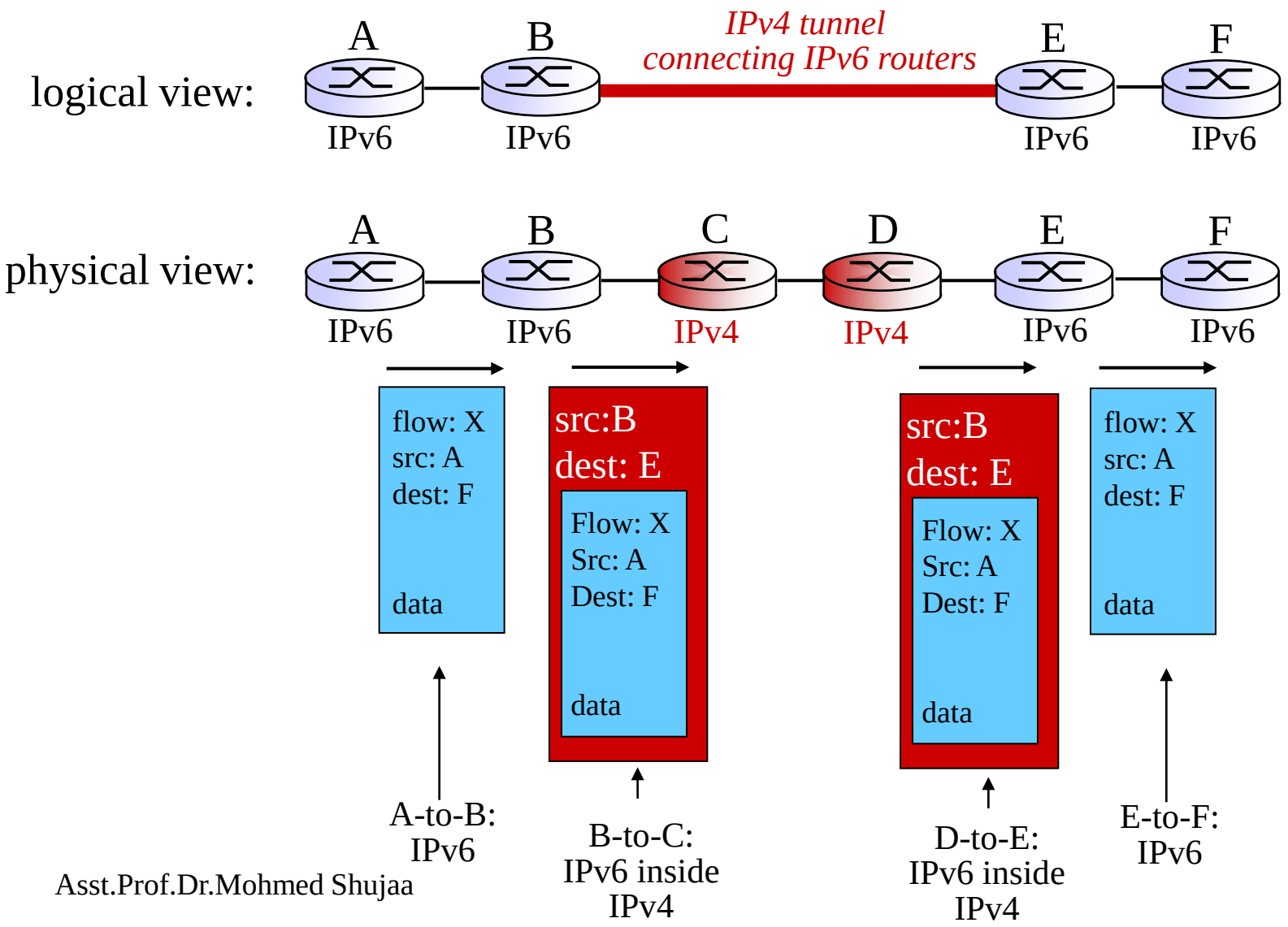
Tunneling Protocol (L2TP)

- Layer Two Tunneling Protocol (L2TP) *is an extension of the Point-to-Point Tunneling Protocol (PPTP)* used by an Internet service provider (ISP) *to enable the operation of a* ***virtual private network*** *(VPN) over the Internet. (secured)*
- **Virtual Local Area Network(VLAN)** switch(es) supporting VLAN capabilities can be configured to define multiple **virtual** LANS over single physical LAN infrastructure (Not secured)
- *tunneling:* IPv6 datagram carried as *payload* in IPv4 datagram among IPv4 routers

Tunneling


logical view:
A
B
IPv4 tunnel
connecting IPv6 routers
E
F
IPv6
IPv6
IPv6
IPv6
physical view:
A
B
C
D
E
F
IPv6
IPv6
IPv4
IPv4
IPv6
IPv6

Tunneling


logical view:
A
B
IPv4 tunnel
connecting IPv6 routers
E
F
IPv6
IPv6
IPv6
IPv6
physical view:
A
B
C
D
E
F
IPv6
IPv6
IPv4
IPv4
IPv6
IPv6
flow: X
src: A
dest: F
data
src:B
dest: E
Flow: X
Src: A
Dest: F
data
src:B
dest: E
Flow: X
Src: A
Dest: F
data
flow: X
src: A
dest: F
data
A-to-B:
IPv6
B-to-C:
IPv6 inside
IPv4
D-to-E:
IPv6 inside
IPv4
E-to-F:
IPv6

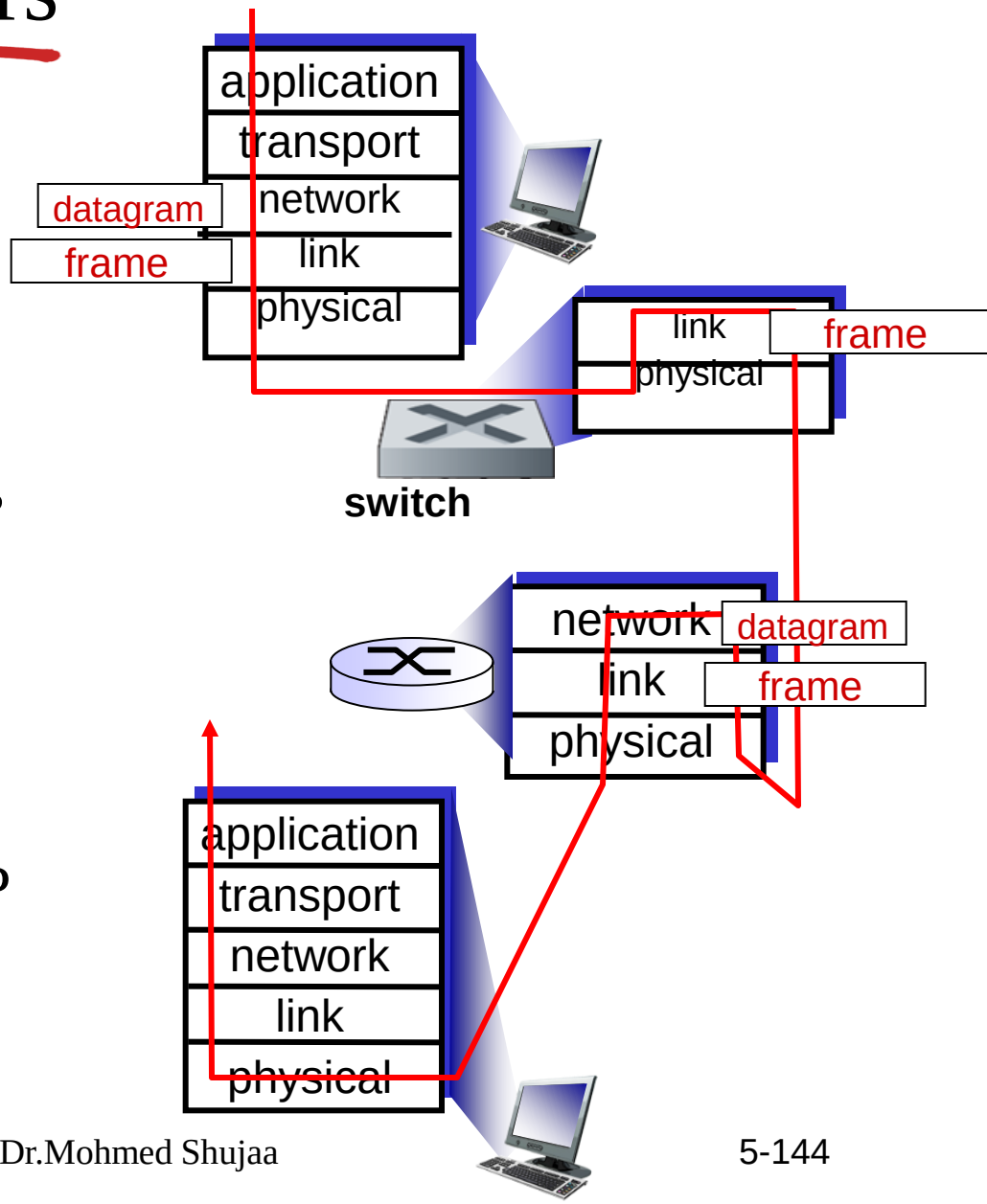
Switches vs. routers

both are store-and-forward:

- *routers:* network-layer devices (examine network-layer headers)
- *switches:* link-layer devices (examine link-layer headers)

both have forwarding tables:

- *routers:* compute tables using routing algorithms, IP addresses
- *switches:* learn forwarding table using flooding, learning, MAC addresses



7 Application

6 Presentation

5 Session

4 Transport

3 Network

2 Data Link

1 Physical

Lecture IX
3.NETWORK LAYER

7 Application

6 Presentation

5 Session

4 Transport

3 Network

2 Data Link

1 Physical

OVERVIEW

1. Routing Algorithms
   - **Shortest Path**
   - **Flooding**
   - **Flow-based**
   - **Distance Vector**
   - **Link State**
   - **Hierarchical**
   - **Broadcast**
   - **Multicast**
   - **Routing in the internet, mobile host**
2. Congestion control
3. IP Addressing **IP: Internet Protocol . datagram format**

IPv4 addressing , ICMP , IPv6

4. Routers protocols: **RIP OSPF BGP**
5. Implementation of **connectionless service**

   Implementation of **connection-oriented service**

Routing Concept

- **ROUTING** is **forwarding** of packets from one network to another network choosing the best path from the routing table.
- **Routing table** consist of only the best routes for every destinations.
- **Types of routing:**

1- Static

2- Default

3- Dynamic

Static Algorithm

- Flooding routing.
- Shortest path routing.
- Flow based routing.

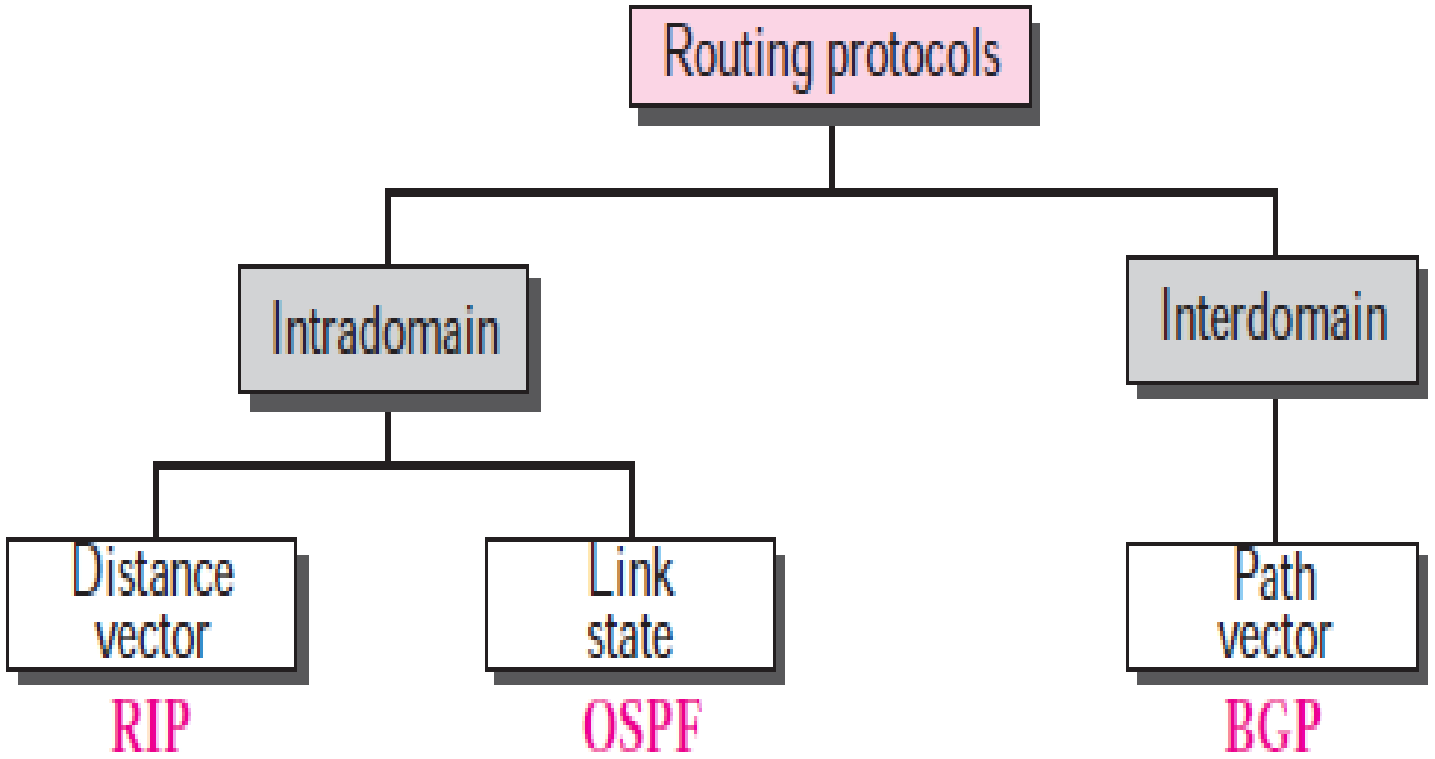
Dynamic Routing Algorithm

- Distance Vector Routing.
- Link state routing.

Routing protocol

- **A routing protocol** is a combination of **rules and procedures** *that lets routers in the internet inform each other of changes.*
- **Routing metric**: *a method* by which routing algorithms determines that *one route is better than another route* **Metric** may be(hop count, bandwidth ,delay, load)

Popular routing protocols



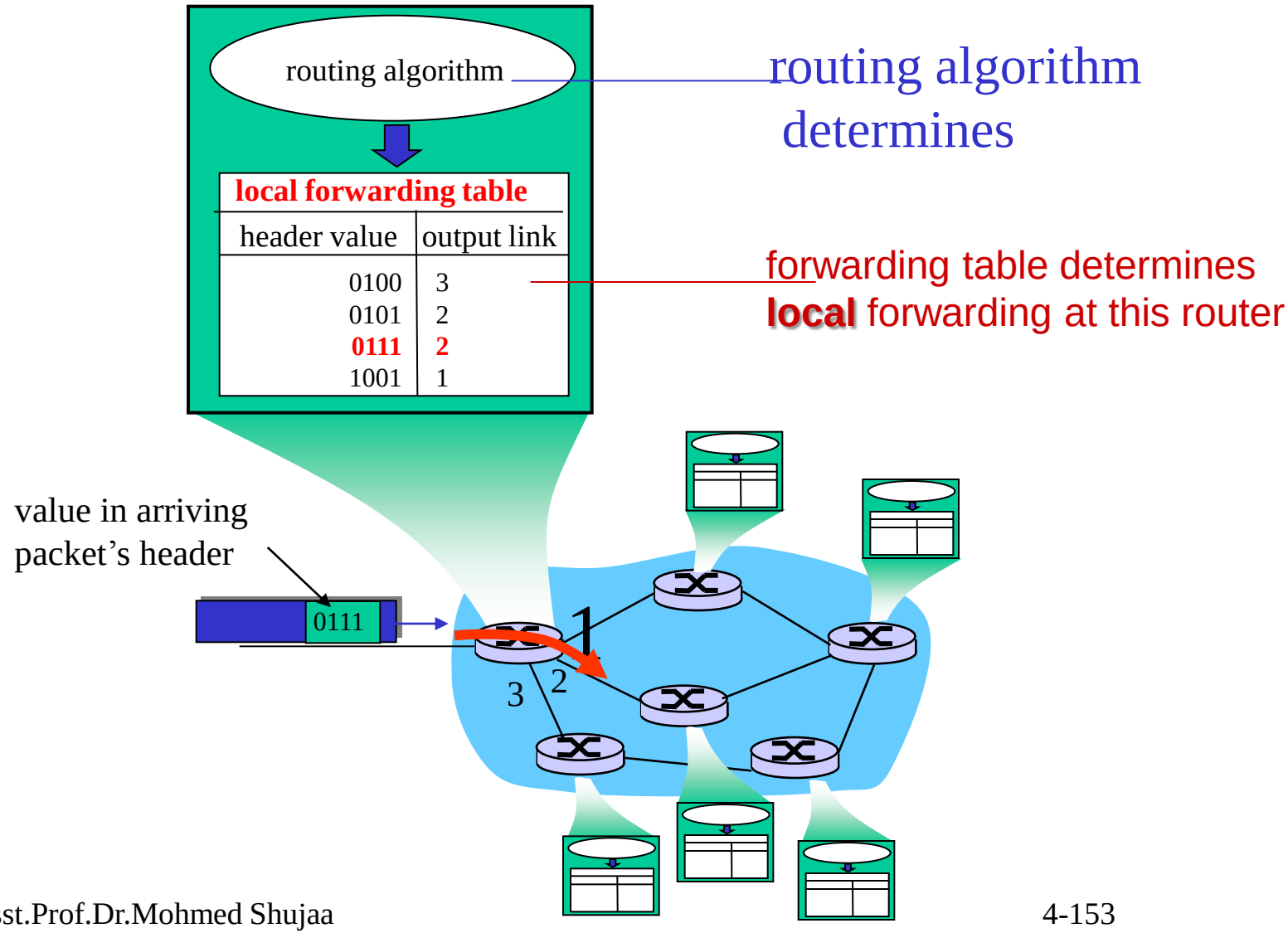
Two key network-layer functions

- *forwarding:* move packets from router's input to appropriate router output

- *routing:* determine route taken by packets from source to destination.

- *routing algorithms*

Interplay between routing and forwarding



Routing algorithm classification

General:

- all routers have complete topology, link cost info
- **"link state" algorithms**

decentralized:

- router knows physically-connected neighbors, link costs to neighbors
- process of computation, exchange info with neighbors
- **"distance vector" algorithms**

static or dynamic?

static:

- routes change slowly over time

dynamic:

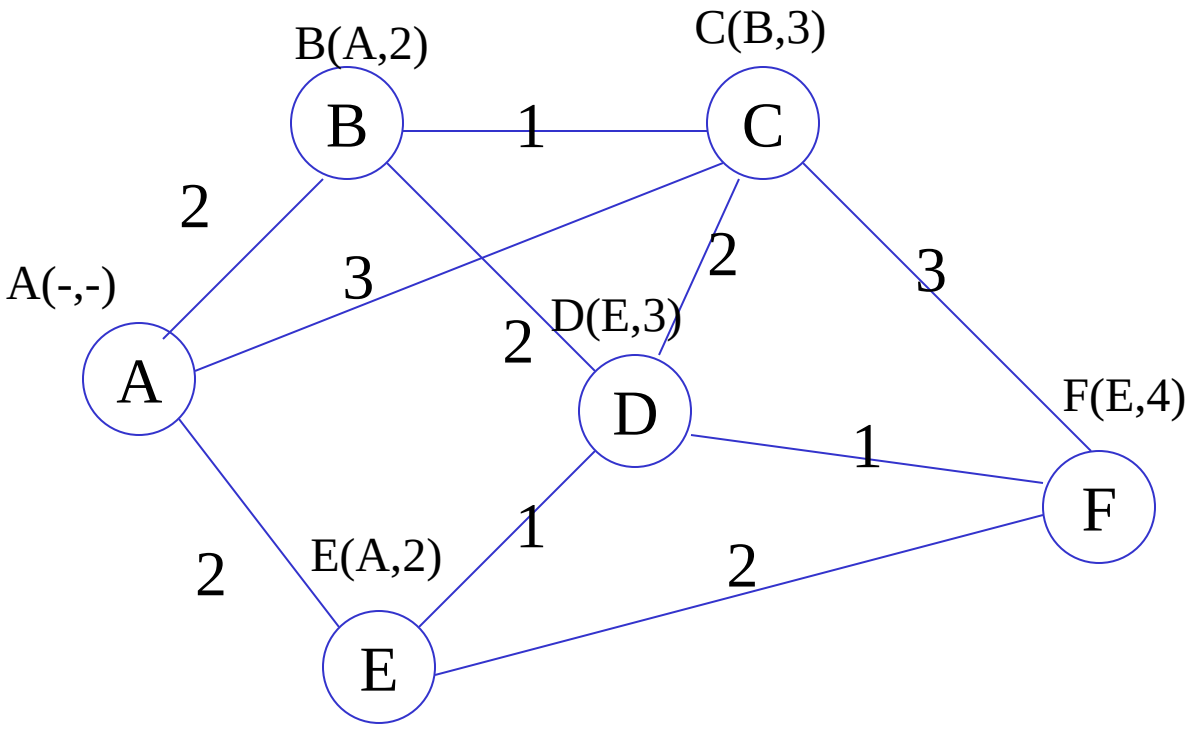
- routes change more quickly
  - *periodic update*
  - *in response to link cost changes*

Lecture X

Algorithems

STATIC ROUTING ALGORITHMS

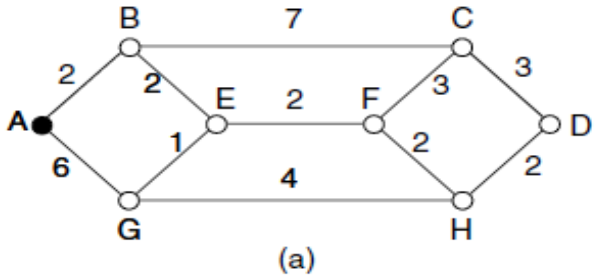
1. Shortest Path : Developed by E.W.Dijkstra Algorithem



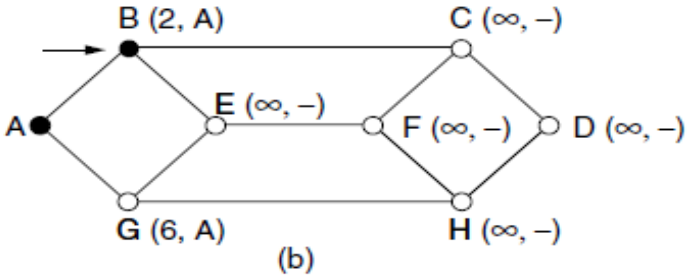
A – E – D – F

A – E – F is the answer. (hops& cost)

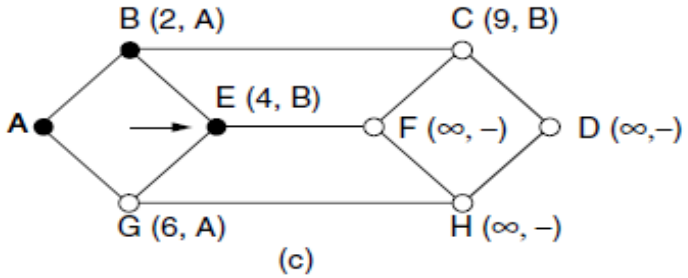
Example: Find the shortest path from a specified source to all other destinations in the network.



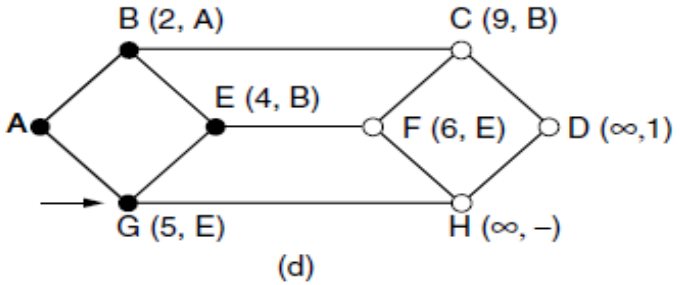
(a)



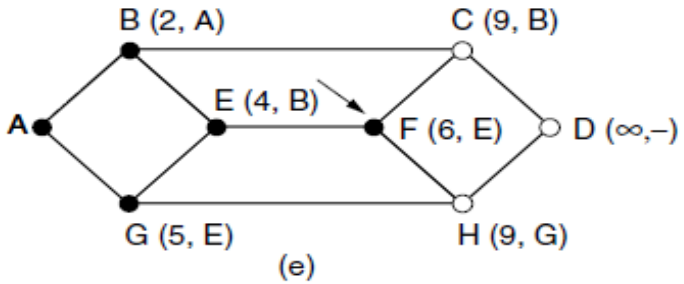
(b)



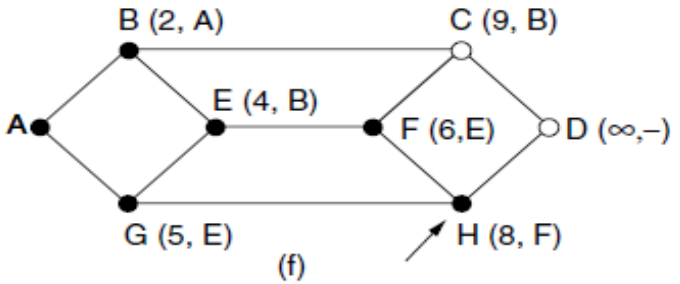
(c)



(d)



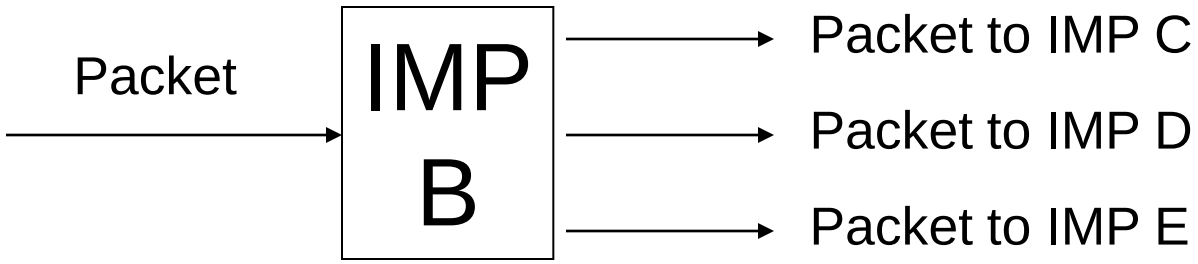
(e)



(f)

STATIC ROUTING ALGORITHMS

2. Flooding :



To prevent packets from circulating indefinitely, a packet has a hop counter. Every time a packet arrives at an IMP, the hop counter is decrease by 1. Once the hop counter of a packet reaches 0, the packet is discarded.

every incoming packet is sent out on every outgoing line except the one it arrived on.

7 Application

6 Presentation

5 Session

4 Transport

3 Network

2 Data Link

1 Physical

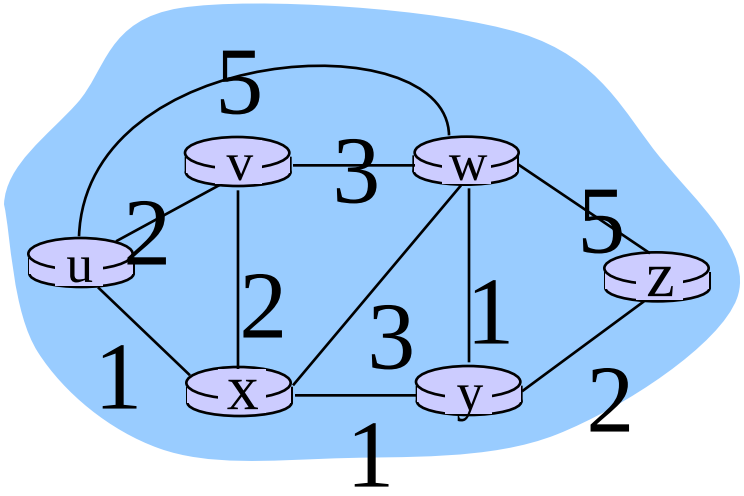
NETWORK DEVICE /ROUTERS

A layer 3 device that is used to interconnect 2 or more logical networks.

Can filter broadcast traffic, preventing broadcast traffic from one network from reaching another network.



Routing Graph abstraction



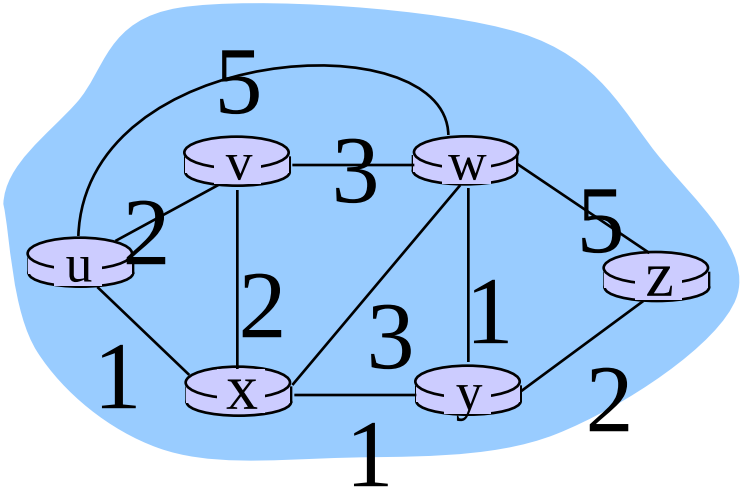
graph: G = (N,E)

N = set of routers = { u, v, w, x, y, z }

E = set of links ={ (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) }

graph abstraction is useful in other network contexts, e.g., P2P, where *N* is set of peers and *E* is set of TCP connections

Graph abstraction: costs



c(x,x') = cost of link (x,x')
e.g., c(w,z) = 5

cost could always be 1, or inversely related to bandwidth, or inversely related to congestion

cost of path $(x_1, x_2, x_3, \ldots, x_p) = c(x_1,x_2) + c(x_2,x_3) + \ldots + c(x_{p-1},x_p)$

key question: what is the least-cost path between u and z ?
routing algorithm: algorithm that finds that least cost path

Lecture XI

LS. Algorithem

A Link-State Routing Algorithm (Dynamic Algorithm)

Dijkstra's algorithm

- net topology, link costs known to all nodes
  - accomplished via "link state broadcast"
  - all nodes have same info
- ***computes least cost paths from one node ("source") to all other nodes***
  - gives *forwarding table* for that node
- iterative: after k iterations, know least cost path to k dest.'s

notation:

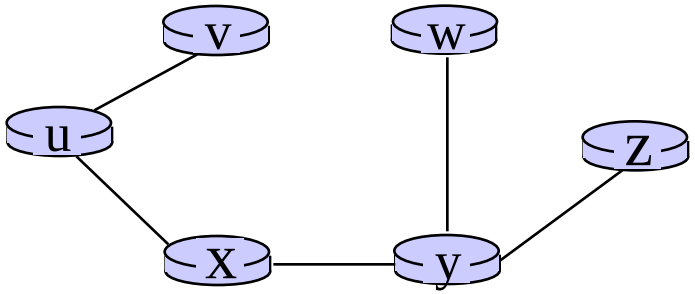
- c(x,y): link cost from node x to y; $= \infty$ if not direct neighbors
- D(v): current value of cost of path from source to dest. v
- p(v): predecessor node along path from source to v
- N': set of nodes whose least cost path definitively known

Link State Routing

1. Discover neighbors, learn network addresses.
2. Set distance/cost metric to each neighbor.
3. Construct packet telling all learned.
4. Send packet to, receive packets from other routers.
5. Compute shortest path to every other router.

Dijkstra's algorithm: example

resulting shortest-path tree from u:

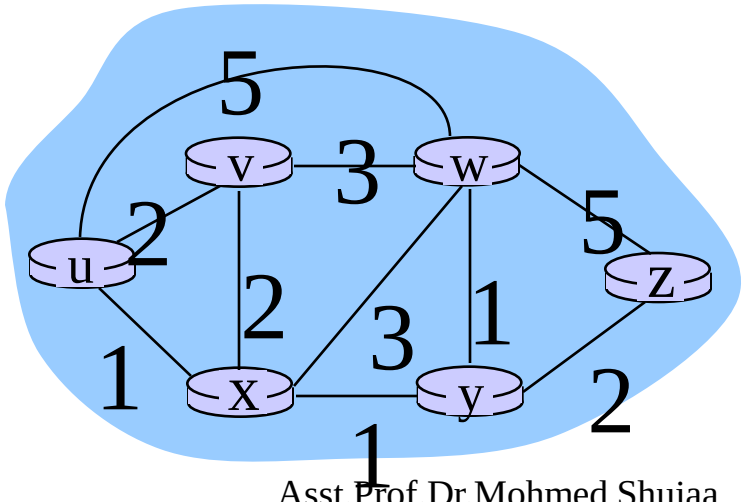


resulting forwarding table in u:

destination	link
v	(u,v)
x	(u,x)
y	(u,x)
w	(u,x)
z	(u,x)

Dijkstra's algorithm: example

Step	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					



Dijkstra's algorithm, discussion

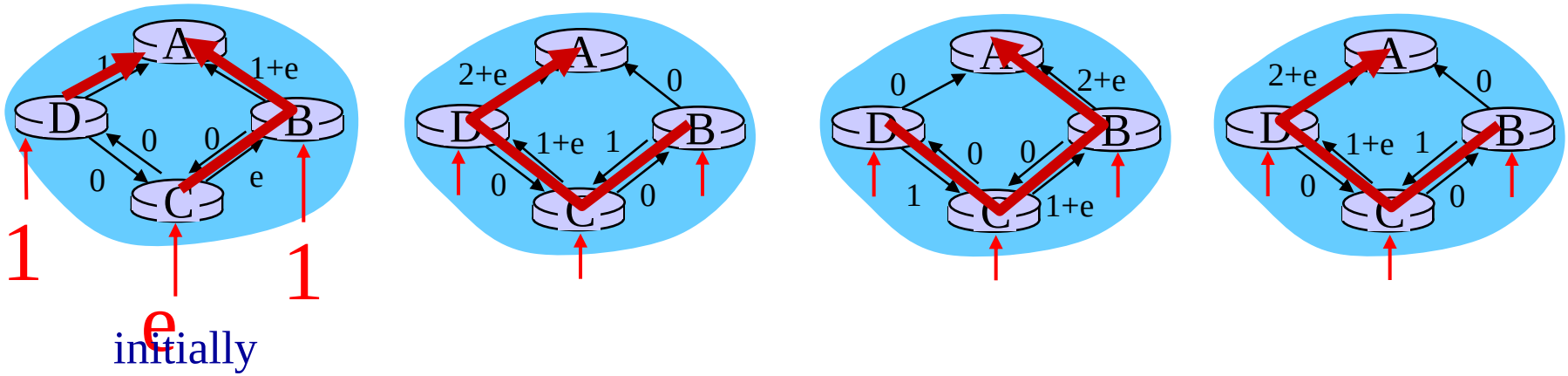
algorithm complexity: n nodes

- each iteration: need to check all nodes, w, not in N
- n(n+1)/2 comparisons

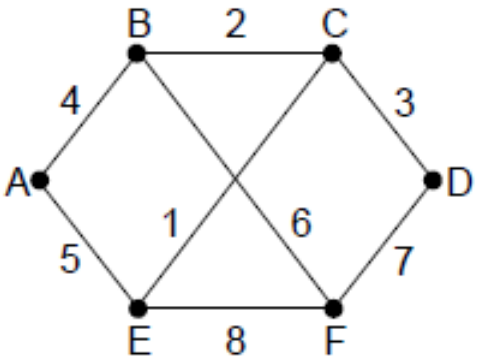
oscillations possible:

- e.g., support link cost equals amount of carried traffic:

given these costs, find new routing resulting in new costs



Building Link State Packets



(a)

Link State Packets

A		B		C		D		E		F	
Seq.		Seq.		Seq.		Seq.		Seq.		Seq.	
Age		Age		Age		Age		Age		Age	
B	4	A	4	B	2	C	3	A	5	B	6
E	5	C	2	D	3	F	7	C	1	D	7
		F	6	E	1			F	8	E	8

(b)

(a) A network. (b) The link state packets for this network.

Lecture XII

D.V. Algorithm

Distance Vector Routing algorithm

- Distance Vector routing is **intra-domain** protocols, **inside** Autonomous system, but not between Autonomous system.(inside one region)
- distance-vector routing are based on the **least-cost** goal.
- Distance Vector developed by **Bellman-Ford** algorithm.
- Bellman **equation** is used to find the **least** cost (**shortest distance**) between a source to destination.
- Base on RIP protocol.

How Does D.V Works

- A **distance vector routing algorithm** operates by having each router **maintain a table** (i.e., a **vector**) giving the best known distance to each destination and which link to use to get there.
- These tables are **updated by exchanging** information with the neighbors router. Every router knows the **best link** to reach each destination.

Distance vector algorithm

Bellman-Ford equation (dynamic programming)

let

$d_x(y)$:= cost of least-cost path from x to y

Then

$$d_x(y) = \min_v \{c(x,v) + d_v(y)\}$$

cost from neighbor v to destination y

cost to neighbor v

min taken over all neighbors v of x

Note: v,x are the neighbors' of source

Distance vector algorithm

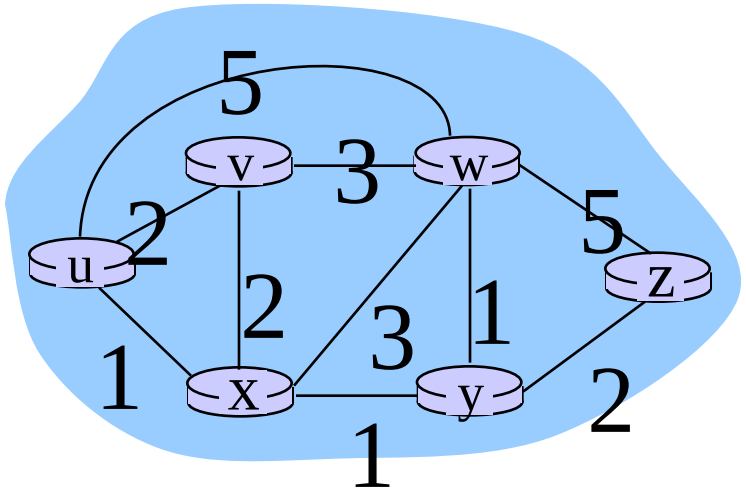
key idea:

- from time-to-time, each node sends its own distance vector estimate to neighbors
- when x receives new DV estimate from neighbor, it updates its own DV using B-F equation:

$$D_x(y) \leftarrow \min_v\{c(x,v) + D_v(y)\} \quad \text{for each node } y \in N$$

- under minor, natural conditions, the estimate $D_x(y)$ *converge to the actual least cost* $d_x(y)$

Bellman-Ford example



clearly Distance V of sources to destination

$d_v(z) = 5,\ d_x(z) = 3,\ d_w(z) = 3$

(neighbors' of source)

then B-F equation says:

$$d_u(z) = \min \{ c(u,v) + d_v(z),\ c(u,x) + d_x(z),\ c(u,w) + d_w(z) \}$$
$$= \min \{2 + 5,\ 1 + 3,\ 5 + 3\} = 4$$

node achieving minimum is next hop in shortest path, used in forwarding table

Distance vector algorithm

each node:

wait for (change in local link cost or msg from neighbor)

recompute estimates

if DV to any dest has changed, *notify* neighbors

$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\}$
$= \min\{2+0 , 7+1\} = 2$

$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\}$
$= \min\{2+1 , 7+0\} = 3$

node x table

from \ cost to	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

node y table

from \ cost to	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

from \ cost to	x	y	z
x	0	2	7
y	2	0	1
z	7	1	0

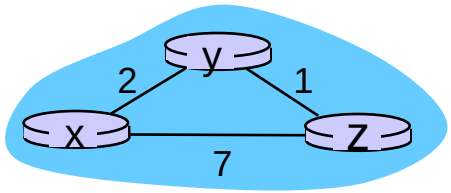
from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

node z table

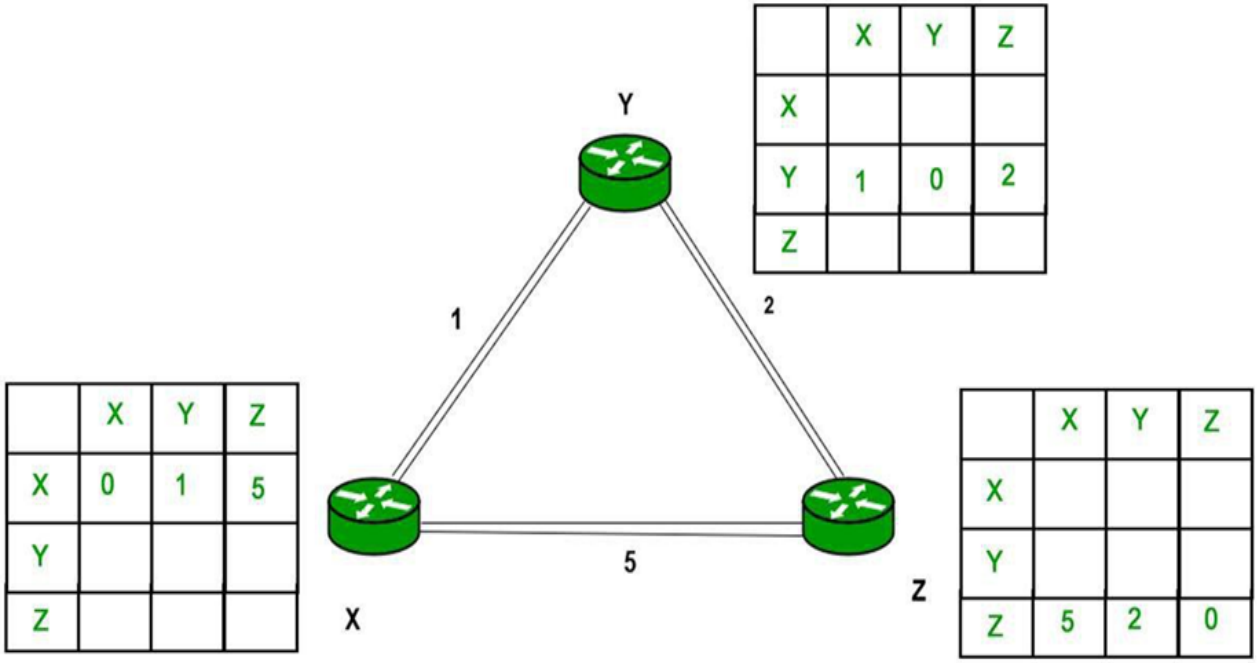
from \ cost to	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

from \ cost to	x	y	z
x	0	2	7
y	2	0	1
z	3	1	0

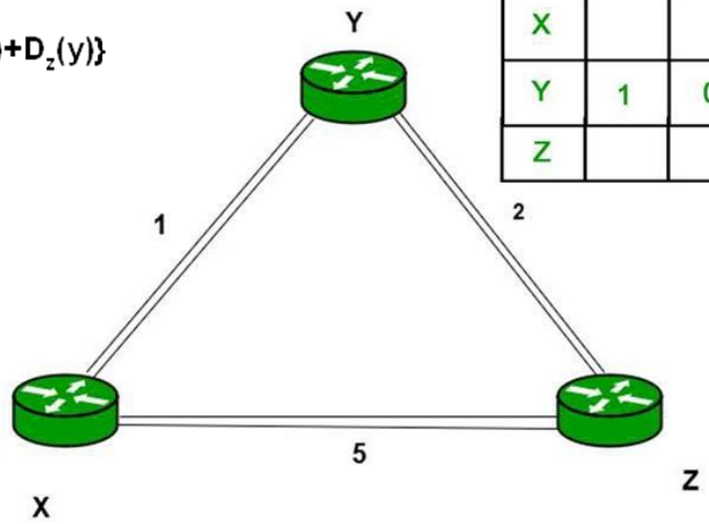
from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0



time


Y
1
2
5
X
Z
X Y Z
X
Y 1 0 2
Z
X Y Z
X 0 1 5
Y
Z
X Y Z
X
Y
Z 5 2 0

$D_x(y)=\min\{c(x,y)+D_y(y), c(x,z)+D_z(y)\}$
$= \min\{1+0 , 5+2\} = 1$

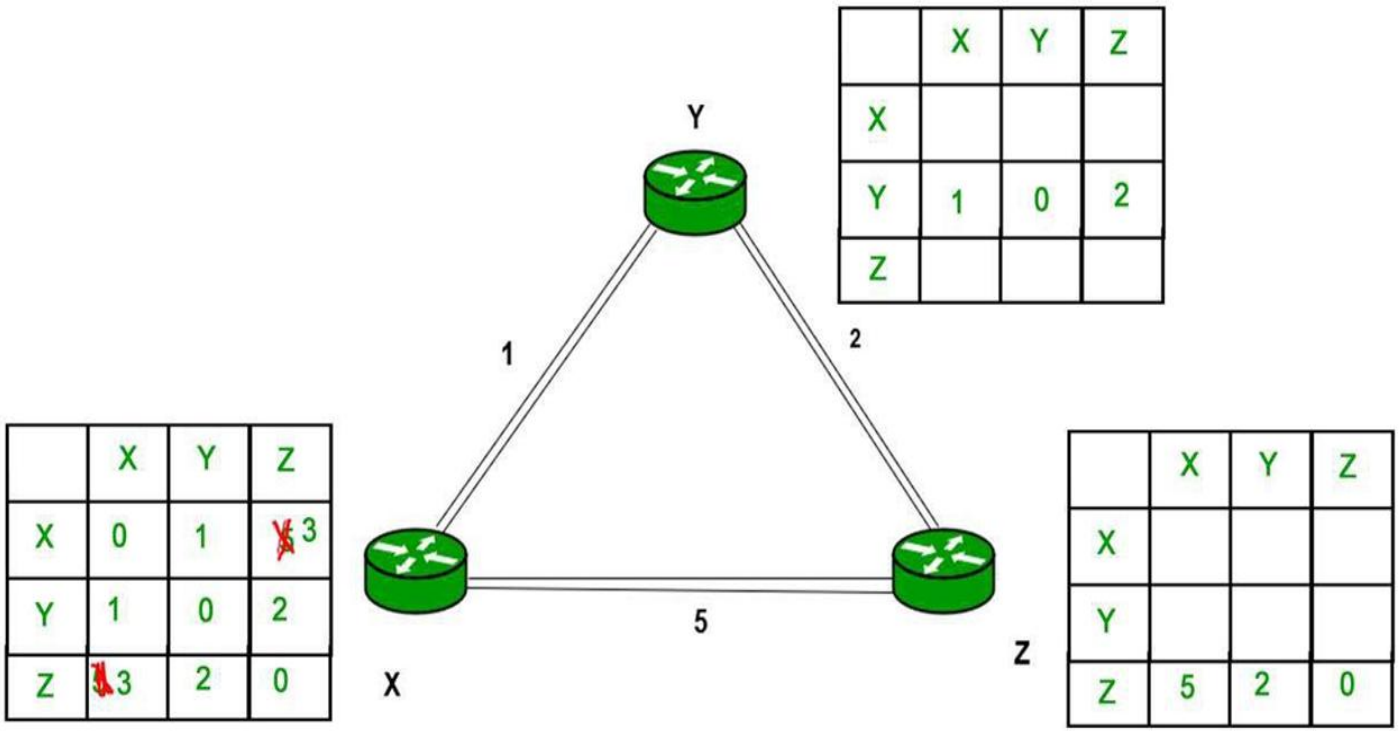


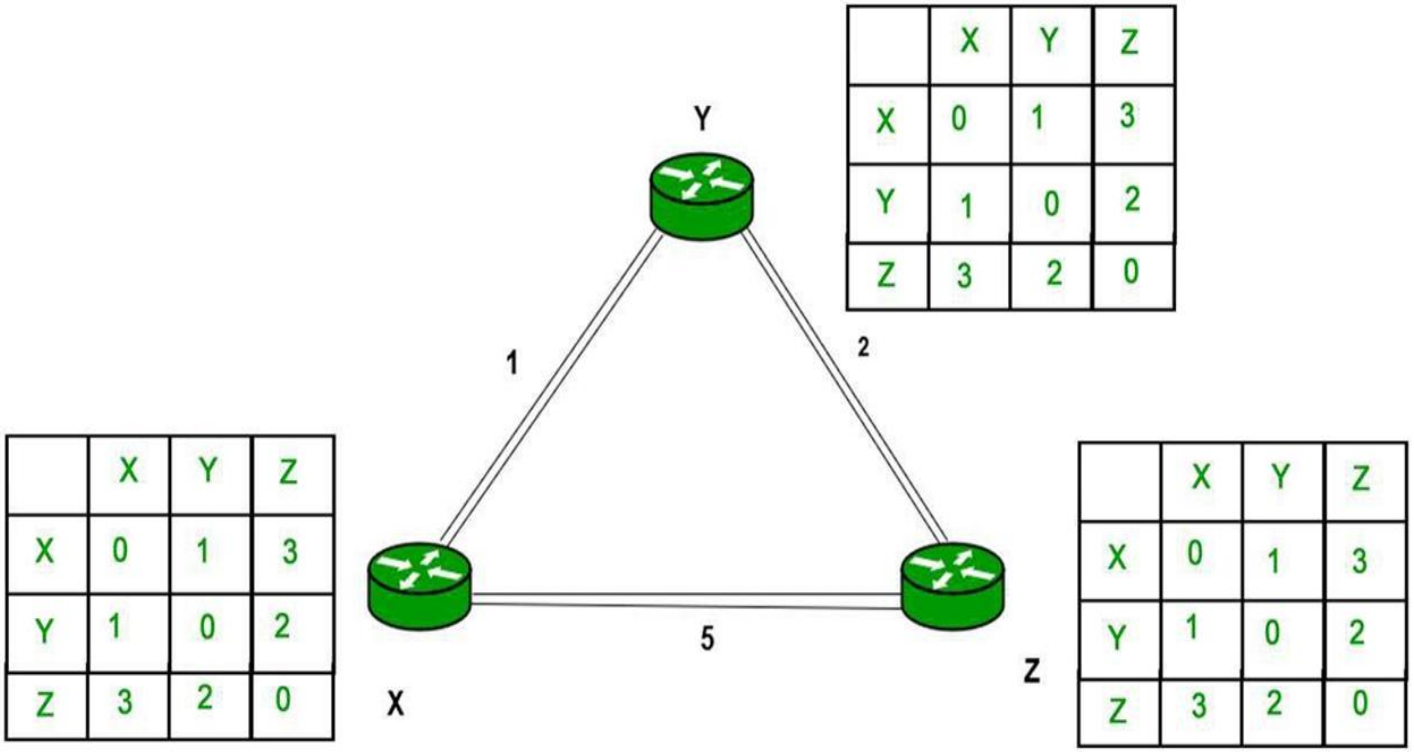
	X	Y	Z
X			
Y	1	0	2
Z			

	X	Y	Z
X	0	1	~~4~~ 3
Y	1	0	2
Z			

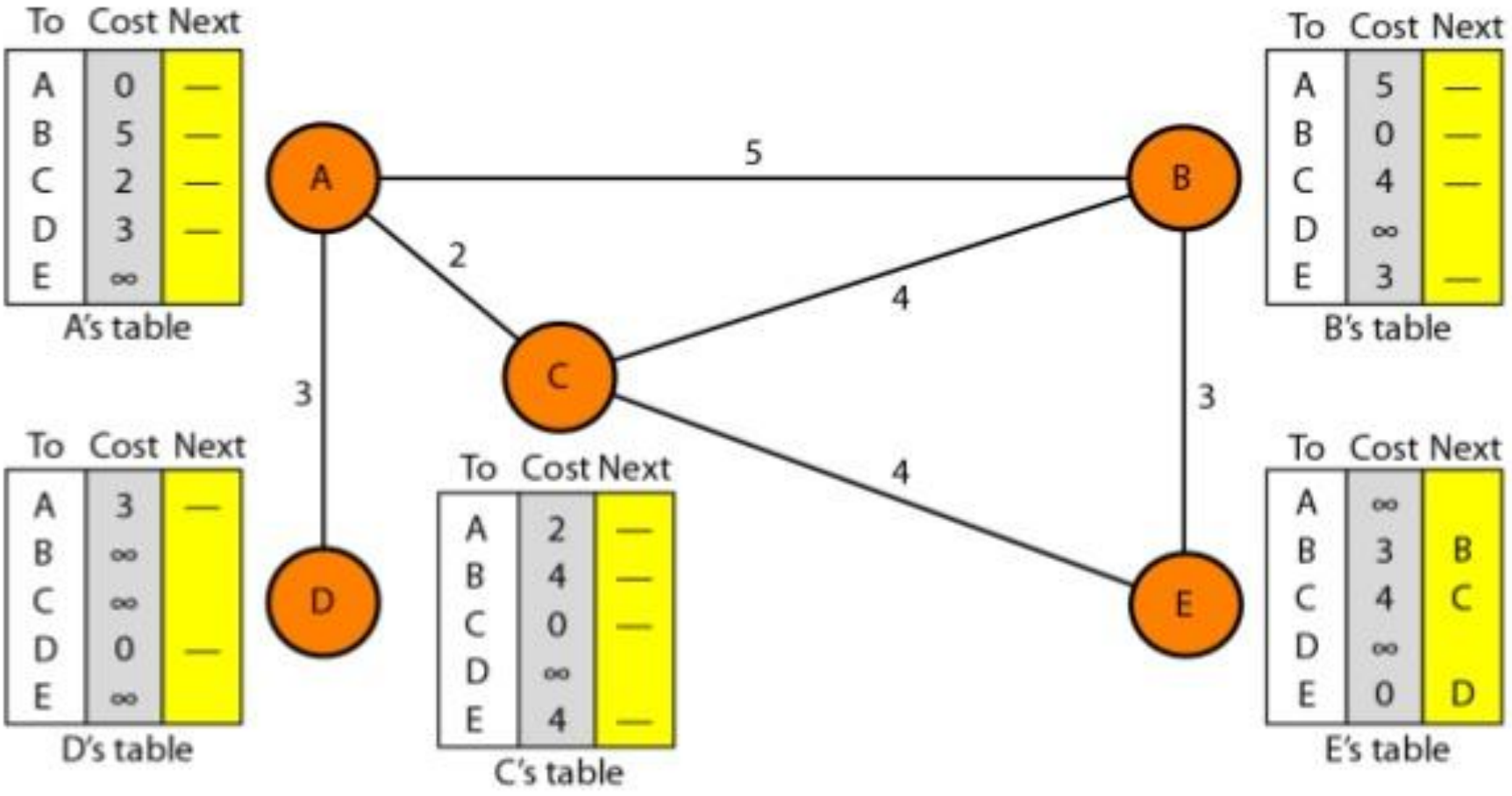
	X	Y	Z
X			
Y			
Z	5	2	0

Dx(z) = min{c(x,y) +
Dy(z), c(x,z) + Dz(z)}
= min{ 2+1, 5+0} = 3


Y
1
2
5
X
Z
X Y Z
X
Y 1 0 2
Z
X Y Z
X 0 1 3
Y 1 0 2
Z 3 2 0
X Y Z
X
Y
Z 5 2 0

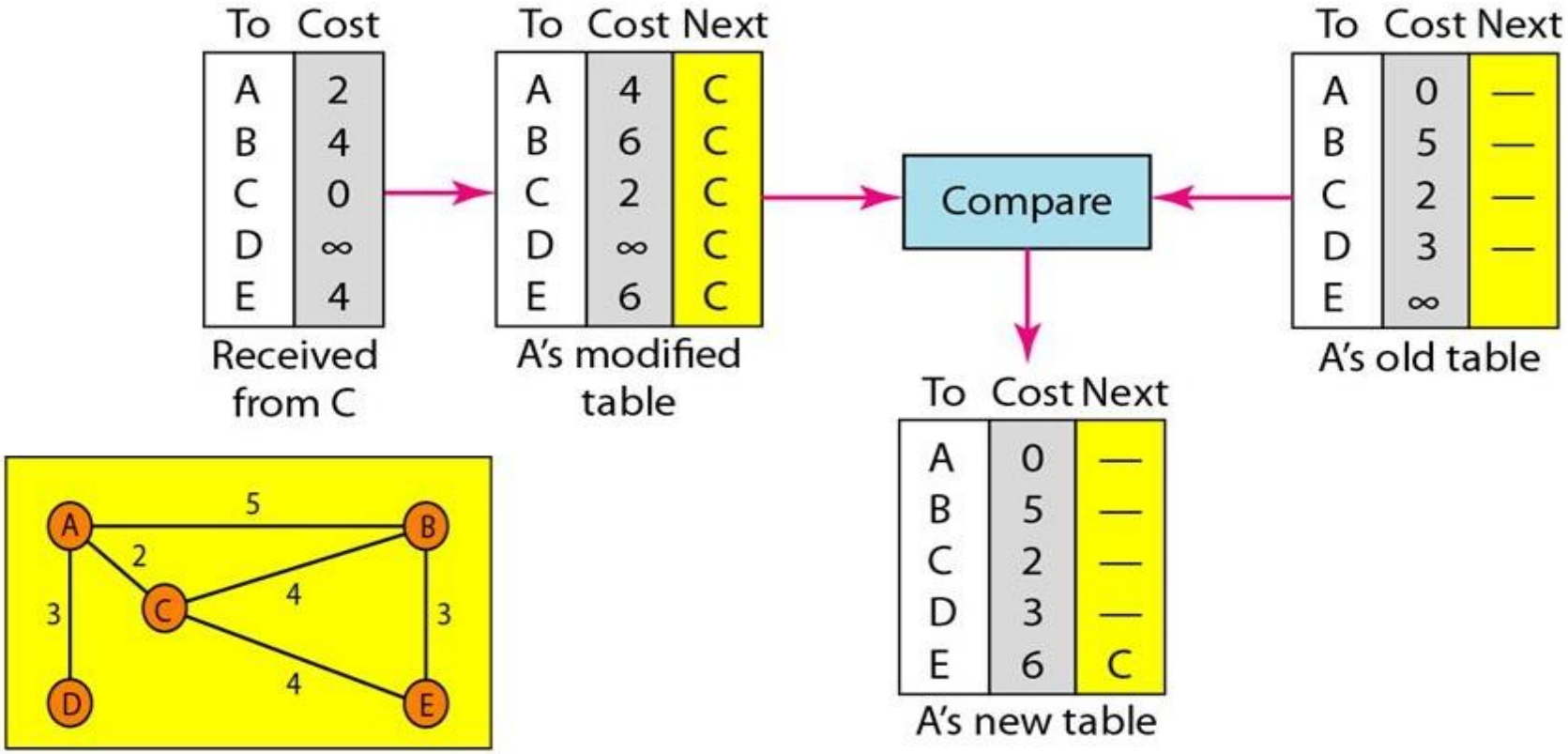

Y
1
2
5
X
Z
X Y Z
X 0 1 3
Y 1 0 2
Z 3 2 0
X Y Z
X 0 1 3
Y 1 0 2
Z 3 2 0
X Y Z
X 0 1 3
Y 1 0 2
Z 3 2 0

Initialization of tables in distance vector routing (DVR)



Distance vector routing

Sharing: Updating in distance vector routing



Comparison of LS and DV algorithms

message complexity

- ***LS:*** with n nodes, E links, O(nE) msgs sent
- ***DV:*** exchange between neighbors only
  - convergence time varies

speed of convergence

- ***LS:*** $O(n^2)$ algorithm requires O(nE) msgs
  - may have oscillations
- ***DV:*** convergence time varies
  - may be routing loops
  - count-to-infinity problem

Robustness (Failure): what happens if router malfunctions?

LS:

- node can advertise incorrect *link* cost
- each node computes only its *own* table

DV:

- DV node can advertise incorrect *path* cost
- each node's table used by others
- error propagate thru network

Lecture XIII

Hieratical routing Algorithm

Hierarchical Routing

- As networks grow in size, the router routing tables grow proportionally.
- Not only the router memory consumed by ever-increasing tables, but more CPU time is needed to scan them and more bandwidth is needed to send status reports about them.
- So router can not have table about the entire network.

Hierarchical Routing

- When hierarchical routing is used, *the routers are divided into what we will call* ***regions*****.**
- *Each router knows all the details about how to route packets to destinations within its* ***own region*** *but knows nothing about the internal structure of other regions*.**(ie EGRP protocol IGRP)**

Hierarchical routing

- routers into regions, "autonomous systems" (AS)
- routers in same AS run same routing protocol called: IGBP

  (Interior Gate Way Protocol)

  - routers in different AS can run different intra-AS routing protocol called:EGBP (Exterior Gate Way Protocol.)

this will be studied next lecture.

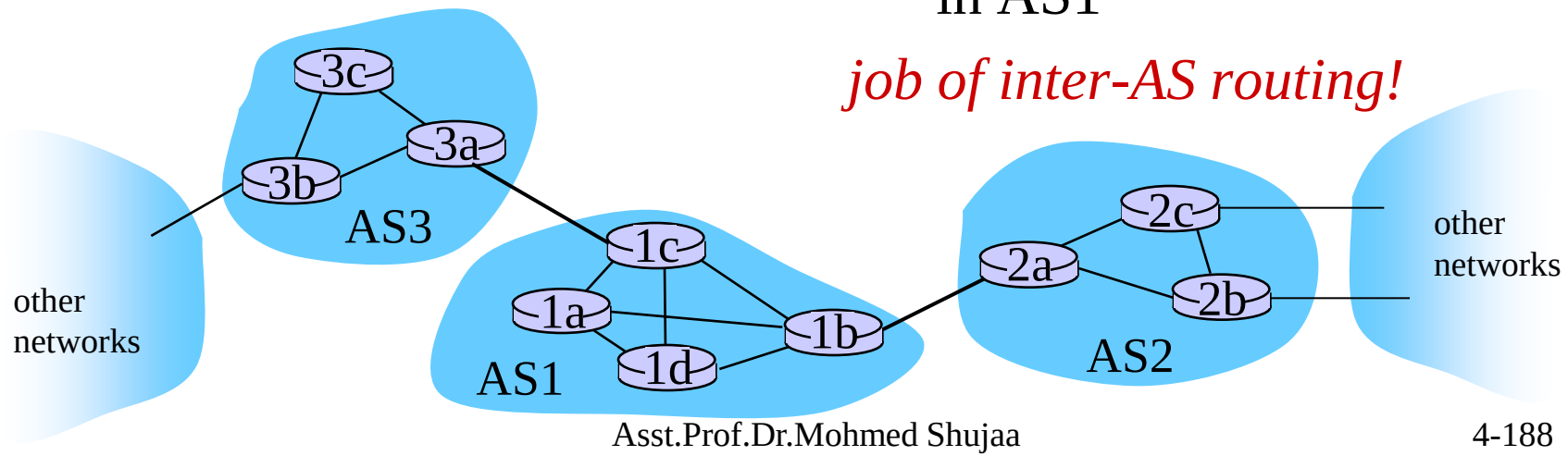
Inter-AS tasks Example:

- suppose router in AS1 receives datagram destined outside of AS1:
  - router should forward packet to gateway router, but which one?

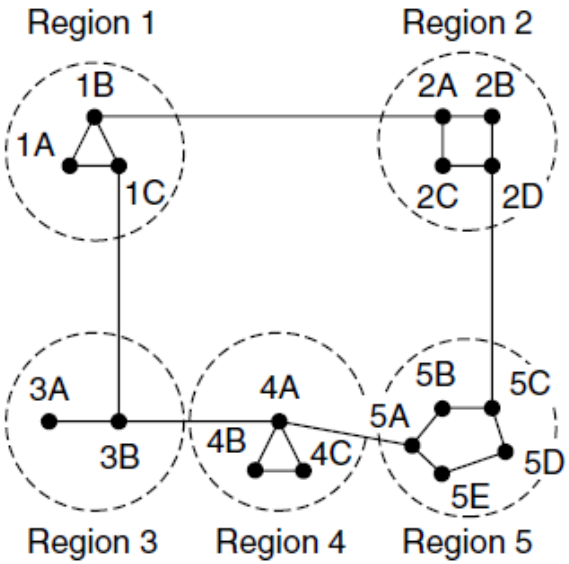
AS1 must:

1. learn which dests are reachable through AS2,and which through AS3
2. propagate this reach ability info to all routers in AS1

job of inter-AS routing!



Hierarchical routing.



(a)

Full table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

(b)

Hierarchical table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

(c)

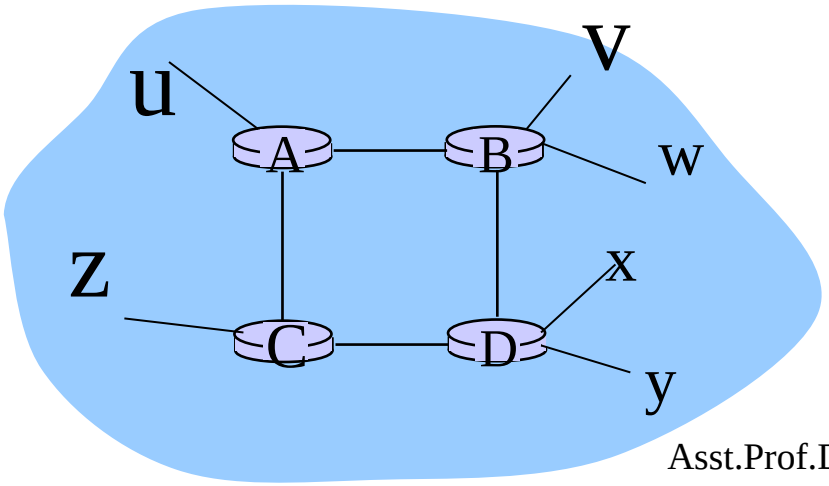
Asst.Prof.Dr.Mohmed Shujaa

Lecture XIV

Network Protocols

RIP (Routing Information Protocol)

- It is a very simple protocol based on **distance vector routing**.
- distance vector algorithm
  - distance metric: # hops (max = 15 hops), each link has cost 1
  - DVs exchanged with neighbors every 30 sec in response message (aka advertisement)
  - each advertisement: list of up to 25 destination *subnets (in IP addressing sense)*



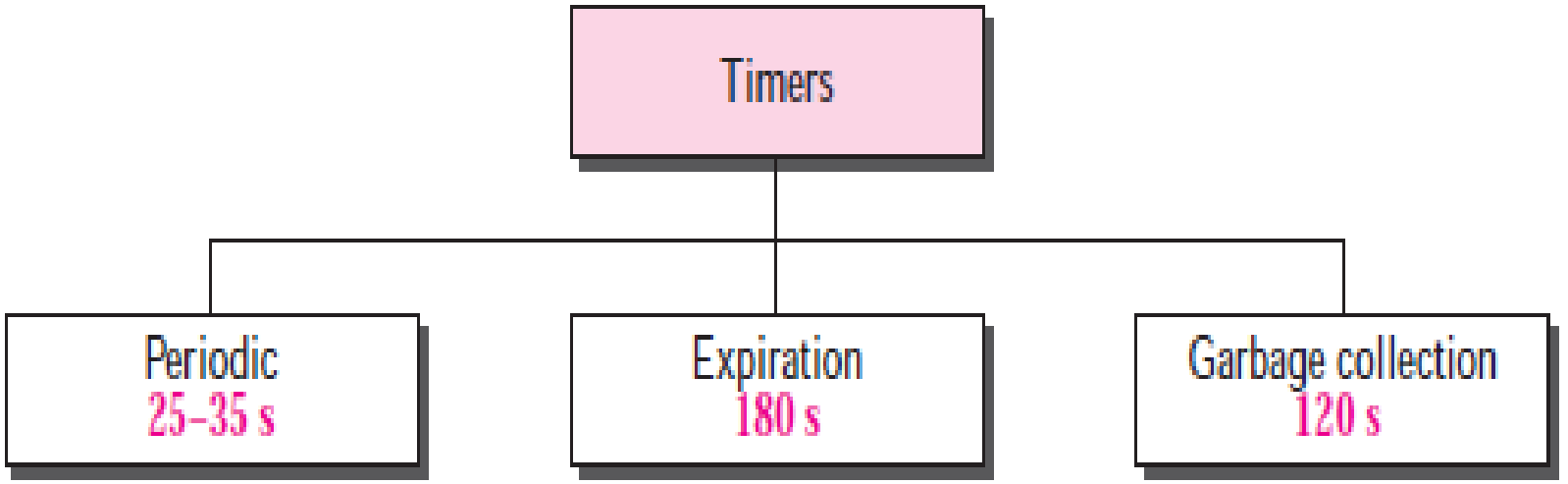
from router A to destination *subnets:*

subnet	hops
u	1
v	2
w	2
x	3
y	3
z	2

RIP Characteristics

- Distance vector routing protocol.
- Uses hop count as a path selection metric.
- Three types of timers.
- Multiple stability features.

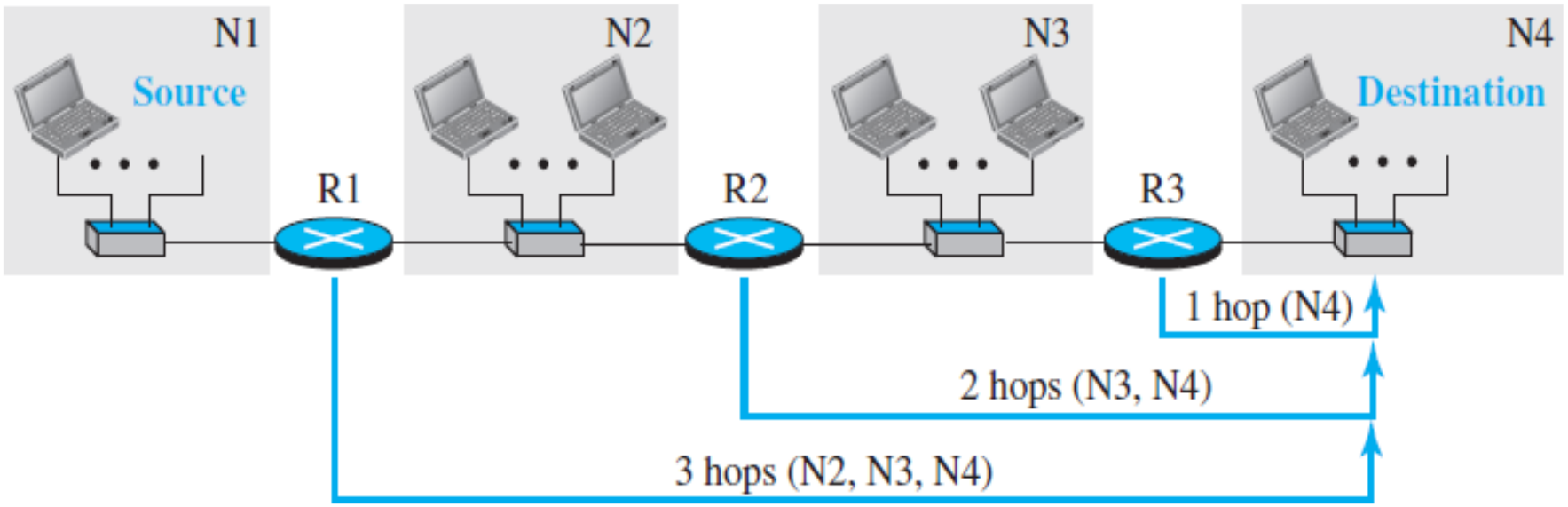
Timers in RIP



Rip timer

- The **periodic timer** **controls the advertising of regular update messages.**
- ***Expiration Timer*** if there is a problem on an internet and no update is received within the allotted 180 s, the route is considered expired and the hop count of the route is set to 16, which means the destination is unreachable.(Because 15 hop max)
- ***Garbage Collection Timer*** When the information about a route becomes invalid.

Hop Count -- 15 Hop Limit

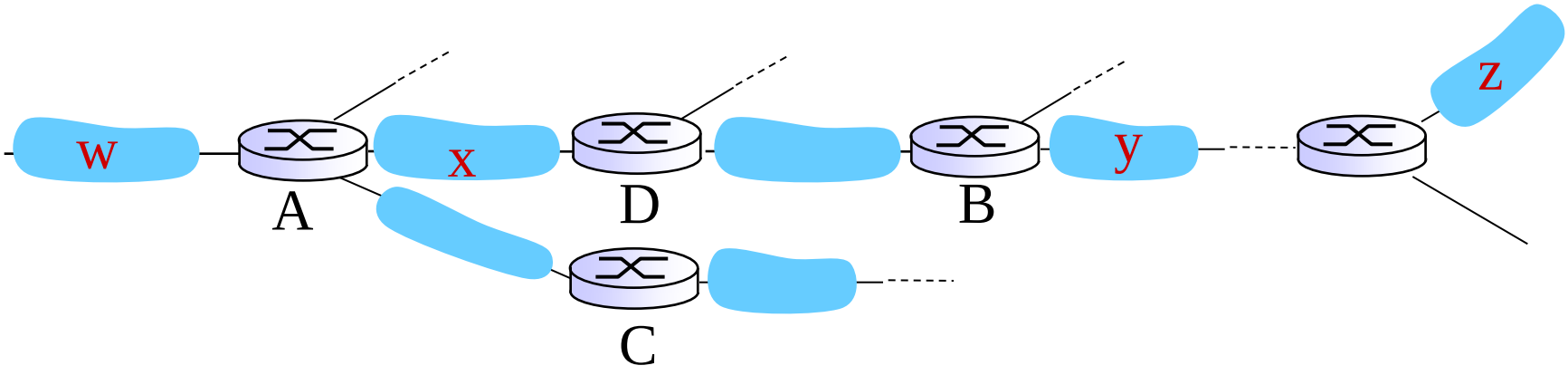


Forwarding table for R1

Destination network	Next router	Cost in hops
N1	——	1
N2	——	1
N3	R2	2
N4	R2	3

Note: After 15 hop the destination unreachable

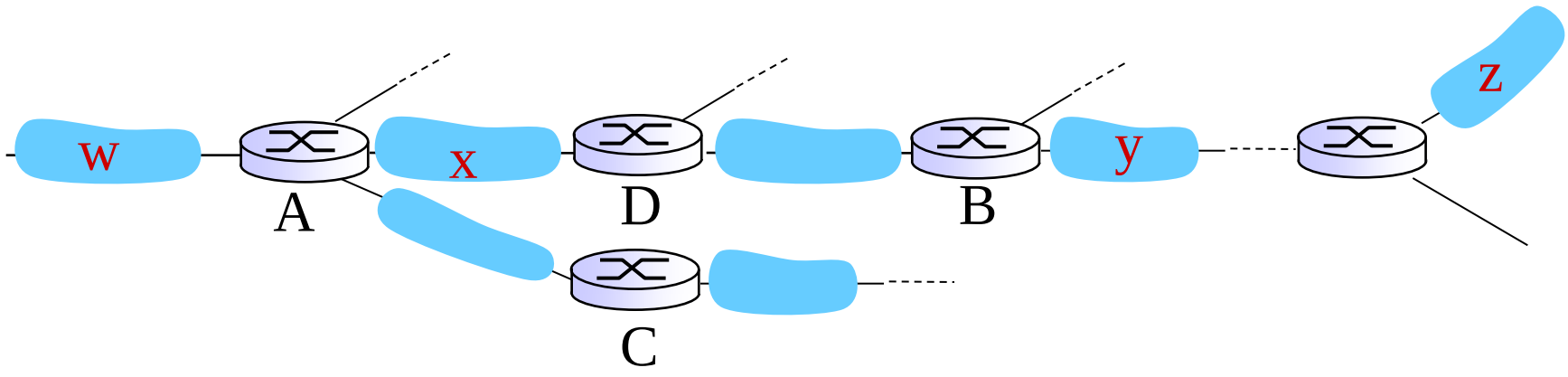
RIP: example



routing table in router D

destination subnet	next router	# hops to dest
w	A	2
y	B	2
z	B	7
x	--	1
….	….	....

RIP: example



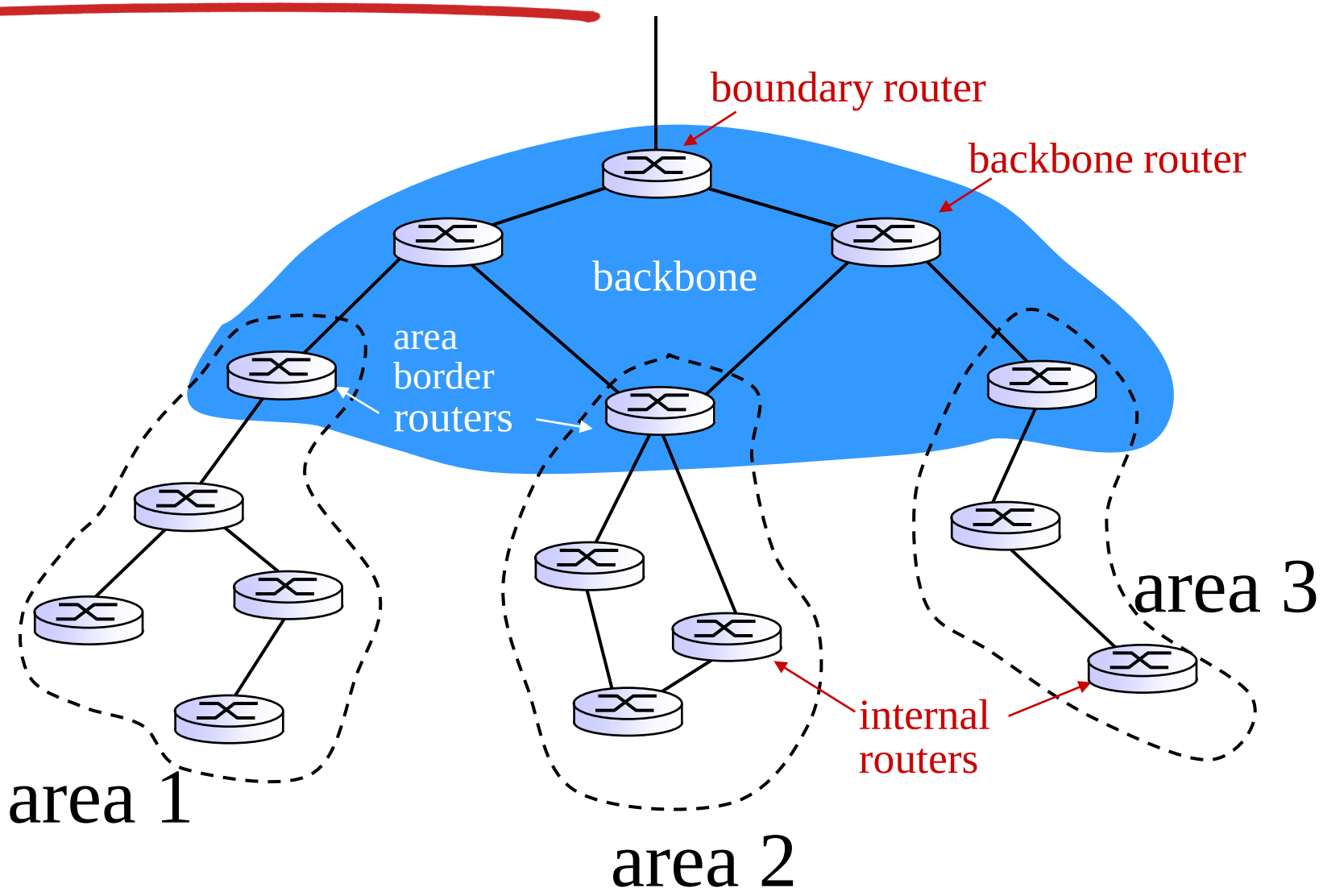
routing table in router D

destination subnet	next router	# hops to dest
w	A	2
y	B	2
z	~~B~~ A	~~7~~ 5
x	--	1
….	….	....

OSPF (Open Shortest Path First)

- "open": publicly available
- uses link state algorithm
  - LS packet dissemination
  - topology map at each node
  - route computation using Dijkstra's algorithm
- OSPF advertisement *carries one entry per neighbor*
- advertisements flooded to *entire* AS
  - carried in OSPF messages directly *over IP* (rather than TCP or UDP
- *security:* all OSPF messages authenticated (to prevent malicious intrusion)
- multiple same-cost paths allowed (only one path in RIP)

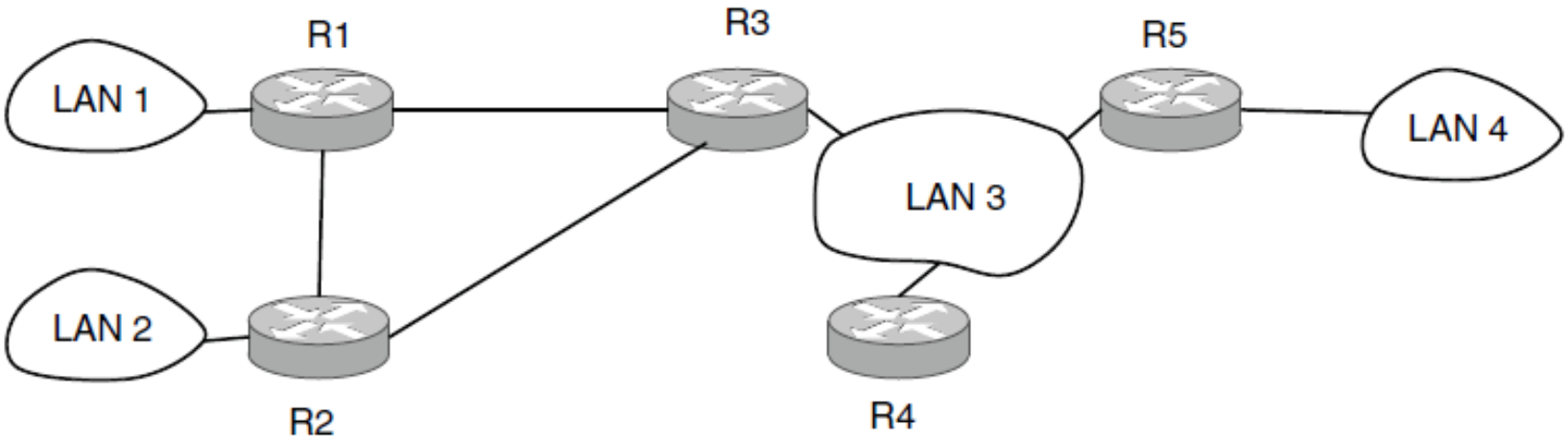
Hierarchical OSPF



Hierarchical OSPF

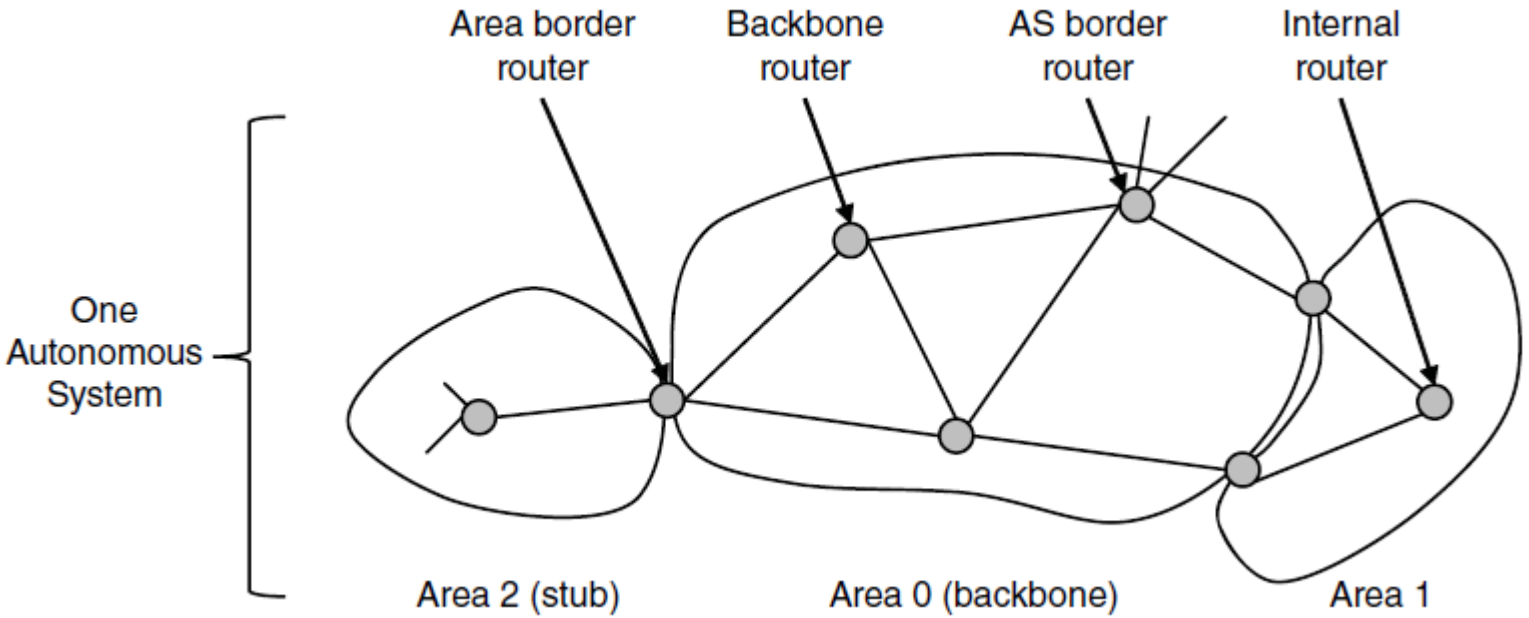
- *two-level hierarchy:* local area, backbone.
  - link-state advertisements only in area
  - each nodes has detailed area topology; only know direction (shortest path) to nets in other areas.
- *area border routers:* "summarize" distances to nets in own area, advertise to other Area Border routers.
- *backbone routers:* run OSPF routing limited to backbone.
- *boundary routers:* connect to other AS's.

OSPF—An Interior Gateway Routing Protocol



An autonomous system

OSPF—An Interior Gateway Routing Protocol



The relation between ASes, backbones, and areas in OSPF.

OSPF—An Interior Gateway Routing Protocol

Message type	Description
Hello	Used to discover who the neighbors are
Link state update	Provides the sender's costs to its neighbors
Link state ack	Acknowledges link state update
Database description	Announces which updates the sender has
Link state request	Requests information from the partner

The five types of OSPF messages

Lecture XV
protocols

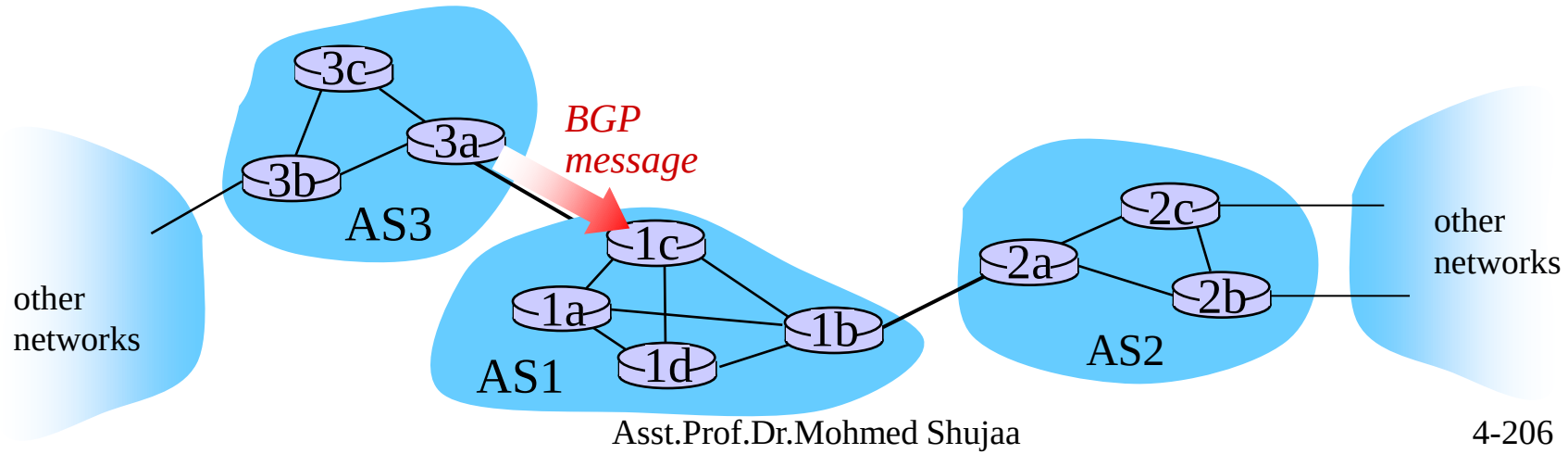
Internet inter-AS routing: BGP

- BGP (Border Gateway Protocol): inter-domain routing protocol
  - “ holds the Internet together”
- BGP provides :
  - eBGP: obtain subnet reach ability information from neighboring ASs.
  - iBGP: propagate reach ability information to all AS-internal routers.
- allows subnet to advertise its existence to rest of Internet:

BGP basics

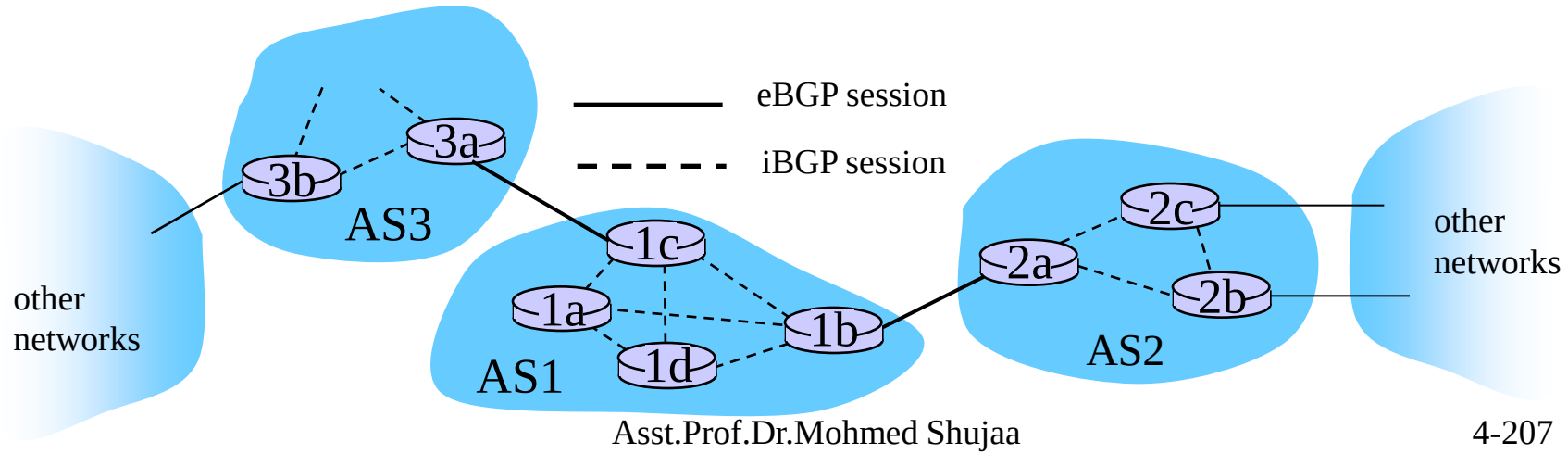
- BGP session: two BGP routers ("peers") exchange BGP messages:
  - advertising *paths* to different destination network prefixes ("path vector" protocol)
  - exchanged over semi-permanent *TCP connections*

- The IBGP used to connect different routers have same AS(same company)
- The EBGP used to connect different routers have different AS(different company)



BGP basics: distributing path information

- using eBGP session between 3a and 1c, AS3 sends prefix reachability info to AS1.
  - 1c can then use iBGP do distribute new prefix info to all routers in AS1
  - 1b can then re-advertise new reachability info to AS2 over 1b-to-2a eBGP session
- when router learns of new prefix, it creates entry for prefix in its forwarding table.



BGP messages

- BGP messages exchanged between peers over TCP connection
- BGP messages:
  - OPEN: opens TCP connection to peer and authenticates sender
  - UPDATE: advertises new path (or withdraws old)
  - KEEP ALIVE: keeps connection alive in absence of UPDATES; also ACKs OPEN request
  - NOTIFICATION: reports errors in previous msg; also used to close connection

DHCP &IP addresses: how to get one

Q: How does a *host* get IP address?

- hard-coded by system admin in a file
  - Windows: control-panel->network->configuration->tcp/ip->properties
  - UNIX: /etc/rc.config

DHCP: Dynamic Host Configuration Protocol: dynamically get address from as server

- "plug-and-play"

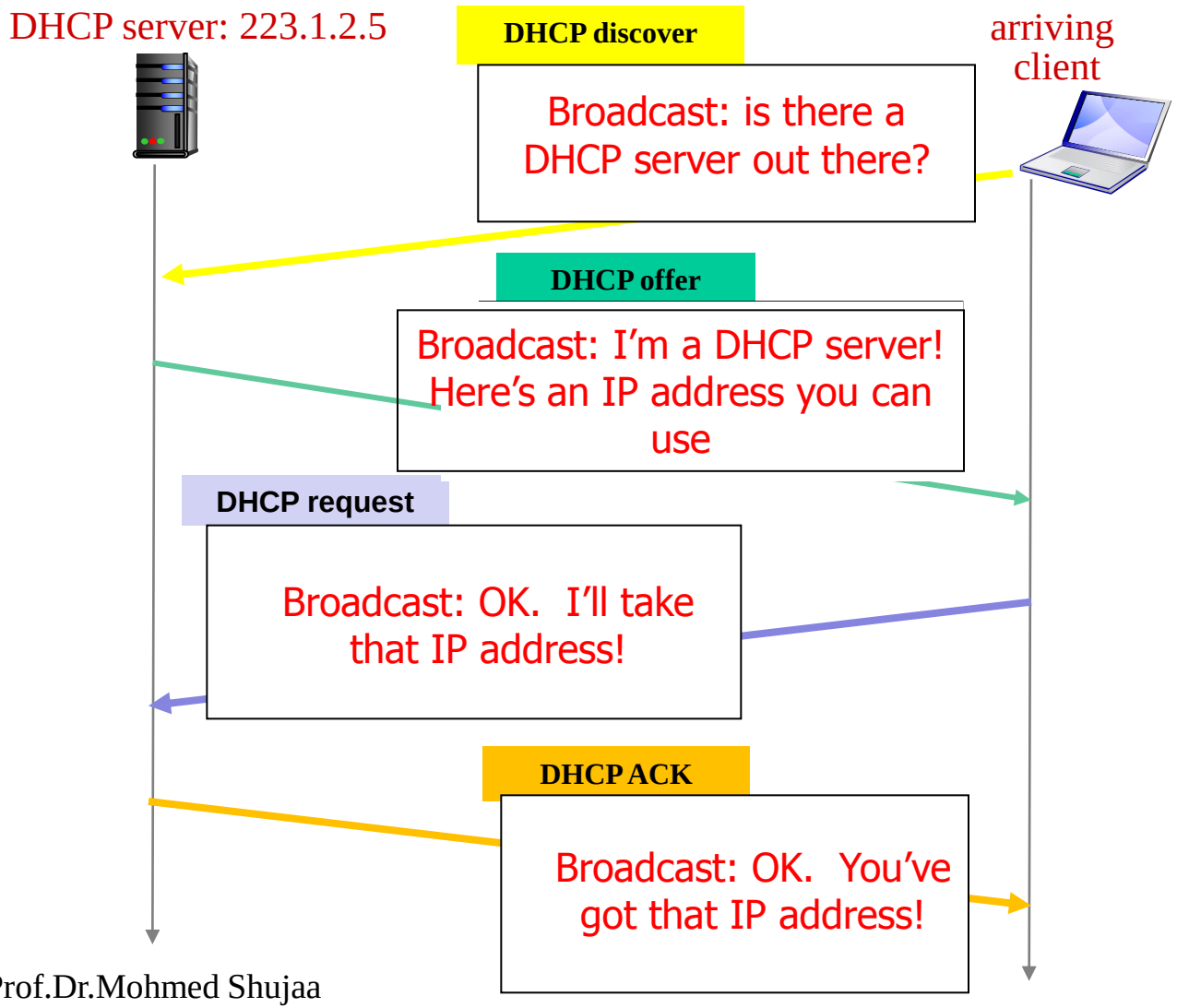
DHCP: Dynamic Host Configuration Protocol

goal: allow host to *dynamically* obtain its IP address from network server when it joins network

DHCP overview: (pool operation)

- host broadcasts "DHCP discover" msg [optional]
- DHCP server responds with "DHCP offer" msg [optional]
- host requests IP address: "DHCP request" msg
- DHCP server sends address: "DHCP ack" msg

DHCP client-server scenario

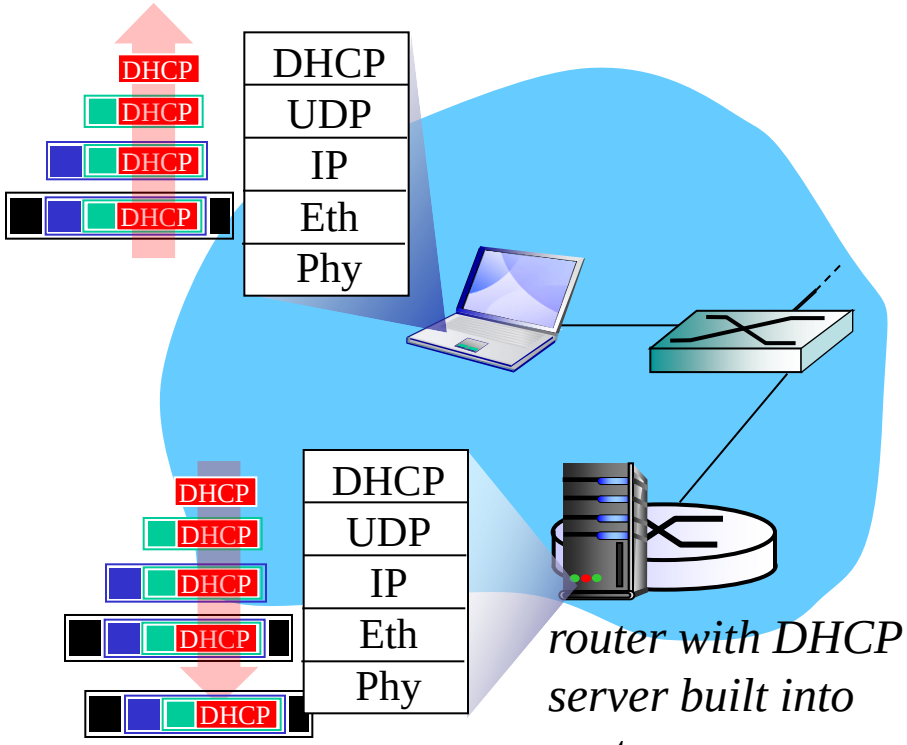


DHCP: more than IP addresses

DHCP can return more than just allocated IP address on subnet:

- address of first-hop router for client
- name and IP address of DNS sever

DHCP: example



router with DHCP server built into router

- DCP server formulates DHCP ACK containing client's IP address, IP address of first-hop router for client, name & IP address of DNS server

- encapsulation of DHCP server, frame forwarded to client, demuxing up to DHCP at client
- client now knows its IP address, name and IP address of DNS server, IP address of its first-hop router

ICMP: internet control message protocol

- used by hosts & routers to communicate network-level information
  - error reporting: unreachable host, network, port, protocol
  - echo request/reply (used by ping)
  - ICMP msgs carried in IP datagrams
- ICMP message: type, code plus first 8 bytes of IP datagram causing error

Type	Code	description
0	0	echo reply (ping)
3	0	dest. network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control - not used)
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

ICMP Applications

- **PING: The ping checks whether a host is alive & reachable or not. This is done by sending an ICMP Echo Request packet to the host, and waiting for an ICMP Echo Reply from the host.**

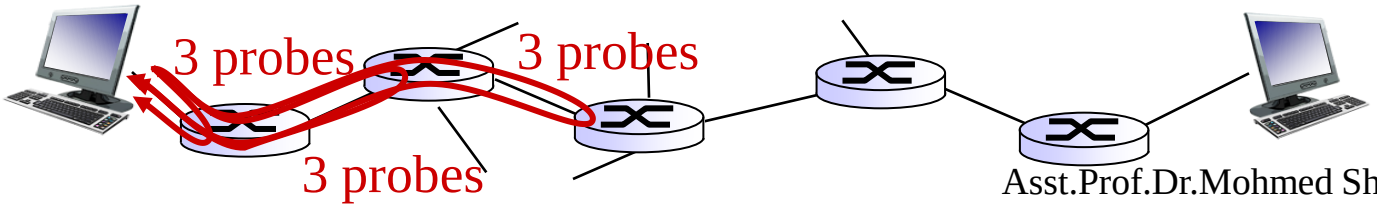
- **TRACE ROUTE: Trace route is a used to records the route through the Internet between your computer and a specified destination computer. It also calculates and displays the amount of time each hop took.**

Trace route and ICMP

- source sends series of UDP segments to dest
  - first set has TTL =1
  - second set has TTL=2, etc.
  - unlikely port number
- when *n*th set of datagrams arrives to nth router:
  - router discards datagrams
  - and sends source ICMP messages (type 11, code 0)
  - ICMP messages includes name of router & IP address
- when ICMP messages arrives, source records RTTs

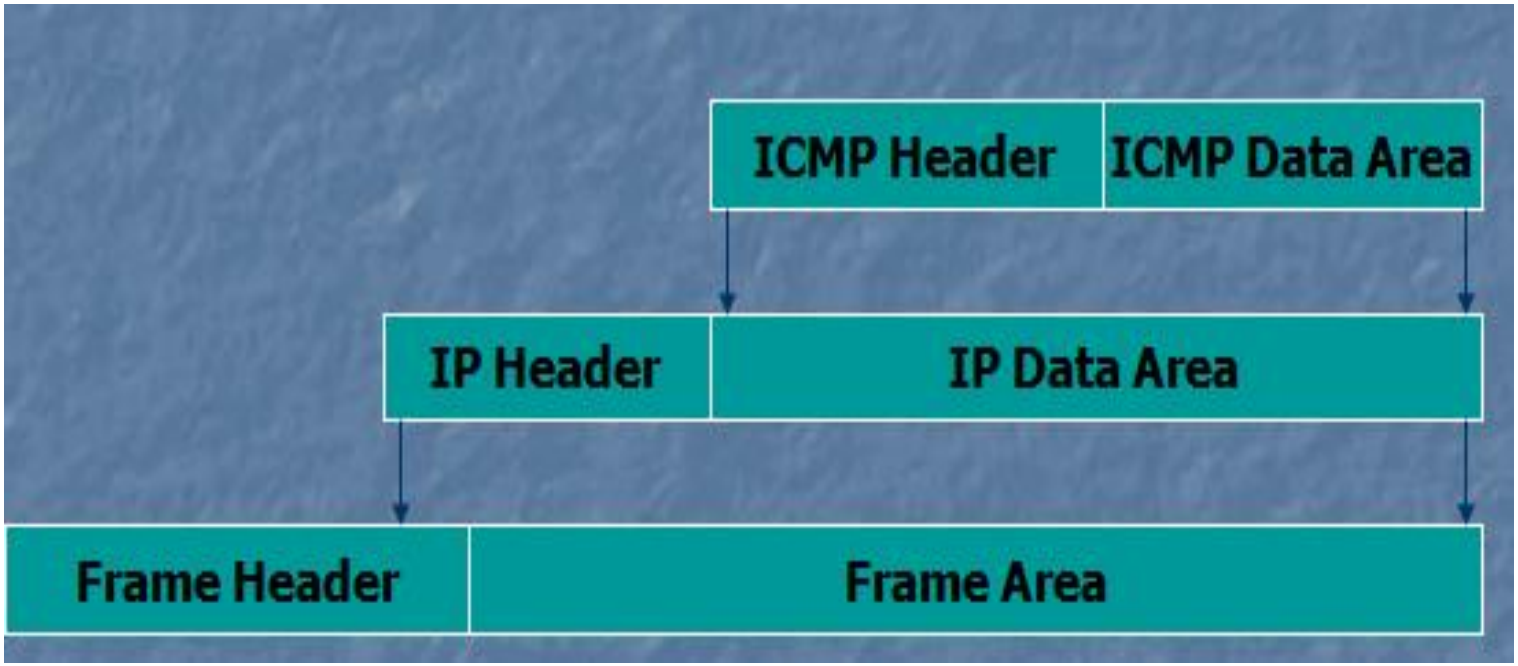
stopping criteria:

- UDP segment eventually arrives at destination host
- destination returns ICMP "port unreachable" message (type 3, code 3)
- source stops



ICMP

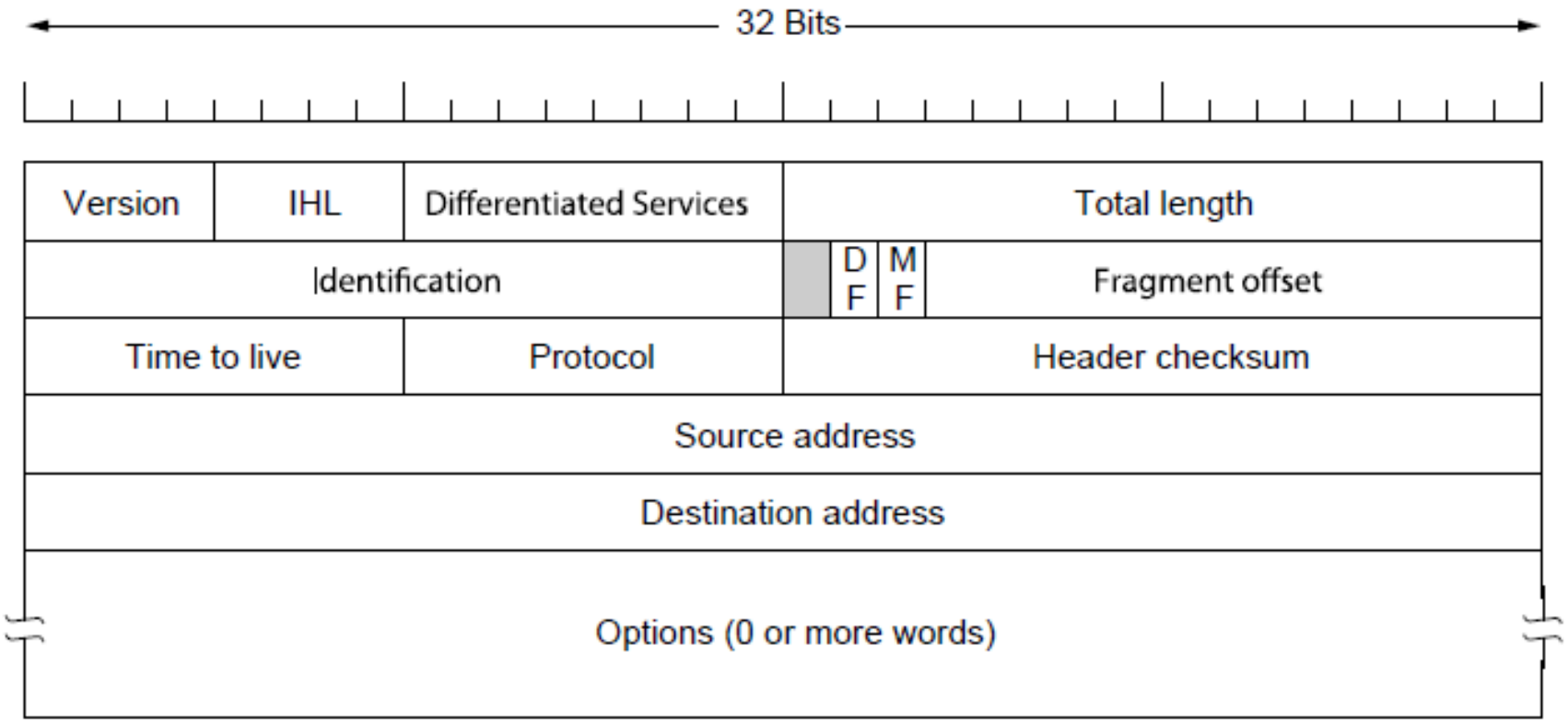
- ICMP messages carried on IP packet.



Lecture XVI

IP Protocols

IPV4 datagram format



IPV4 datagram format

- **Version**: The 4-bit field defines the version of the ipv4 protocol.

- **Header Length** : 4-bit field defines total length of datagram header in 4-bit words When the option filed is at the maximum size ,the value of this filed is 15 ⇒HLEN=60 .

IPV4 datagram format

Total Length

- 16-bit field defines total datagram length in bytes, including header .
- 16 bits ⇒ maximum size = 65,535 bytes (class depends)
- some physical networks are not able to encapsulate a datagram of 65,535 bytes, so datagram must be fragmented to be able to pass through those networks.

IPV4 datagram format

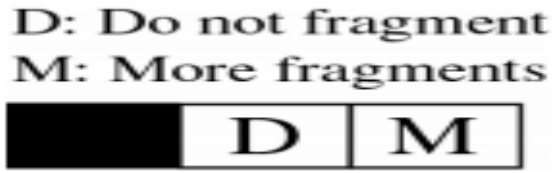
- **Identification :**16-bit field – uniquely identifies datagram originating from source host.

- N.B. when datagram is fragmented identification field is copied into all fragments.

IPV4 datagram format

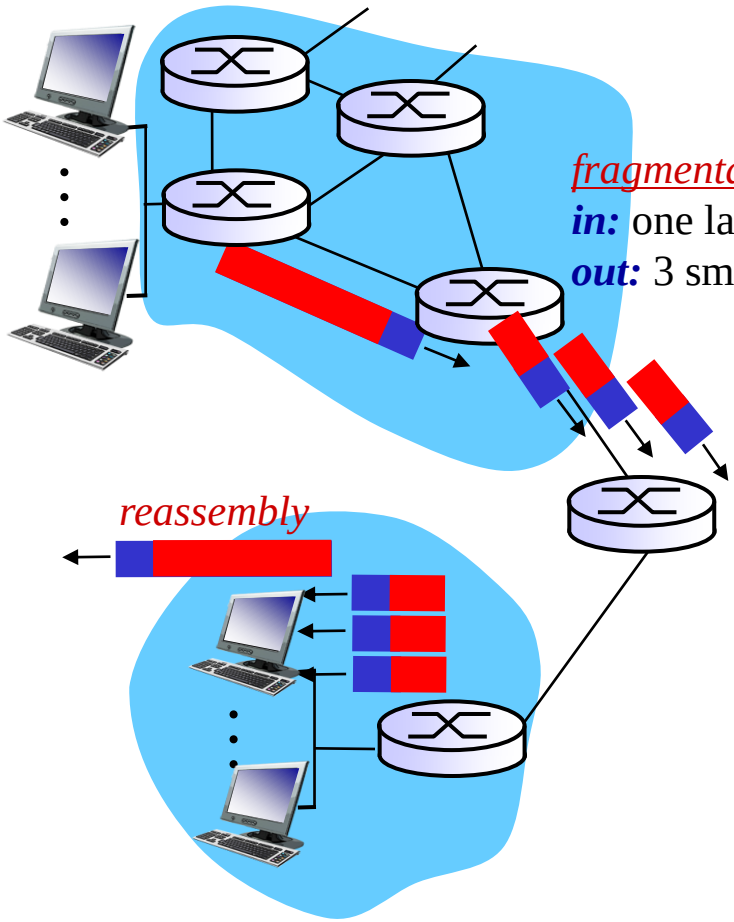
Flags :

- 3-bit field .
- 1st bit is reserved .
- 2nd bit is called "do not fragment" bit , if its value is 1, machine must NOT fragment datagram , if its value is 0,the datagram can be fragment if necessary.
- 3rd bit is called "more fragment" bit ,if its value is 1, datagram is not last fragment – there are more fragments after this one , if its value is 0, this is last or only fragment.



IP fragmentation,

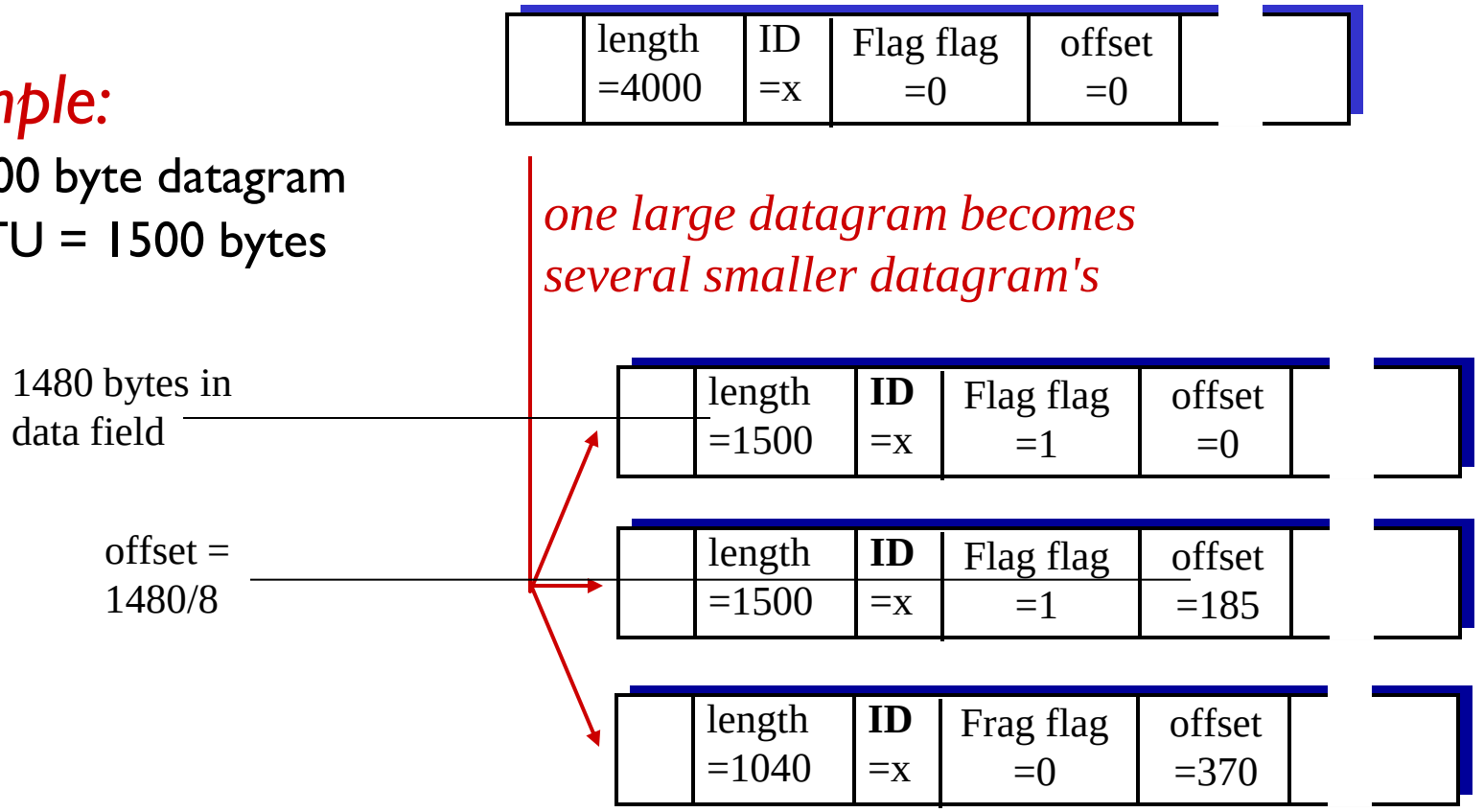
- network links have MTU (max transfer unit (size) - largest possible link-level frame
- large IP datagram divided ("fragmented") within net
  - IP header bits used to identify, order related fragments



IP fragmentation, reassembly

example:

- 4000 byte datagram
- MTU = 1500 bytes



Fragment Bytes ID Offset Flag

1st fragment 1,480 bytes in

the data field of

the IP datagram

identification 777 offset 0 (meaning the data should be inserted beginning at byte 0)

2nd fragment 1,480 bytes

of data

identification 777 offset 185 (meaning the data should be inserted beginning at byte 1,480. Note that 185 · 8 1,480)

3rd fragment 1,020 bytes

(3,980–1,480–1,480) of data

identification 777 offset 370 (meaning the data should be inserted beginning at byte 2,960. Note that 370 · 8 2,960)

flag 1 (meaning there is more)

flag 1 (meaning there is more)

flag 0 (meaning this is the last fragment)

IPV4 datagram format

Time-To-Live protocol (TTL) :

- 8-bit field controls max number of hops visited by datagram and/or time spend in the network .

- field is decremented by one each time datagram is processed by a router – when TTL reaches 0, datagram must be dropped .

IPV4 datagram format

Protocol : 8-bit Identifies the higher-level protocol.

protocol number is glue that binds network & transport layer, while port number is glue that binds transport & application layer.

values: 1 = ICMP , 2 = IGMP , 6 = TCP ,

17 = UDP , 89 = OSPF.

Header Checksum : 16-bit field – aids in detecting errors in header only .

IPV4 datagram format

- **Source and Destination IP Addresses** : 32-bit fields must remain unchanged until IP datagram reaches its final destination.

- **Options :** 32-bit field(s) not required for every datagram allows expansion of IP header .

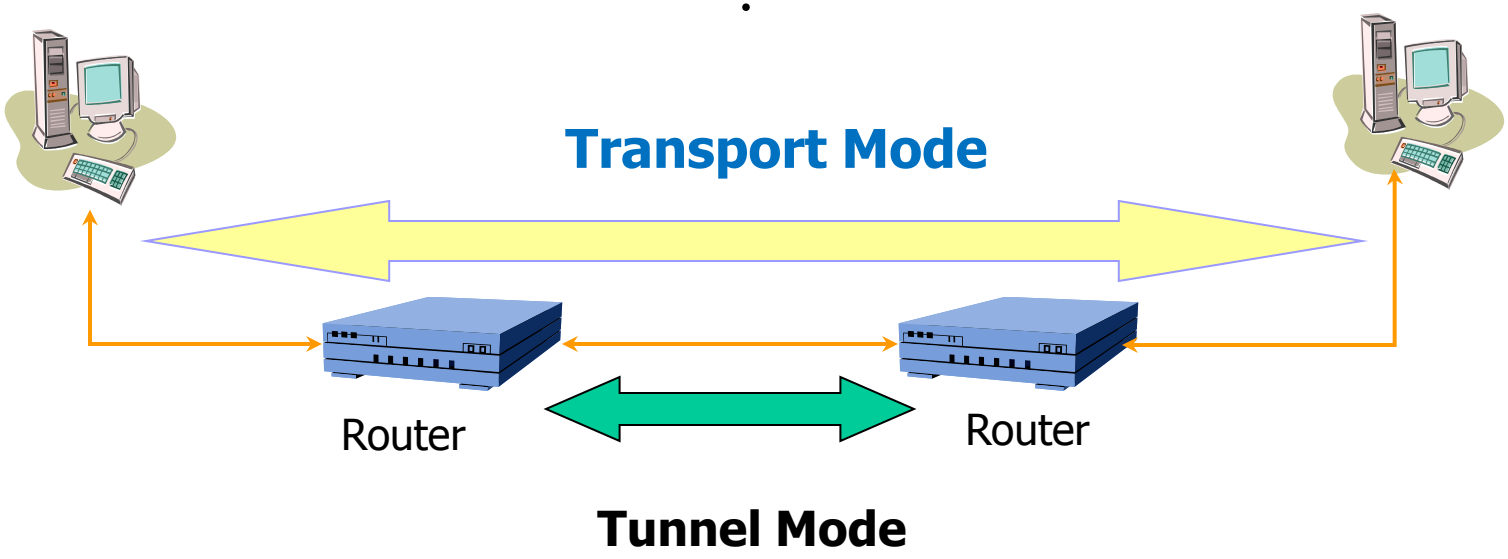
IP Security (IPSec) protocol

- IP Security (IPSec) is a collection of protocols to provide security for a packet at the network level.

IPSec protocol mode

IP Security operates in one of two different modes: the transport mode or tunnel mode.

.



Lecture XVII

IP protocols

IPv6

- *initial motivation:* 32-bit IPv4 address space soon to be completely allocated.
- *additional motivation*:
  - header format -speed processing/forwarding
  - header changes to facilitate QoS, billions of hosts
  - *checksum*: removed entirely to reduce processing time at each hop, Reduce routing table size
  - Aid multicasting

IPv6 datagram format:

  - fixed-length 40 byte header
  - no fragmentation allowed

IPv6 datagram format

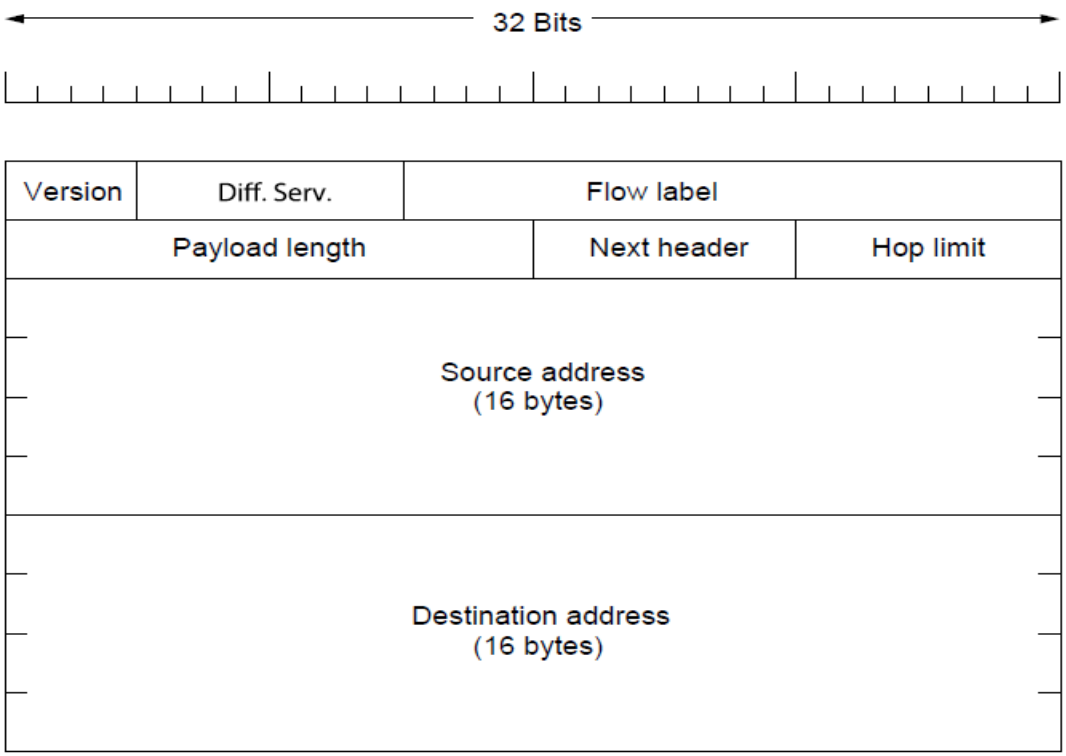
Priority or Diff.serv: identify priority among datagram in flow

flow Label: identify datagram in same "flow."

next header: identify upper layer protocol for data

Source add: 16byte = 128 bit

Destination add: 128 bit

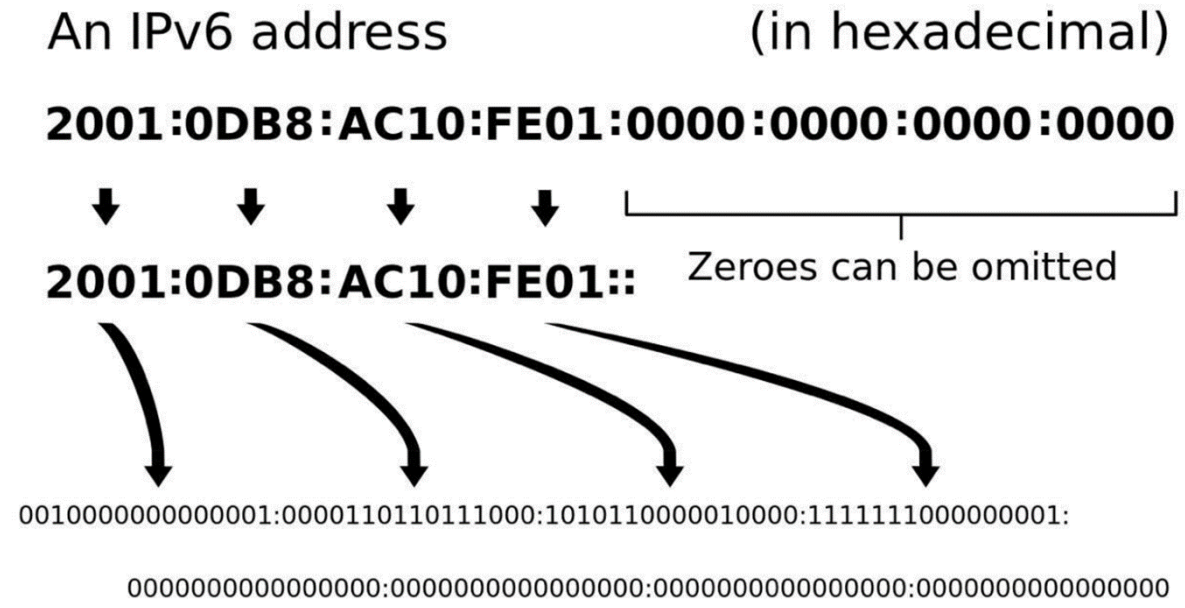


IP Version 6

IPV6 is written in hexadecimal and consists of 8 groups, consists 4 hex digits

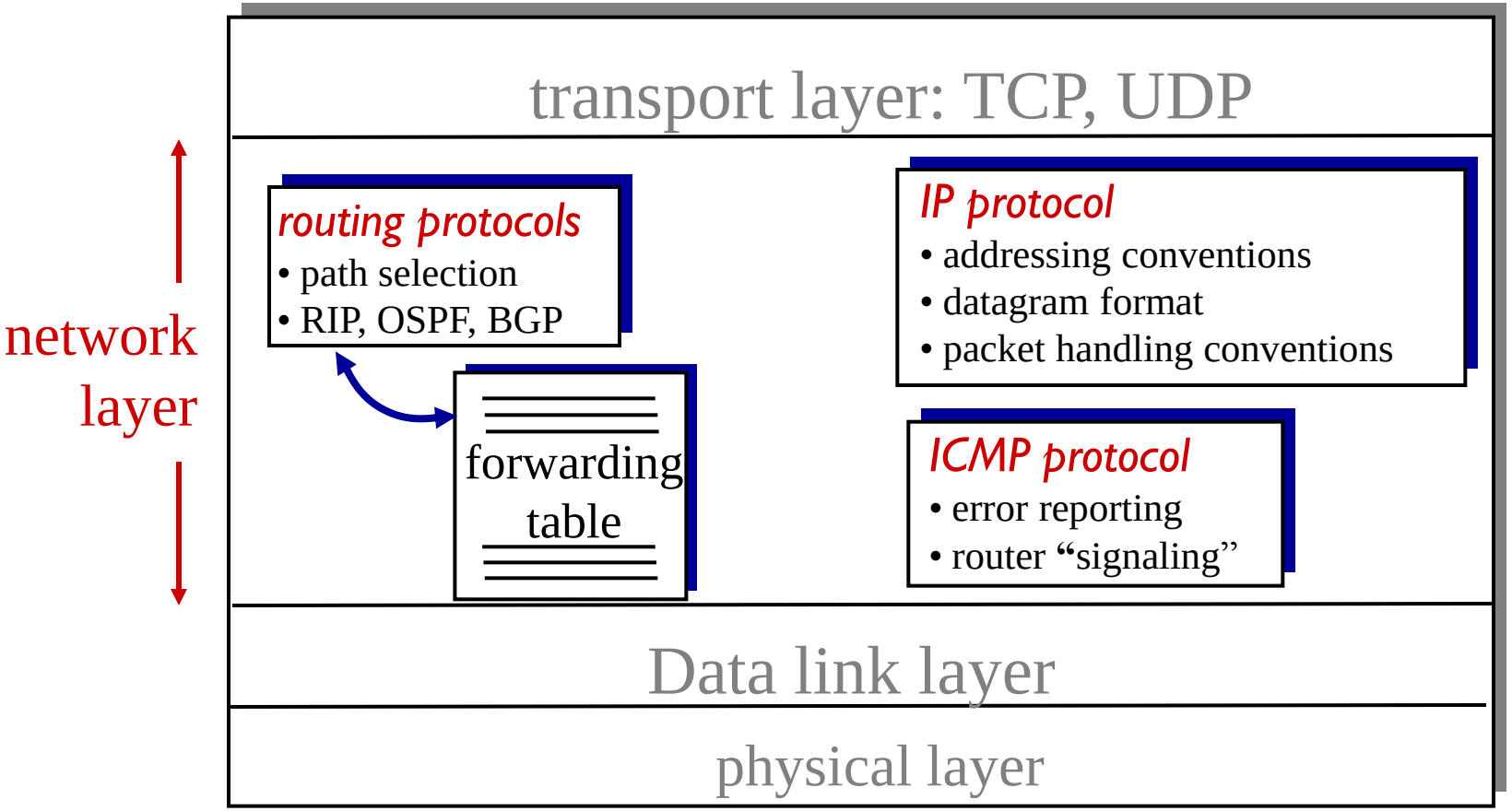
Binary (128 bits)	1111111011110110	…	1111111100000000
Colon Hexadecimal	FEF6:BA98:7654:3210:ADEF:BBFF:2922:FF00		

IPv6


An IPv6 address
(in hexadecimal)
2001:0DB8:AC10:FE01:0000:0000:0000:0000
2001:0DB8:AC10:FE01::
Zeroes can be omitted
0010000000000001:0000110110111000:1010110000010000:1111111000000001:
0000000000000000:0000000000000000:0000000000000000:0000000000000000

The Internet network layer

host, router network layer functions:



Switches vs. routers

both are store-and-forward:

- *routers:* network-layer devices (examine network-layer headers)
- *switches*: link-layer devices (examine link-layer headers)

both have forwarding tables:

- *routers:* compute tables using routing algorithms, IP addresses
- *switches:* learn forwarding table using flooding, learning, MAC addresses

Lecture XVIII

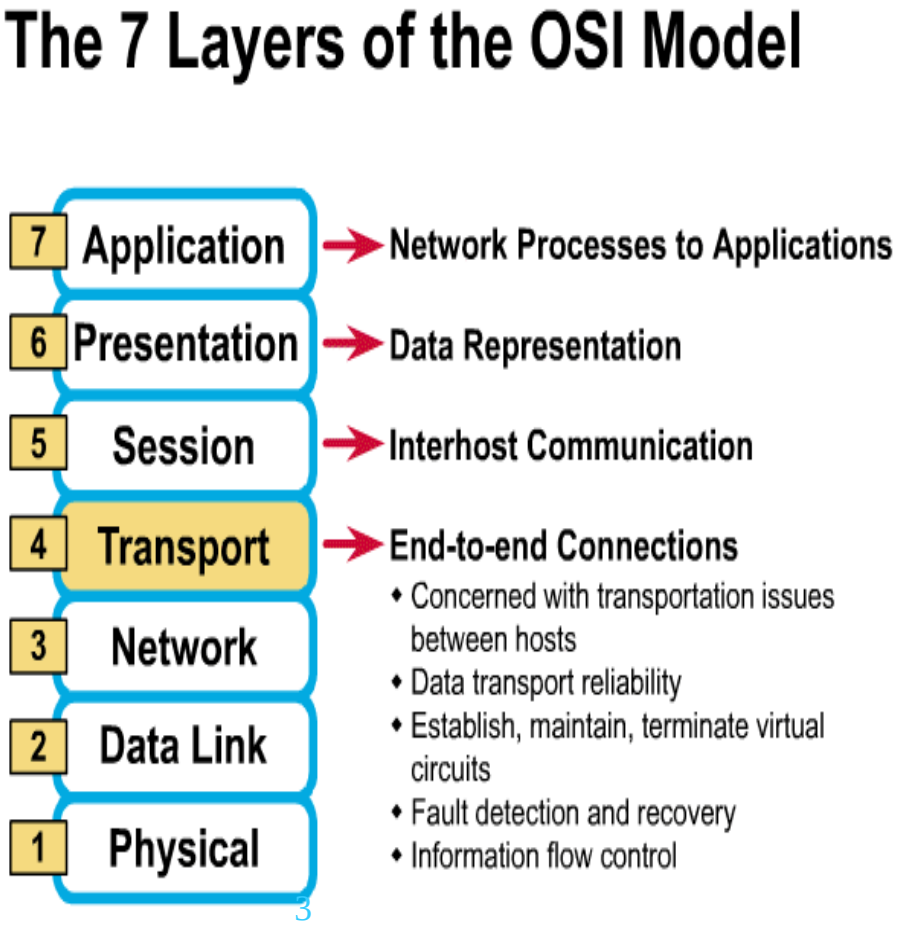
Transport Protocols

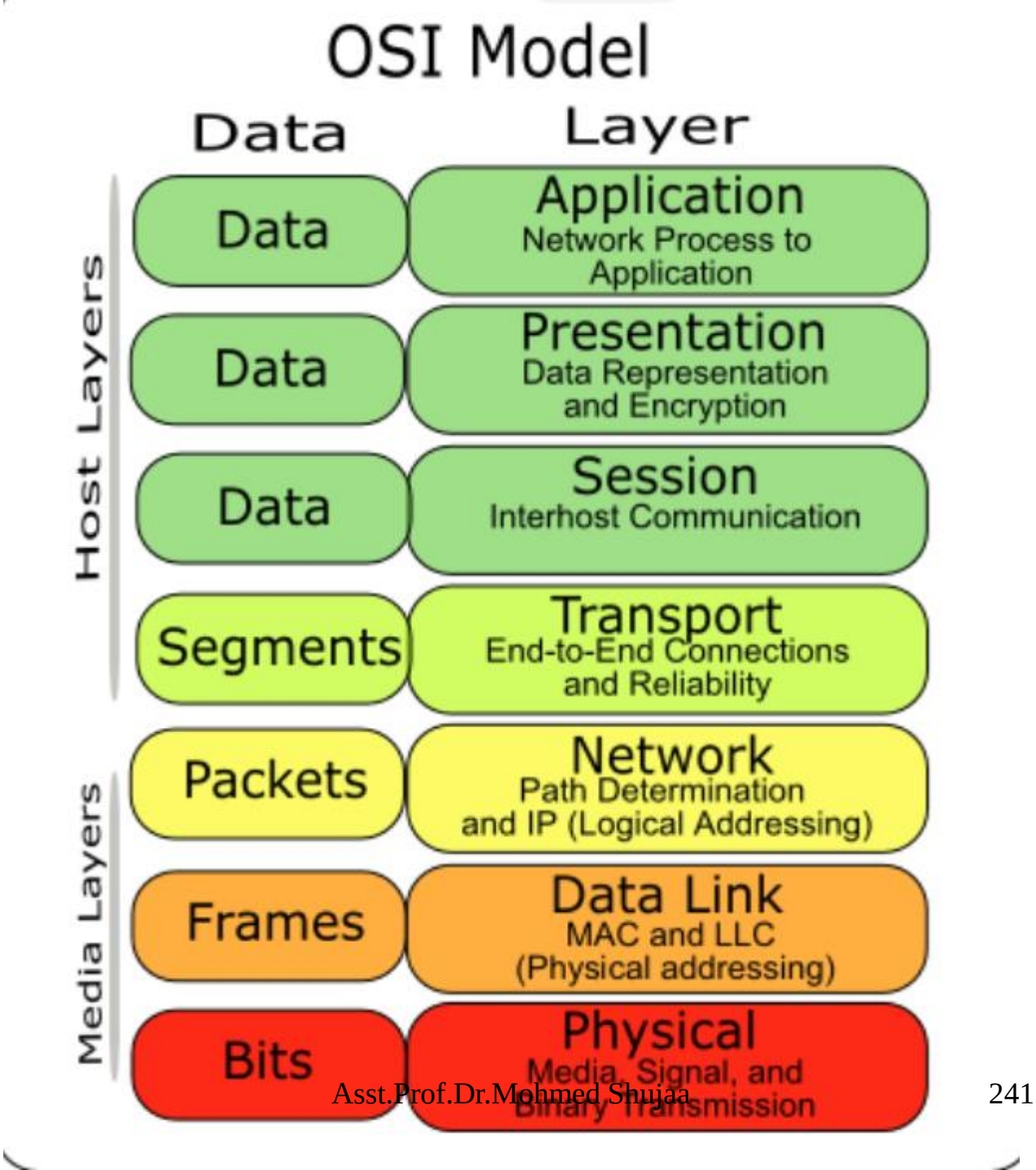
4. Transport Layer

- Provides reliable data delivery
- It's the TCP in TCP/IP

- Receives info from upper layers and segments it into packets

- Can provide error detection and correction

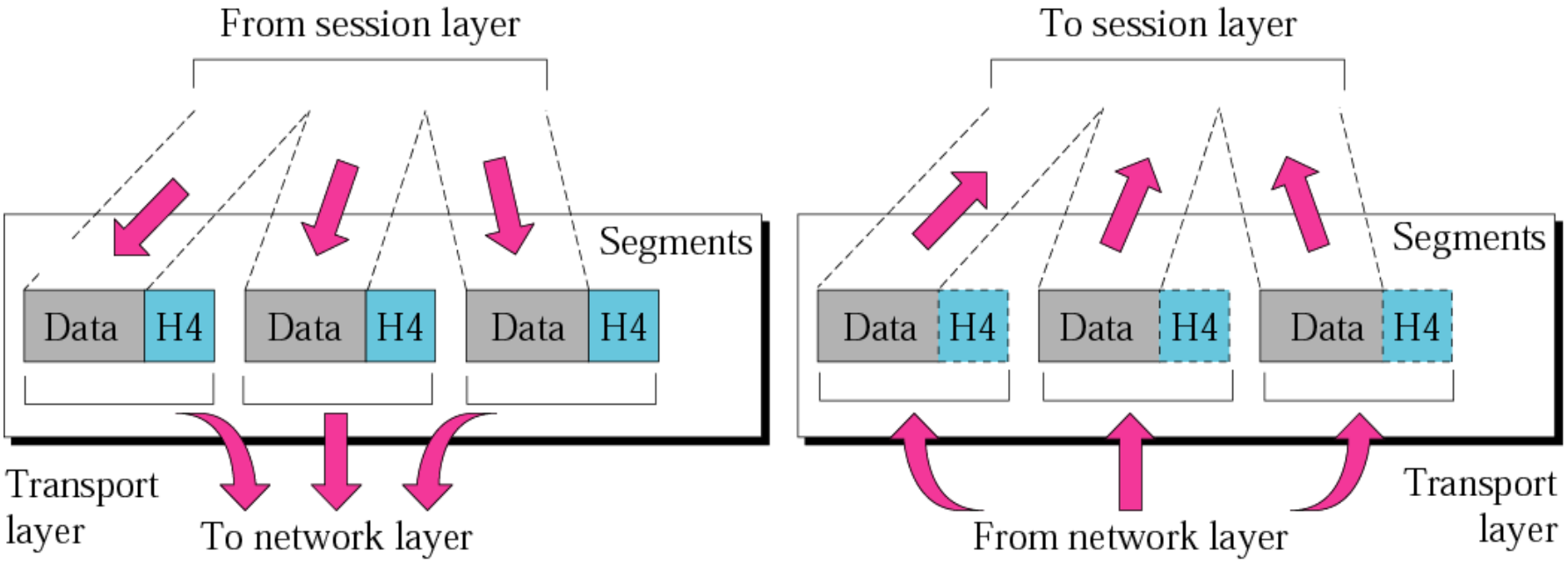



OSI Model
Data
Layer
Host Layers
Media Layers
Data
Application
Network Process to Application
Data
Presentation
Data Representation and Encryption
Data
Session
Interhost Communication
Segments
Transport
End-to-End Connections and Reliability
Packets
Network
Path Determination and IP (Logical Addressing)
Frames
Data Link
MAC and LLC (Physical addressing)
Bits
Physical
Media, Signal, and Binary Transmission

OSI model

Layer	Name	Example protocols
7	Application Layer	HTTP, FTP, DNS, SNMP, Telnet
6	Presentation Layer	SSL, TLS
5	Session Layer	NetBIOS, PPTP
4	Transport Layer	TCP, UDP
3	Network Layer	IP, ARP, ICMP, IPSec
2	Data Link Layer	PPP, ATM, Ethernet
1	Physical Layer	Ethernet, USB, Bluetooth, IEEE802.11

Transport layer



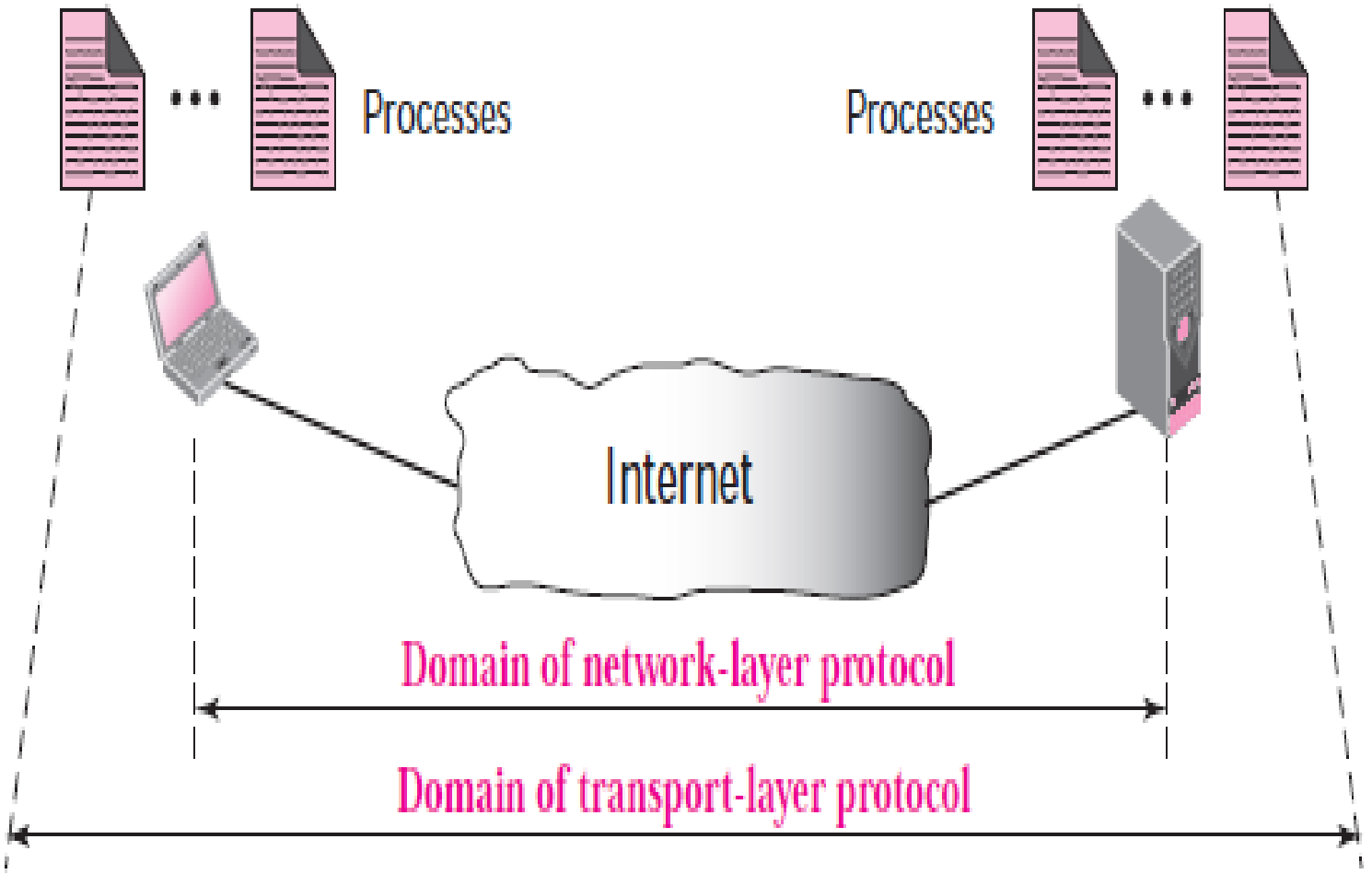
The transport layer is responsible for the delivery of a message from one process to another.

Transport vs. network layer

- *network layer:* logical communication between hosts

- *transport layer:* logical communication between processes
  - Sender: breaks application messages into segments, and passes to network layer
  - Receiver: reassembles segments into messages, passes to application layer

Transport layer Protocols



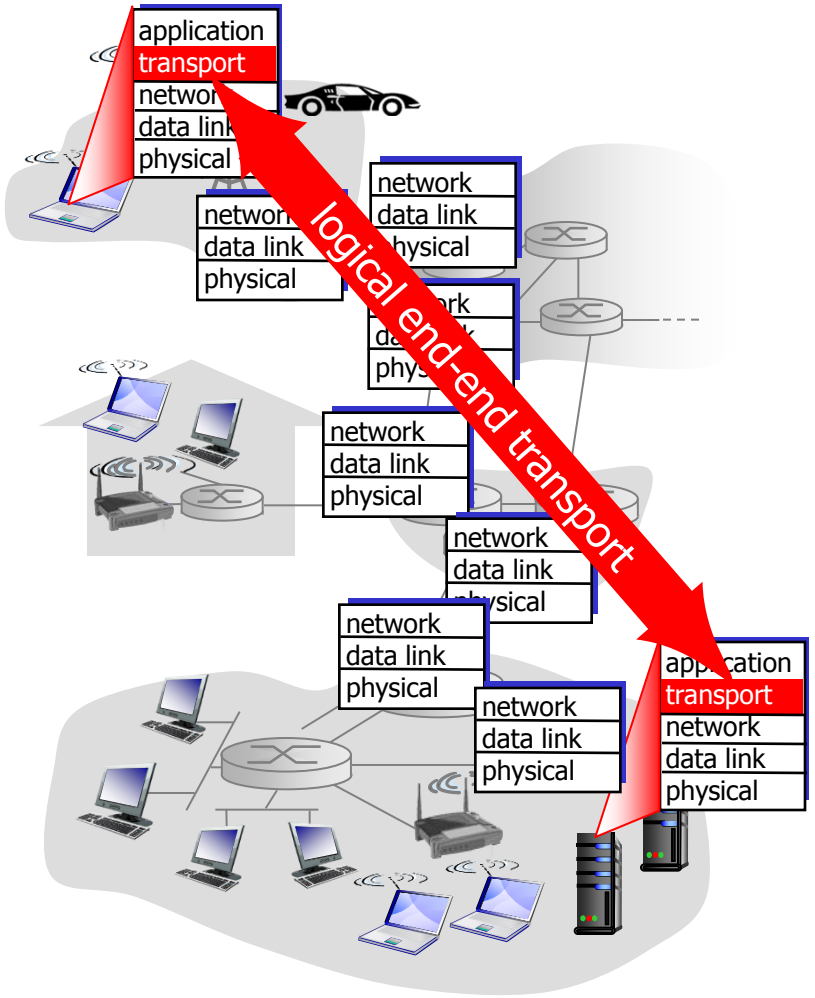
Lecture XIX

Transport protocols

UDP, TCP

transport-layer protocols

- reliable, in-order delivery (TCP)
  - congestion control
  - flow control
  - connection setup
- unreliable, unordered delivery: (UDP)
- services not available:
  - delay guarantees
  - bandwidth guarantees

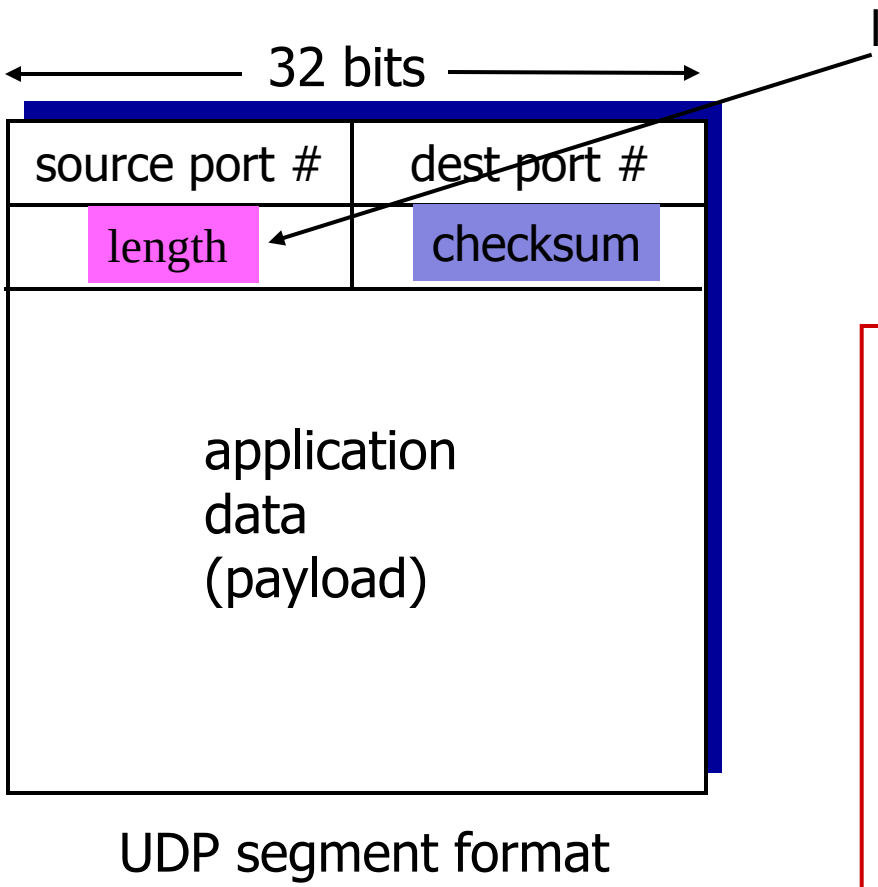


UDP: User Datagram Protocol

- "Internet transport protocol
- *segments* may be:
  - lost
  - delivered out-of-order to app
- *connectionless:*
  - no handshaking between UDP sender, receiver
  - each UDP segment handled independently of others

- UDP use:
  - multimedia apps
  - DNS
  - SNMP

UDP: segment header



UDP segment format

why is there a UDP?

- **no connection establishment (which can add delay)**
- **simple: no connection state at sender, receiver**
- **small header size**
- **no congestion control: UDP can blast away as fast as desired**

UDP checksum

Goal: detect "errors" (e.g., flipped bits) in transmitted segment

sender:

- treat segment contents, including header fields, as sequence of 16-bit integers
- checksum: addition (one's complement sum) of segment contents
- sender puts checksum value into UDP checksum field

receiver:

- compute checksum of received segment
- check if computed checksum equals checksum field value:
  - NO - error detected
  - YES - no error detected.

  ….

TCP

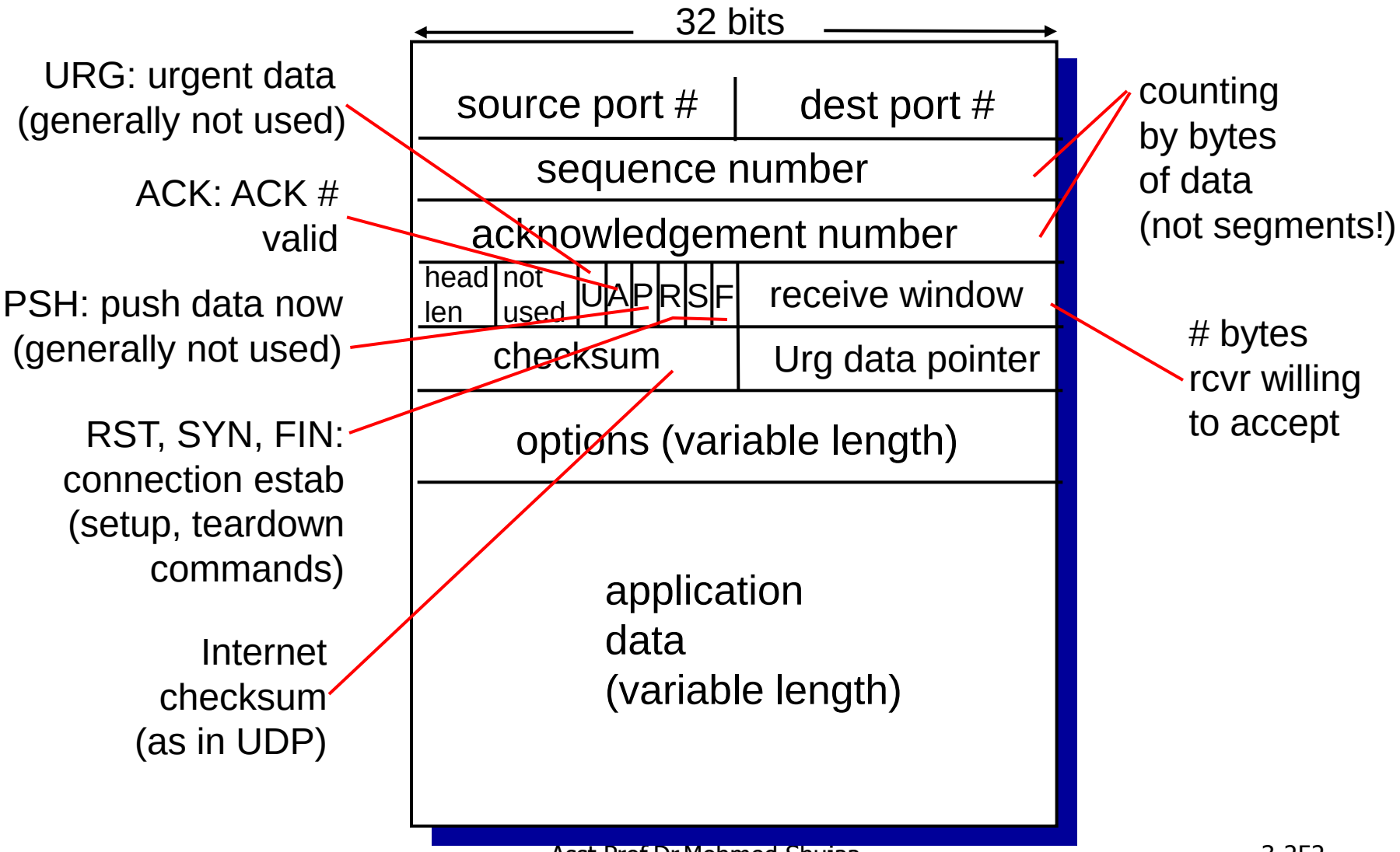
- point-to-point:
  - one sender, one receiver
- Reliable.
  - no “message boundaries”
- pipelined:
  - TCP congestion and flow control set window size

❖ full duplex data:
- bi-directional data flow in same connection
- MSS: maximum segment size

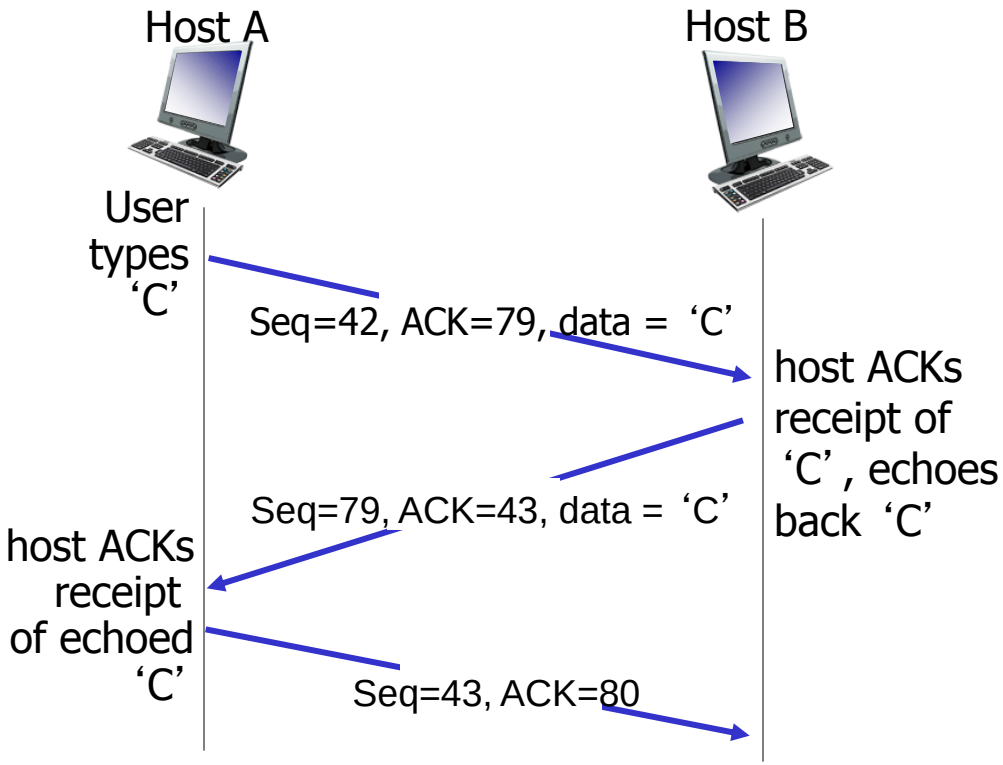
❖ connection-oriented:
- handshaking (exchange of control msgs) in its sender, receiver state before data exchange

❖ flow controlled:
- sender will not overwhelm receiver

TCP segment structure



TCP seq. numbers, ACKs



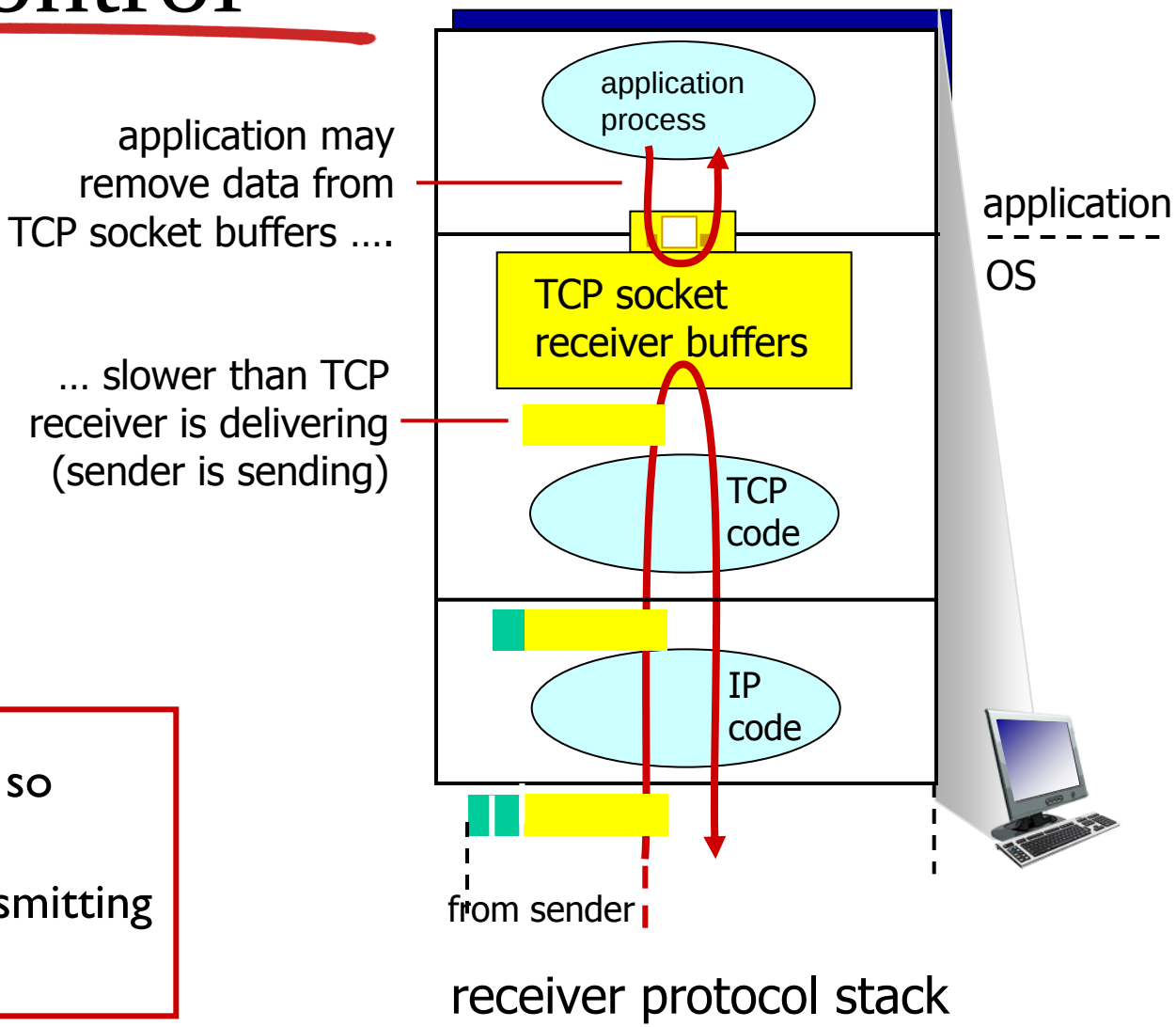
simple telnet scenario

TCP round trip time RTT, timeout

Q: how to set TCP timeout value?

- *longer than RTT*
  - but RTT varies
- *too short:* premature timeout, unnecessary re-transmissions.
- *too long:* slow reaction to segment loss

TCP flow control

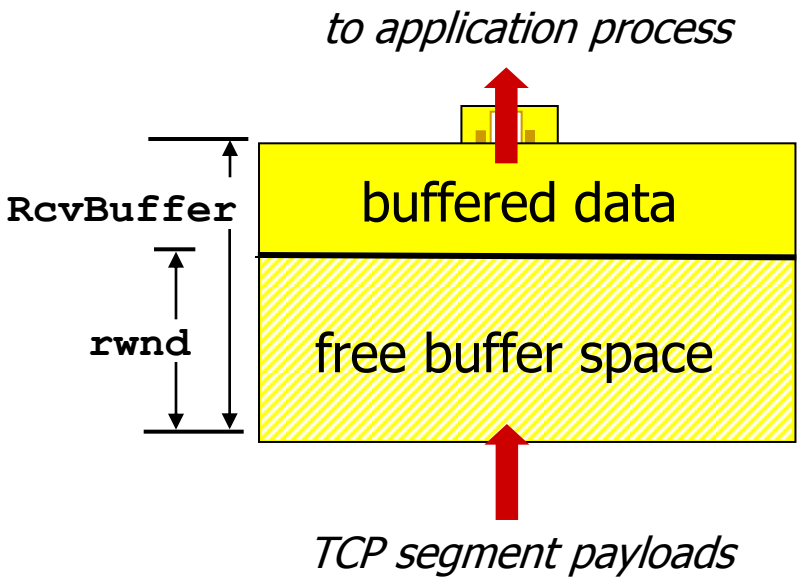


flow control

receiver controls sender, so sender won't overflow receiver's buffer by transmitting too much, too fast

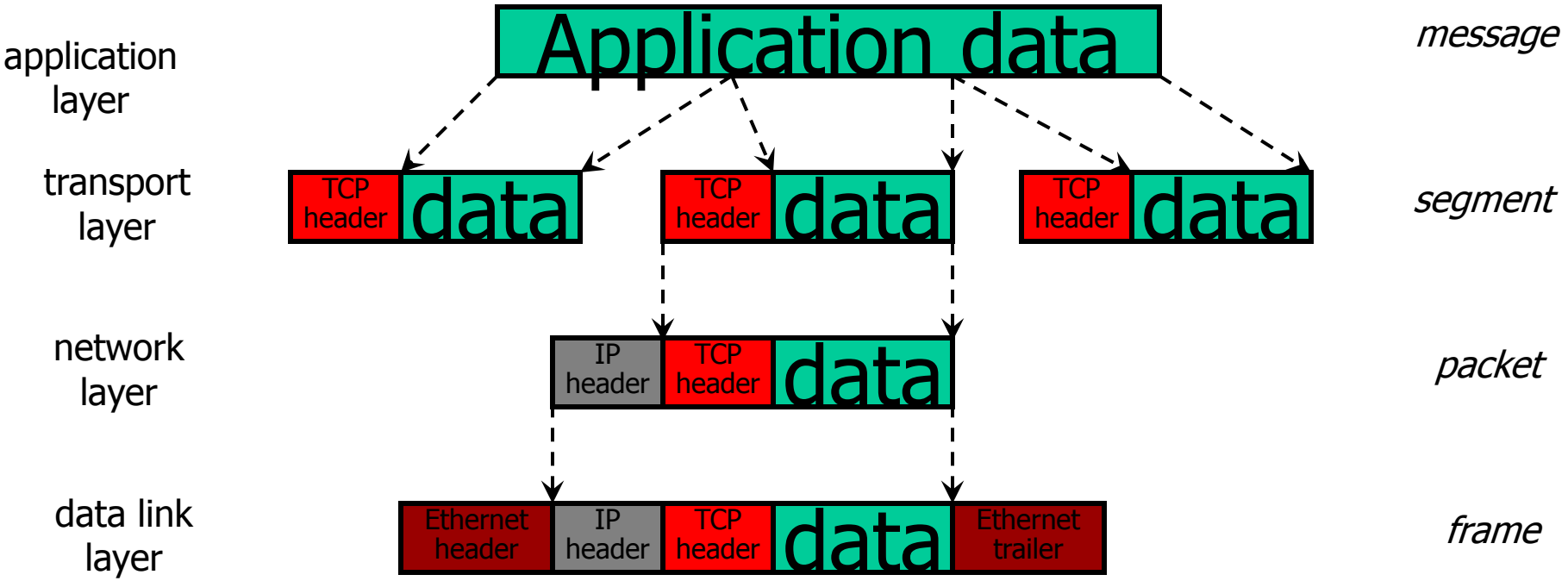
TCP flow control

- receiver "advertises" free buffer space by including **`rwnd`** value in TCP header of receiver-to-sender segments
- sender limits amount of un-ack. data to receiver's **`rwnd`** value.
- guarantees receive buffer will not overflow



receiver-side buffering

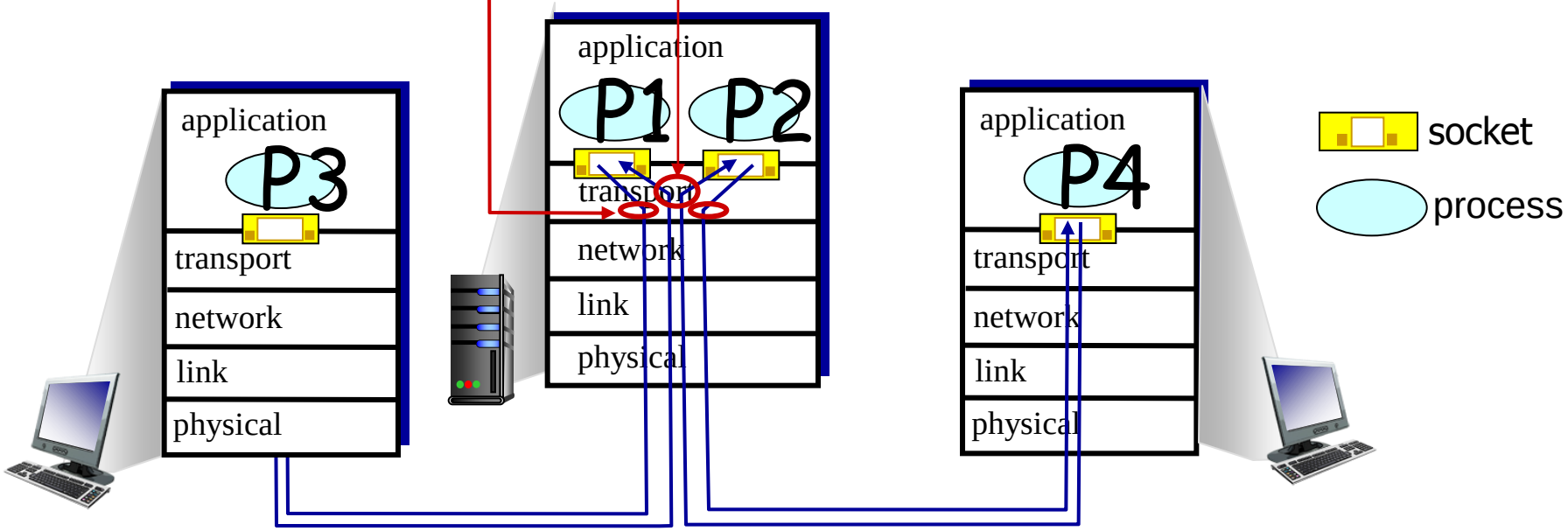
Data Formats



Multiplexing/demultiplexing

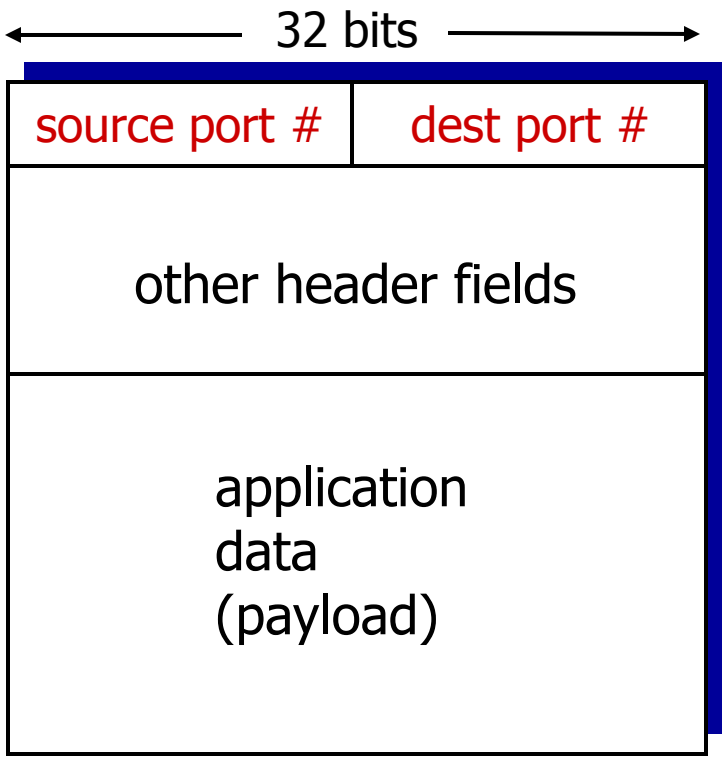
multiplexing at sender:
handle data from multiple sockets, add transport header (later used for demultiplexing)

demultiplexing at receiver:
use header info to deliver received segments to correct socket



How demultiplexing works

- host receives IP datagrams
  - each datagram has source IP address, destination IP address
  - each datagram carries one transport-layer segment
  - each segment has source, destination, port number
- host uses *IP addresses & port numbers* to direct segment to appropriate socket



TCP/UDP segment format

Lecture XX

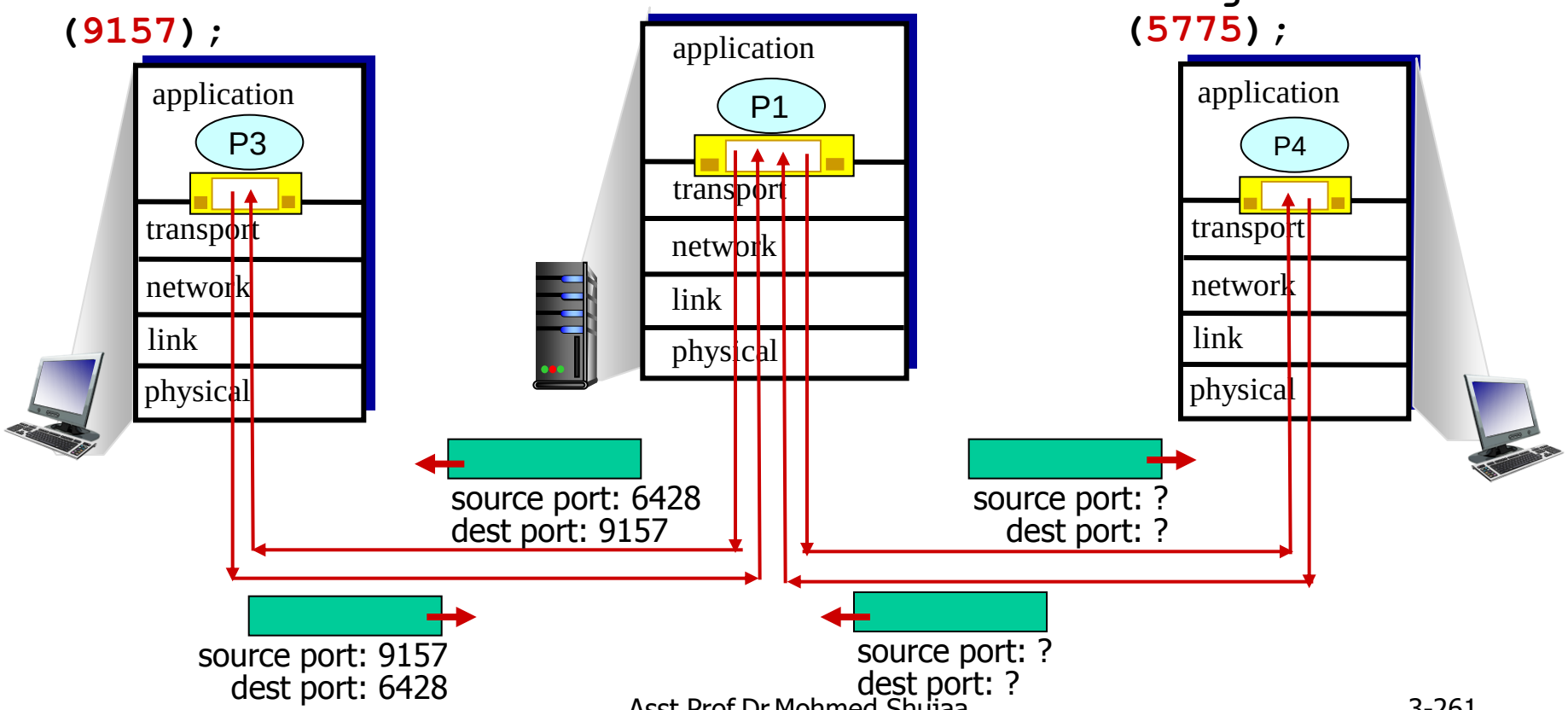
Services

Connectionless demux: example

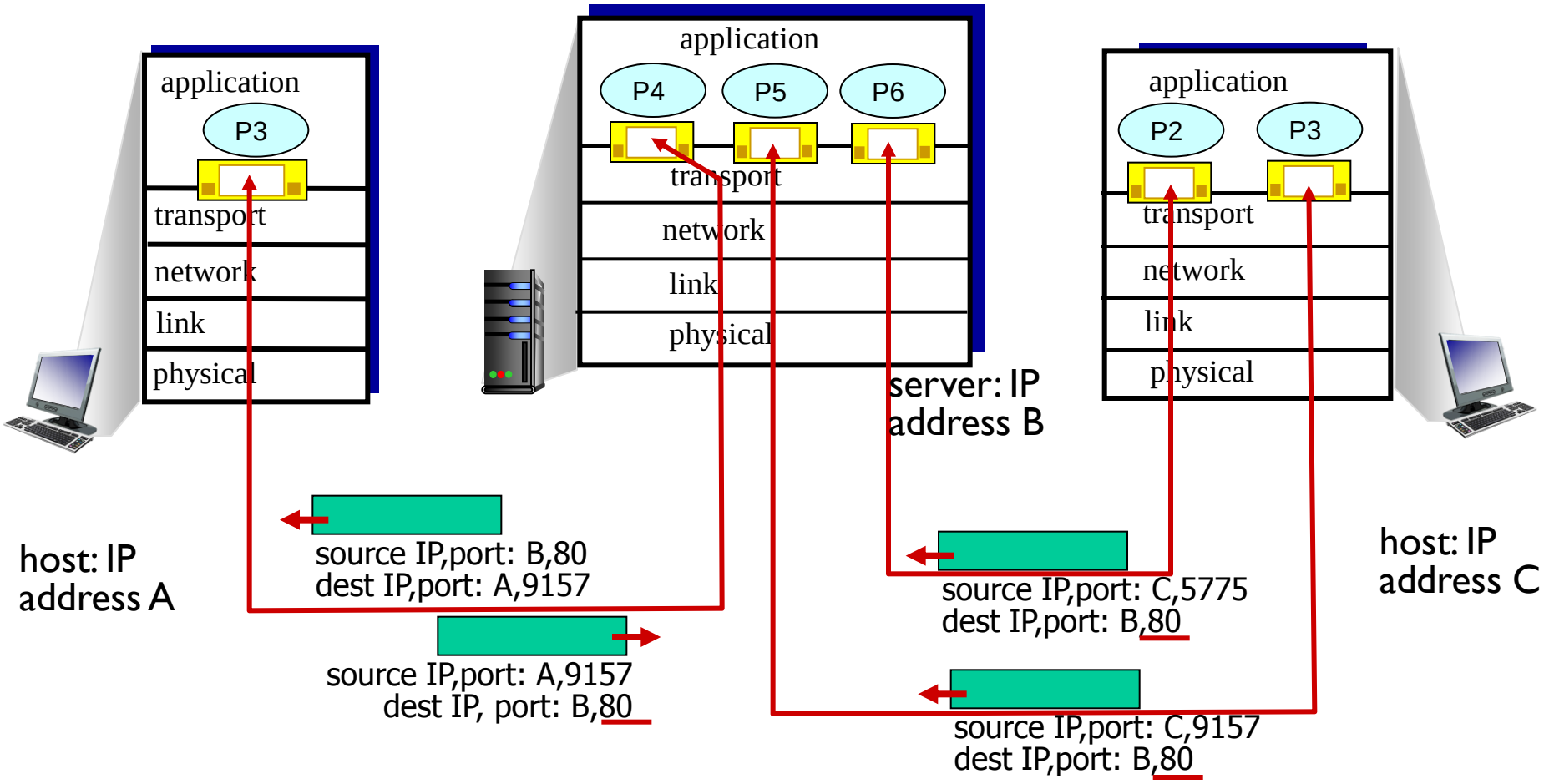
```
DatagramSocket
 serverSocket = new
 DatagramSocket
 (6428);
```

```
DatagramSocket
 mySocket2 = new
 DatagramSocket
 (9157);
```

```
DatagramSocket
 mySocket1 = new
 DatagramSocket
 (5775);
```



Connection-oriented demux: example



three segments, all destined to IP address: B, dest port: 80 are demultiplexed to *different* sockets

Connectionless demultiplexing-UDP

❖ *recall:* created socket has host-local port #:

```
DatagramSocket mySocket1
= new DatagramSocket(12534);
```

❖ *recall:* when creating datagram to send into UDP socket, must specify

- destination IP address
- destination port #

❖ when host receives UDP segment:

- checks destination port # in segment
- directs UDP segment to socket with that port #



IP datagrams with *same dest. port #,* but different source IP addresses and/or source port numbers will be directed to *same socket* at dest

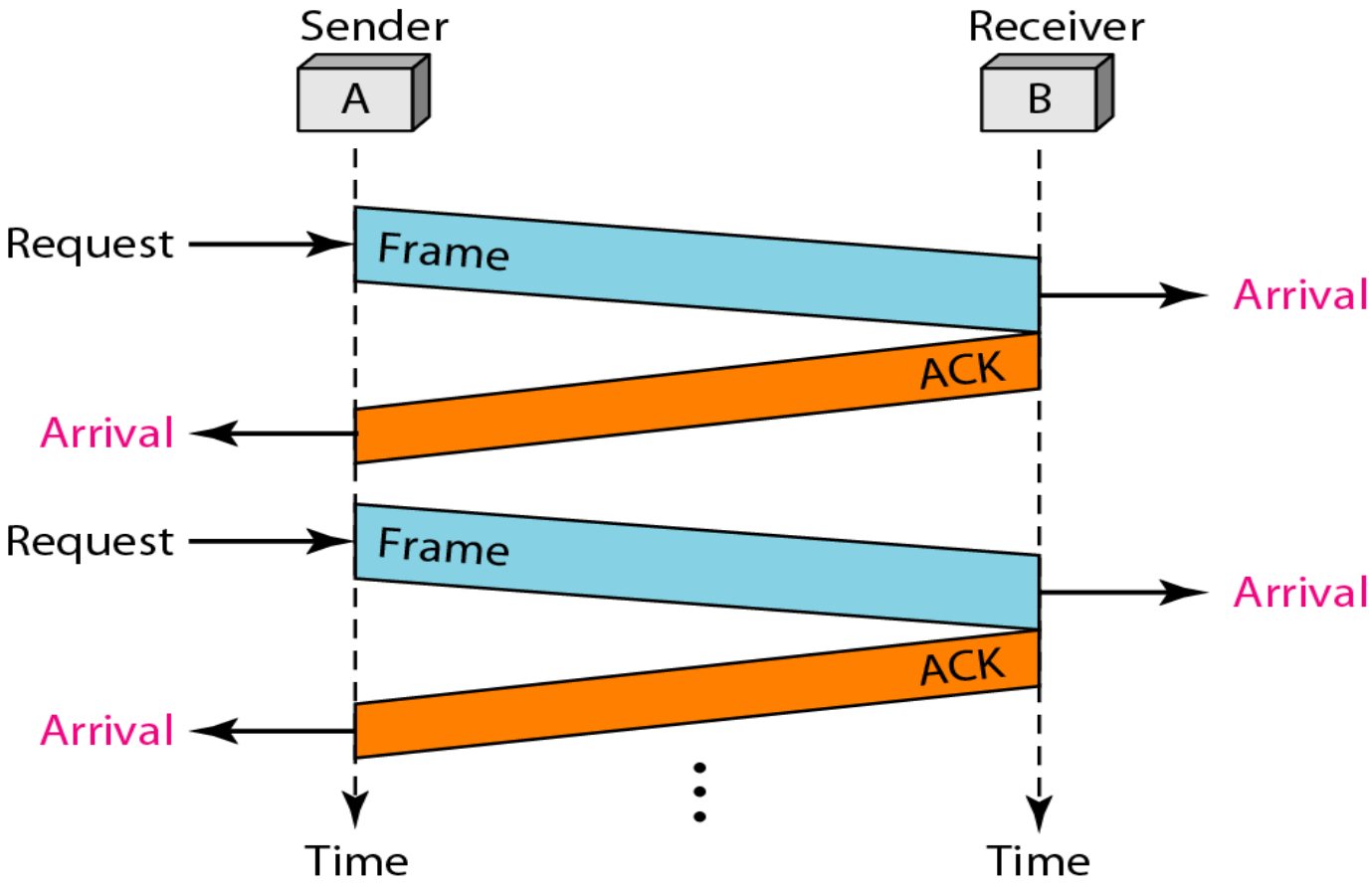
Connection-oriented demux-TCP

- TCP socket identified by 4-tuple:
  - source IP address
  - source port number
  - dest IP address
  - dest port number
- demux: receiver uses all four values to direct segment to appropriate socket
- server host may support many simultaneous TCP sockets:
  - each socket identified by its own 4-tuple
- web servers have different sockets for each connecting client

Stop-and-Wait

- This is the **simplest** form of flow control.
- After receiving the frame, the receiver indicates its willingness to accept another frame by sending back an **ACK** frame acknowledging the frame just received.
- The sender must **wait** until it receives the ACK frame **before** sending the next data frame.

Stop-and-Wait



Disadvantage of stop and wait

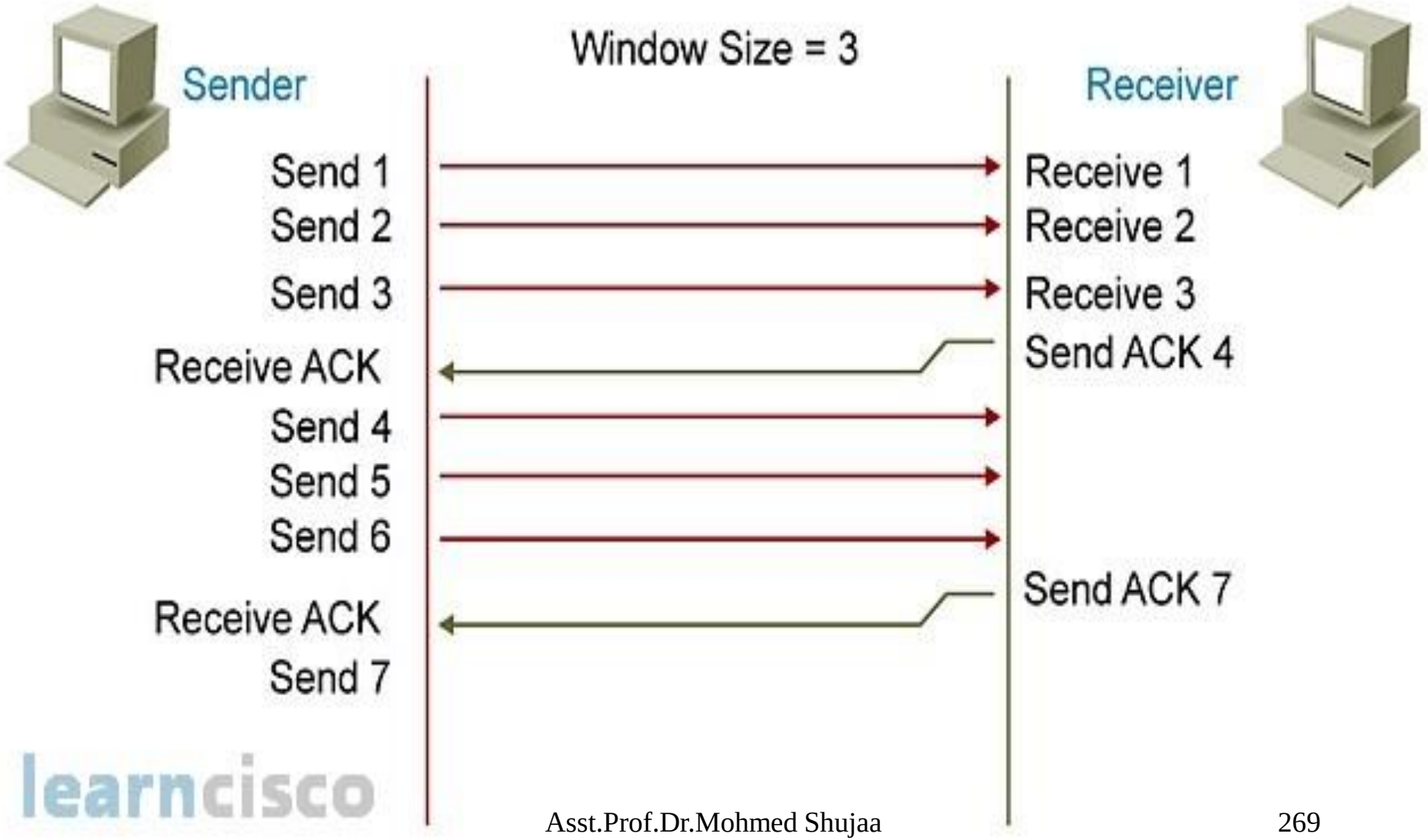
- only **one frame** sent and wait to ACK1
- **not** good for transmission **media.**

multiple frames should be transmit while send is wait to one ACK this called sliding window

Sliding Window protocols

- To keep track of the frames, sender station sends **sequentially** numbered frames .
- **Sliding Window** meaning number of frames that sender can transmit in same time.

Sliding Window Flow control



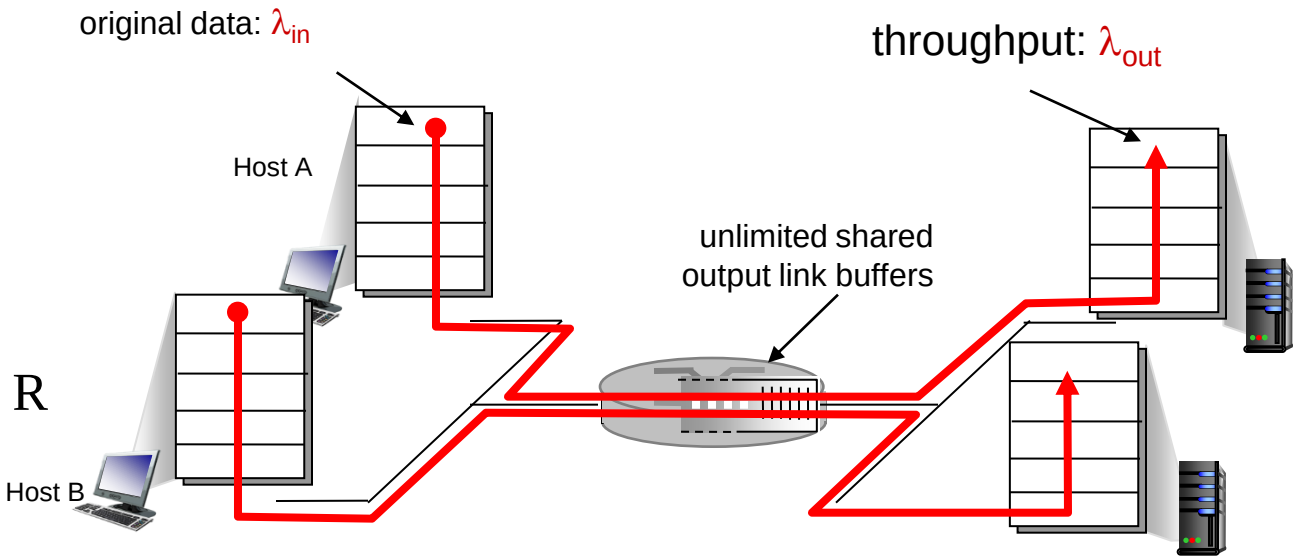
Congestion Control

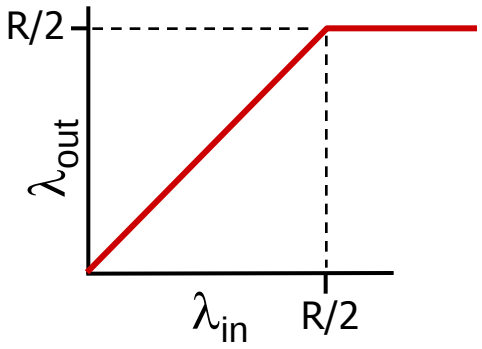
***Reasons of congestions*:**

- Load on the network. “too many sources sending too much data too fast for *network* to handle”
- Low Bandwidth.
- Slow Processor.
- ***Results:***
  - lost packets (buffer overflow at routers)
  - long delays (queuing in router buffers)

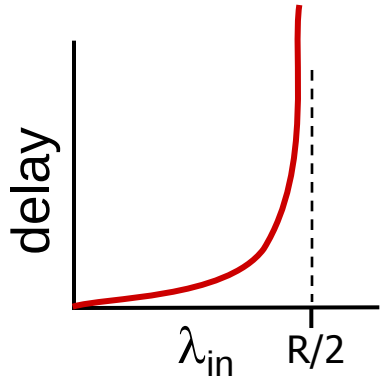
Causes/ congestion:

- two senders, two receivers
- one router, infinite buffers
- output link capacity: R
- no retransmission





- maximum per-connection throughput: R/2

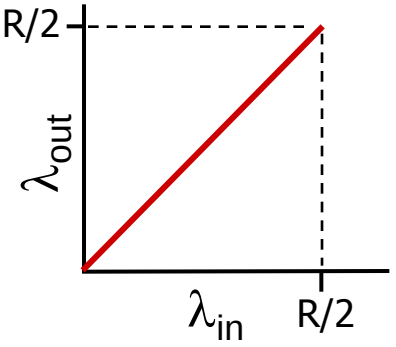


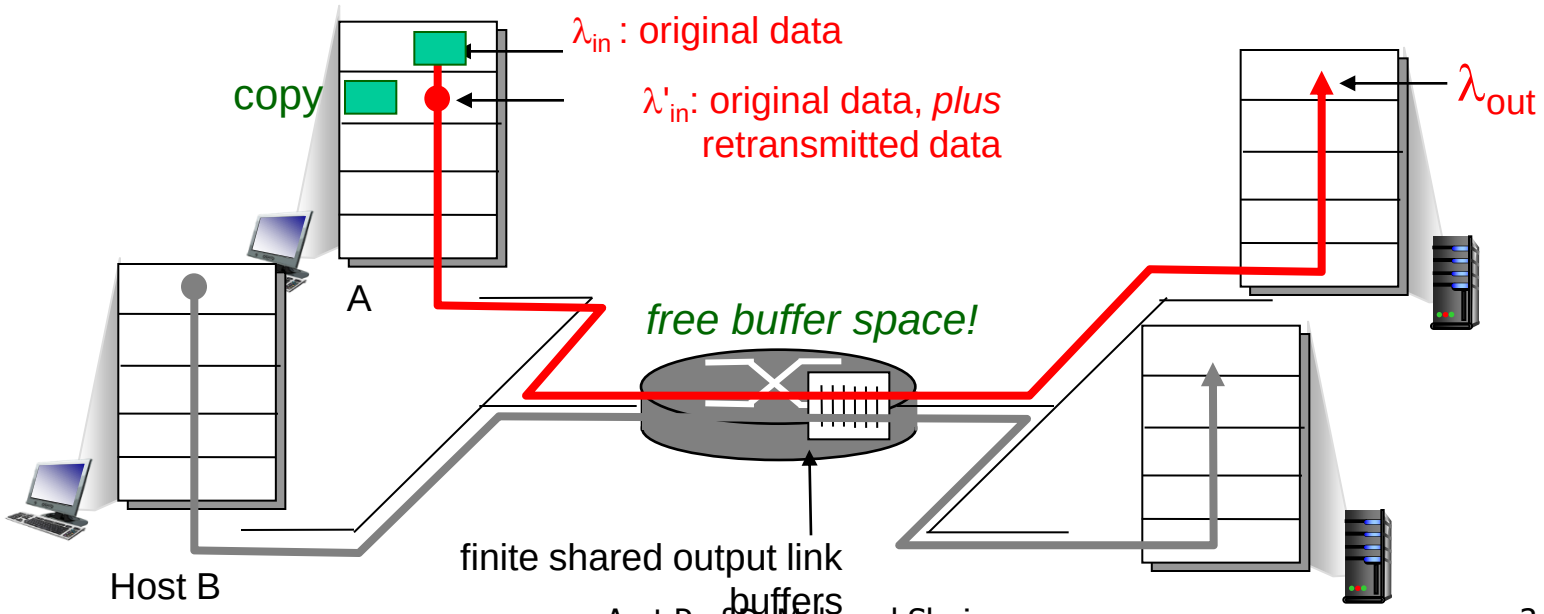
- large delays as arrival rate, λ_{in}, approaches capacity

Causes/costs of congestion: scenario 1

idealization: perfect knowledge

- sender sends only when router buffers available

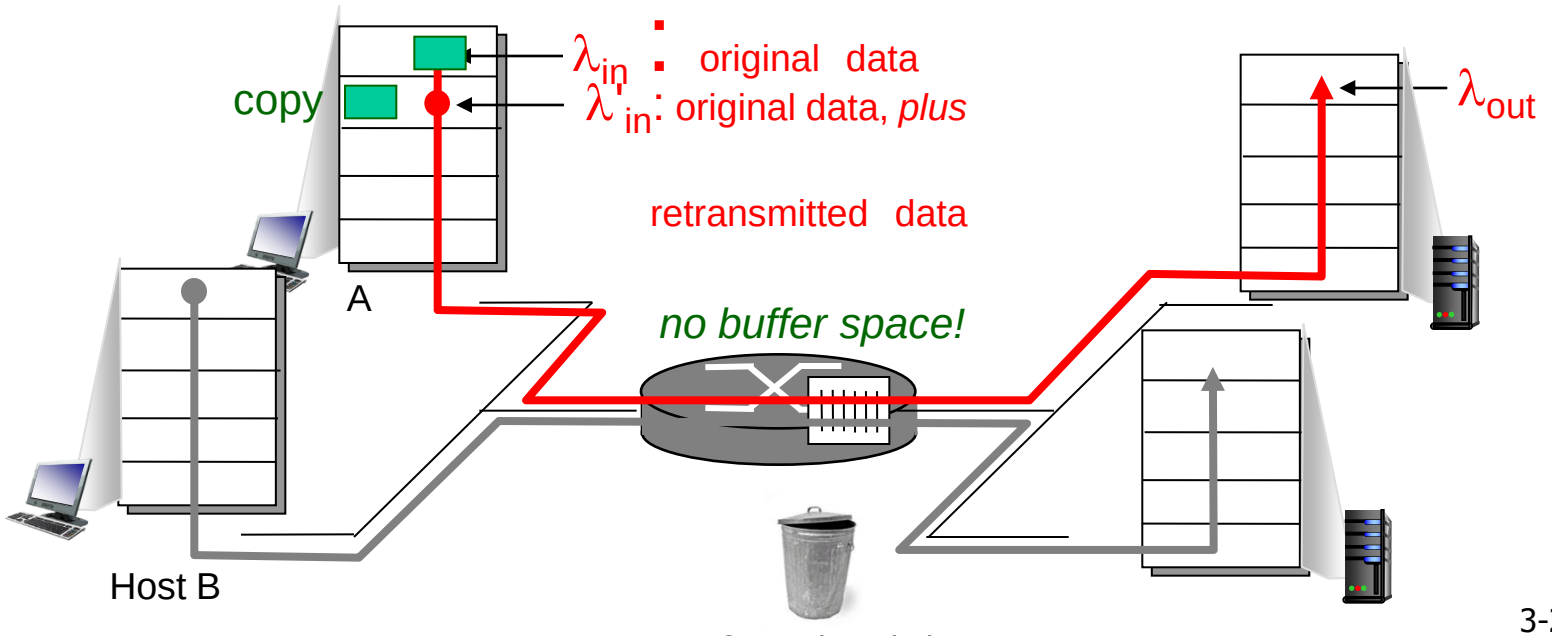




Causes/costs of congestion: scenario 2

Idealization: known loss packets can be lost, dropped at router due to full buffers

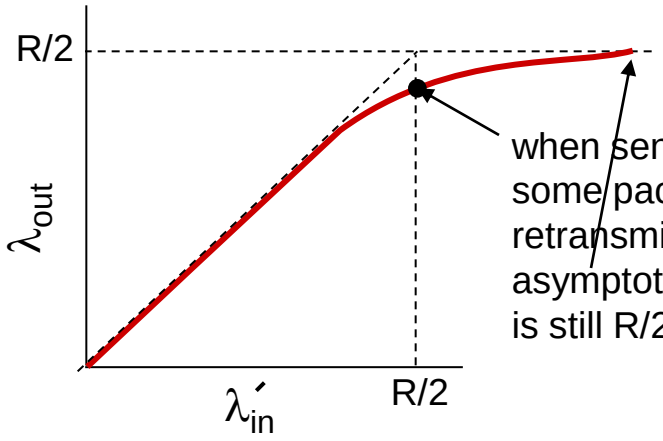
- sender only resends if packet *known* to be lost

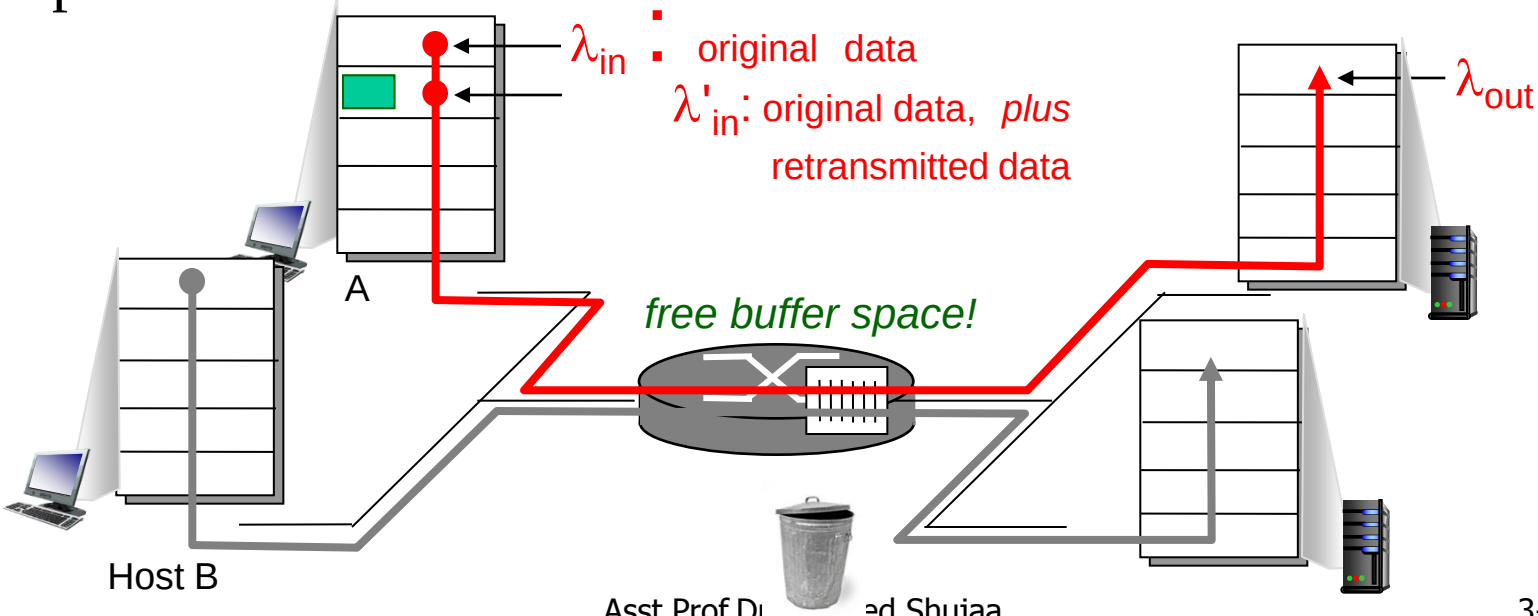


Causes/costs of congestion: scenario 2

Idealization: known loss packets can be lost, dropped at router due to full buffers

- sender only resends if packet *known* to be lost

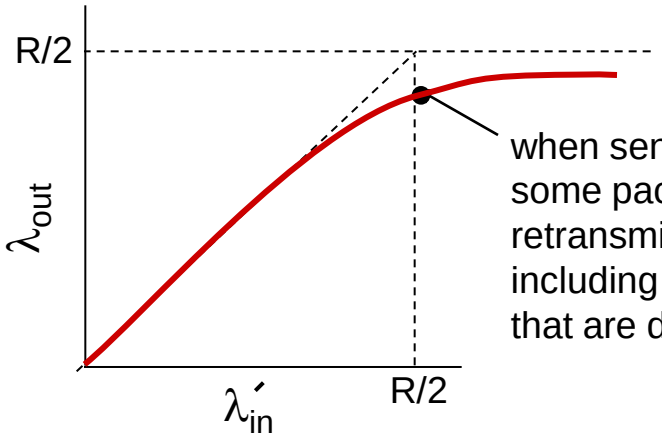


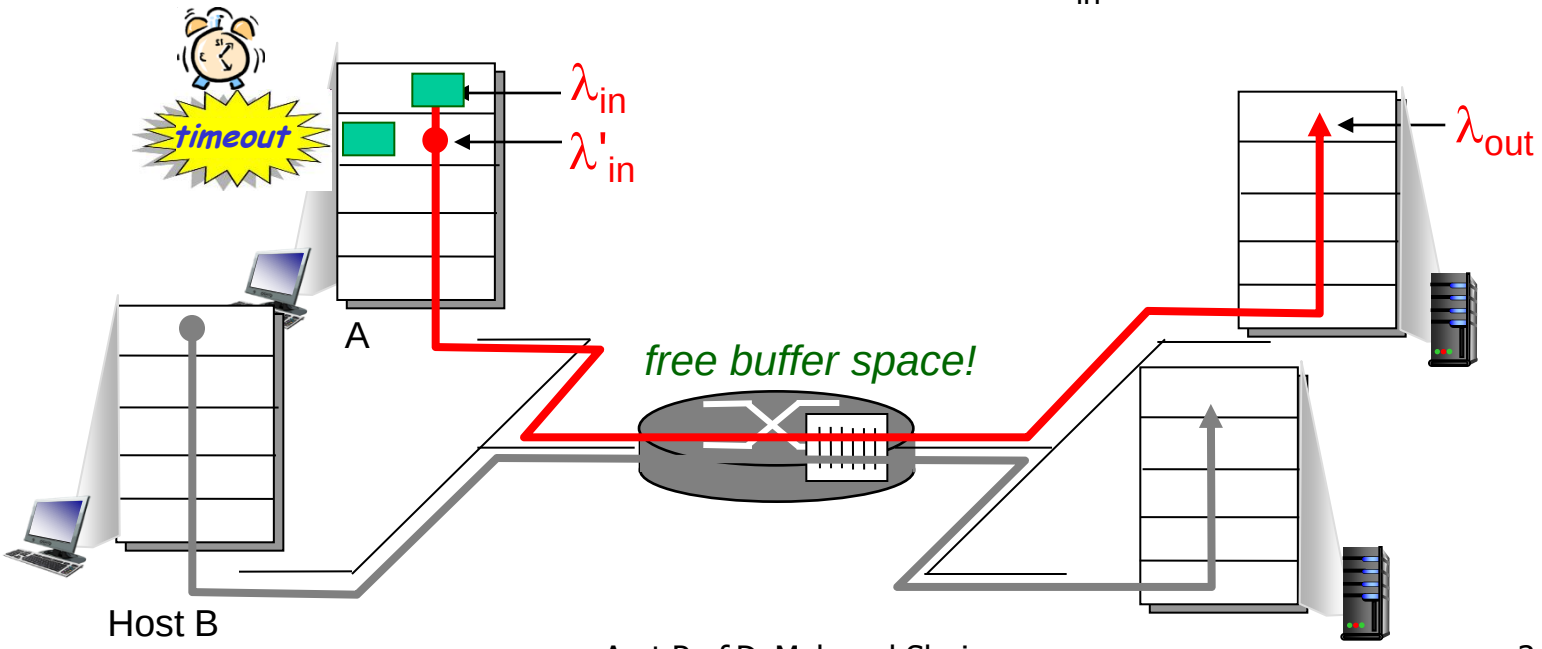


Causes/costs of congestion: scenario 3

Realistic: duplicates

- packets can be lost, dropped at router due to full buffers
- sender times out prematurely, sending *two* copies, both of which are delivered

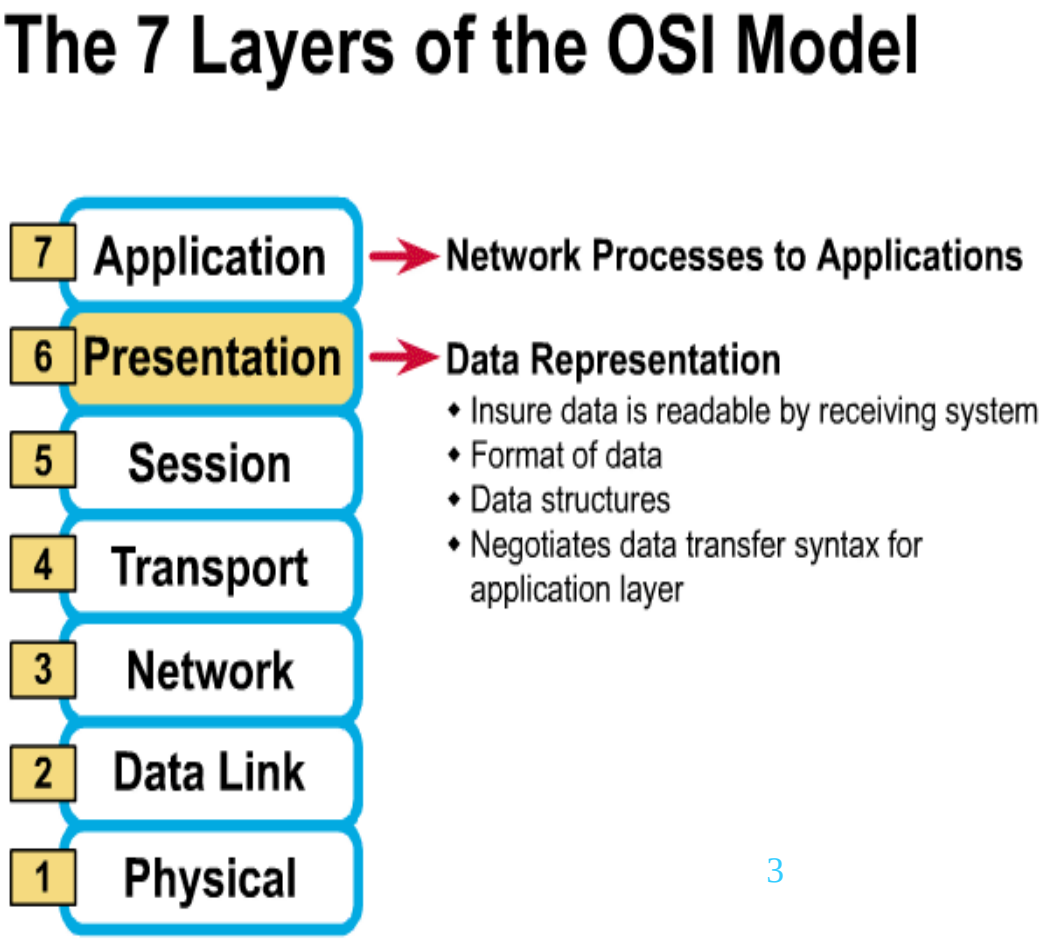




Lecture XXI

Presentation protocols

6.Presentation Layer



3

OSI model

Layer	Name	Example protocols
7	Application Layer	HTTP, FTP, DNS, SNMP, Telnet
6	Presentation Layer	SSL, TLS
5	Session Layer	NetBIOS, PPTP
4	Transport Layer	TCP, UDP
3	Network Layer	IP, ARP, ICMP, IPSec
2	Data Link Layer	PPP, ATM, Ethernet
1	Physical Layer	Ethernet, USB, Bluetooth, IEEE802.11

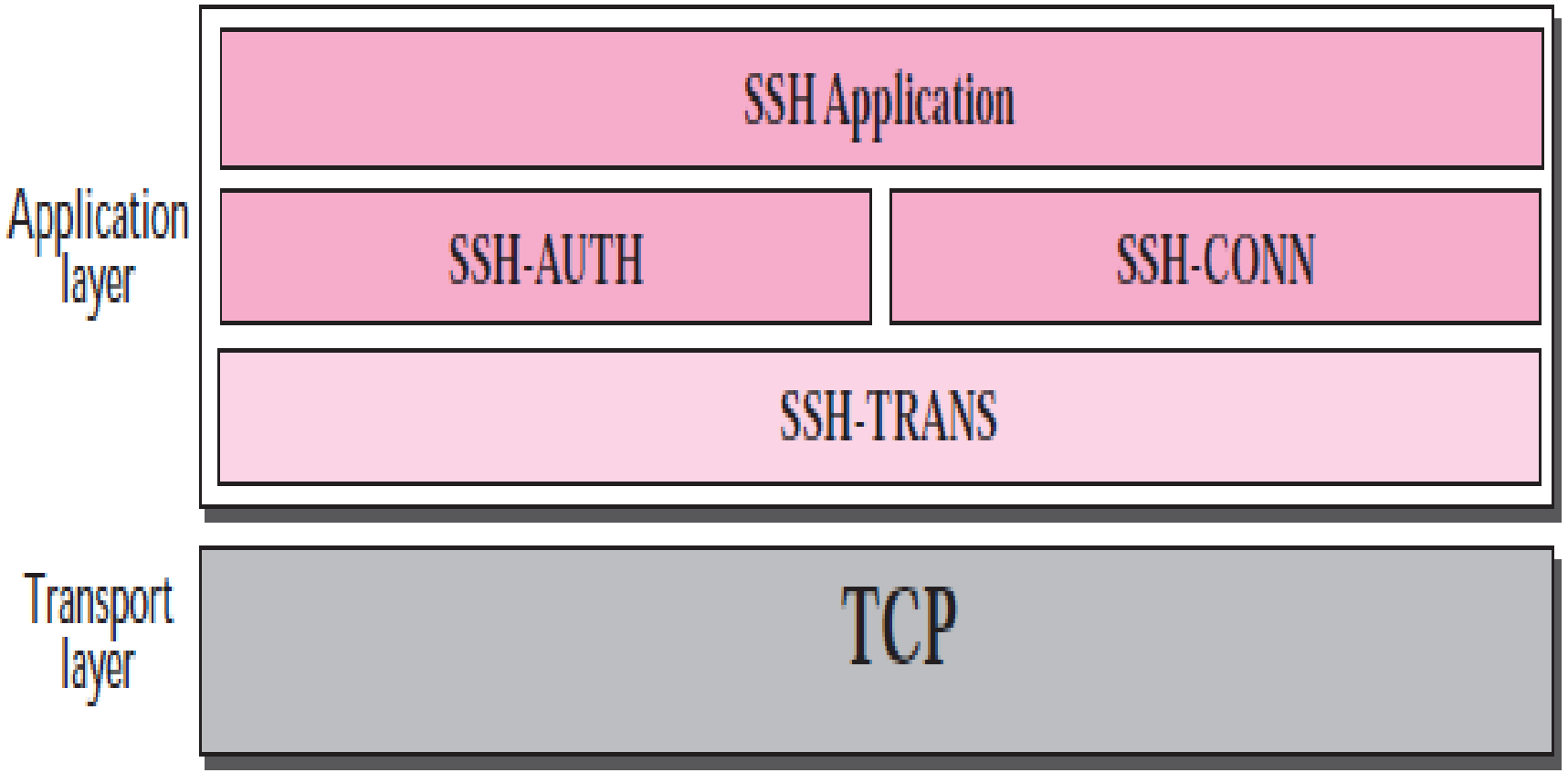
SECURE SHELL PROTOCOL (SSH)

- Another popular **remote login** application program is **Secure Shell (SSH).**
- SSH, like TELNET, uses **TCP** as the underlying transport protocol, but SSH is **more secure** and provides more services than TELNET.
- Covers authentication, encryption.
- Solve the security issues at **remote login** of **Telnet.**
- Solve the security issues during **file transfer** at **FTP**.

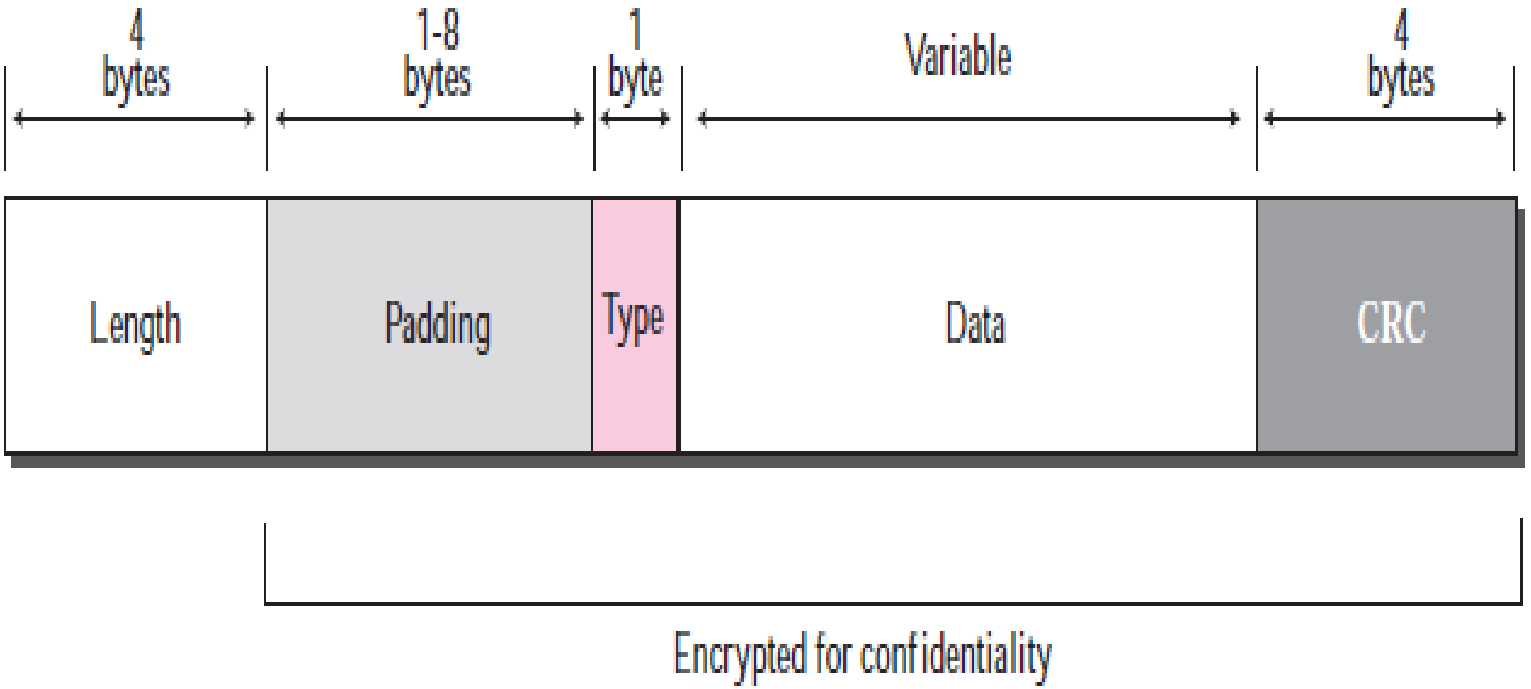
SSL vs SSH

- SSL is **TCP**-based and always used in **WEB applications**, with **HTTP**.
- SSH is **TCP**-based and always used with **Telnet** and **FTP**

SSH components



Format of the SSH Packets



SSH Packet format

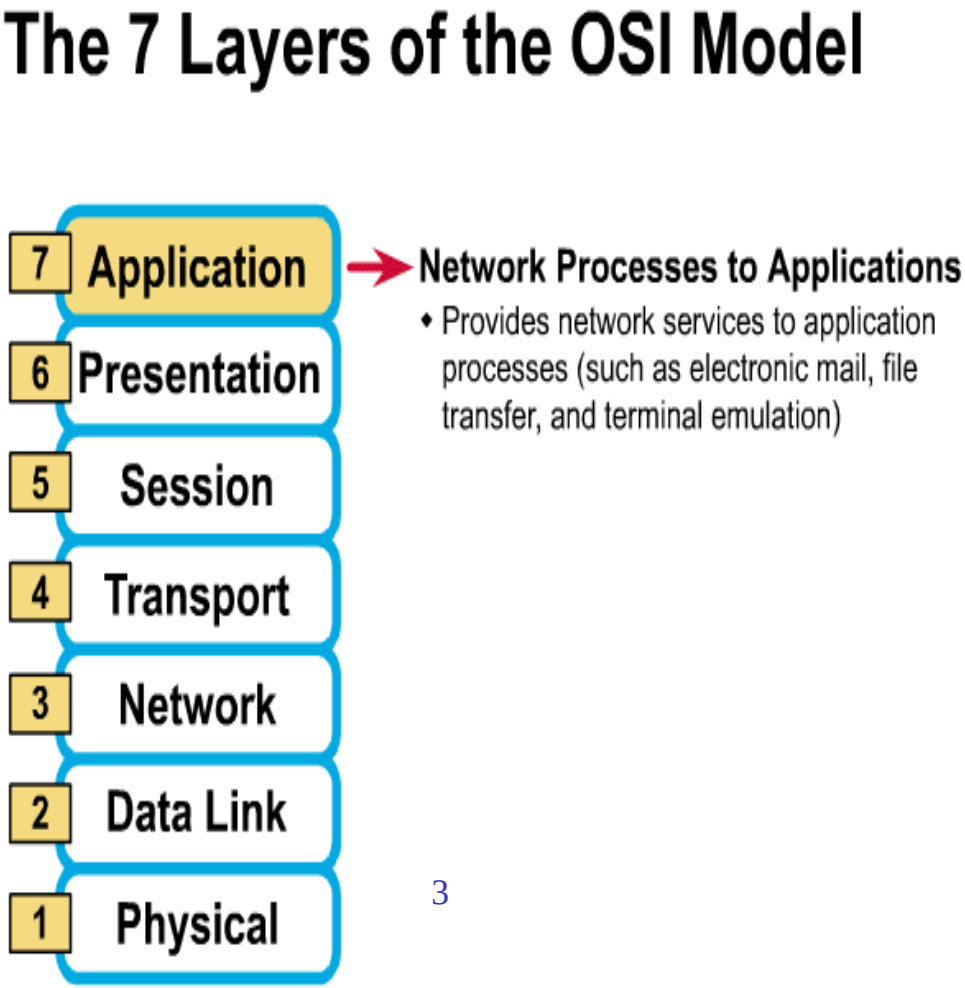
- **Length.** **4-byte field defines the length of the packet including the type, the** data, and the CRC field, **but not** the padding and the length field.
- **Padding. One to eight bytes of padding is added to the packet to make the attack** on the security provision more difficult.
- **Type.** This one-byte field defines the type of the packet used by SSH protocols.
- **CRC.** The cyclic redundancy check filed is used for **error detection**

Lecture XXII

Application protocols

7. application

- Gives end-user applications access to network resources
- Where is it on my computer?
  - Workstation or Server Service in MS Windows



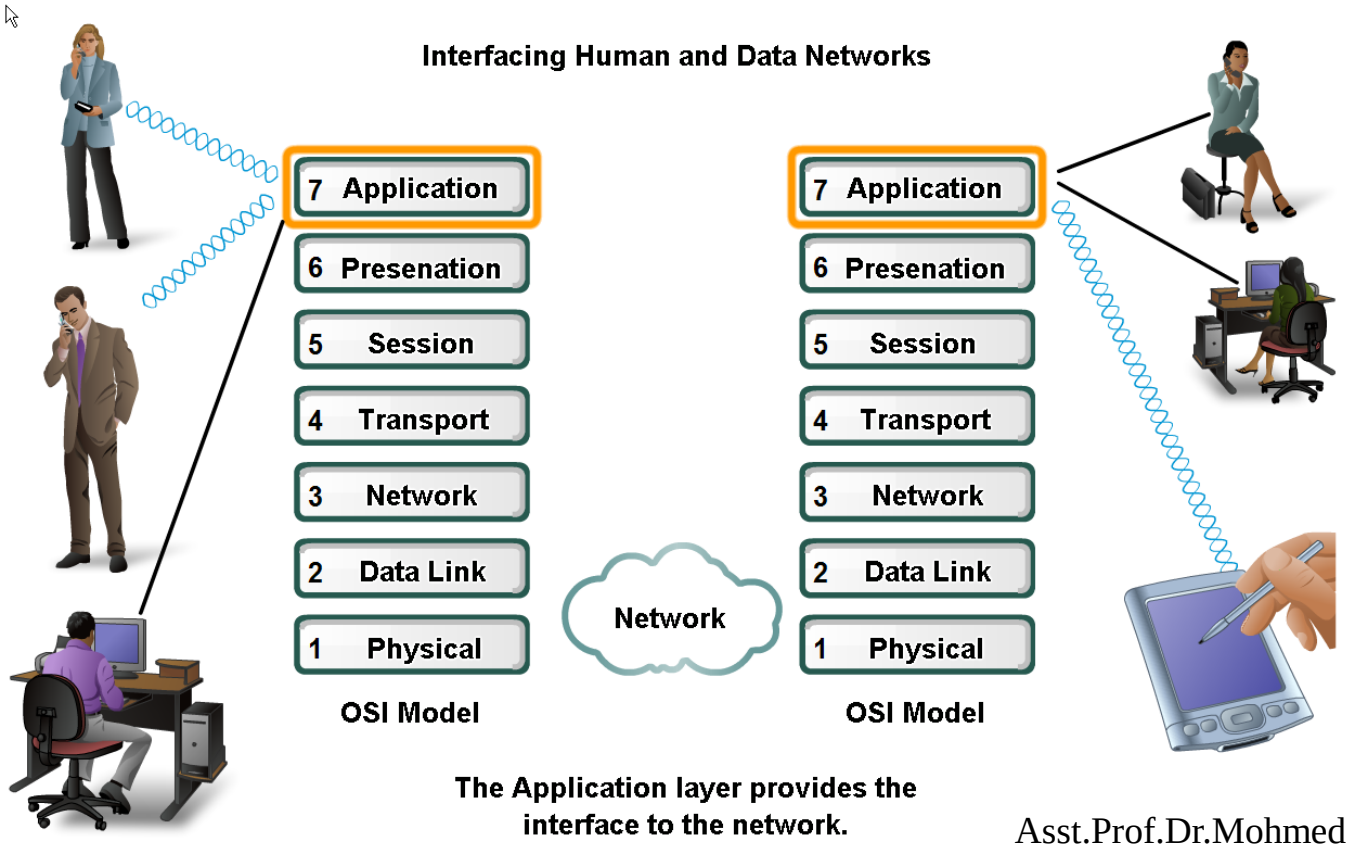
3

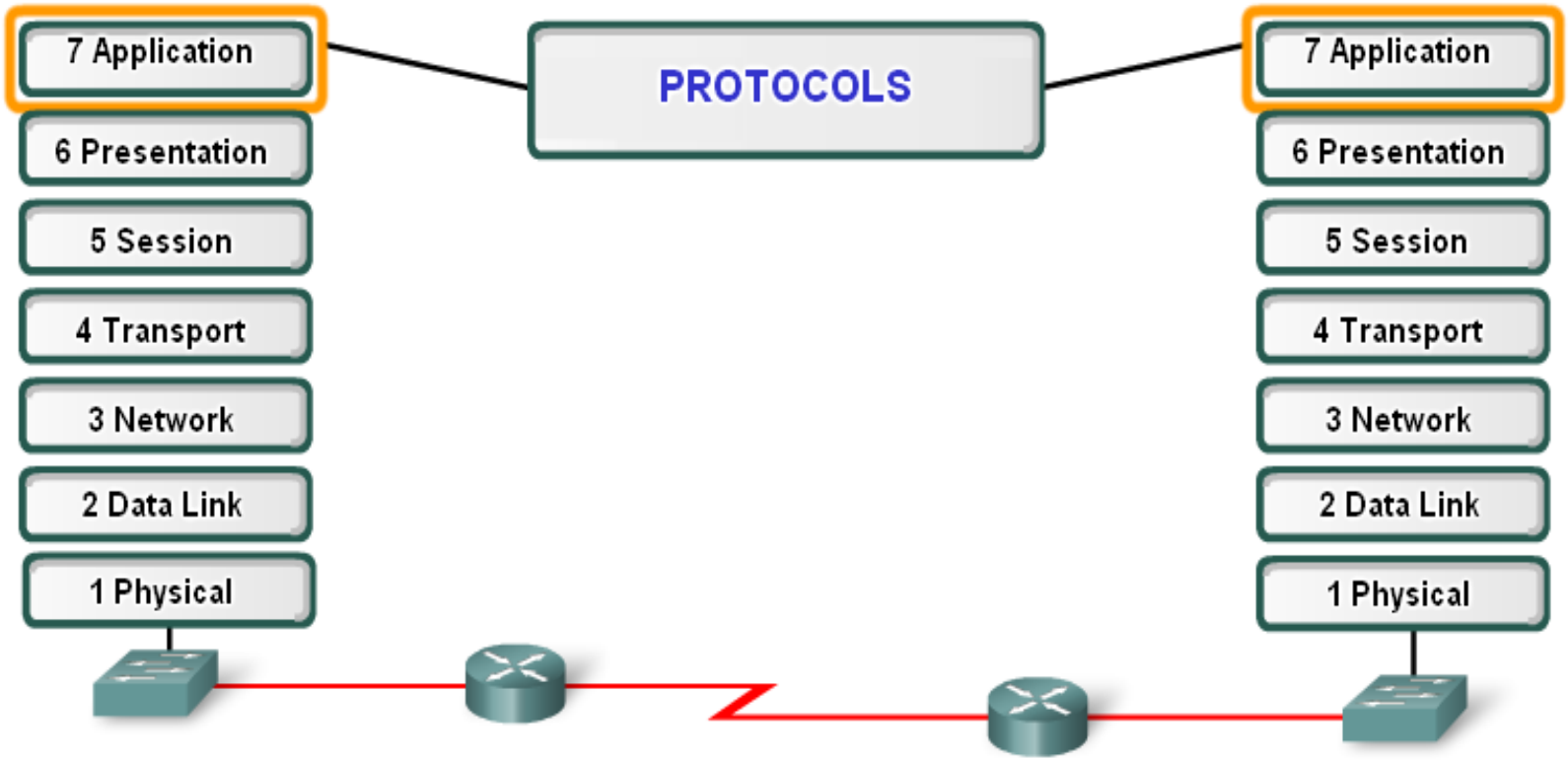
OSI model

Layer	Name	Example protocols
7	Application Layer	HTTP, FTP, DNS, SNMP, Telnet
6	Presentation Layer	SSL, TLS
5	Session Layer	NetBIOS, PPTP
4	Transport Layer	TCP, UDP
3	Network Layer	IP, ARP, ICMP, IPSec
2	Data Link Layer	PPP, ATM, Ethernet
1	Physical Layer	Ethernet, USB, Bluetooth, IEEE802.11

Applications Layer – allows user to interface with the network!

Application layer





Application layer protocols provide the rules for communication between applications.

Protocols:

- Define processes on either end of the communication
- Define the types of messages
- Define the syntax of messages
- Define the meaning of any informational fields
- Define how messages are sent and the expected response
- Define interaction with the next lower layer

Protocols

- **DNS** – Matches domain names with IP addresses
- **HTTP** – Used to transfer data between clients/servers using a web browser
- **SMTP & POP3** – used to send email messages from clients to servers over the internet
- **FTP** – allows the download/upload of files between a client/server
- **Telnet** – allows users to login to a host from a remote location and take control as if they were sitting at the machine (virtual connection)
- **DHCP** – assigns IP addresses, subnet masks, default gateways, DNS servers, etc. To users as they login the network

Application Architecture & data requests occur

- Client/server model
- Application layer services and protocols
- Peer-to-peer networking and applications

Lecture XXIII

services

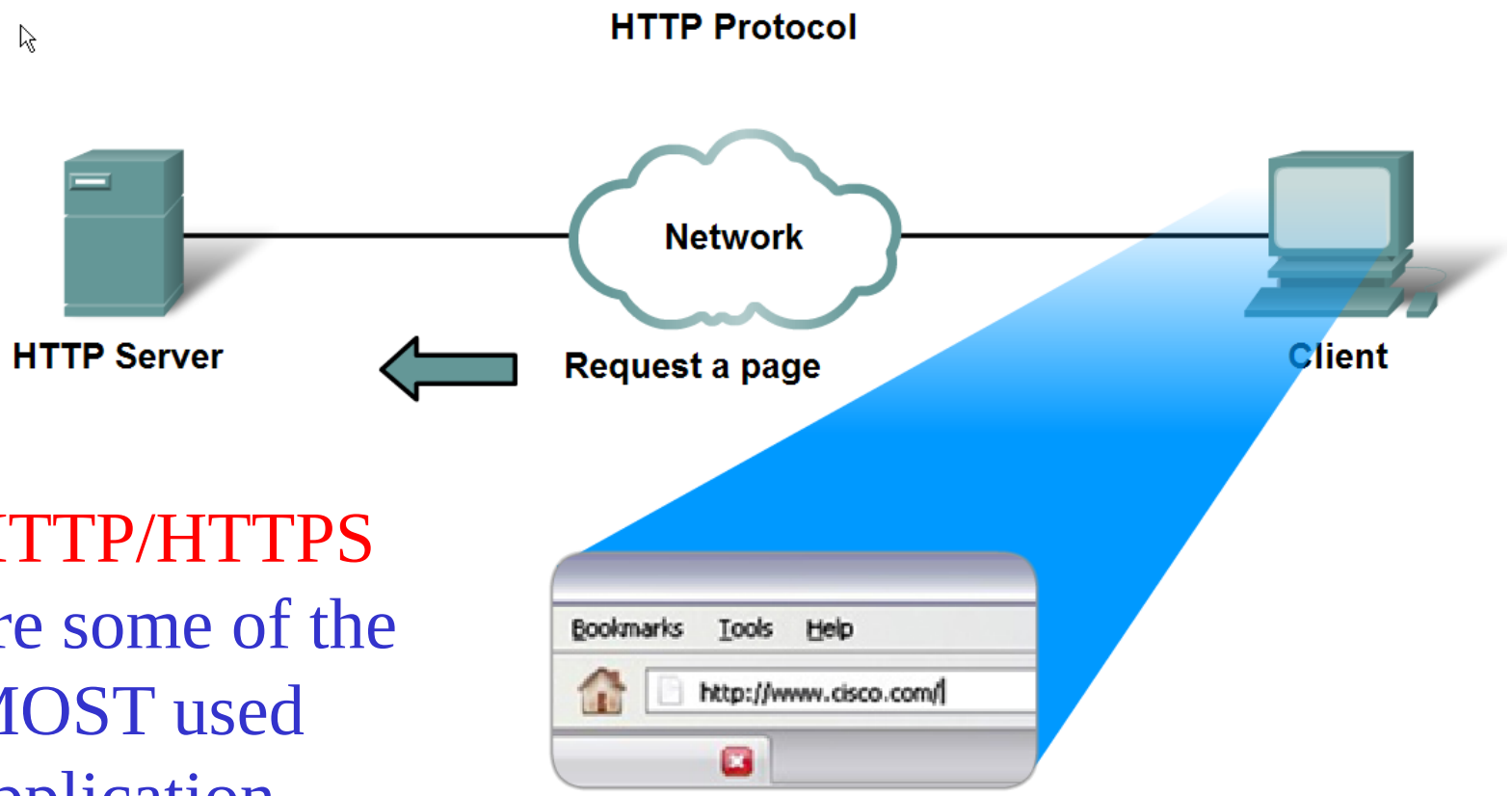
Common Port Numbers

- TCP
  - FTP – 20-21
  - Telnet – 23
  - SMTP – 25
  - DNS – 53 (Both TCP & UDP)
  - HTTP – 80
- UDP
  - DHCP – 67 & 68
  - POP – 110

Internet apps: application, transport protocols

Application	Application layer protocol	Underlying transport protocol
e-mail	SMTP [RFC 2821]	TCP
remote terminal access	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
file transfer	FTP [RFC 959]	TCP
streaming multimedia	HTTP (eg Youtube), RTP [RFC 1889]	TCP or UDP
Internet telephony	SIP, RTP, proprietary (e.g., Skype)	typically UDP

WWW Service and HTTP



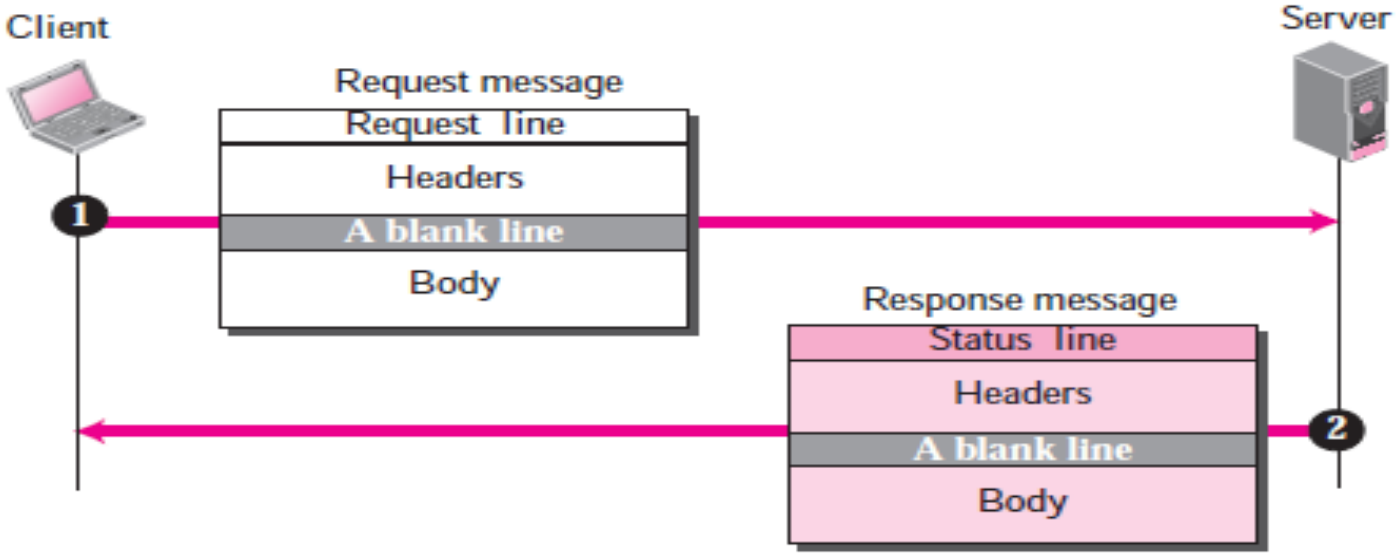
HTTP/HTTPS are some of the MOST used application protocols!

Hypertext Transfer Protocol HTTP

- The **(HTTP) is a protocol used mainly to access data on** the World Wide Web.
- HTTP use TCP connection and port 80.
- the HTTP messages are not destined to be read by humans; they are read and interpreted by the HTTP server and HTTP client (browser).

HTTP…

- HTTP itself is a stateless protocol, which means that the server does not keep information about the client. The client initializes the transaction by sending a *request.* The server replies by sending a *response*.



HTTPS protocol

- **HTTPS = HTTP + SSL**
- HTTPS stands for Hypertext Transfer Protocol over **Secure Socket Layer** (SSL), or HTTP over SSL.
- HTTPS by default uses port 443.
- URL's beginning with HTTPS indicate that the connection is encrypted using SSL.

Disadvantage of HTTPS

- HTTPS is slightly slower than HTTP because of the encryption of data.

HTTP vs HTTPS

- **URL begins with "http://" in case of HTTP while the URL begins with "https://" in case of HTTPS.**
- **HTTP is unsecured while HTTPS is secured.**
- **HTTP uses port 80 for communication while HTTPS uses port 443 for communication.**
- **HTTP operates at Application Layer while HTTPS operates at Transport and application Layer.**
- **No encryption is there in HTTP while HTTPS uses encryption.**
- **No certificates required in HTTP while certificates required in HTTPS.**

WWW Service and HTTP

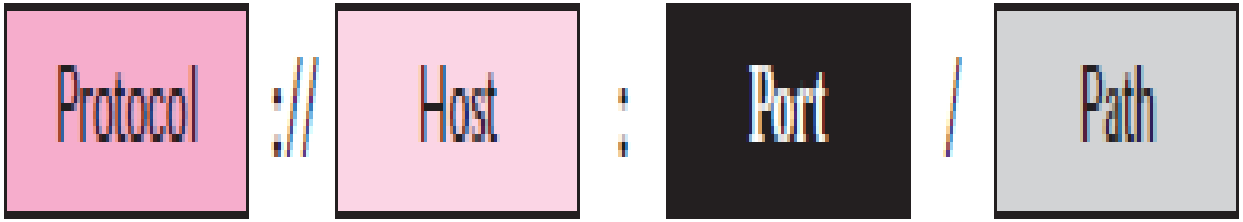
- **Steps**:

1) URL is typed in the address bar.
2) Browser checks with DNS server to convert it to an IP address
3) Connects to the server requested
4) Using **HTTP** or **HTTPS** protocol requirements, the browser sends a GET request to the server to ask for the desired html document (usually index.html)
5) The server sends the HTML code for the web page to the browser.
6) The browser interprets the HTML code and formats the page to fit the browser window.
7) See the next slide for an example.

Lecture XXIV
protocoles

URL

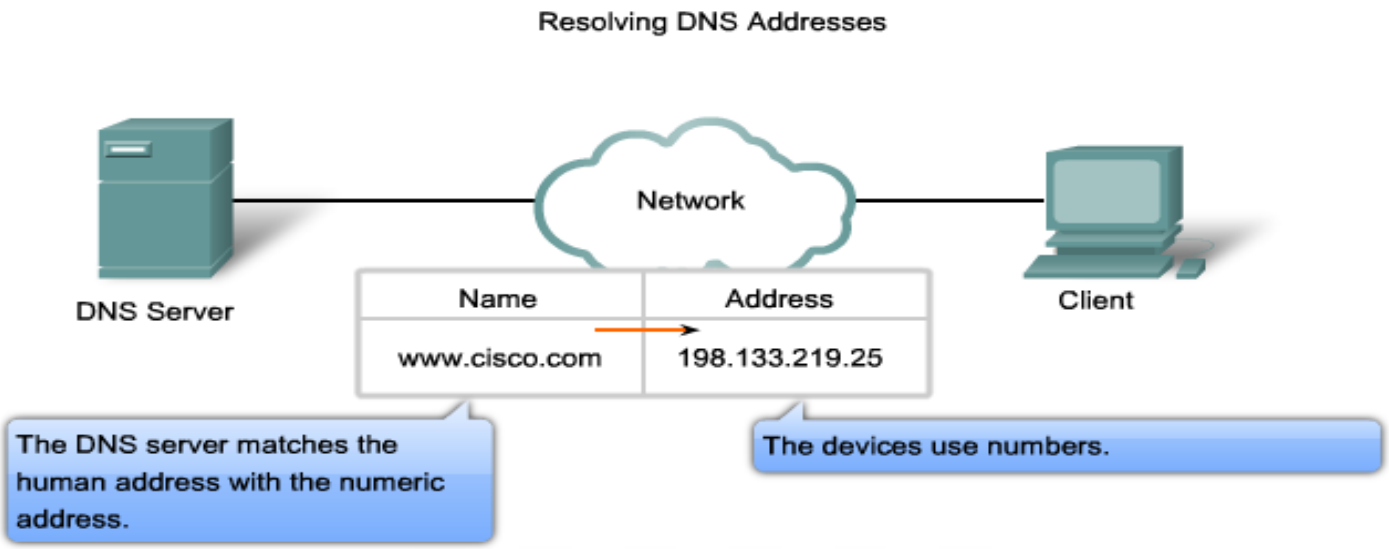
- The URL defines four things: *protocol*, *host computer*, *port*, *and path*.



➢ URL information located at Request message of HTTP

➢ The **uniform resource locator (URL) is a standard locator for specifying any kind of information on the Internet.**

DNS Services and Protocol



DNS (domain name service) Servers resolve names to IP addresses. It would be difficult to remember the IP address of every website we like to visit, but we can remember names.

DNS work &Problems

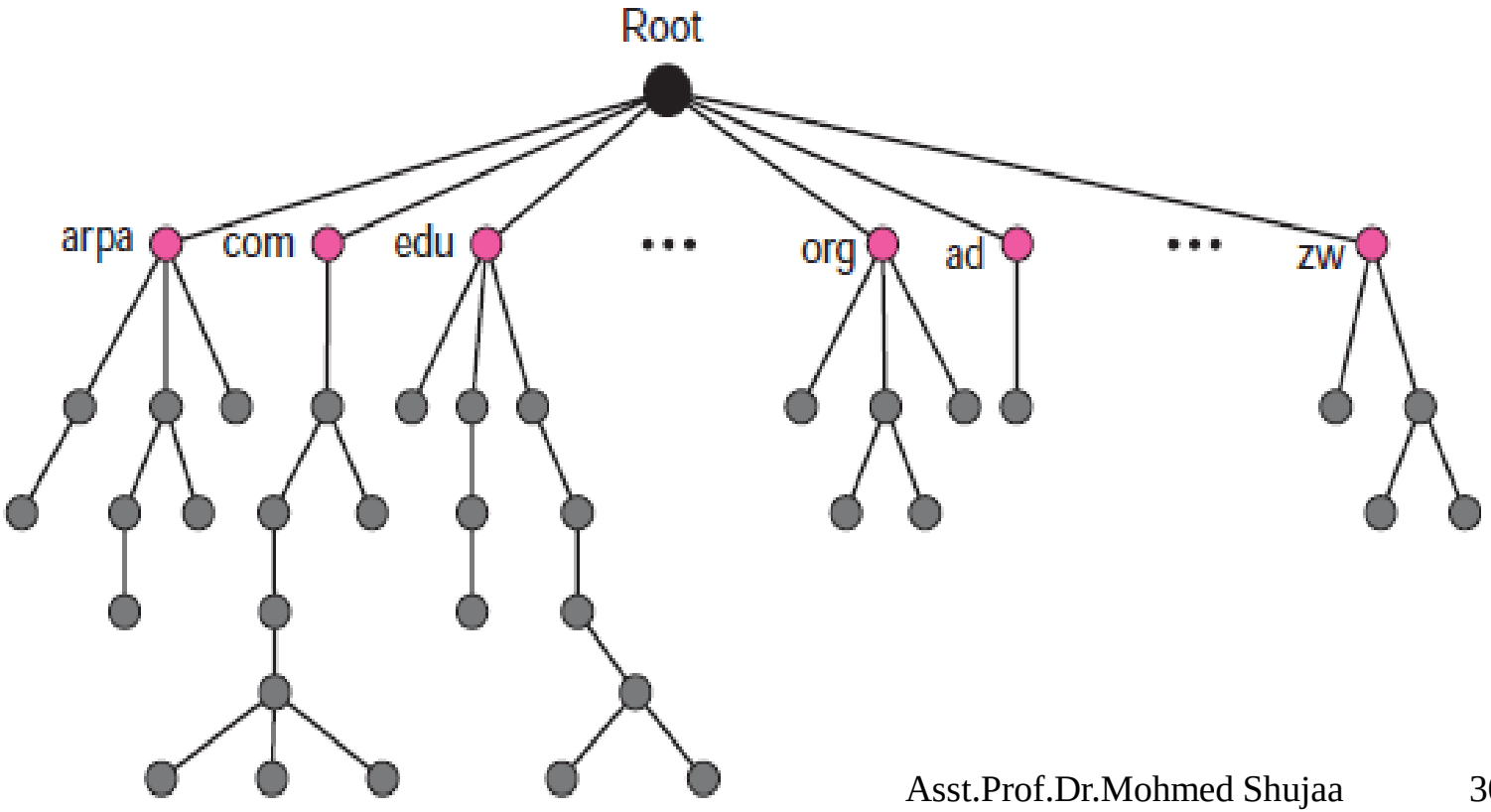
- Once DNS server **can't** response to all of the demands that coming from all over the world.
- **The problem** is how we can distribute the traffic among more DNS servers.
- The problem solved by **Domain Name Space**.
- A **name space** distributed the traffic by two ways: **flat or hierarchical.**

Types of Domain Name Space

- **Flat name space, a name is assigned to an address. A name in this space is a** sequence of characters without structure. **not used in Internet .**
- **Hierarchical name space, each name is made of several parts. The first part can** define the nature of the organization, the second part can define the name of an organization, the third part can define departments in the organization. **Use at internet**.

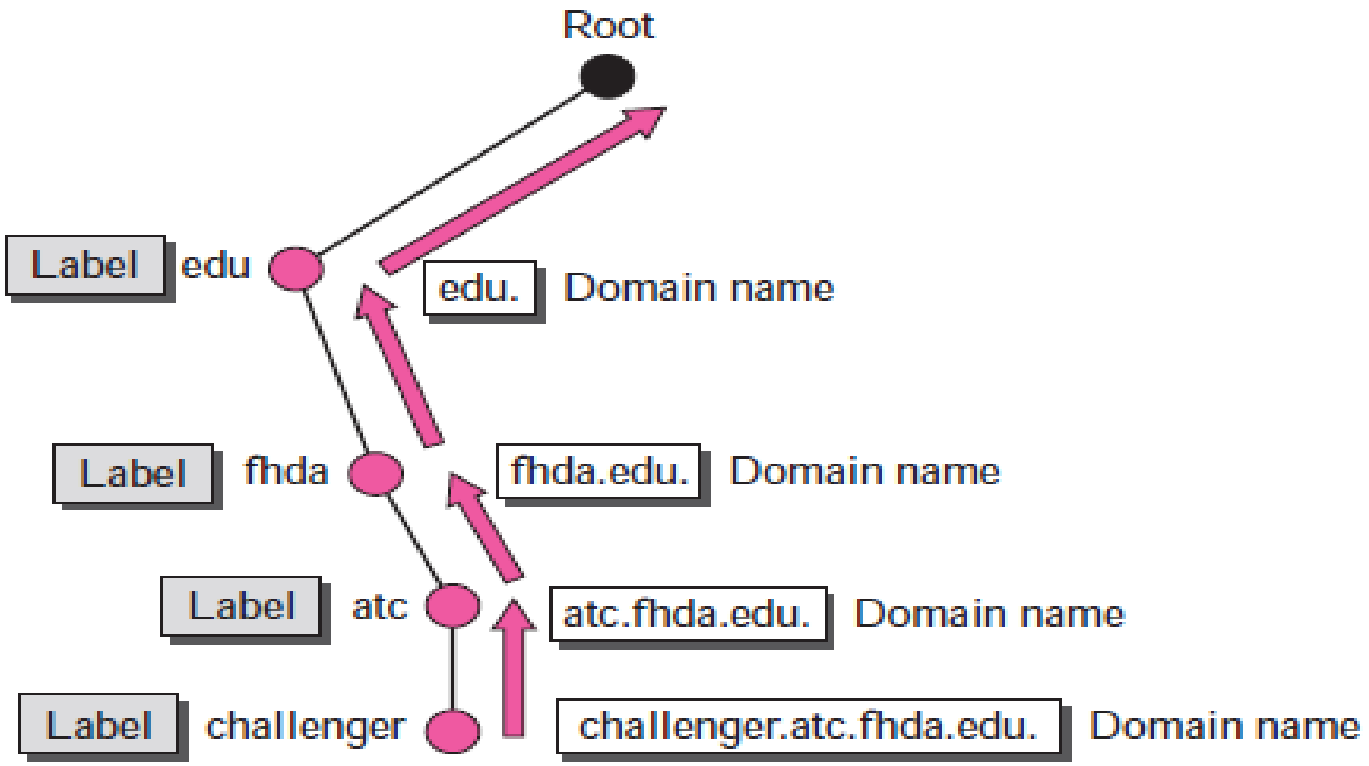
hierarchical Structure

- hierarchical tree structure with one root



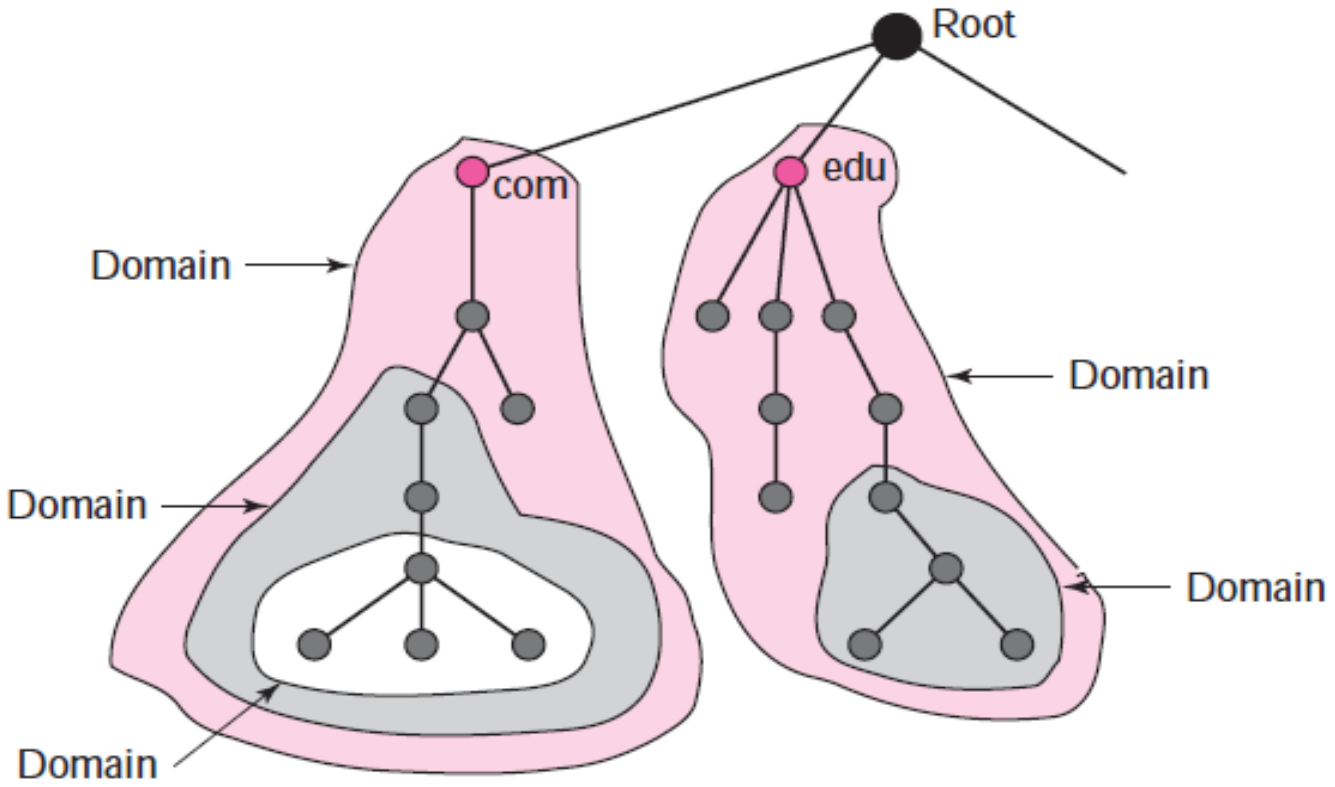
Hierarchical Structure

- The tree can have only 128 levels: level 0 (root) to level 127.

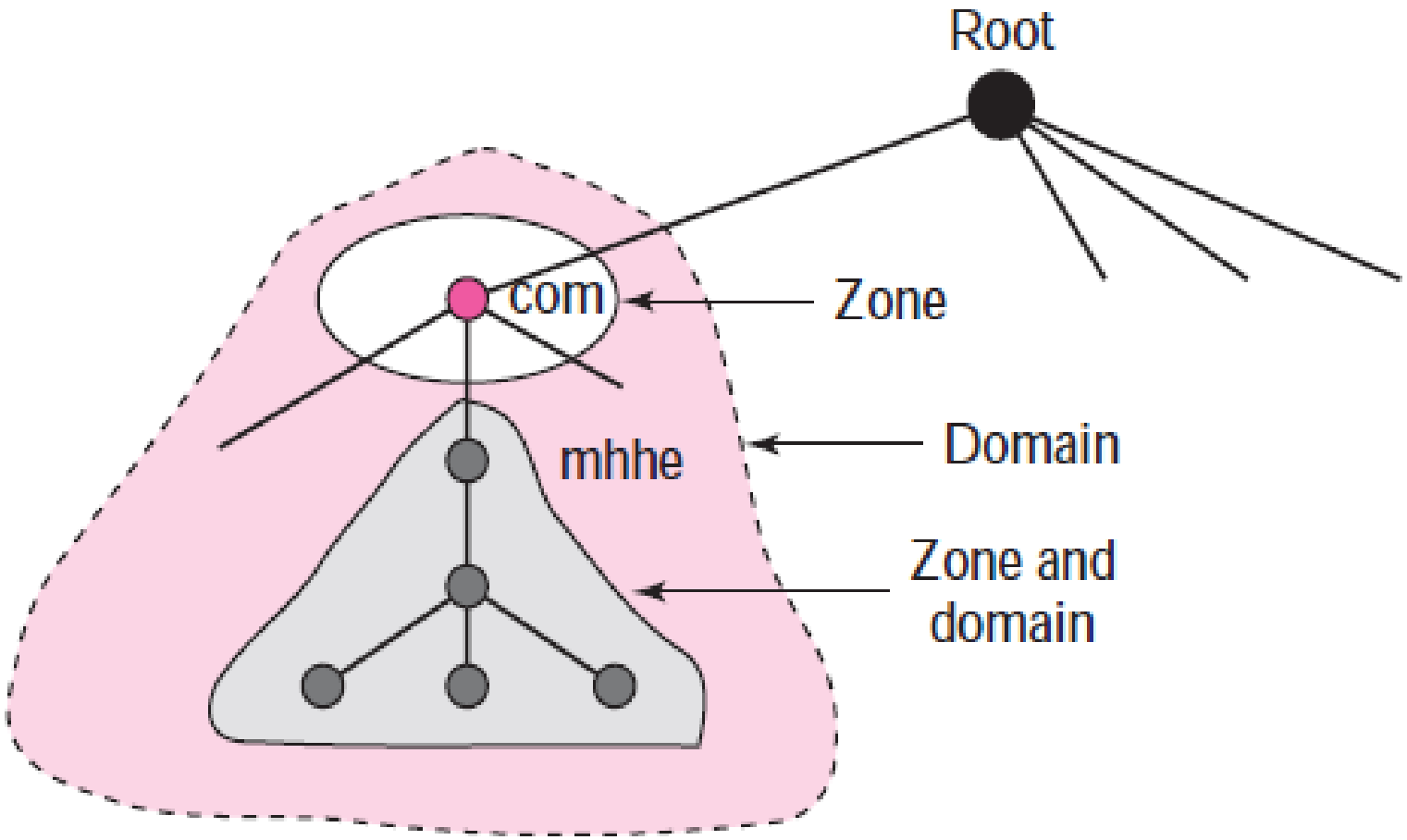


Domain

- A **domain is a sub tree of the domain name space.**

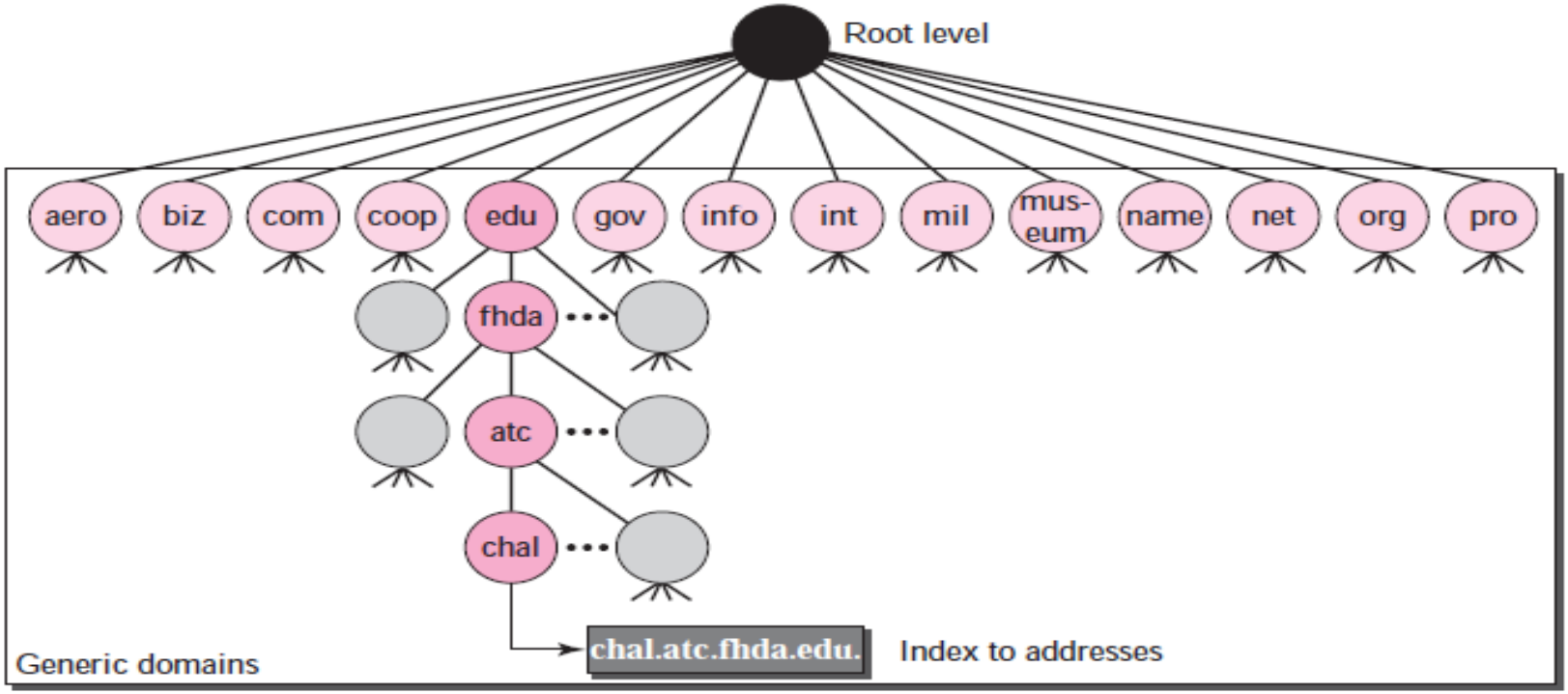


Zone in name space



Generic domains

- There are **fourteen** generic domains, each specifying an organization type.

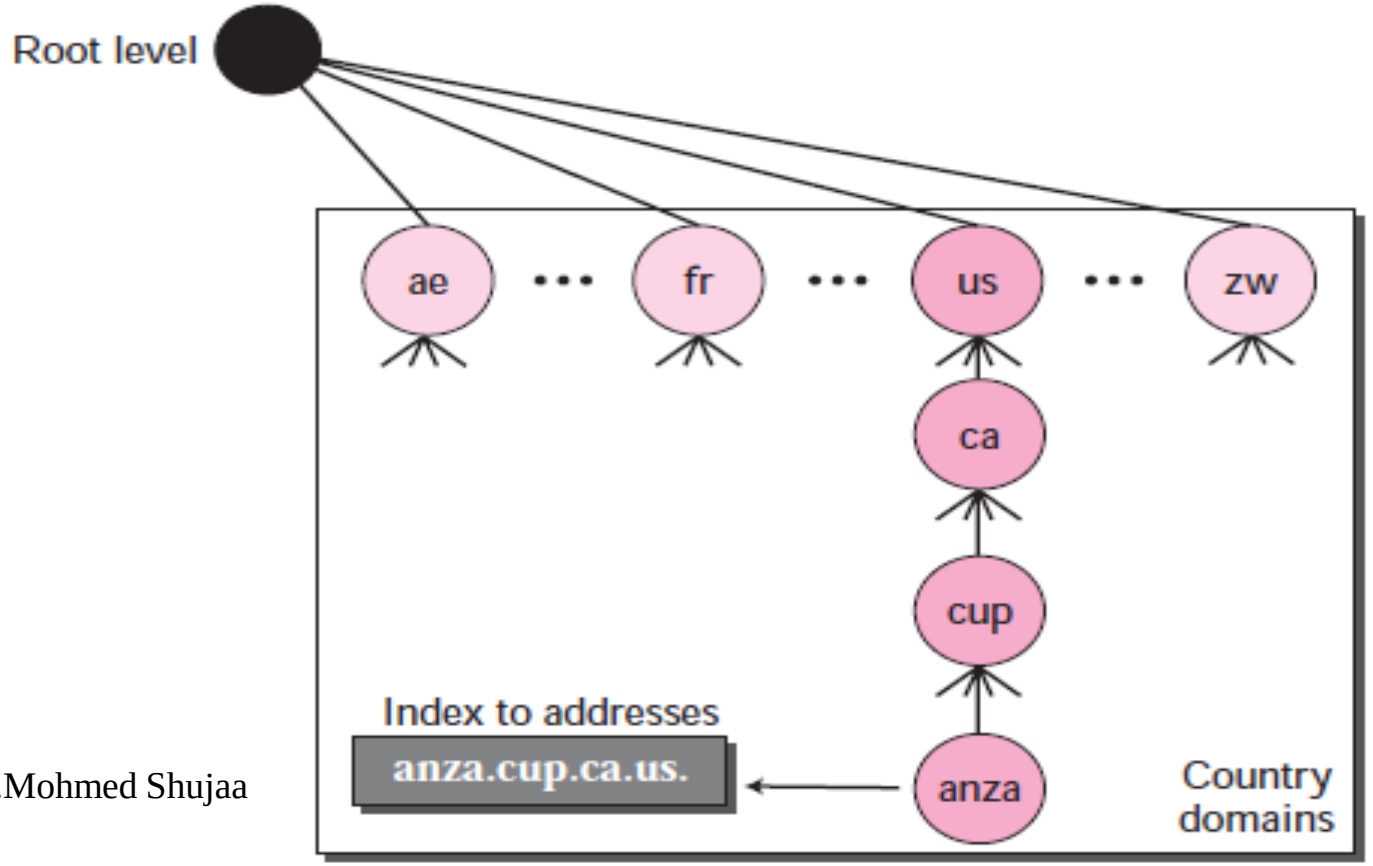


Generic domain labels

Label	*Description*
aero	Airlines and aerospace companies
biz	Businesses or firms (similar to "com")
com	Commercial organizations
coop	Cooperative business organizations
edu	Educational institutions
gov	Government institutions
info	Information service providers
int	International organizations
mil	Military groups
museum	Museums and other non-profit organizations
name	Personal names (individuals)
net	Network support centers
org	Nonprofit organizations
pro	Professional individual organizations

Country domains

- The country domains section uses **two character** country abbreviations.
- Second labels can be organizational and so on.



Lecture XXVI

Protocol services

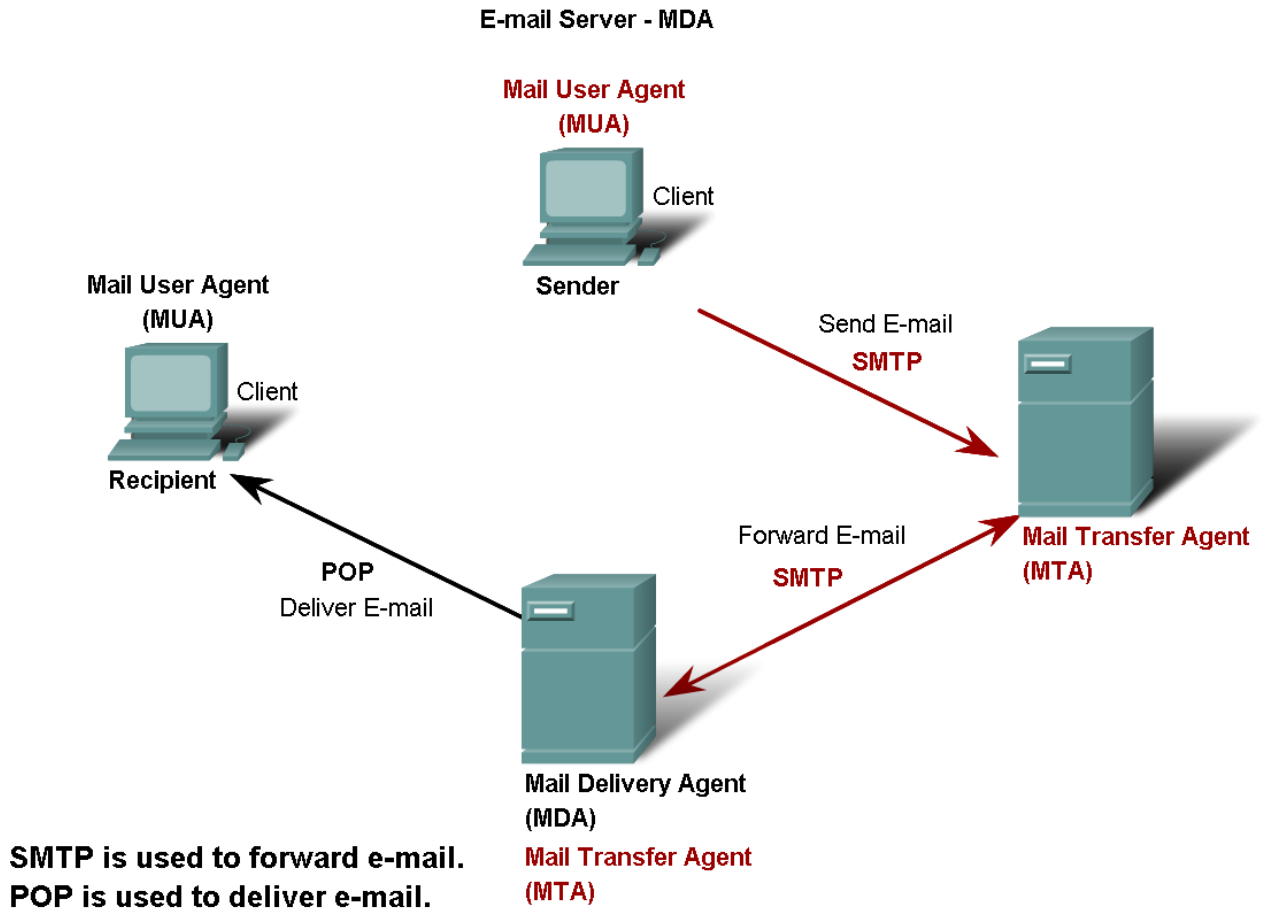
E-mail services and SMTP/POP protocols

- What do servers require?

1) **SMTP**!(simple mail transfer protocol)
2) Also operates
    1. Mail Transfer Agent (MTA) – used to forward email
    2. Receives email from the clients MUA mail user agency
    3. SMTP is used to route email between SERVERS!
    4. Passes email to the MDA for final mail delivery

3) For two e-mail servers to talk – MUST run SMTP and MTA in order to transfer mail between the 2 servers!
4) POP: post office protocol: is used to deliver e mail

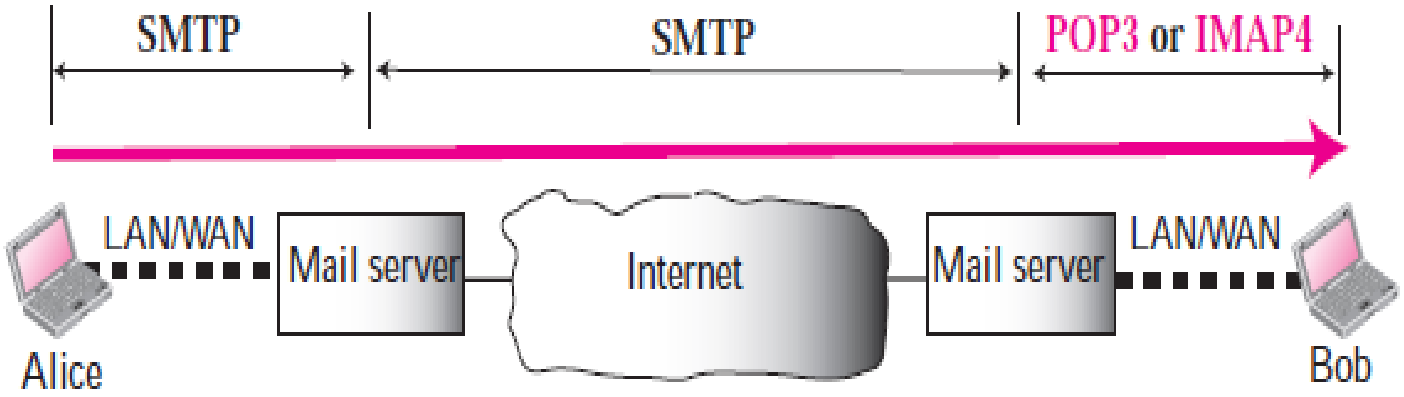
E-mail services and SMTP/POP protocols



MESSAGE ACCESS AGENT: POP AND IMAP

The first and the second stages of mail delivery use SMTP. However, SMTP is not involved in the third stage because **SMTP** is a ***push protocol***; *it pushes the message* from the client to the server.

POP: Post Office Protocol IMAP: Internet Mail Access Protocol



POP3 and IMAP

More about POP3

- Previous example uses "download and delete" mode.
- cannot re-read e-mail if he changes client
- "Download-and-keep": copies of messages on different clients
- POP3 is stateless across sessions

IMAP

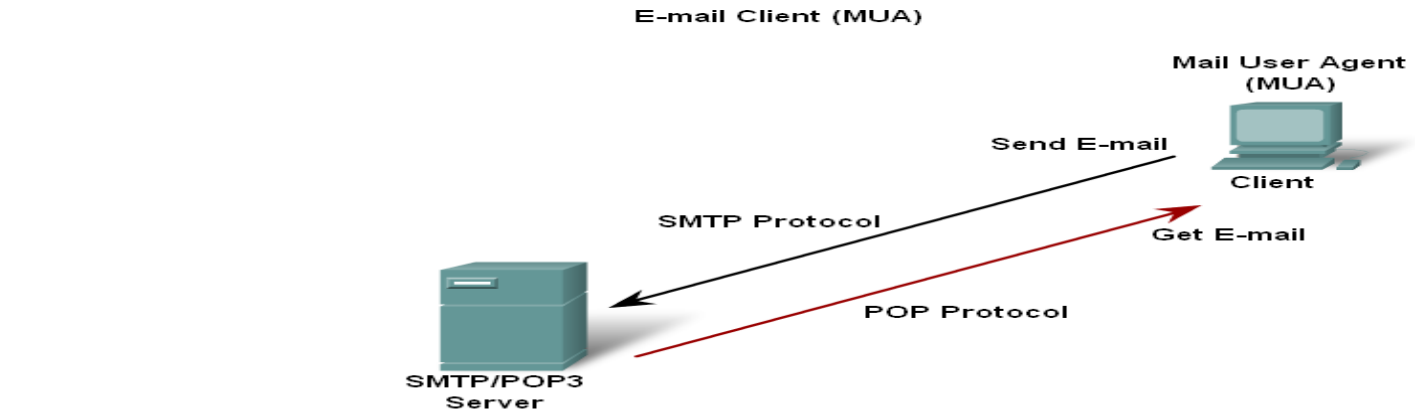
- Keep all messages in one place: the server
- Allows user to organize messages in folders
- IMAP keeps user state across sessions:
  - names of folders and mappings between message IDs and folder name

E-mail Server Processes - MTA and MDA and the SMTP protocol

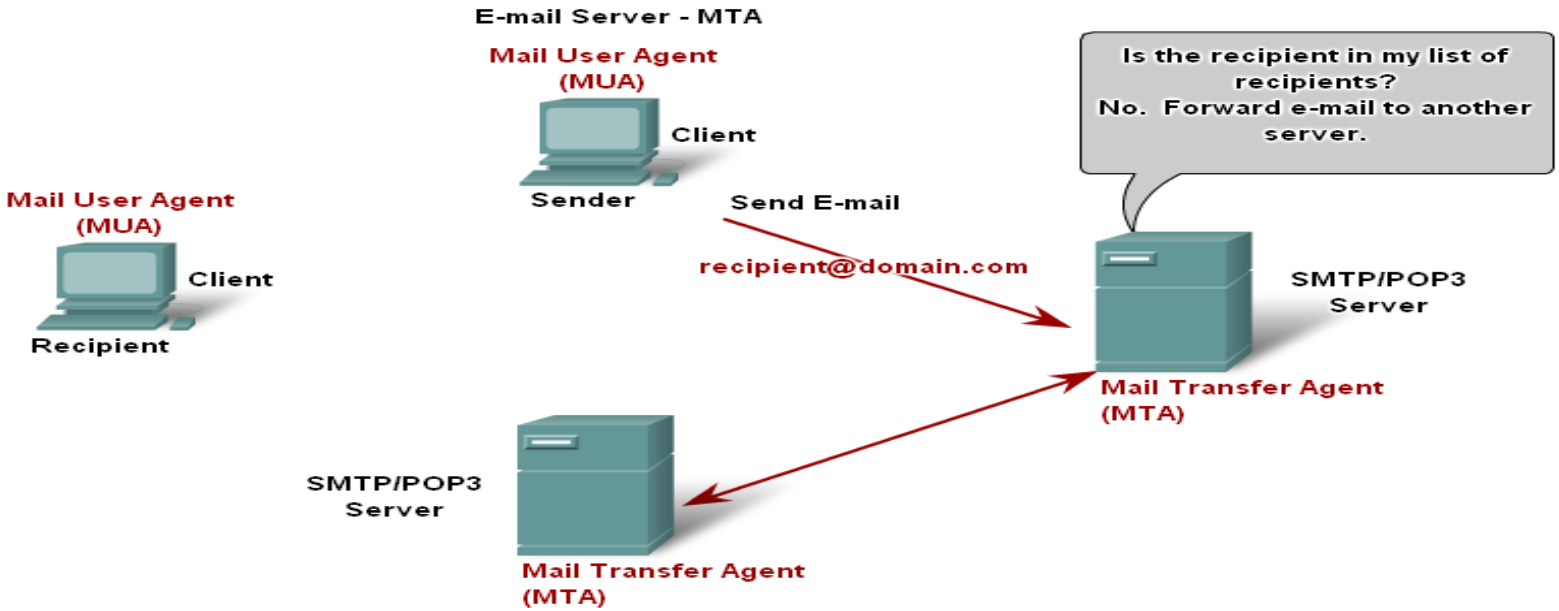
- The e-mail server operates two separate processes:
- 1- Mail Transfer Agent (MTA)
- 2- Mail Delivery Agent (MDA)

The Mail Transfer Agent (MTA) process is used *to forward e-mail.*

The MTA receives messages from the MUA or from another MTA on another e-mail server. Based on the message header, it determines how a message has to be forwarded to reach its destination. If the mail is addressed to a user whose mailbox is on the local server, the mail is passed to the MDA. If the mail is for a user not on the local server, the MTA routes the e-mail to the MTA on the appropriate server.



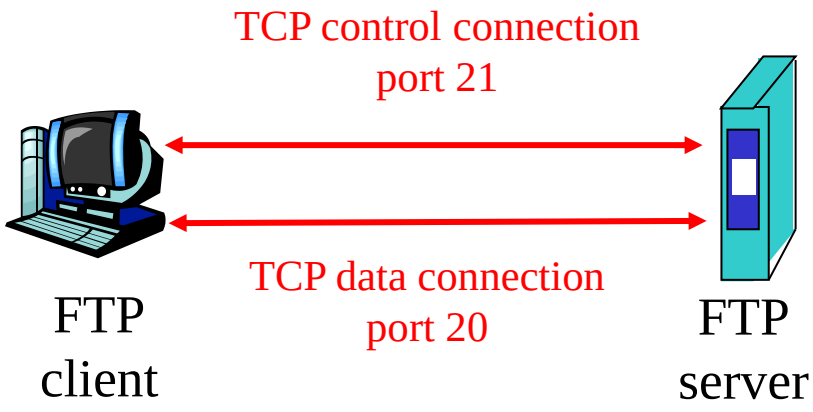
Clients send e-mails to a server using SMTP and receive e-mails using POP3.



The Mail Transfer Agent process governs e-mail handling between servers and servers.

FTP protocol

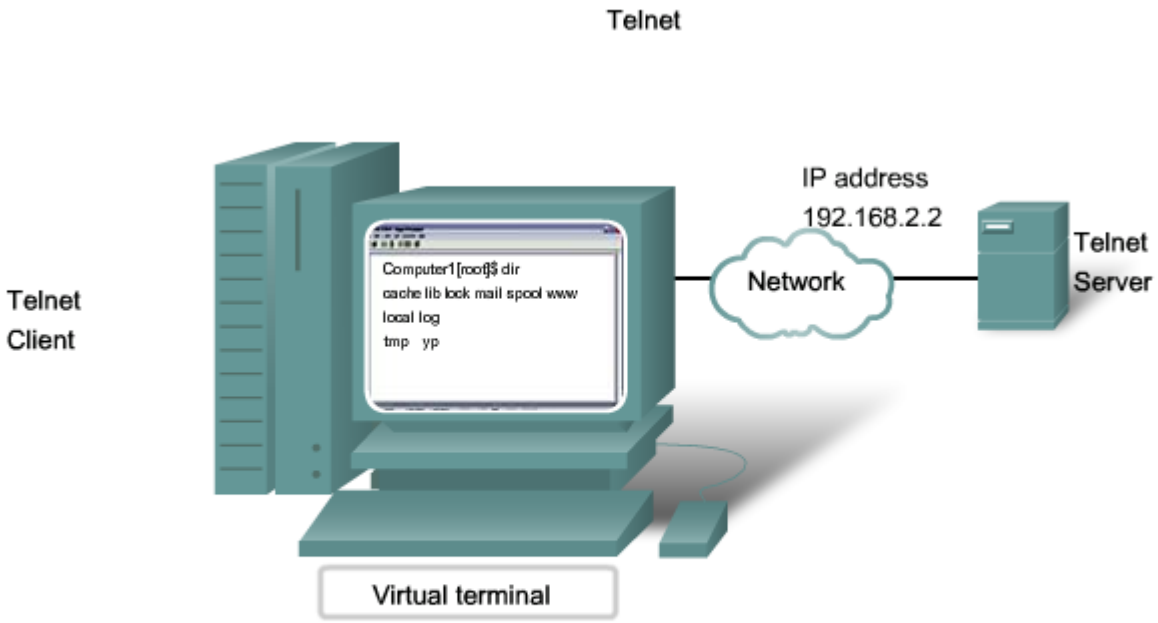
- Commonly used application layer protocol
- Allows for the transfer of files between clients/servers.
- Requires 2 connections to the server
  1) Commands – uses TCP port 21
  2) Actual data – uses TCP port 20



Telnet protocol

- protocols and services in the TCP/IP protocol suite.
- Allows users to emulate (simulation) text-based terminal devices over the network using software. Virtual connection
- Can be run from the command prompt on a PC.
- You can use the device as if you were sitting there with all the rights and priorities that you username will offer you.
- Disadvantages: Doesn't support encryption like SSH. All data is transferred as plain text. It can be easily intercepted and understood.

Telnet



Telnet provides a way to use a computer, connected via the network, to access a network device as if the keyboard and monitor were directly connected to the device.

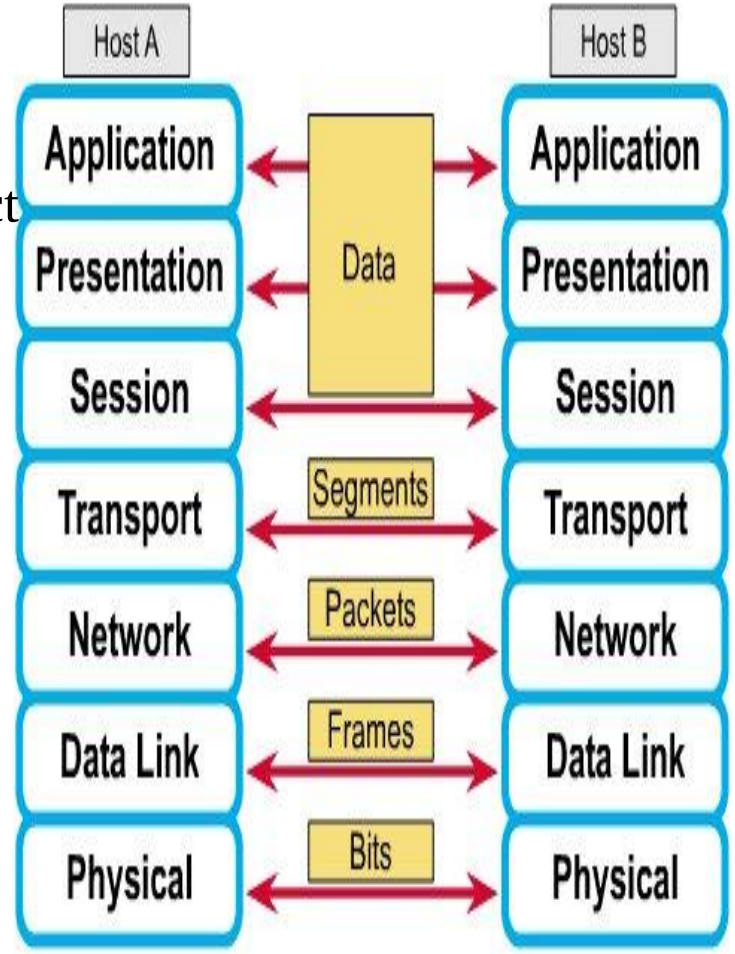
Lecture XXVII

PDU &TCP MODEL

How Does It All Work Together

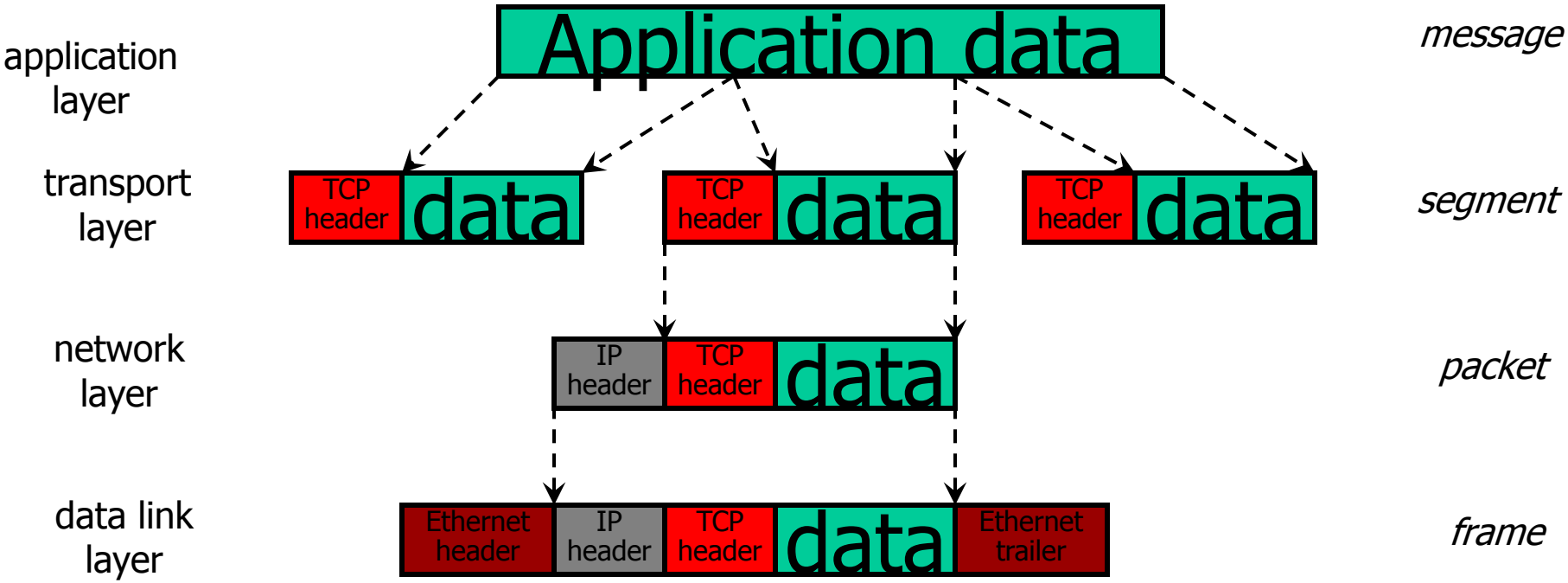
Protocol Data Unit (PDU)

- Each layer contains a
  - PDU's are used for peer-to-peer contact between corresponding layers.



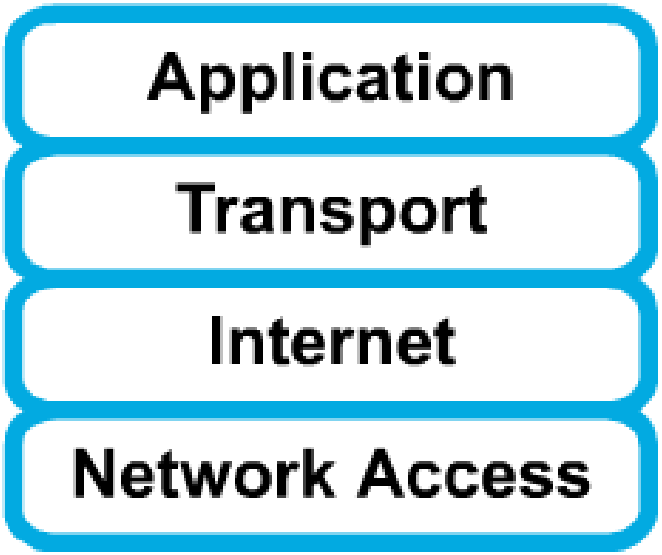
2

Data Formats



4 layers of the TCP/IP model

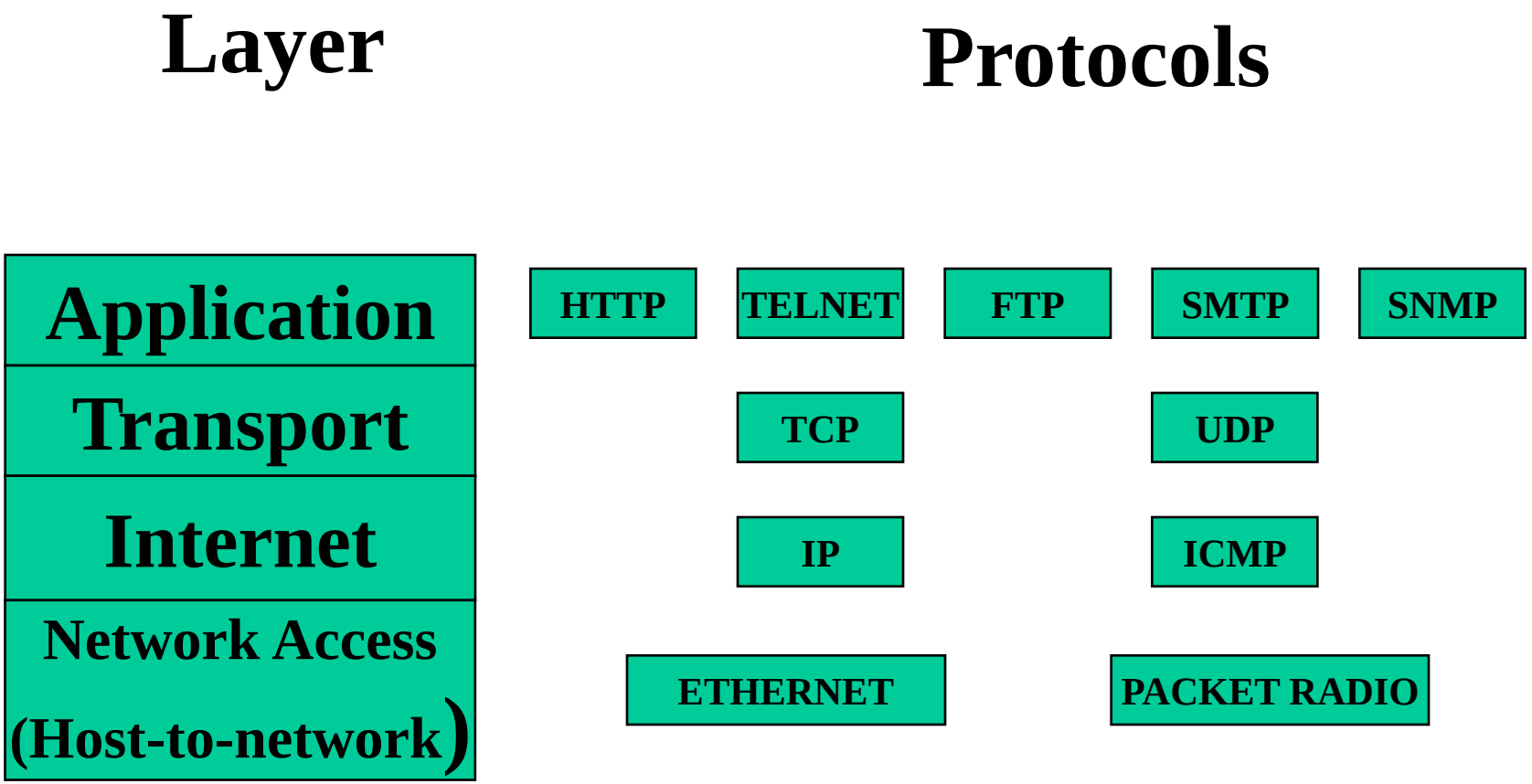
- Layer 4: Application
- Layer 3: Transport
- Layer 2: Internet
- Layer 1: Network access



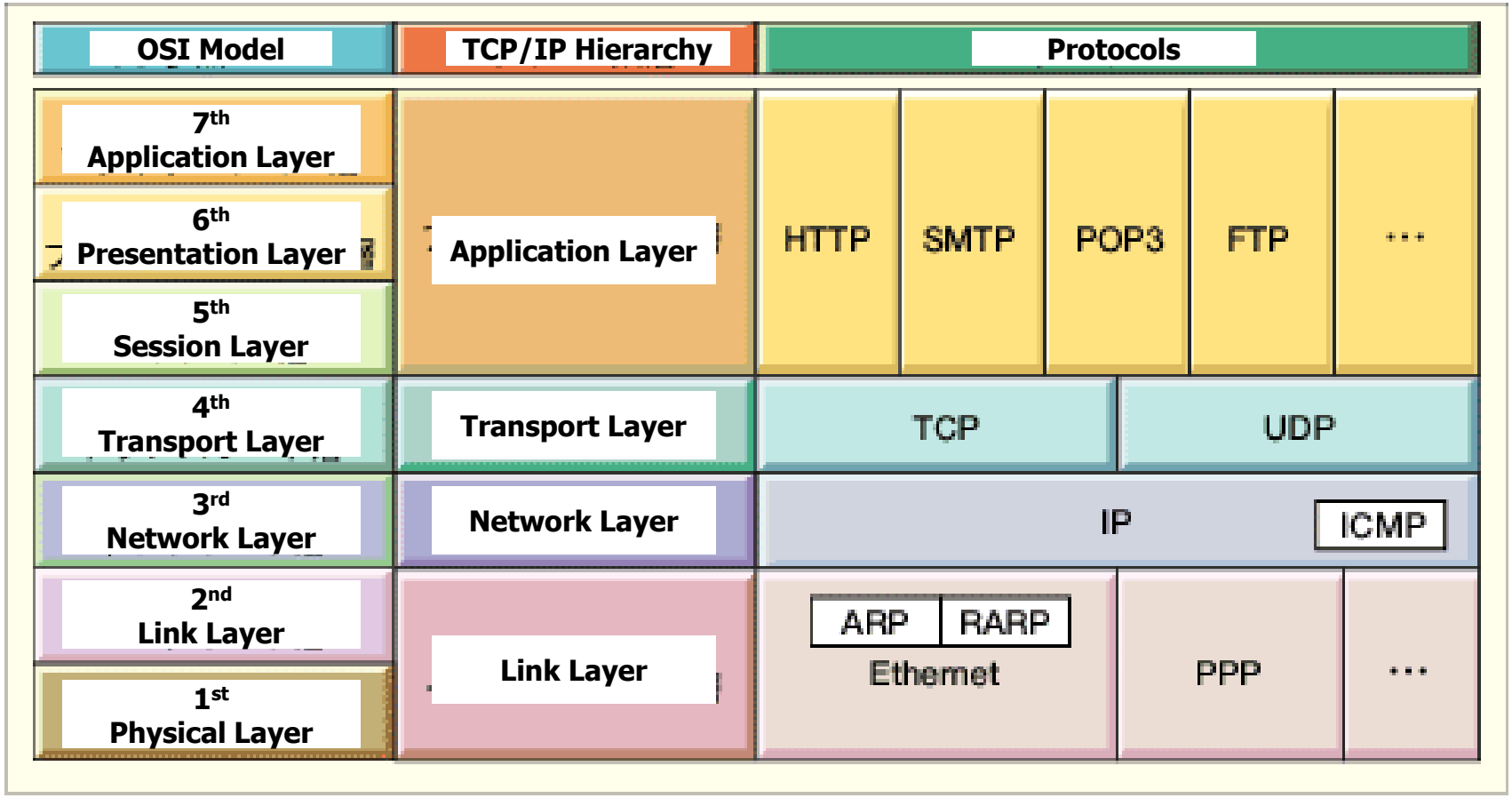
It is important to note that some of the layers in the TCP/IP model have the same name as layers in the OSI model.

Do not confuse the layers of the two models.

TCP/IP Reference Model & protocol stack



Comparing TCP/IP with OSI



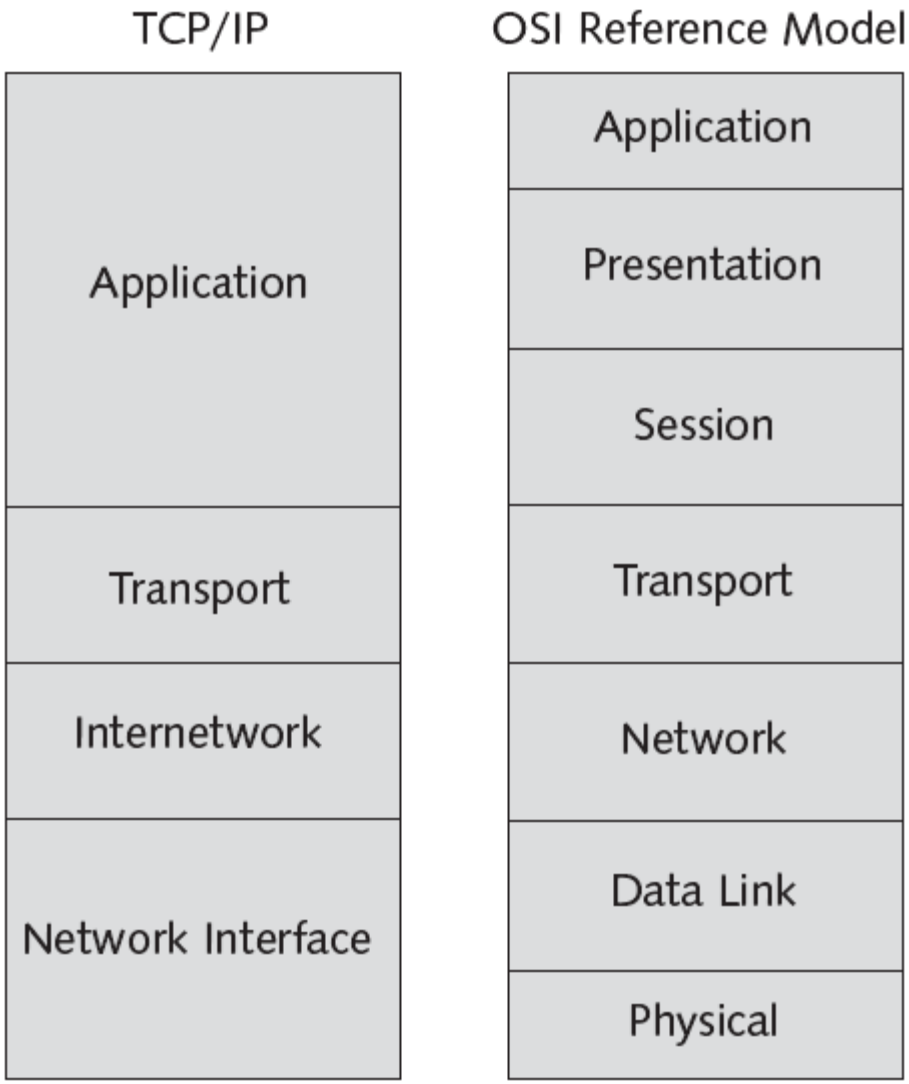


Figure 3-1 Protocol architecture comparison

End Of Lectures
Good Luck