# Unit No:5

# Link Layer and Local Area Networks

Marwadi University

Department of Computer Engineering

Computer Networks (3150710)

Dr. Sushil Kumar Singh
Associate Professor

Syllabus

# Link layer, LANs: outline

5.1 introduction, services

5.2 error detection, correction

5.3 multiple access protocols

5.4 LANs

- addressing, ARP
- Ethernet
- switches
- VLANS

5.5 link virtualization: MPLS

5.6 data center networking

5.7 a day in the life of a web request

Goals

# Chapter 5: Link layer

*our goals:*

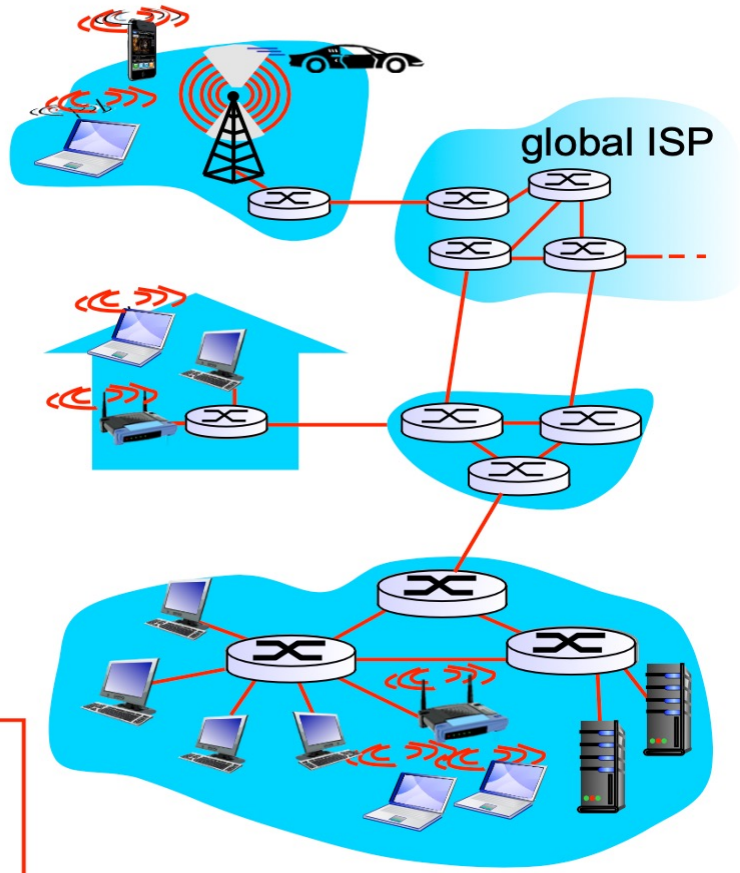
- understand principles behind link layer services:
  - error detection, correction
  - sharing a broadcast channel: multiple access
  - link layer addressing
  - local area networks: Ethernet, VLANs
- instantiation, implementation of various link layer technologies

Link Layer and LANs

# Link layer: introduction

*terminology:*

- hosts and routers: nodes
- communication channels that connect adjacent nodes along communication path: links
  - wired links
  - wireless links
  - LANs
- layer-2 packet: frame, encapsulates datagram




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

Link Layer and LANs

# Link layer: context

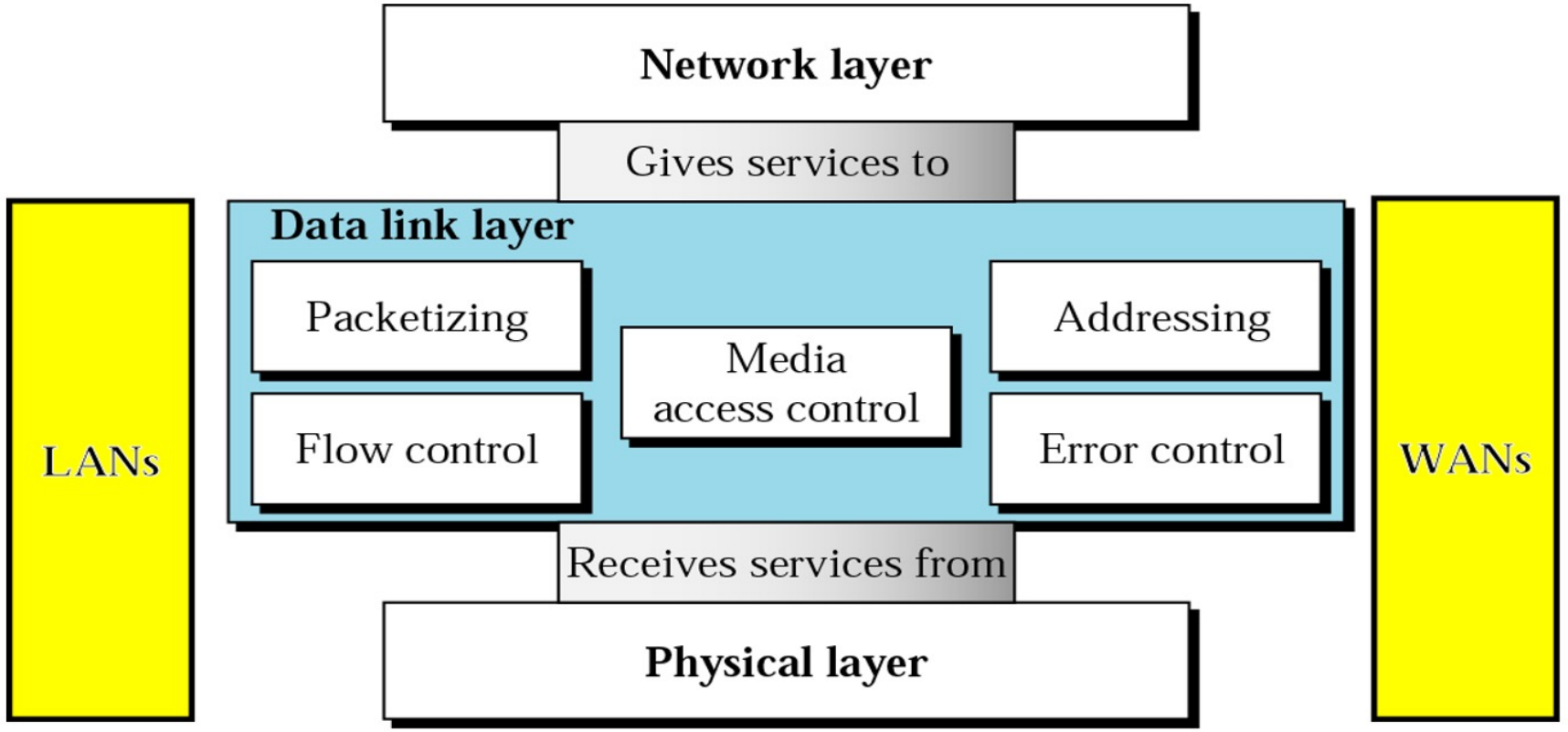
- datagram transferred by different link protocols over different links:
  - e.g., Ethernet on first link, frame relay on intermediate links, 802.11 on last link
- each link protocol provides different services
  - e.g., may or may not provide rdt over link

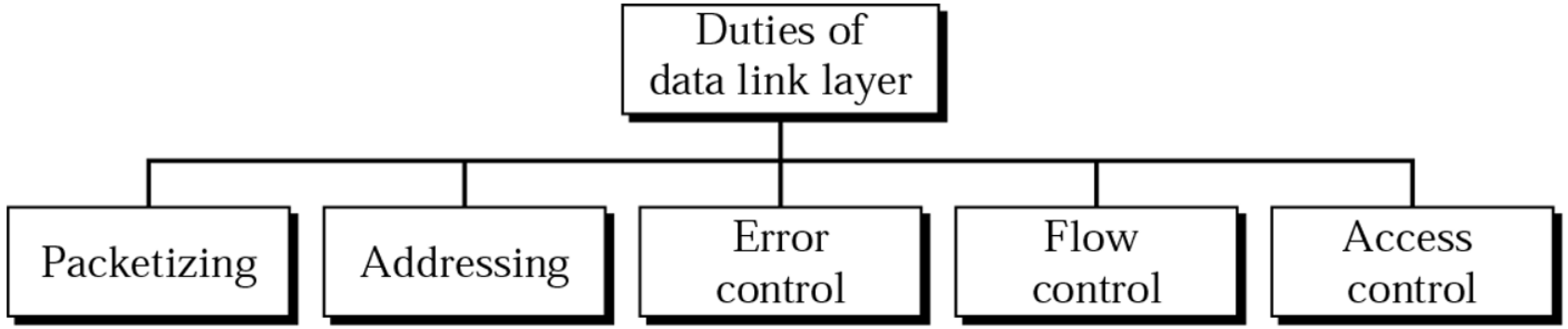
*transportation analogy:*

- trip from Princeton to Lausanne
  - limo: Princeton to JFK
  - plane: JFK to Geneva
  - train: Geneva to Lausanne
- tourist = datagram
- transport segment = communication link
- transportation mode = link layer protocol
- travel agent = routing algorithm

# Link Layer and LANs

(Position of Data Link Layer and Duties)



Link Layer and LANs

(Services and Duties)

# Link layer services

- *framing, link access:*
  - encapsulate datagram into frame, adding header, trailer
  - channel access if shared medium
  - "MAC" addresses used in frame headers to identify source, dest
    - different from IP address!
- *reliable delivery between adjacent nodes*
  - we learned how to do this already (chapter 3)!
  - seldom used on low bit-error link (fiber, some twisted pair)
  - wireless links: high error rates
    - *Q:* why both link-level and end-end reliability?

Link Layer and LANs

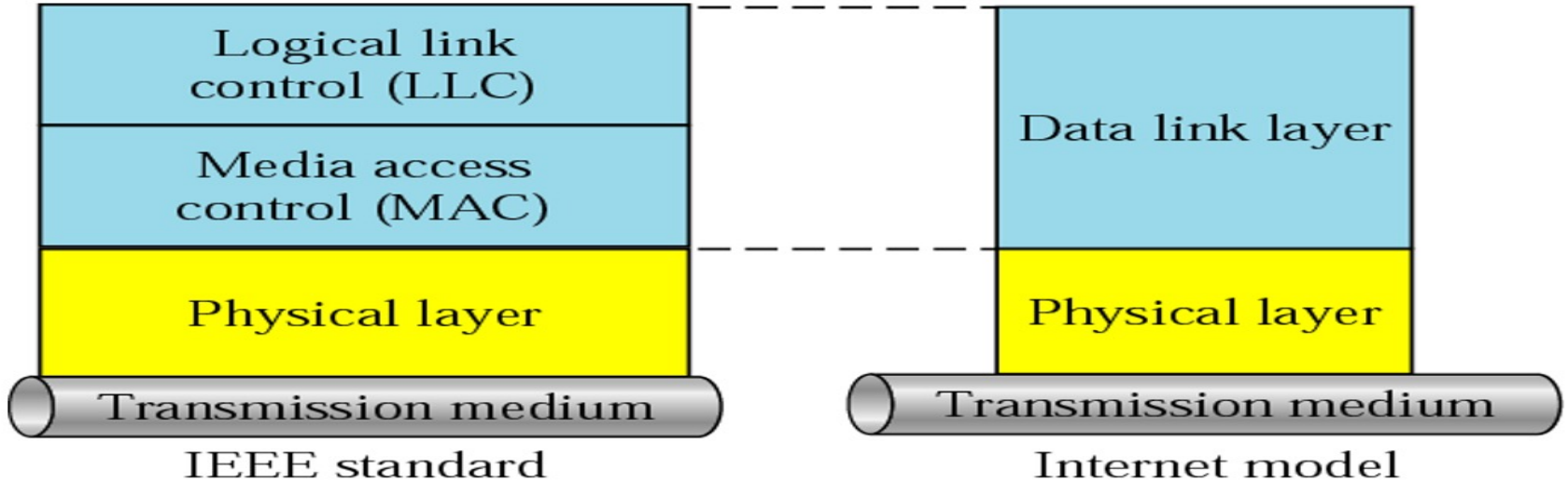
(Services and Duties)

# Link layer services (more)

- *flow control:*
  - pacing between adjacent sending and receiving nodes
- *error detection:*
  - errors caused by signal attenuation, noise.
  - receiver detects presence of errors:
    - signals sender for retransmission or drops frame
- *error correction:*
  - receiver identifies *and corrects* bit error(s) without resorting to retransmission
- *half-duplex and full-duplex*
  - with half duplex, nodes at both ends of link can transmit, but not at same time

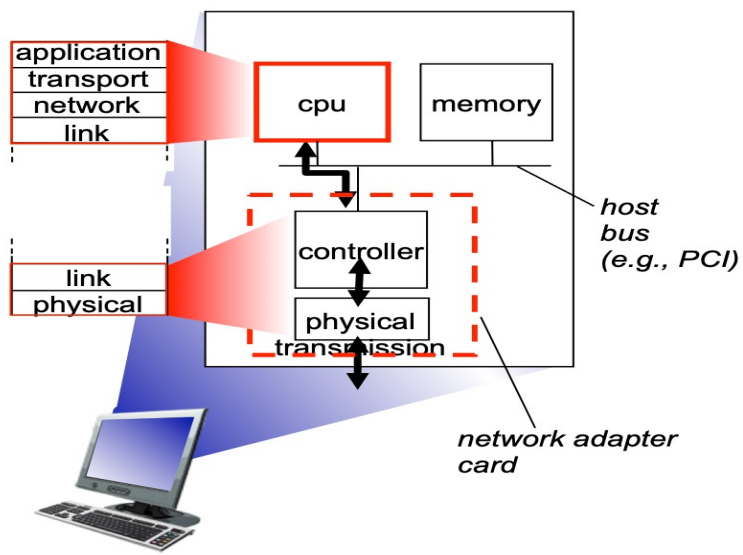
# Link Layer and LANs

(LLC and MAC Sub Layers)




## Where is the link layer implemented?

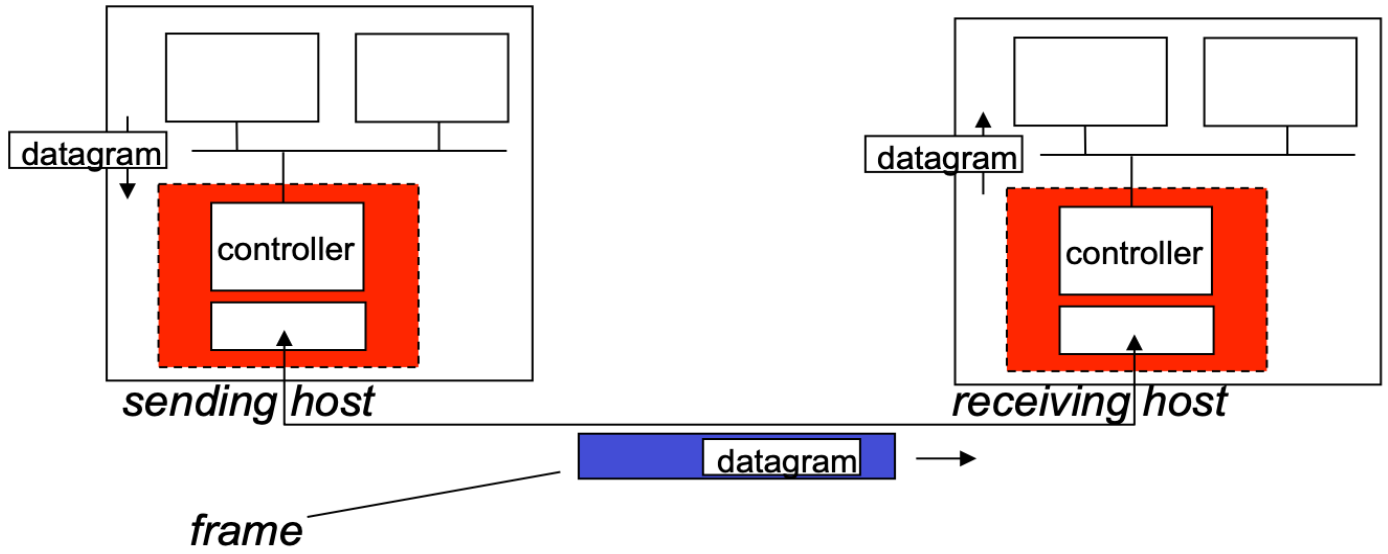
- in each and every host
- link layer implemented in "adaptor" (aka *network interface card* NIC) or on a chip
  - Ethernet card, 802.11 card; Ethernet chipset
  - implements link, physical layer
- attaches into host's system buses
- combination of hardware, software, firmware



Link Layer and LANs

(Adaptors Communicating)

# Adaptors communicating

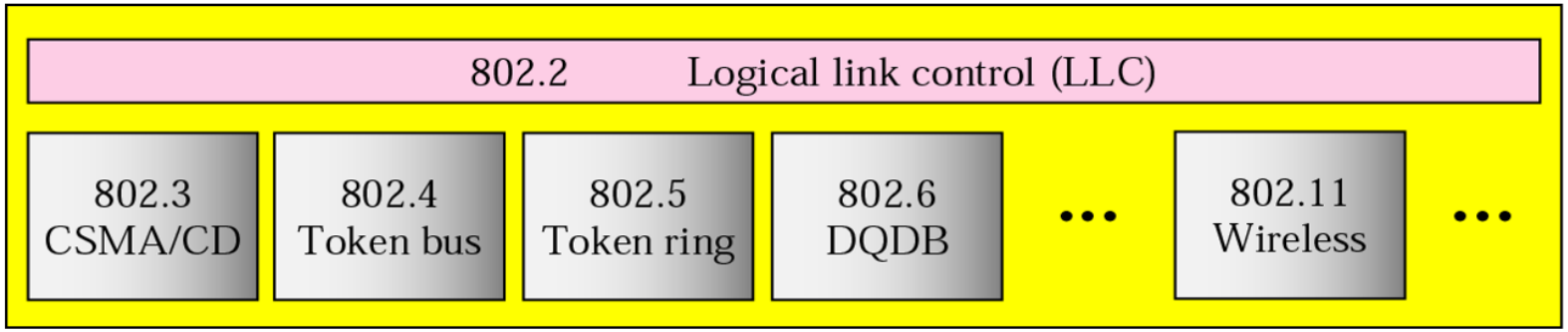



- sending side:
  - encapsulates datagram in frame
  - adds error checking bits, rdt, flow control, etc.
- receiving side
  - looks for errors, rdt, flow control, etc
  - extracts datagram, passes to upper layer at receiving side

# Link Layer and LANs

# (IEEE Standards for LANs)

## IEEE standards for LANs




Project 802

Link Layer and LANs

# Link layer, LANs: outline

5.1 introduction, services

5.2 error detection, correction

5.3 multiple access protocols

5.4 LANs

- addressing, ARP
- Ethernet
- switches
- VLANS

5.5 link virtualization: MPLS

5.6 data center networking

5.7 a day in the life of a web request

***Error Detection and Correction***

Link Layer and LANs

(Error Detection and Correction)

# Types of Error

**Single-Bit Error**

**Burst Error**

**Note:**

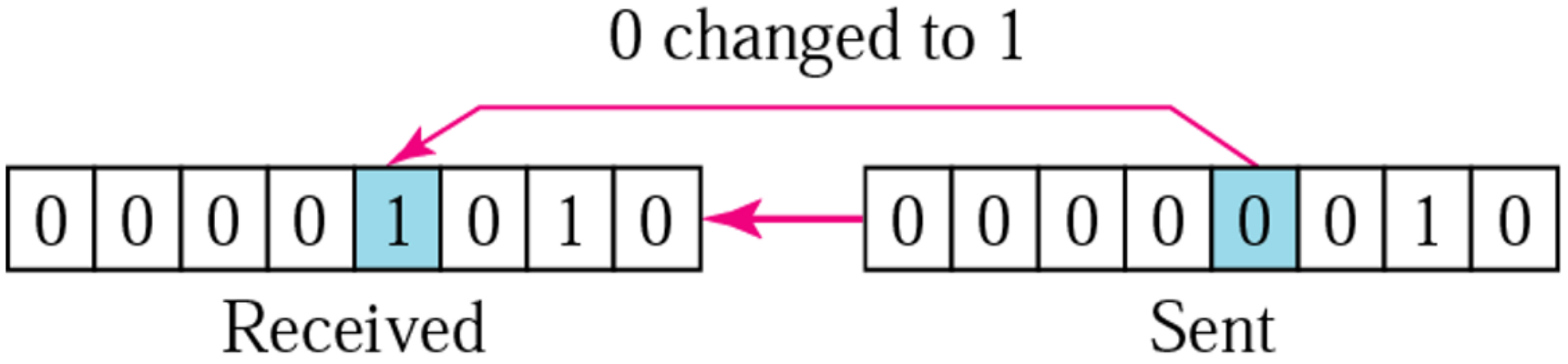
***Data can be corrupted during transmission. For reliable communication, errors must be detected and corrected.***

# Single Bit Error

Link Layer and LANs

(Error Detection and Correction)


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

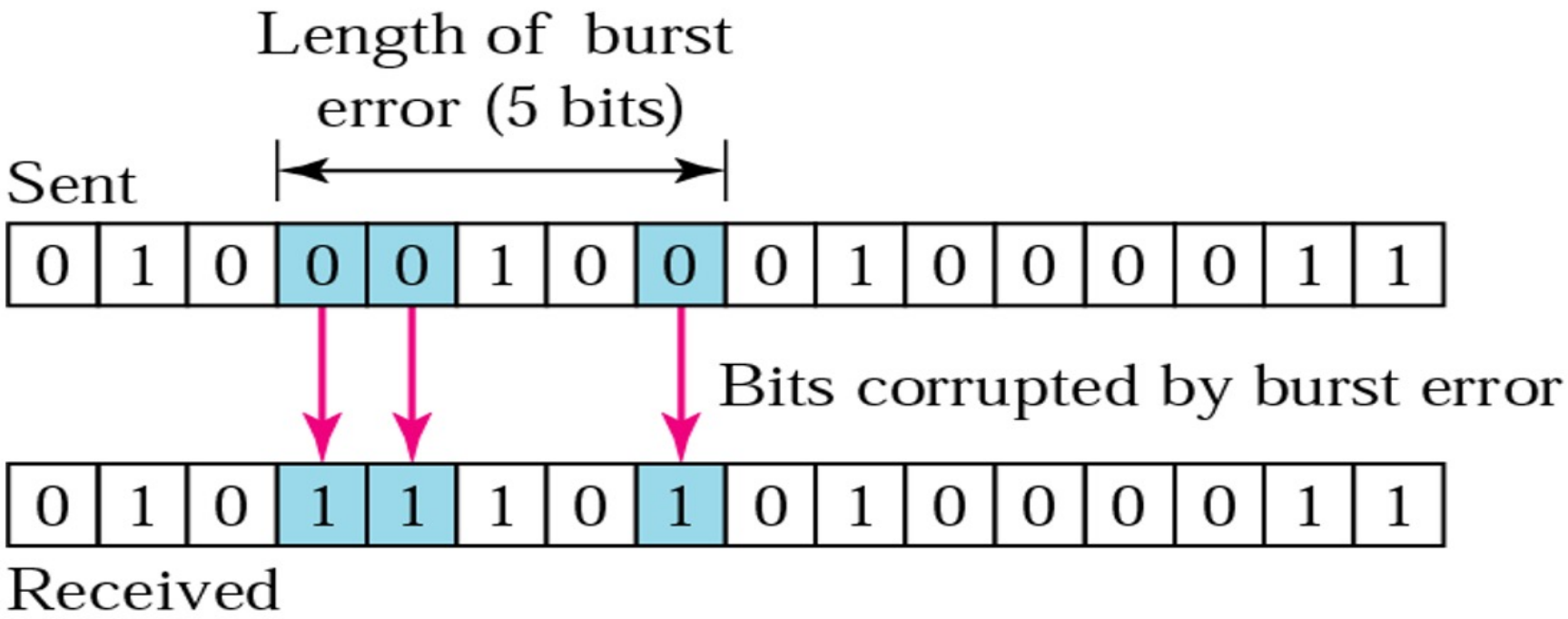


# Link Layer and LANs

## (Error Detection and Correction)

**Note:**

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



Link Layer and LANs

(Error Detection and Correction)

# Detection

**Redundancy**

**Parity Check**

**Cyclic Redundancy Check (CRC)**

**Checksum**

# Link Layer and LANs

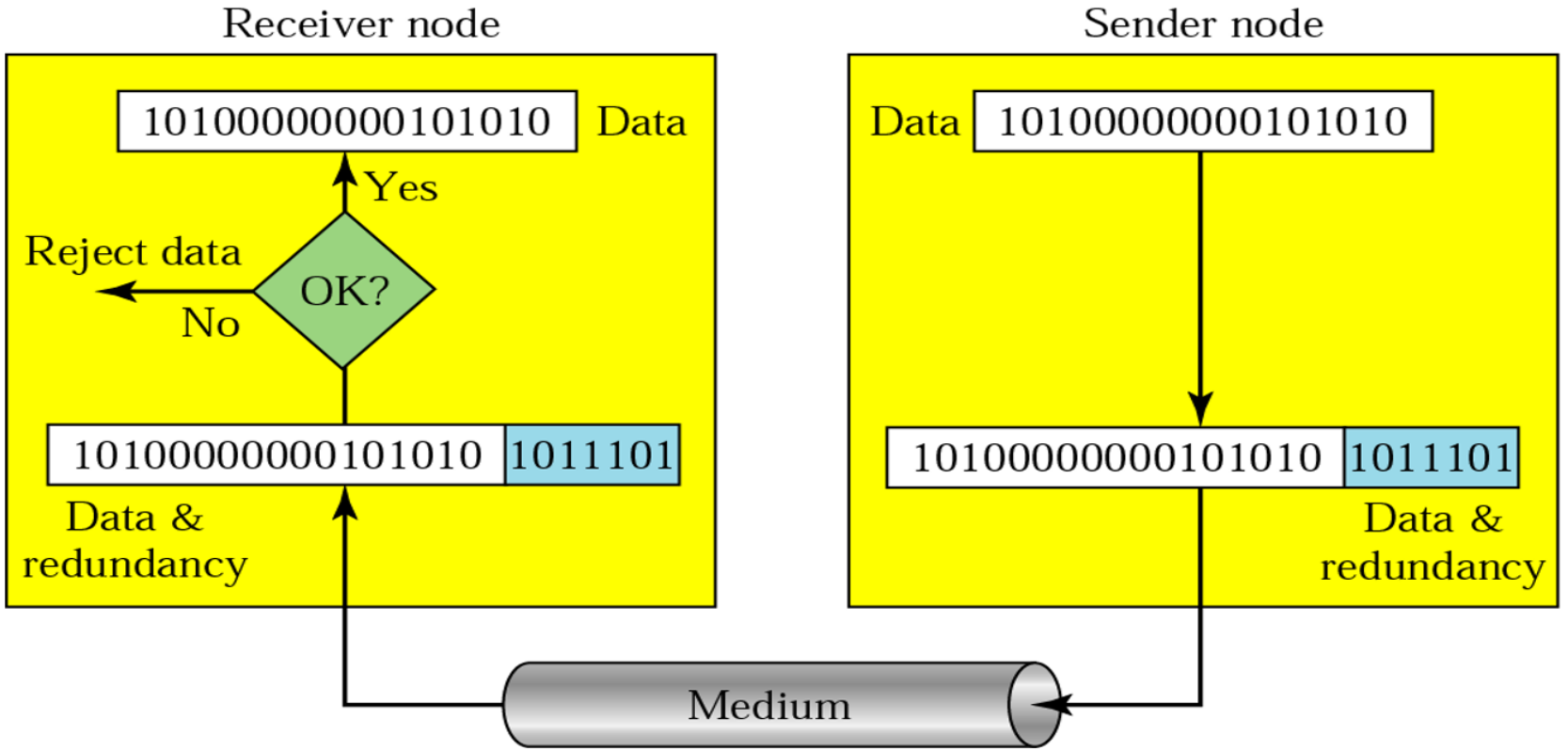
(Error Detection and Correction)

**Note:**

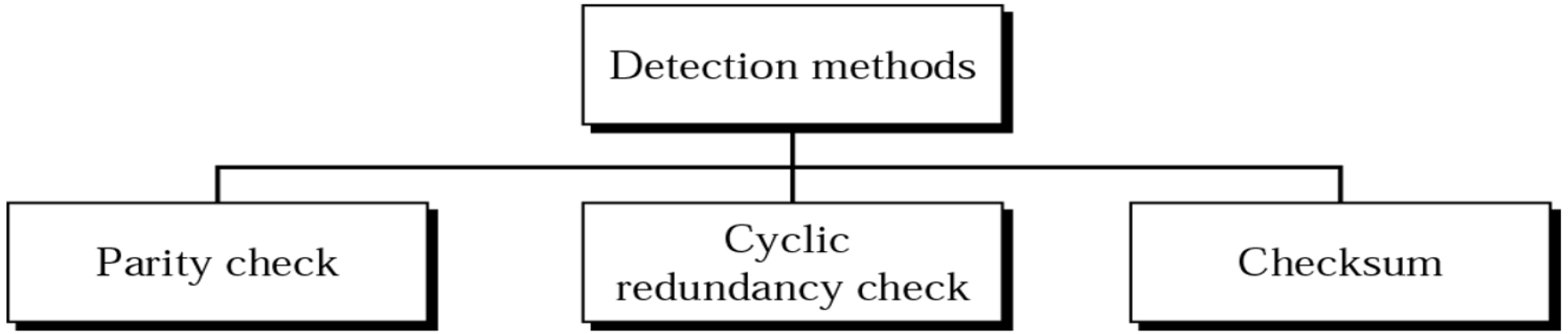
***Error detection uses the concept of redundancy, which means adding extra bits for detecting errors at the destination.***

# Link Layer and LANs

## (Error Detection and Correction)



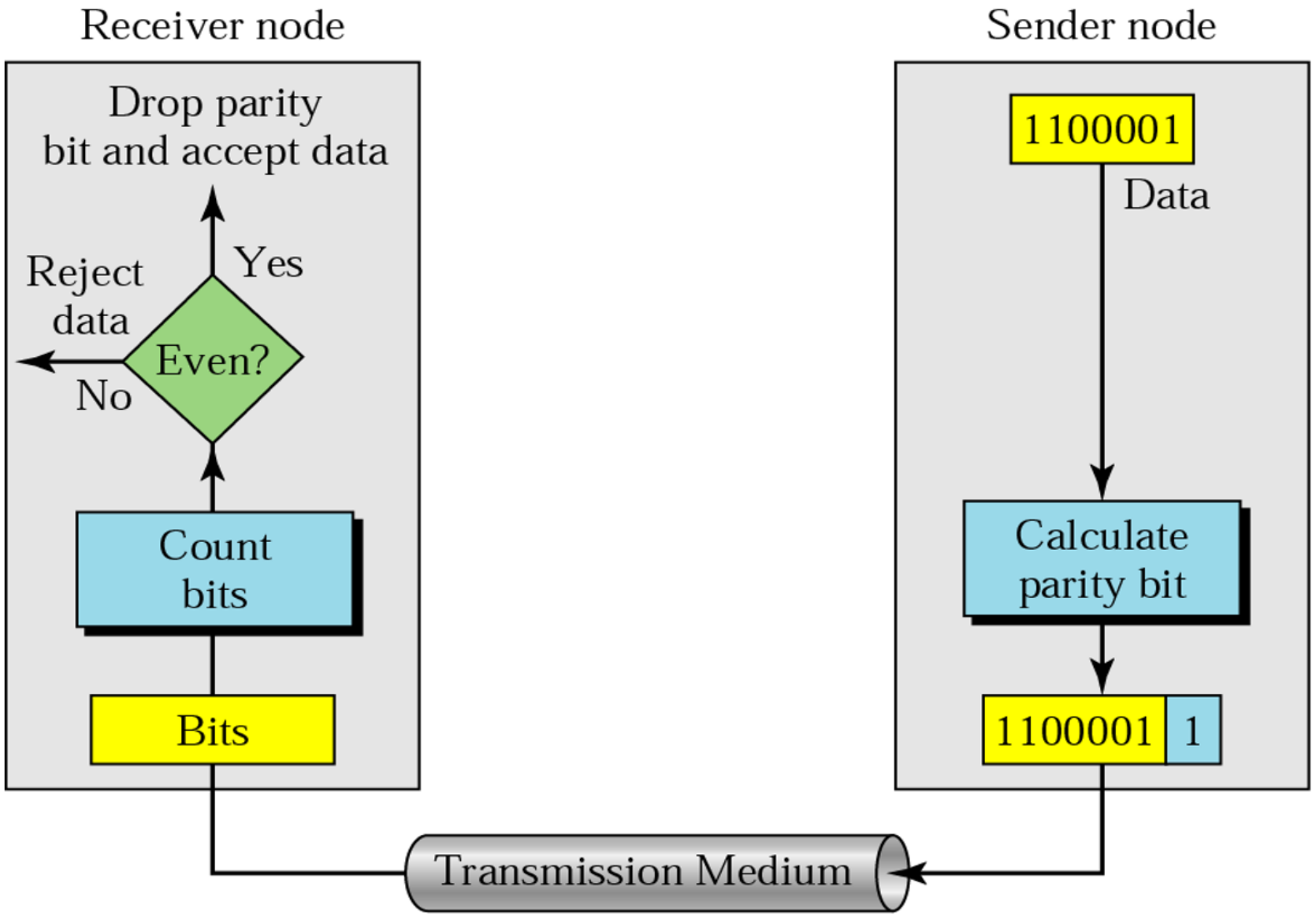

**Redundancy**



# Link Layer and LANs

## (Error Detection and Correction)

## Even Parity Concept



# Link Layer and LANs

## (Error Detection and Correction)

**Note:**

***In parity check, a parity bit is added to every data unit so that the total number of 1s is even (or odd for odd-parity).***

Link Layer and LANs

(Error Detection and Correction)

## *Example 1*

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

1110111**0** 1101111**0** 1110010**0** 1101100**0** 1100100**1**

## *Example 2*

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

11101110 11011110 11100100 11011000 11001001

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

Link Layer and LANs

(Error Detection and Correction)

## *Example 3*

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

11111110 11011110 11101100 11011000 11001001

The receiver counts the 1s in each character and comes up with even and odd numbers (7, 6, 5, 4, 4). The receiver knows that the data are corrupted, discards them, and asks for retransmission.

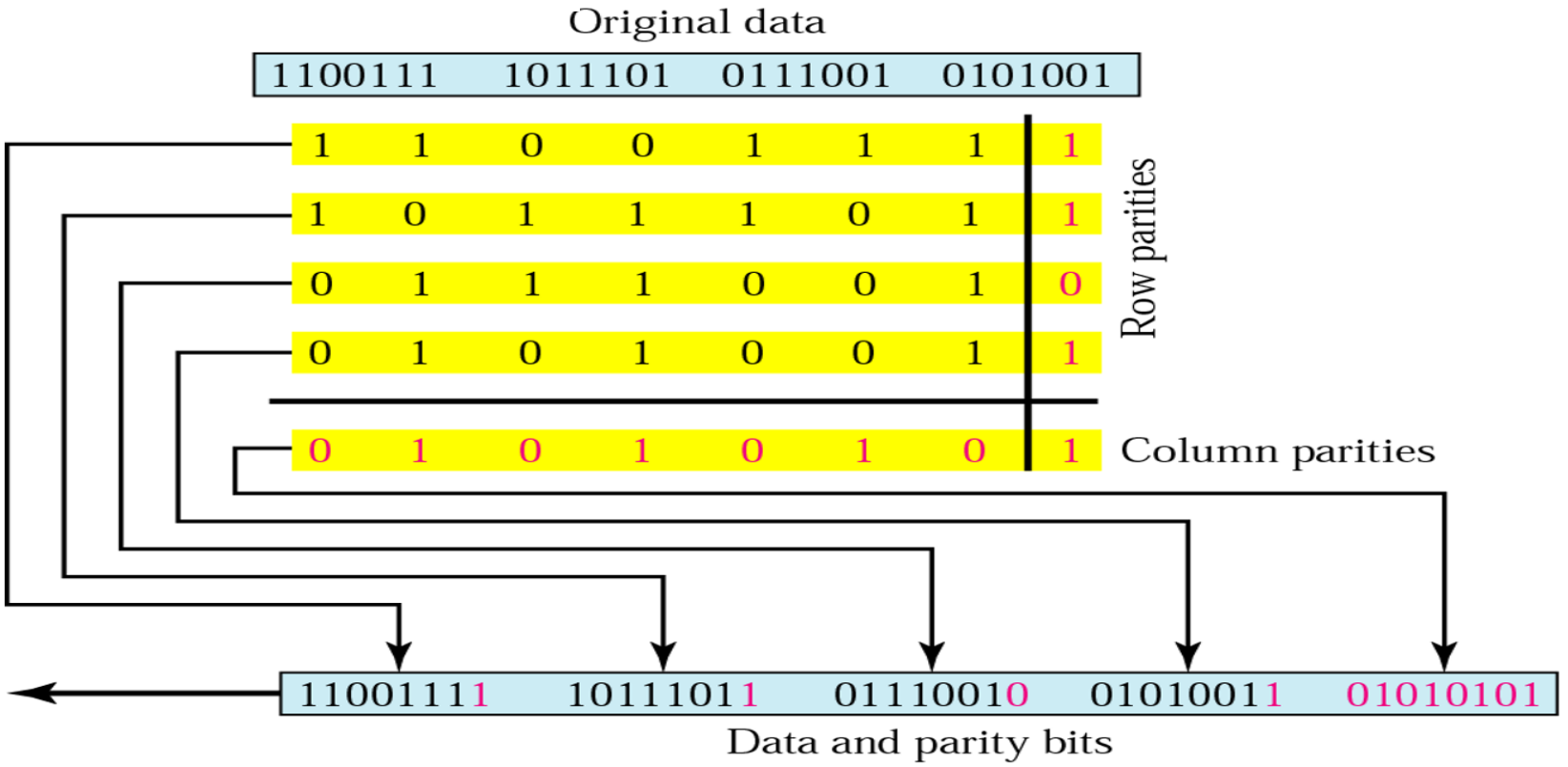
**Note:**

***Simple parity check can detect all single-bit errors. It can detect burst errors only if the total number of errors in each data unit is odd.***

# Link Layer and LANs

## (Error Detection and Correction)

***Two-dimensional parity***

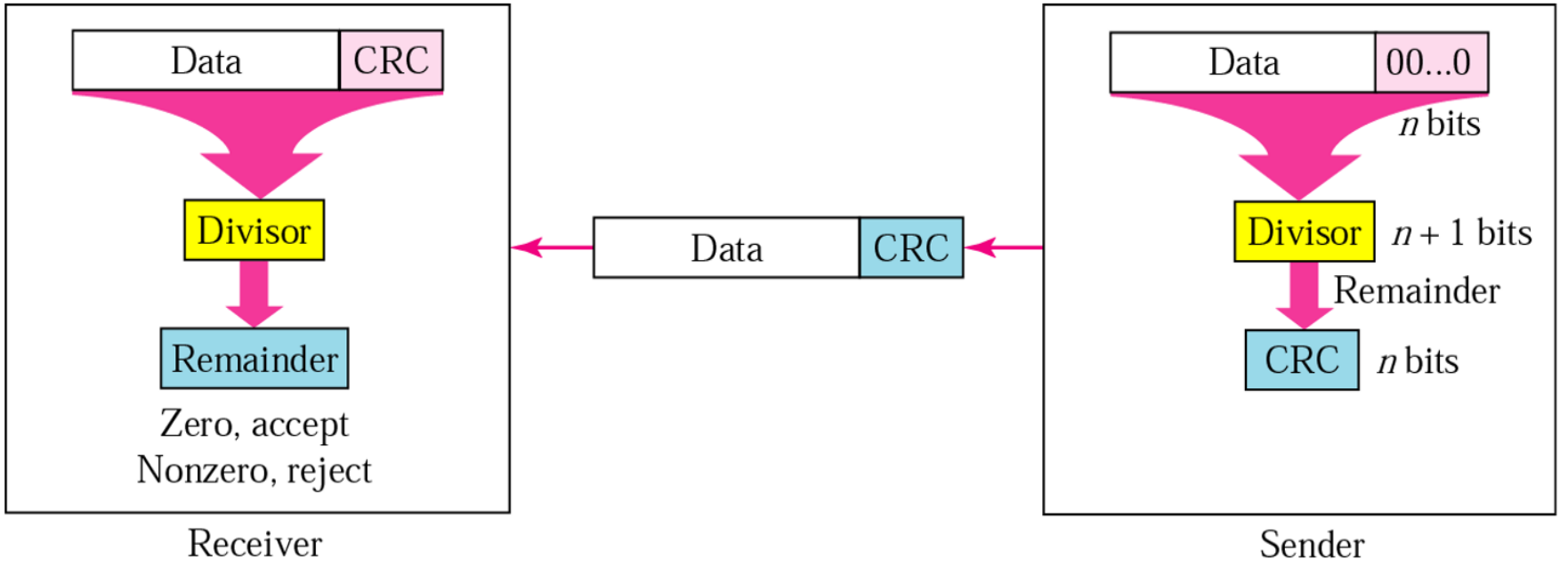


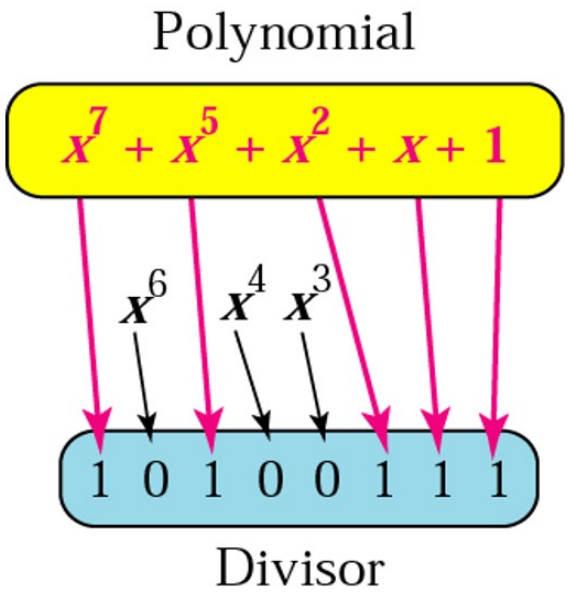

***In two-dimensional parity check, a block of bits is divided into rows and a redundant row of bits is added to the whole block.***

# Link Layer and LANs

## (Error Detection and Correction)

### CRC generator and checker



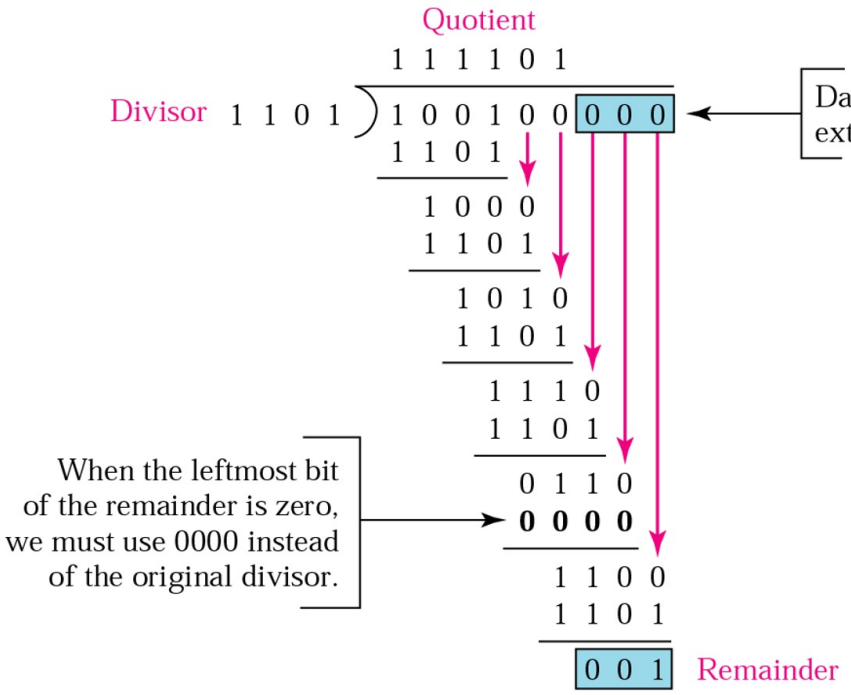

### Table 10.1 Standard polynomials

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

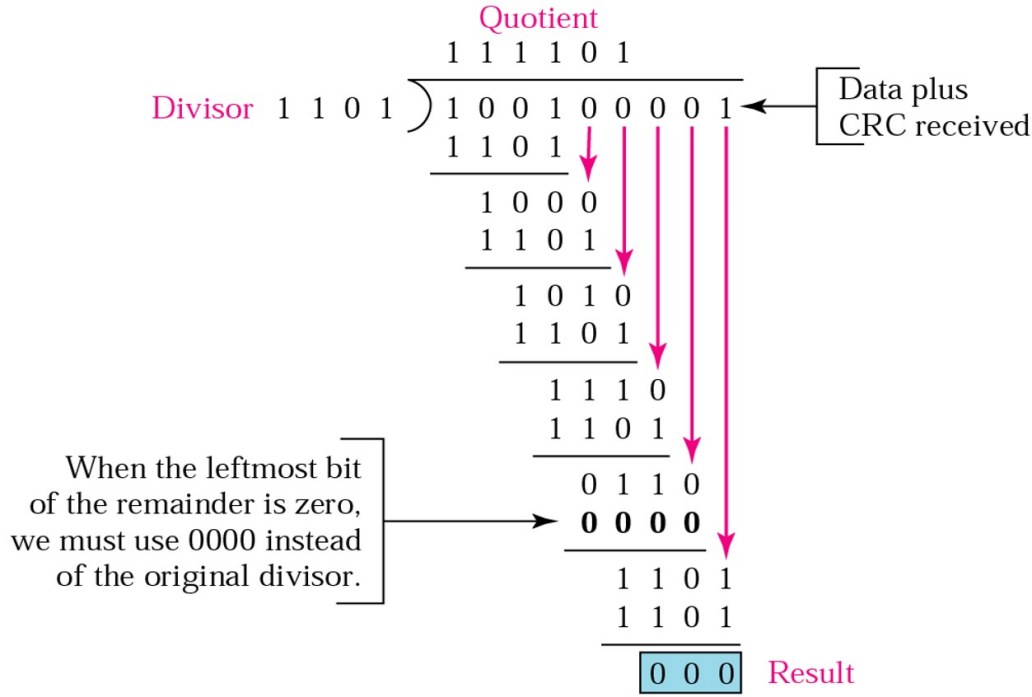
# Link Layer and LANs

# (Error Detection and Correction)

## CRC Generator



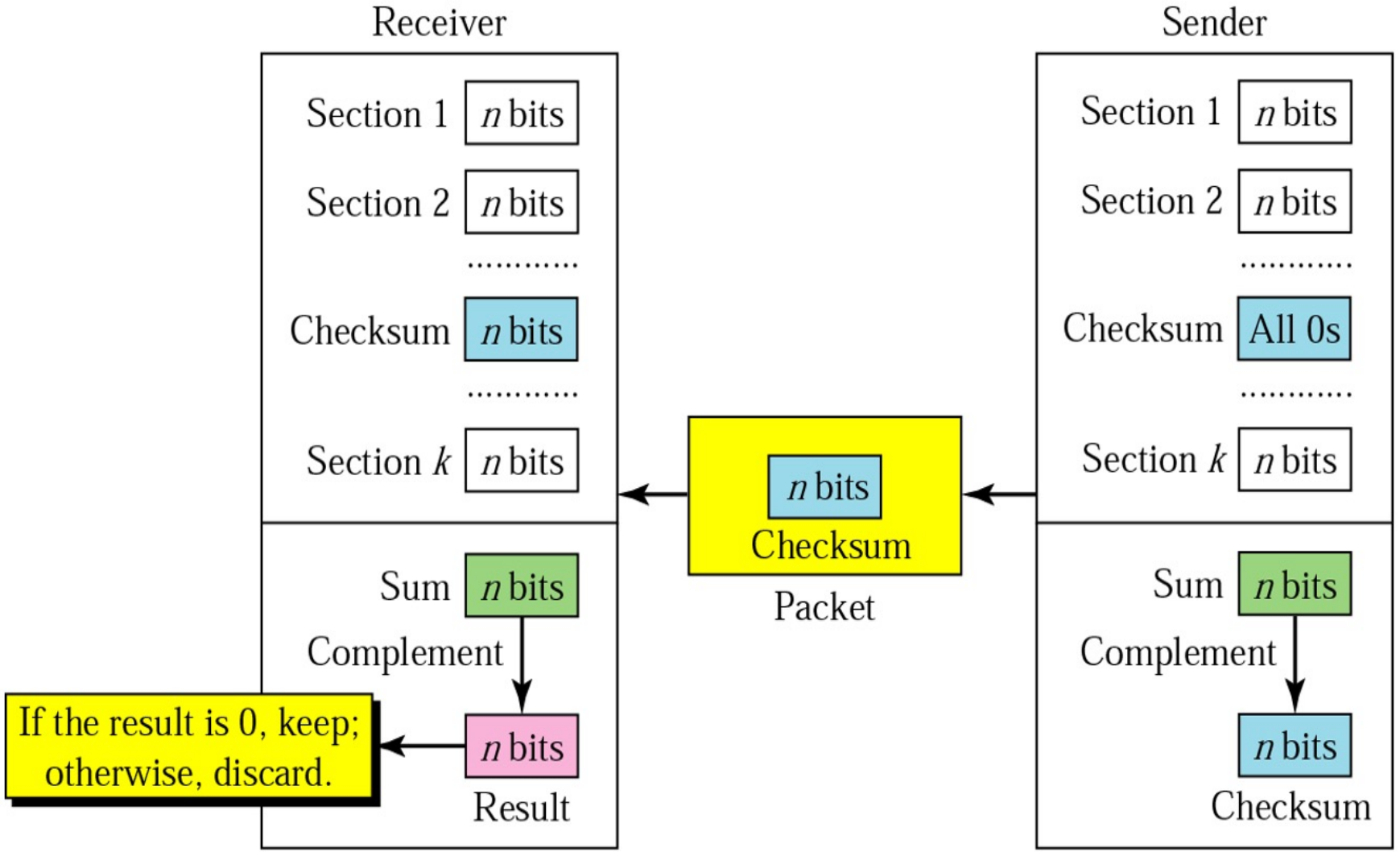

## CRC Checker



# Link Layer and LANs

## (Error Detection and Correction)

## *Checksum*



# Link Layer and LANs

## (Error Detection and Correction)

**Note:**

**The sender follows these steps:**

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

**Note:**

**The receiver follows these steps:**

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

# Checksum Generator

Link Layer and LANs

(Error Detection and Correction)

## *Example 7*

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**

# Checksum Checker

Link Layer and LANs

(Error Detection and Correction)

## Example 8

Now suppose the receiver receives the pattern sent in Example 7 and 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. |

Link Layer and LANs

(Error Detection and Correction)

# Correction

**Retransmission**

**Forward Error Correction**

**Burst Error Correction**

Link Layer and LANs

(Error Detection and Correction)

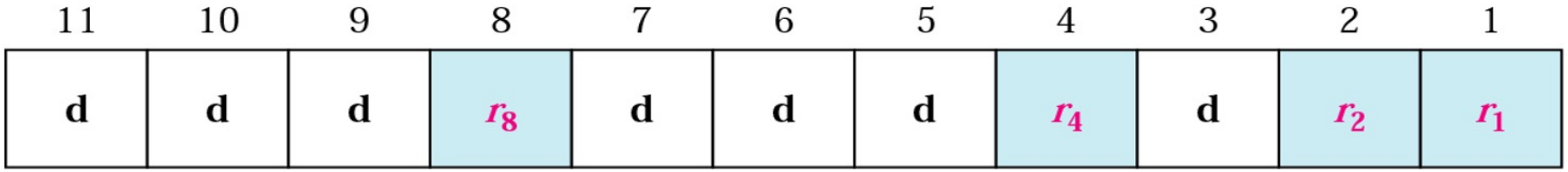
## *Table 10.2 Data and redundancy bits*

| Number of data bits m | Number of redundancy bits r | Total bits m + r |
|---|---|---|
| 1 | 2 | 3 |
| 2 | 3 | 5 |
| 3 | 3 | 6 |
| 4 | 3 | 7 |
| 5 | 4 | 9 |
| 6 | 4 | 10 |
| 7 | 4 | 11 |

# Link Layer and LANs

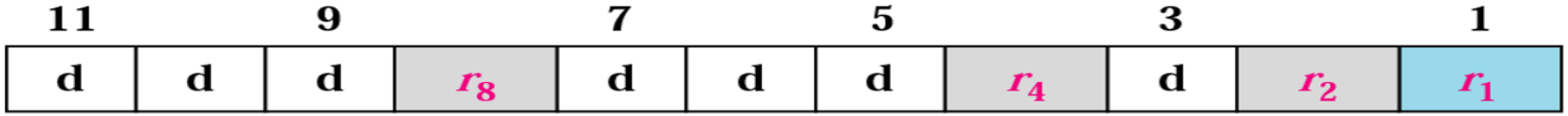
## (Error Detection and Correction)

## r = Redundancy Bits, d = Data Bits

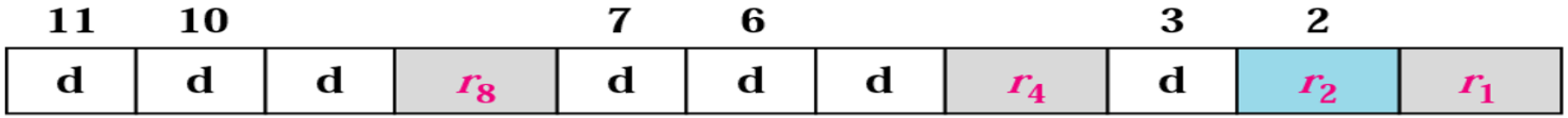



## Redundancy Bit Calculation

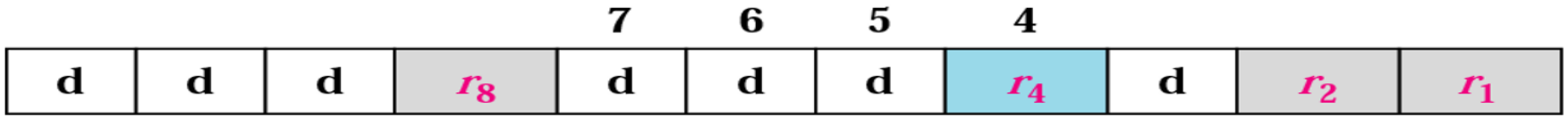
$r_1$ will take care of these bits.



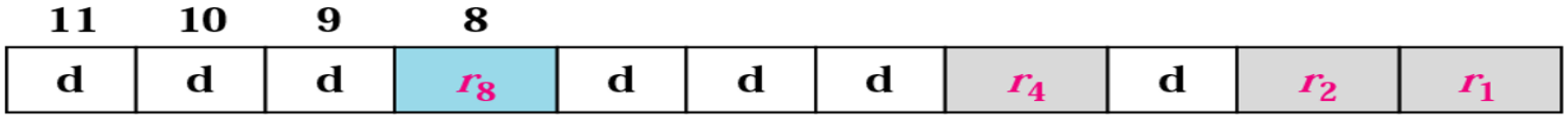

$r_2$ will take care of these bits.




$r_4$ will take care of these bits.



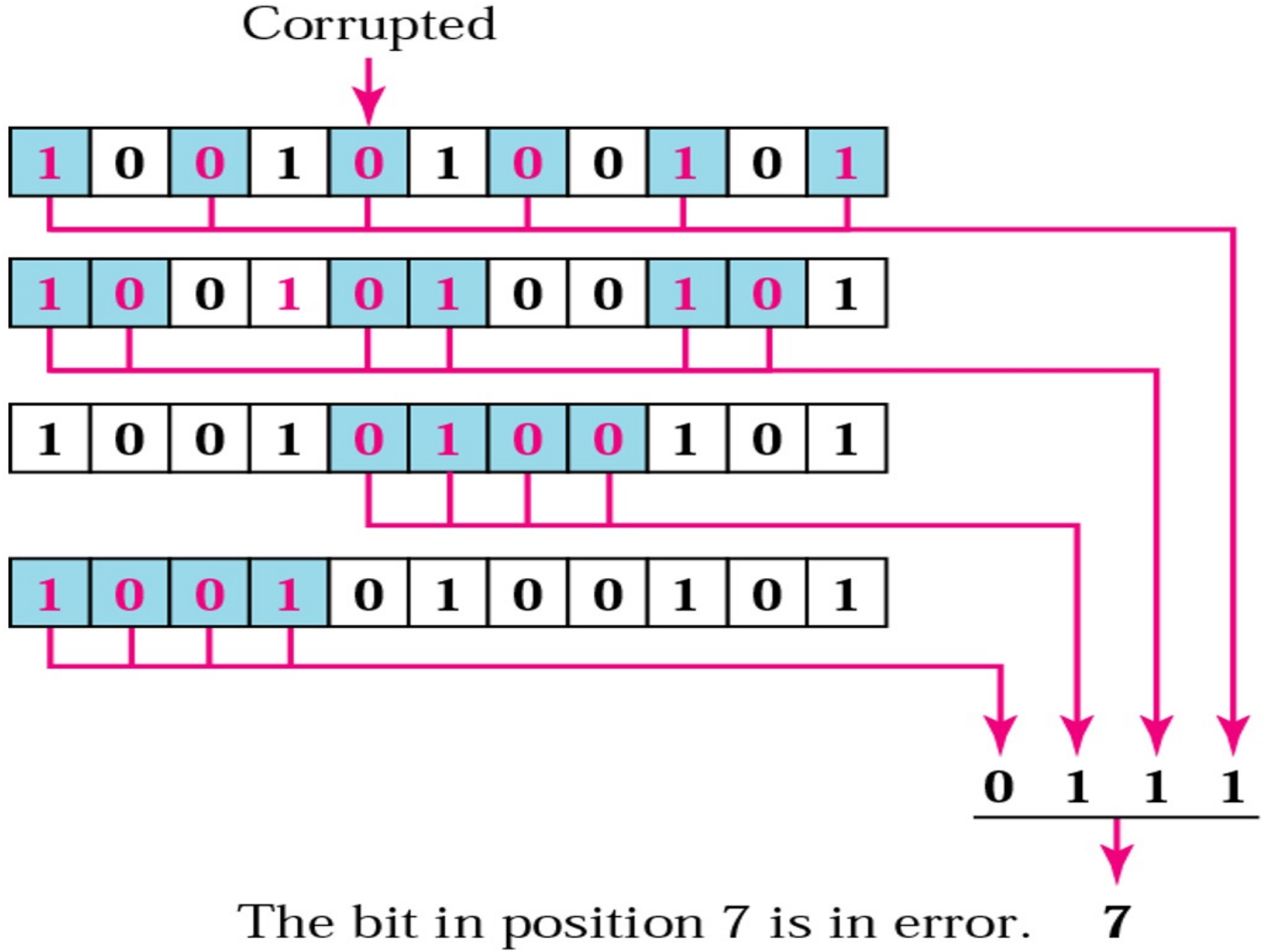

$r_8$ will take care of these bits.



# Link Layer and LANs

## (Error Detection and Correction)

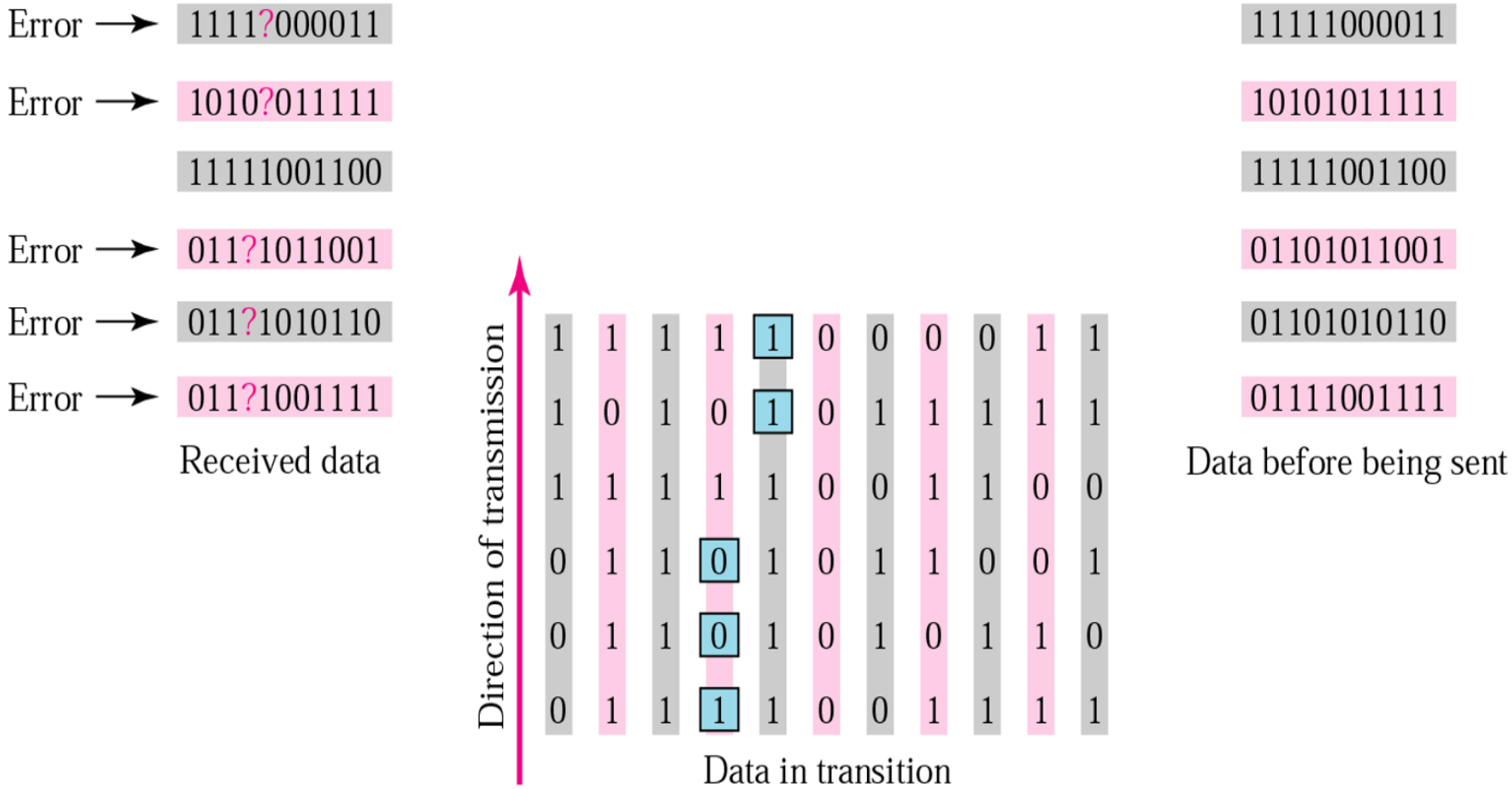
***Error detection using Hamming code***



# Link Layer and LANs

## (Error Detection and Correction)

### *Burst error correction example*

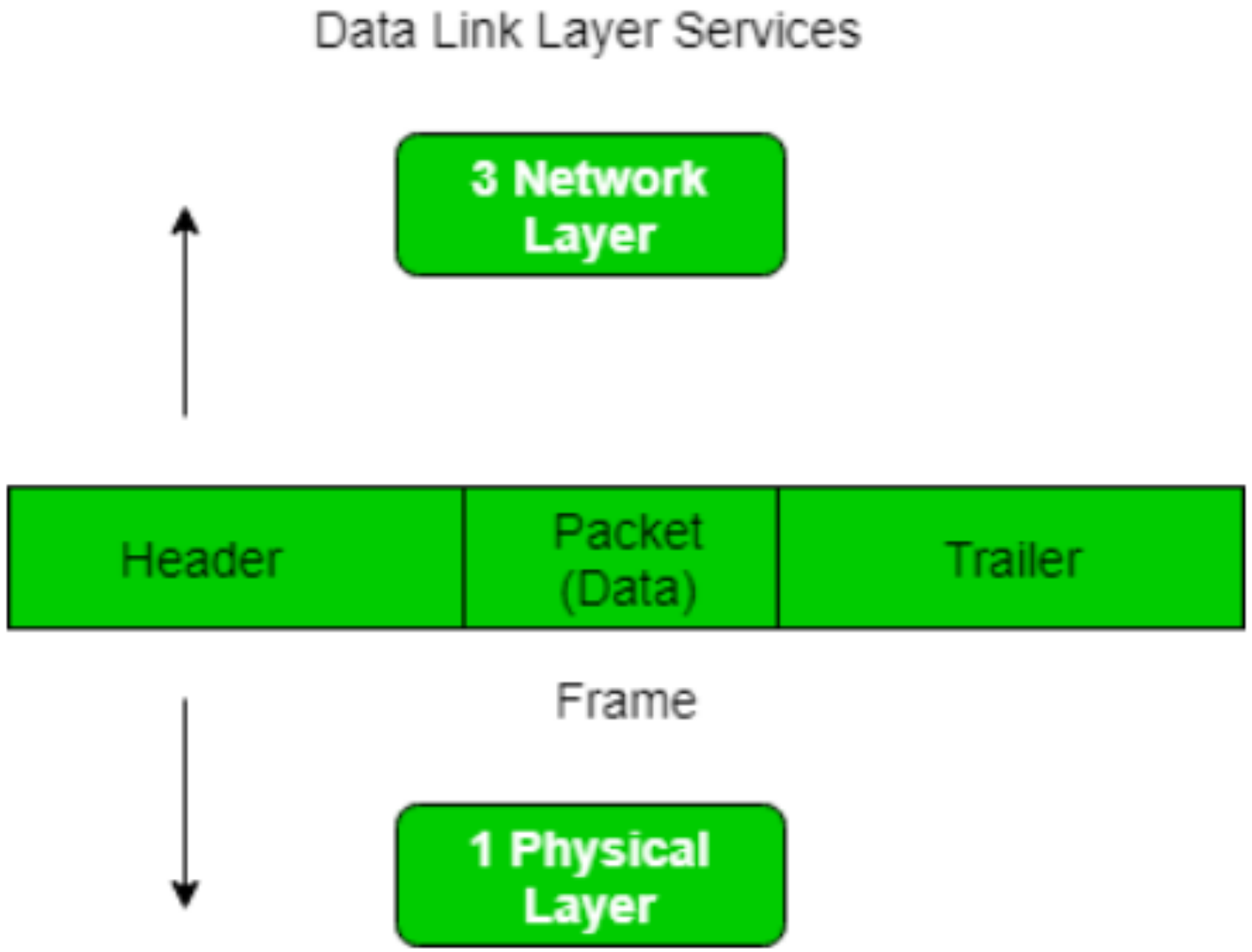


# Framing

Link Layer and LANs
Framing

- Frames are the units of digital transmission, particularly in computer networks and telecommunications.
- Frames are comparable to the packets of energy called photons in the case of light energy. Frame is continuously used in the Time Division Multiplexing process.
- Framing is a point-to-point connection between two computers or devices consisting of a wire in which data is transmitted as a stream of bits.

# Link Layer and LANs
## Framing



Link Layer and LANs

Framing

**Types of Framing**

## Types of Framing

1. Character Count.

2. Starting & Ending Bytes (FLAG) With Byte Stuffing.

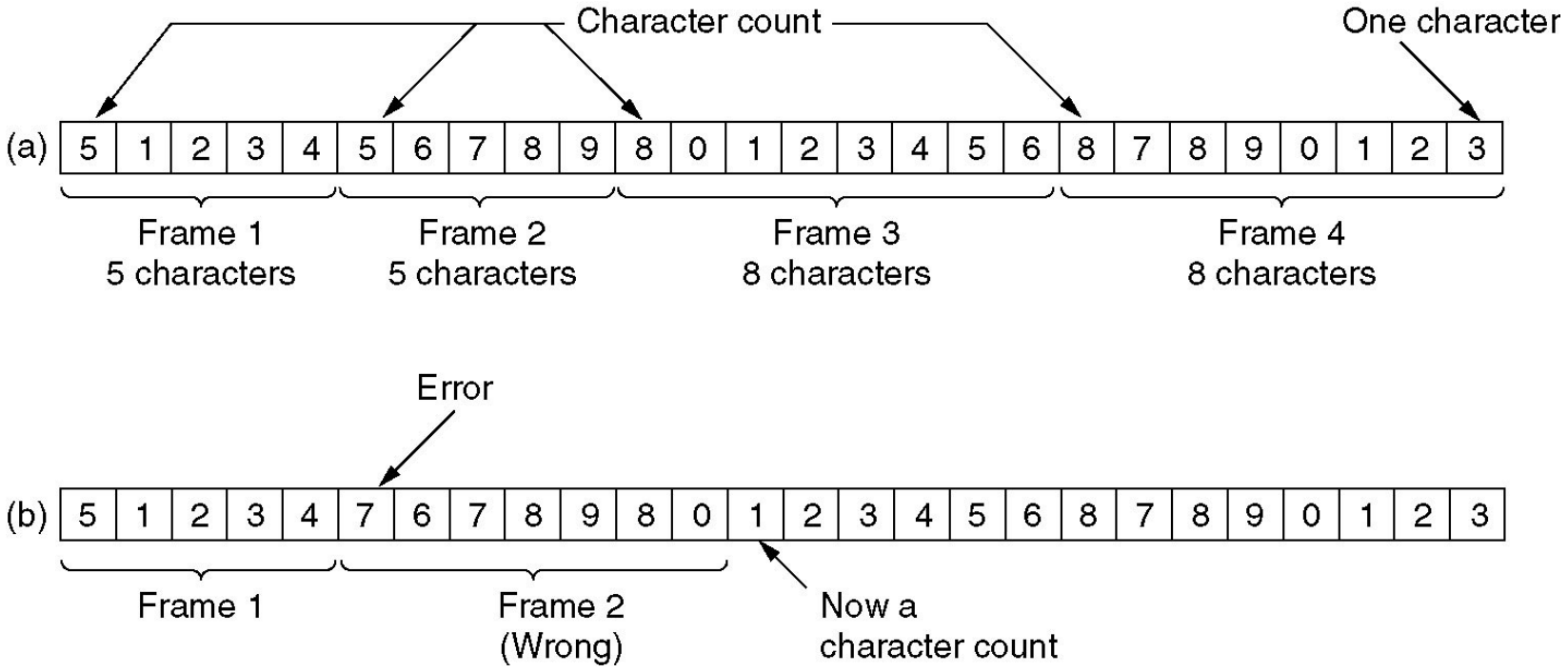
3. Starting & Ending Bit Pattern (Flag) With Bit Stuffing.

# Link Layer and LANs

## Framing

## Character Count - Framing

- This method specifies the number of characters that are present in a particular frame.
- This information is specified by using a special field in the header frame.

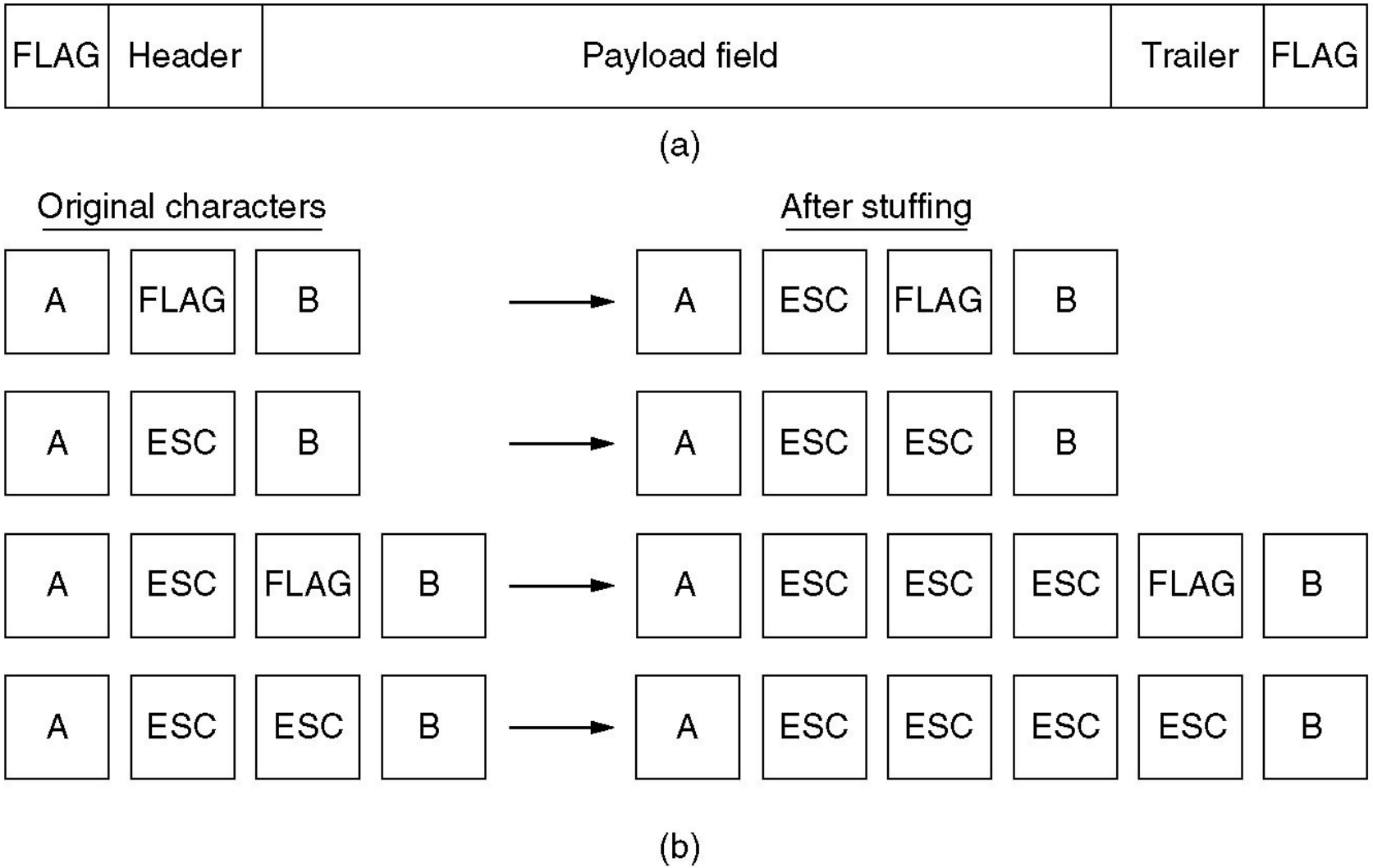



(a) Without errors. (b) With one error.

# Link Layer and LANs
## Framing
## Character Count - Framing Disadvantage

- The trouble with this algorithm is that a transmission error can garble the count.
- For example, if the character count of 5 in the second frame becomes 7, the destination will get out of synchronization and cannot locate the start of the next frame.
- Even if the checksum is incorrect so the destination knows that the frame is bad, it still has no way of telling where the next frame starts.
- Sending a frame back to the source asking for a retransmission does not help either since the destination does not know how many characters to skip over to get to the start of the retransmission.
- For this reason, the character count method is rarely used anymore.

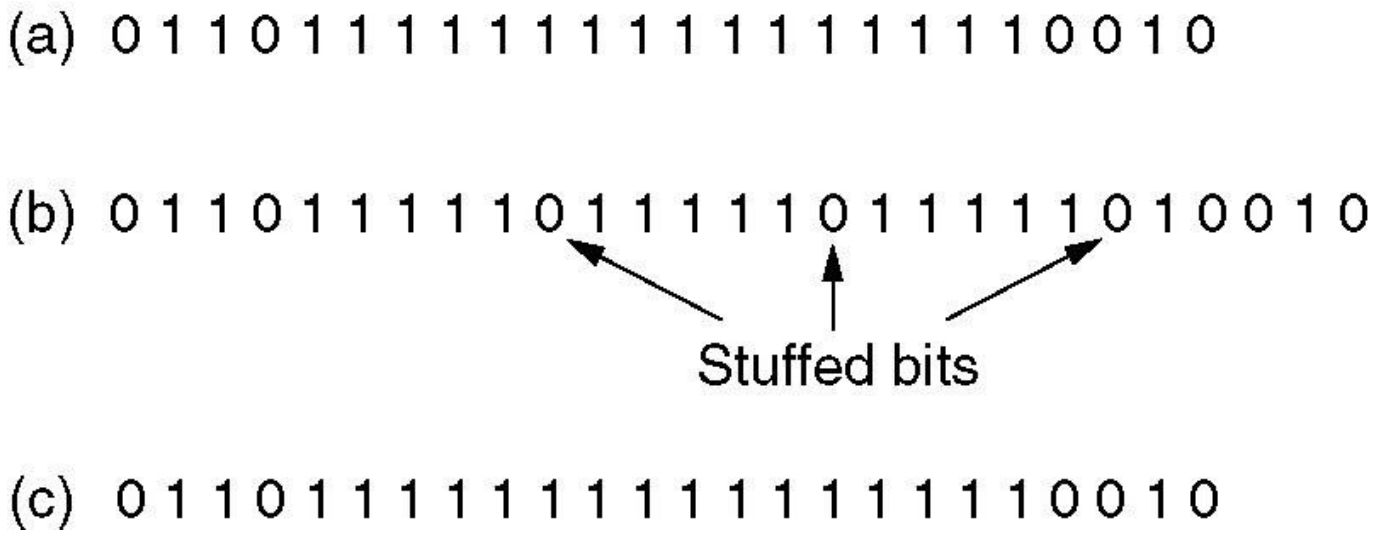
# Link Layer and LANs
## Framing
## Byte Stuffing – Framing



# Link Layer and LANs
## Framing
## Byte Stuffing – Framing

- In this method, the start and end of the frame are recognized with the help of flag bytes. Two consecutive flag bytes indicate the end of one frame and the start of the next one. The flag bytes used in figure 2 used is named as “ESC” flag bytes.
- A frame delimited by flag bytes. This framing method is only applicable in 8-bit character codes, which are a major disadvantage of this method as not all character codes use 8-bit characters, e.g., Unicode.
- Four examples of byte sequences before and after stuffing are shown in the previous diagram.

# Link Layer and LANs
## Framing
## Bit Stuffing



# Link Layer and LANs
## Framing
## Bit Stuffing

- Allows frame to contain an arbitrary number of bits and arbitrary character size. The frames are separated by separating flags.
- Each frame begins and ends with a special bit pattern, 01111110, called a flag byte. When five consecutive l's are encountered in the data, it automatically stuffs a '0' bit into the outgoing bit stream.
- In this method, frames contain an arbitrary number of bits and allow character codes with an arbitrary number of bits per character. In his case, each frame starts and ends with a special bit pattern, 01111110.
- In the data, a 0 bit is automatically stuffed into the outgoing bit stream whenever the sender's data link layer finds five consecutive 1s.

# Link Layer and LANs
## Framing
## Bit Stuffing

- This bit stuffing is similar to byte stuffing, in which an escape byte is stuffed into the outgoing character stream before a flag byte in the data.
- When the receiver sees five consecutive incoming i bits, followed by an o bit, it automatically destuffs (i.e., deletes) the 0 bit. Bit Stuffing is completely transparent to the network layer as byte stuffing.
- This framing method finds its application in networks where the change of data into code on the physical medium contains some repeated or duplicate data. For example, some LANs encode bit of data by using 2 physical bits.

Link Layer and LANs

Framing

Bit Stuffing

Solve this

- A bit string, 01111011111101111110, needs to be transmitted at the data link layer. What is the string actually transmitted after bit stuffing?

The output is "011110111110011111010"

# Link Layer and LANs

## Framing

## Problems in Framing

- **Detecting start of the frame:** When a frame is transmitted, every station must be able to detect it. Station detect frames by looking out for special sequence of bits that marks the beginning of the frame i.e. SFD (Starting Frame Delimiter).
- **How do station detect a frame:** Every station listen to link for SFD pattern through a sequential circuit. If SFD is detected, sequential circuit alerts station. Station checks destination address to accept or reject frame.
- **Detecting end of frame:** When to stop reading the frame.

Link Layer and LANs
Multiple Access Protocols

# Link layer, LANs: outline

5.1 introduction, services

5.2 error detection, correction

5.3 multiple access protocols

5.4 LANs

- addressing, ARP
- Ethernet
- switches
- VLANS

5.5 link virtualization: MPLS

5.6 data center networking

5.7 a day in the life of a web request

# Multiple Access Protocols

Link Layer and LANs
Multiple Access Protocols

The Data Link Layer is responsible for transmitting data between two nodes.

Its main functions are-

- Data Link Control
- Multiple Access Control


**Data Link Control**

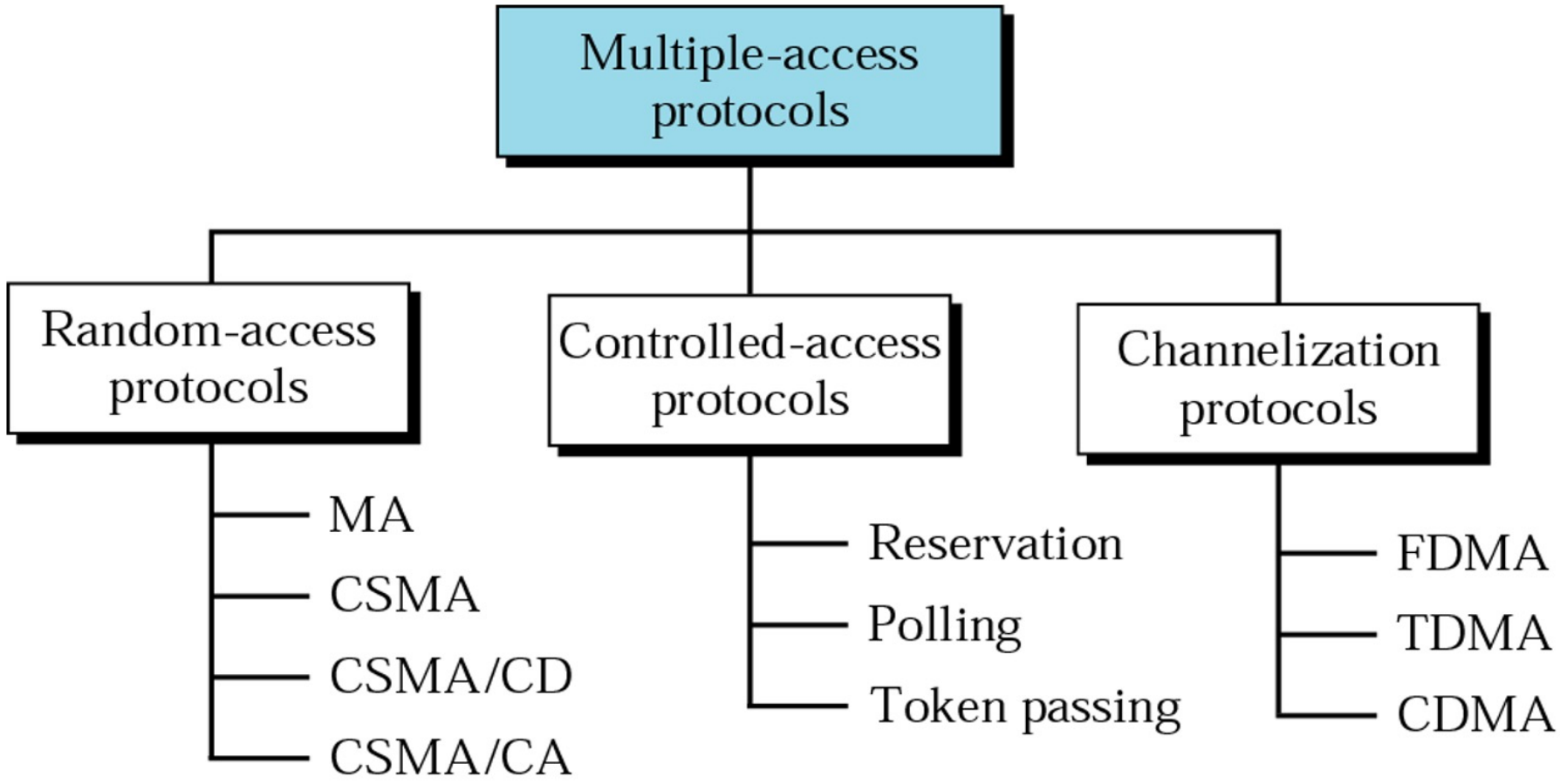
- The data link control is responsible for the reliable transmission of messages over transmission channels by using techniques like framing, error control, and flow control.

**Multiple Access Control**

- If no dedicated link is present, then multiple stations can access the channel simultaneously. Hence, multiple access protocols are required to decrease collision and avoid crosstalk.

- Thus, protocols are required for sharing data on non-dedicated channels.

# Link Layer and LANs
# Multiple Access Protocols



# Link Layer and LANs
# Multiple Access Protocols

## I) Random Access Protocol

- In this, all stations have the same superiority that is, no station has more priority than another station. Any station can send data depending on the medium's state( idle or busy).

- It has two features:

1. There is no fixed time for sending data
2. There is no fixed sequence of stations sending data.

## A) ALOHA (MA)

- It was designed for wireless LAN but is also applicable for shared medium.
- In this, multiple stations can transmit data at the same time and can hence lead to collision and data being garbled.

### 1) Pure ALOHA

- When a station sends data, it waits for an acknowledgment. If the acknowledgment doesn't come within the allotted time, then the station waits for a random amount of time calls back-off time (Tb), and re-sends the data.

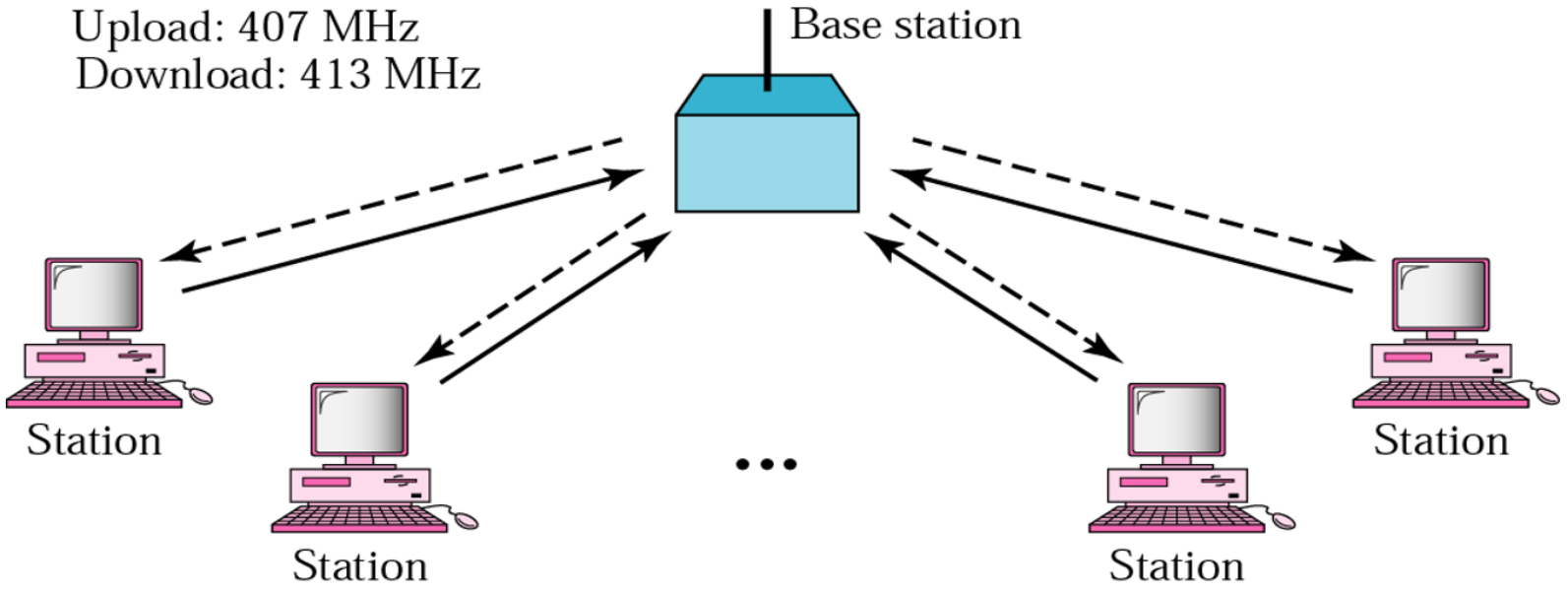
```
Vulnerable Time = 2* Frame transmission time
Throughput =  G exp{-2*G}
Maximum throughput = 0.184 for G=0.5
```

# Link Layer and LANs
# Multiple Access Protocols

## 2) Slotted ALOHA

- It is similar to pure aloha, except that we divide time into slots, and data sending is allowed only at the beginning of these slots. If a station misses out on the allowed time, it must wait for the next slot. This reduces the probability of collision.

```
Vulnerable Time =  Frame transmission time
Throughput =  G exp{-*G}
Maximum throughput = 0.368 for G=1
```



# Link Layer and LANs
# Multiple Access Protocols

**B) CSMA**

- Carrier Sense Multiple Access ensures fewer collisions as the station is required to first sense the medium (for idle or busy) before transmitting data. If it is idle, then it sends data, otherwise, it waits till the channel becomes idle.

- **1-persistent:** The node senses the channel, if idle it sends the data; otherwise, it continuously keeps on checking the medium for being idle and transmits unconditionally(with 1 probability) as soon as the channel gets idle.

- **Non-Persistent:** The node senses the channel, if idle it sends the data, otherwise it checks the medium after a random amount of time (not continuously) and transmits when found idle.

- **P-persistent:** The node senses the medium, if idle it sends the data with p probability. If the data is not transmitted ((1-p) probability) then it waits for some time and checks the medium again, now if it is found idle then it sends with p probability. This repeat continues until the frame is sent. It is used in Wifi and packet radio systems.

# Link Layer and LANs
# Multiple Access Protocols

**C) CSMA/ CD**

- Carrier sense multiple access with collision detection. Stations can terminate transmission of data if collision is detected.

**D) CSMA/ CA**

- Carrier sense multiple access with collision avoidance. The process of collision detection involves the sender receiving acknowledgment signals.
- If there is just one signal(its own), then the data is successfully sent, but if there are two signals (its own and the one with which it has collided), then it means a collision has occurred.
- CSMA/CA avoids collision by:
    - **Interframe space**
    - **Contention Window**
    - **Acknowledgment**

# Link Layer and LANs
# Multiple Access Protocols

## II) Controlled Access:

- In controlled access, the stations seek information from one another to find which station has the right to send. It allows only one node to send at a time to avoid the collision of messages on a shared medium.

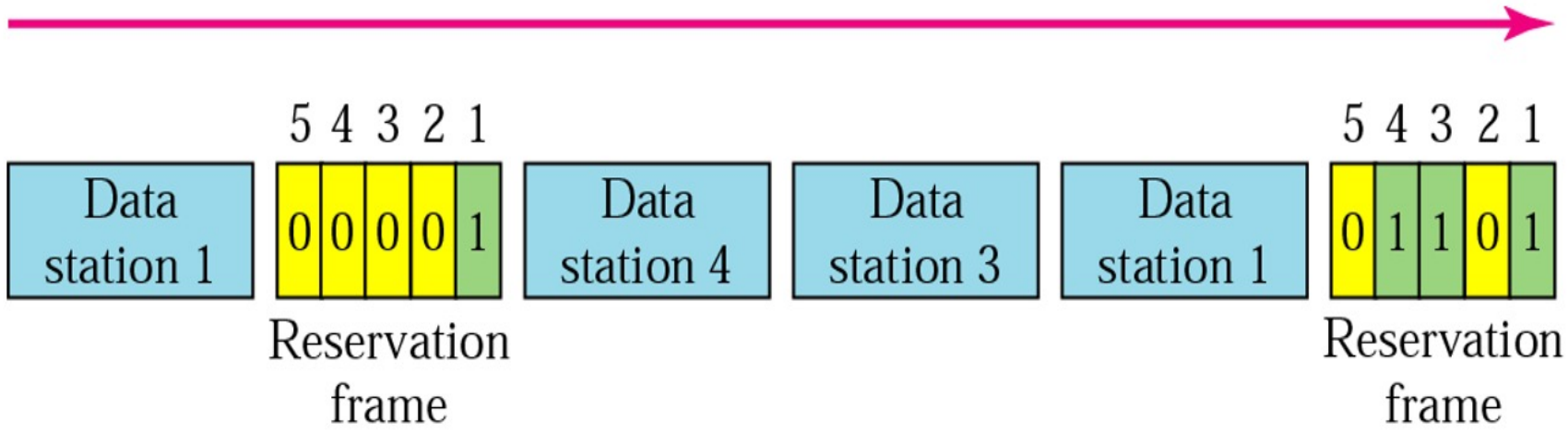
- The three controlled-access methods are:
  1. Reservation
  2. Polling.----------POLL/ SEL
  3. Token Passing

## II) Channelization:

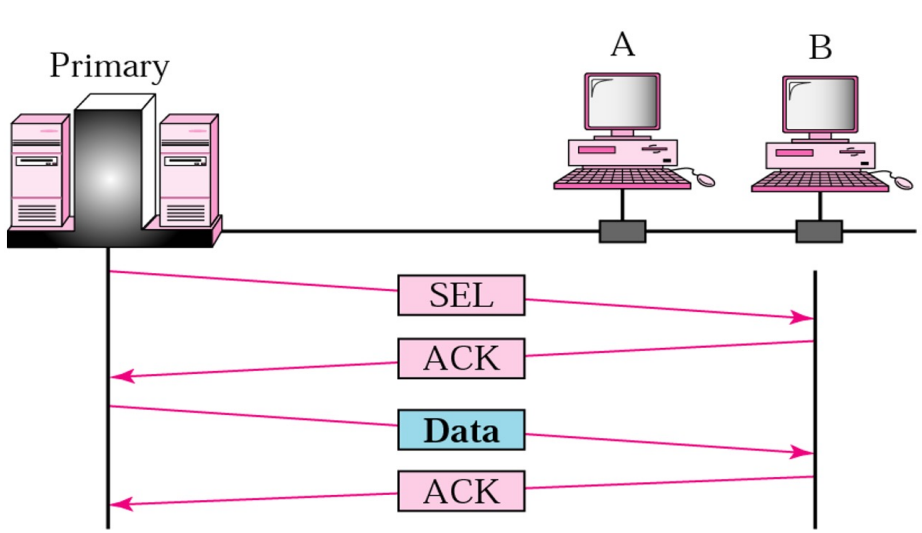
- In this, the available bandwidth of the link is shared in time, frequency, and code to multiple stations to access the channel simultaneously.
- The three controlled-access methods are:
  1. Frequency Division Multiple Access (FDMA)
  2. Time Division Multiple Access (TDMA)
  3. Code Division Multiple Access (CDMA)

# Link Layer and LANs
## Multiple Access Protocols (Controlled Access Protocol)

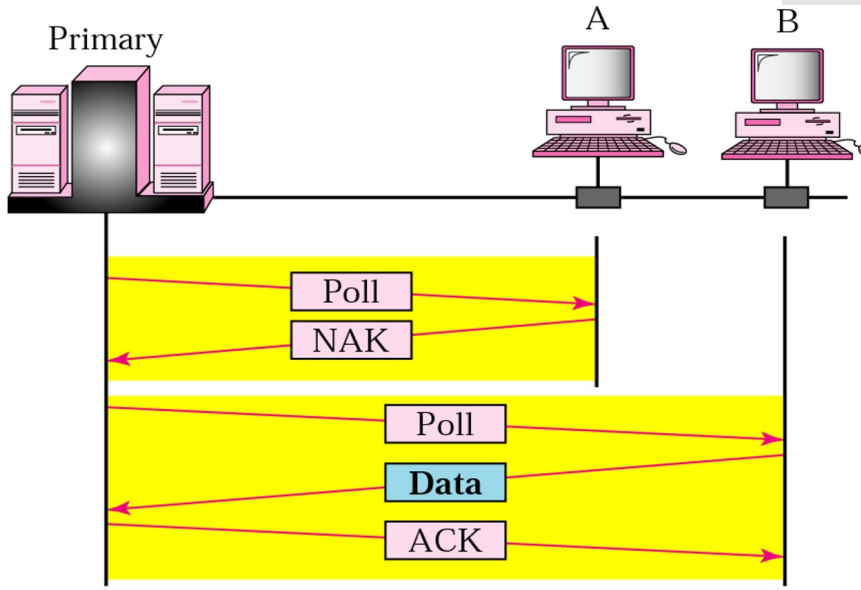
### 1. Reservation Access




### 2. Polling



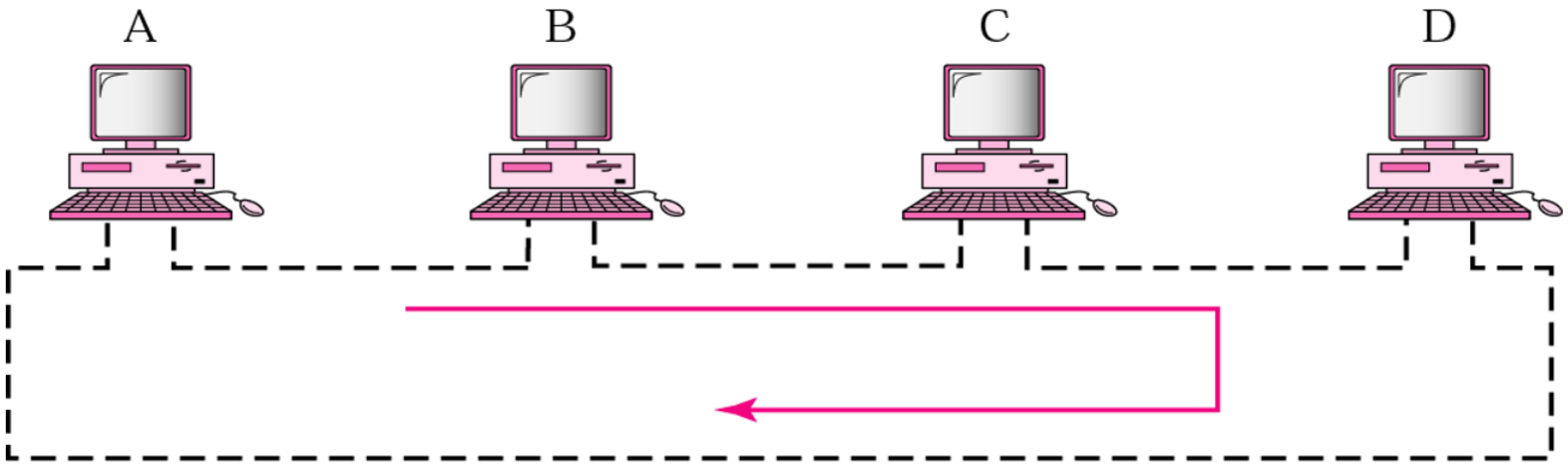

SEL

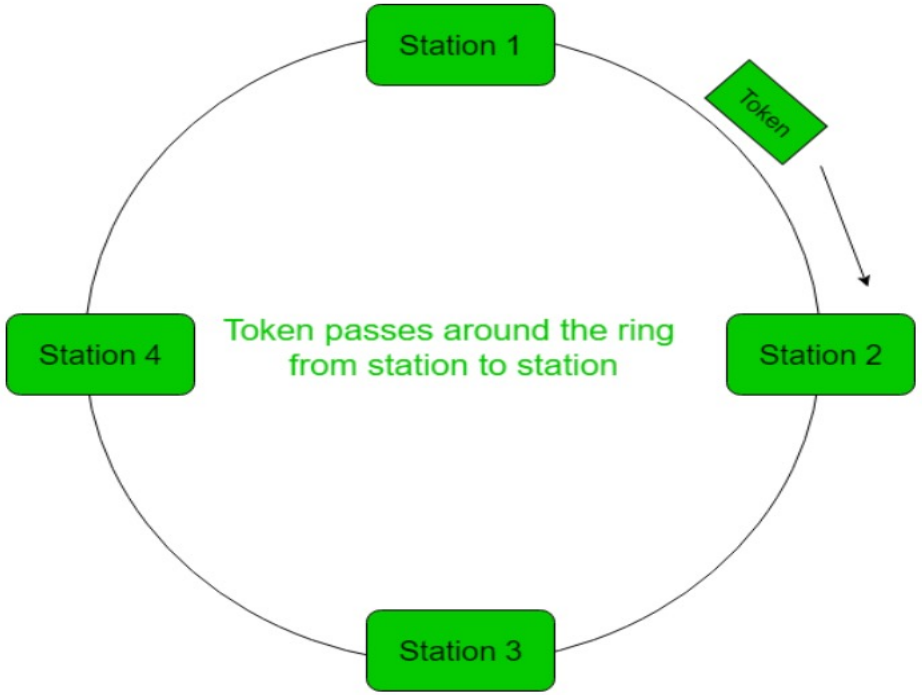



POLL

# Link Layer and LANs
## Multiple Access Protocols (Controlled Access Protocol)

### 3. Token Passing



# Link Layer and LANs

## Multiple Access Protocols (Channelized Access Protocol)

**Note:**

***In FDMA, the bandwidth is divided into channels.***

**Note:**

***In TDMA, the bandwidth is just one channel that is timeshared.***

**Note:**

***In CDMA, one channel carries all transmissions simultaneously.***

Link Layer and LANs
Multiple Access Protocols

# CSMA (carrier sense multiple access)

*CSMA:* listen before transmit:

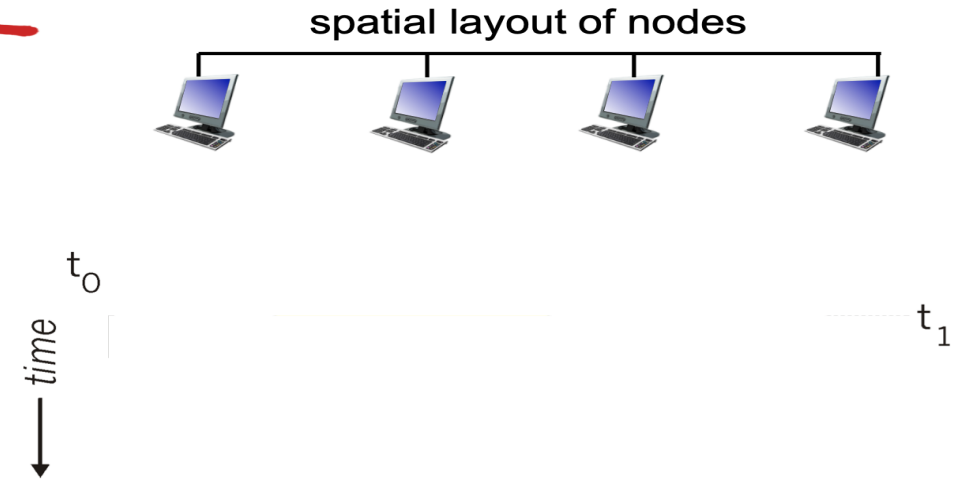
if channel sensed idle: transmit entire frame

- 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
  - distance & propagation delay play role in in determining collision probability



Link Layer and LANs
Multiple Access Protocols

# 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

Link Layer and LANs
Multiple Access Protocols

# 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:*
   - after *m*th collision, NIC chooses *K* at random from $\{0, 1, 2, \ldots, 2^m-1\}$. NIC waits $K \cdot 512$ bit times, returns to Step 2
   - longer backoff interval with more collisions

Link Layer and LANs
(Local Area Networks- LANs)

# Link layer, LANs: outline

5.1 introduction, services

5.2 error detection, correction

5.3 multiple access protocols

5.4 LANs

- addressing, ARP
- Ethernet
- switches
- VLANS

5.5 link virtualization: MPLS

5.6 data center networking

5.7 a day in the life of a web request

Link Layer and LANs
(MAC Addresses and ARP)

# MAC addresses and ARP

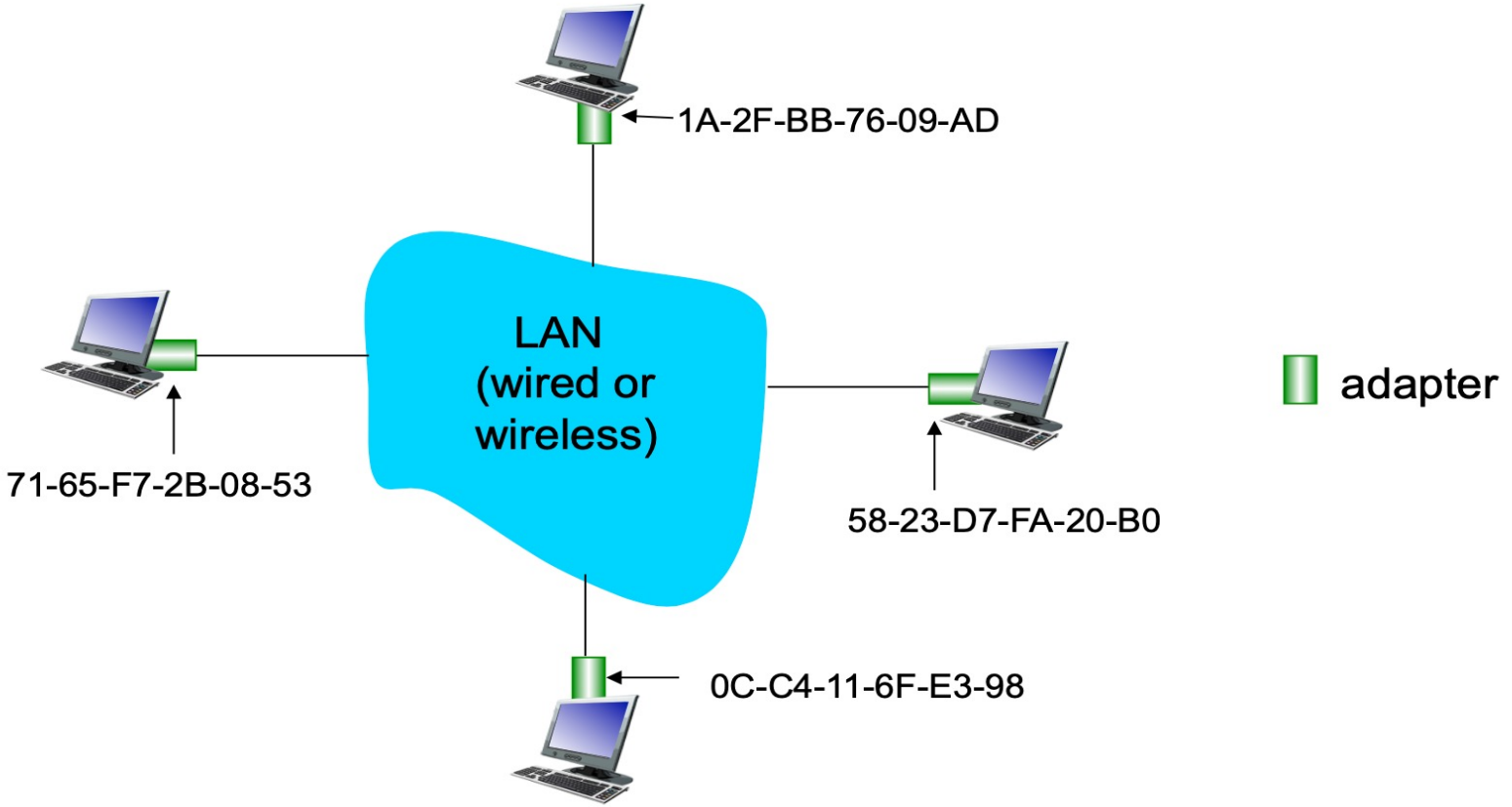
- 32-bit IP address:
  - *network-layer* address for interface
  - used for layer 3 (network layer) forwarding
- MAC (or LAN or physical or Ethernet) address:
  - function: *used 'locally" to get frame from one interface to another physically-connected interface (same network, in IP-addressing sense)*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

hexadecimal (base 16) notation
(each "number" represents 4 bits)

Link Layer and LANs
(LAN Addresses and ARP)

# LAN addresses and ARP

each adapter on LAN has unique *LAN* address



Link Layer and LANs
(LAN Addresses)

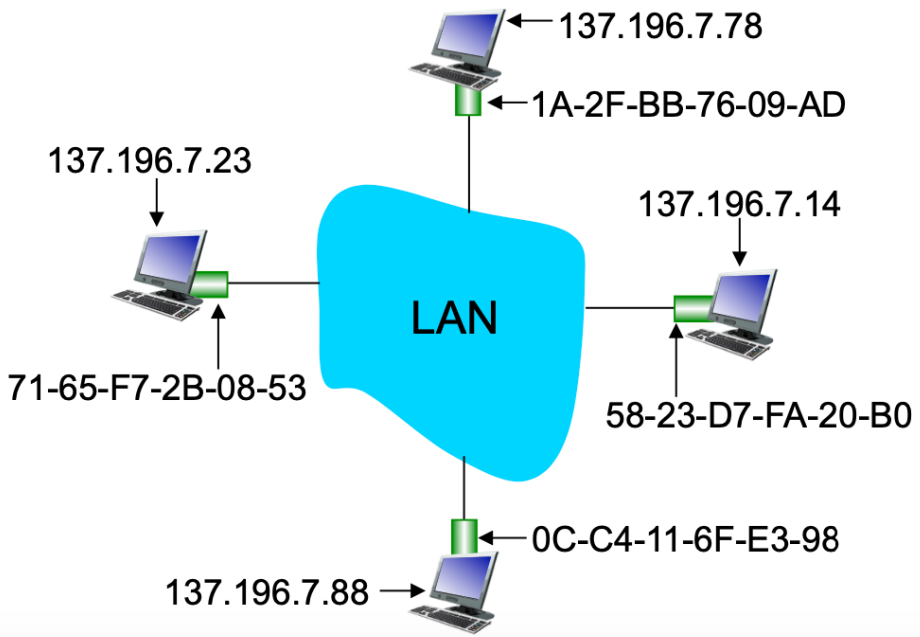
# LAN addresses (more)

- MAC address allocation administered by IEEE
- manufacturer buys portion of MAC address space (to assure uniqueness)
- analogy:
  - 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

Link Layer and LANs
(ARP- Address Resolution Protocol)

# ARP: address resolution protocol

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



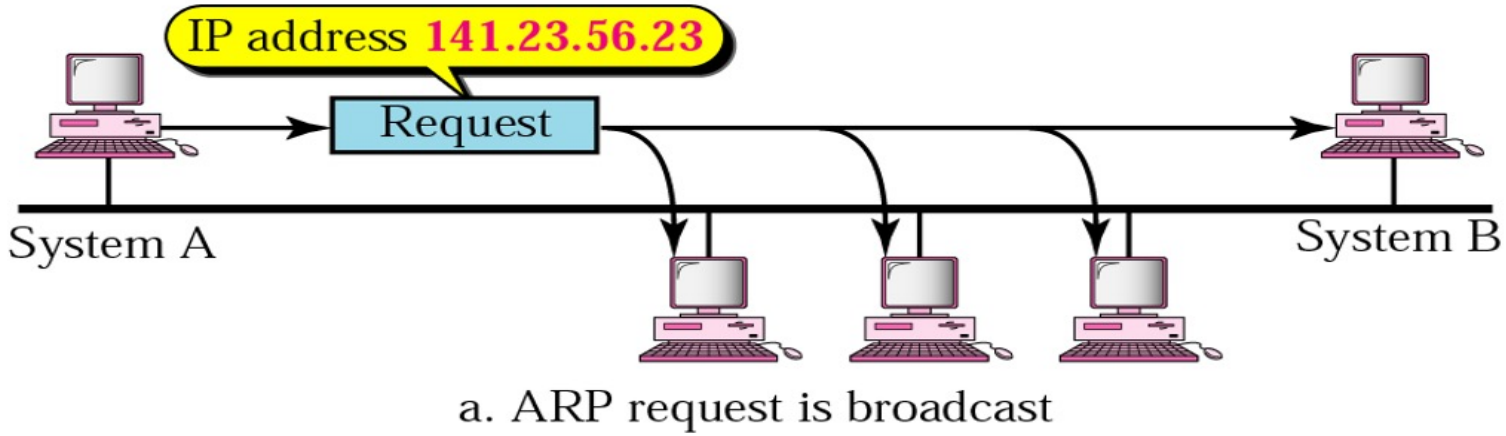

*ARP table:* each IP node (host, router) on LAN has table

- IP/MAC address mappings for some LAN nodes:

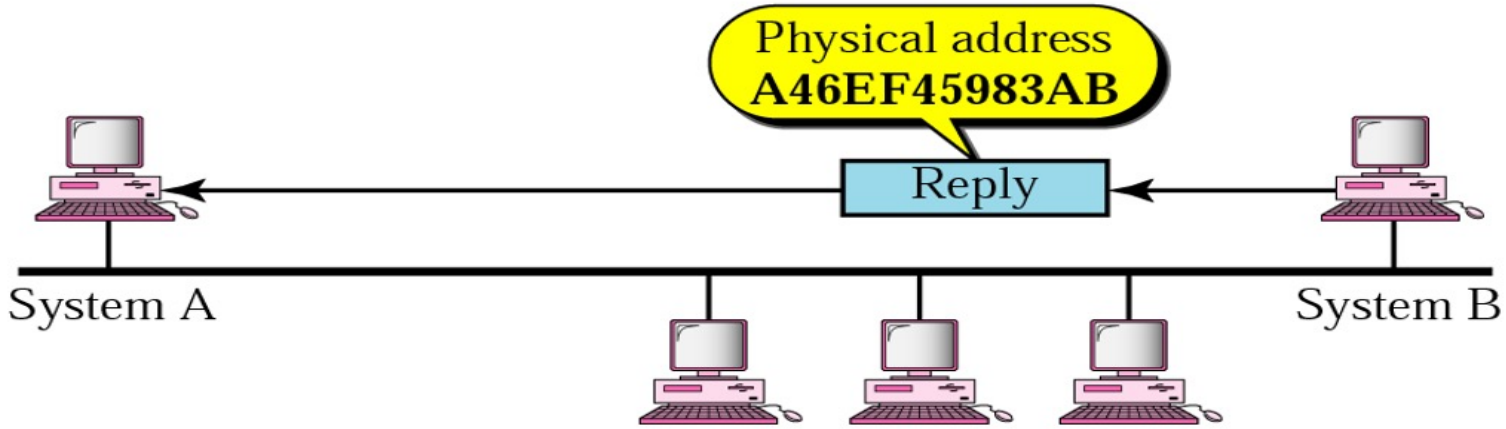
  < IP address; MAC address; TTL>
- TTL (Time To Live): time after which address mapping will be forgotten (typically 20 min)

# Link Layer and LANs (ARP- Address Resolution Protocol)

## *ARP Operation*



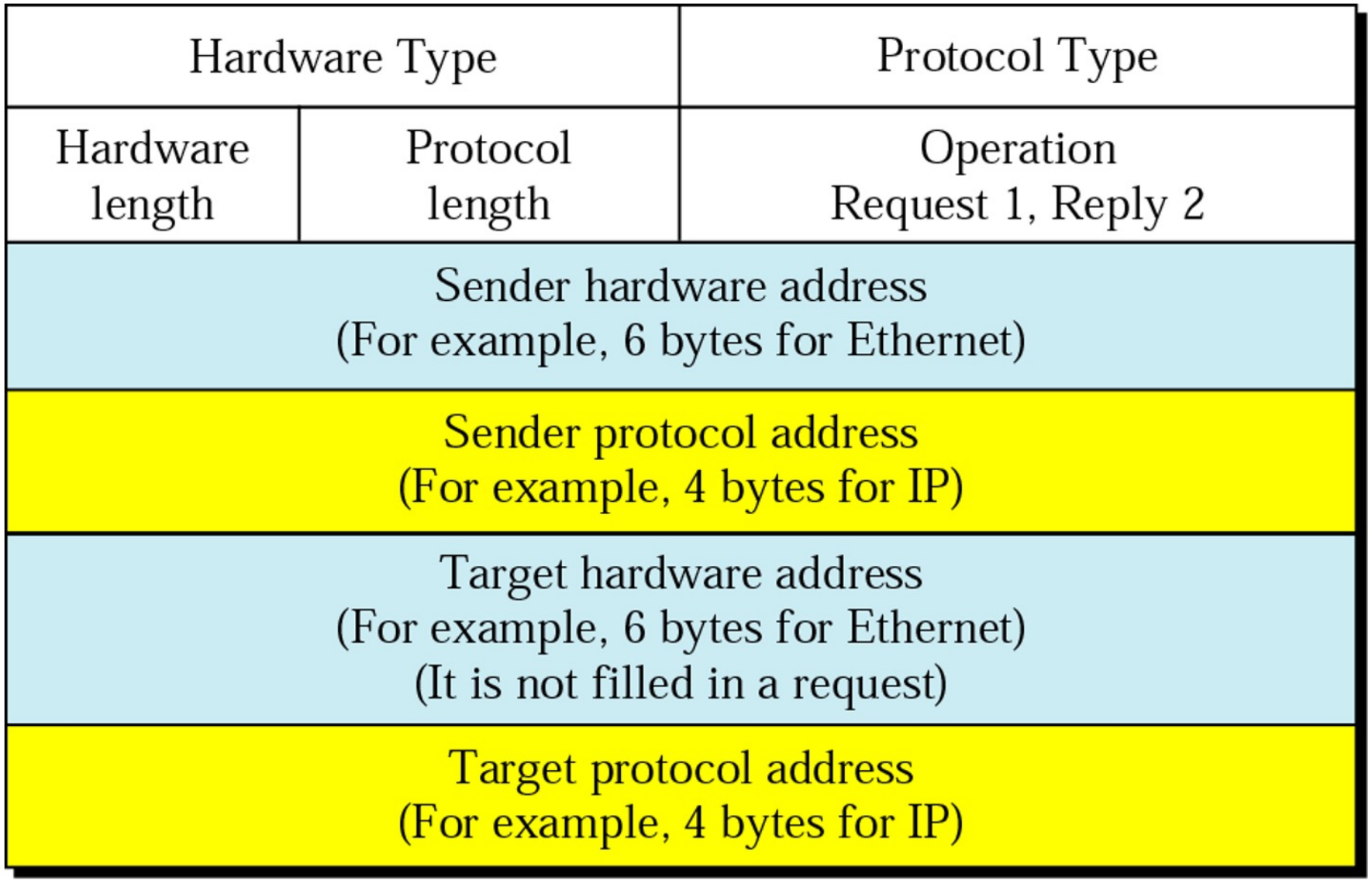

a. ARP request is broadcast




b. ARP reply is unicast

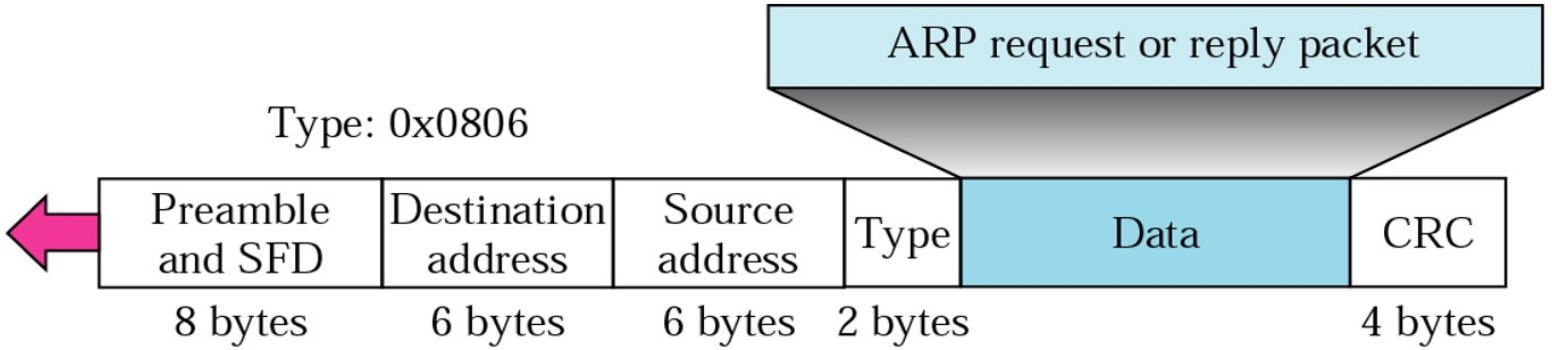
# Link Layer and LANs (ARP- Address Resolution Protocol)

## *ARP Packet*



# Link Layer and LANs (ARP- Address Resolution Protocol)

## Encapsulation of ARP packet




**Note:**

***An ARP request is broadcast; an ARP reply is unicast.***

# Link Layer and LANs (ARP- Address Resolution Protocol)

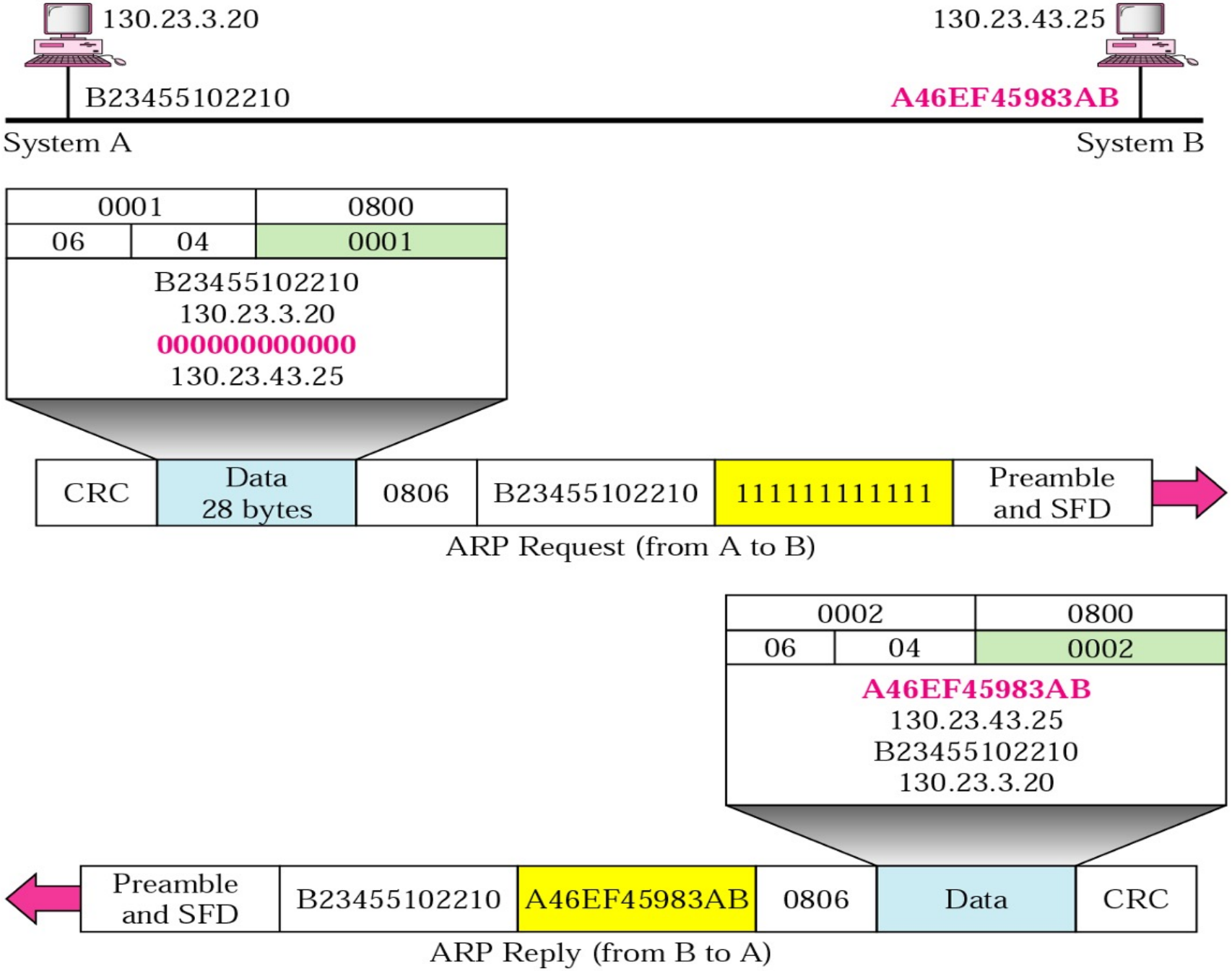
## *Example 1*

A host with IP address 130.23.3.20 and physical address B23455102210 has a packet to send to another host with IP address 130.23.43.25 and physical address A46EF45983AB. The two hosts are on the same Ethernet network. Show the ARP request and reply packets encapsulated in Ethernet frames.

## *Solution*

**Figure 20.6 shows the ARP request and reply packets. Note that the ARP data field in this case is 28 bytes, and that the individual addresses do not fit in the 4-byte boundary. That is why we do not show the regular 4-byte boundaries for these addresses. Note that we use hexadecimal for every field except the IP addresses.**

# Link Layer and LANs (ARP- Address Resolution Protocol)



Link Layer and LANs
(ARP- Address Resolution Protocol)

# 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)
  - 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*

Link Layer and LANs
(Ethernet)

# Link layer, LANs: outline

5.1 introduction, services

5.2 error detection, correction

5.3 multiple access protocols

5.4 LANs

- addressing, ARP
- Ethernet
- switches
- VLANS

5.5 link virtualization: MPLS

5.6 data center networking

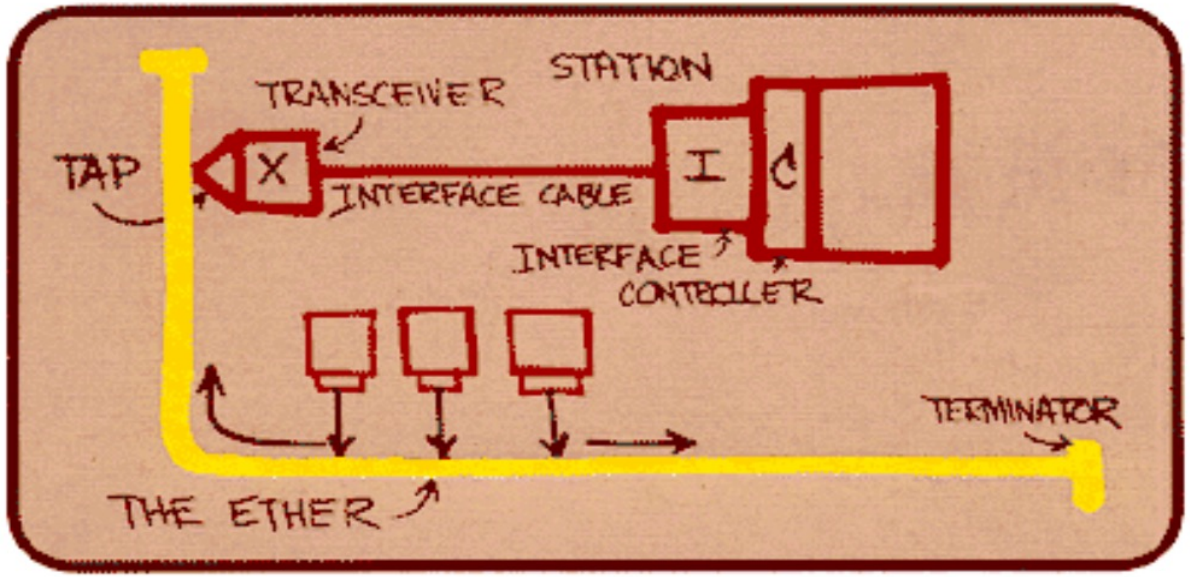
5.7 a day in the life of a web request

Link Layer and LANs (Ethernet)

# Ethernet

"dominant" wired LAN technology:

- cheap $20 for NIC
- first widely used LAN technology
- simpler, cheaper than token LANs and ATM
- kept up with speed race: 10 Mbps – 10 Gbps

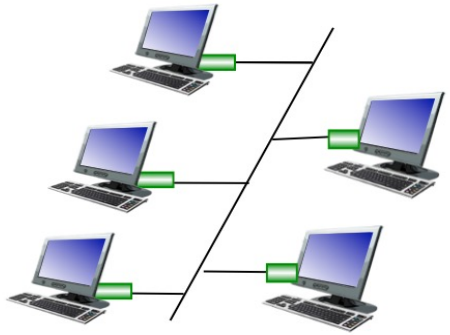



*Metcalfe's Ethernet sketch*

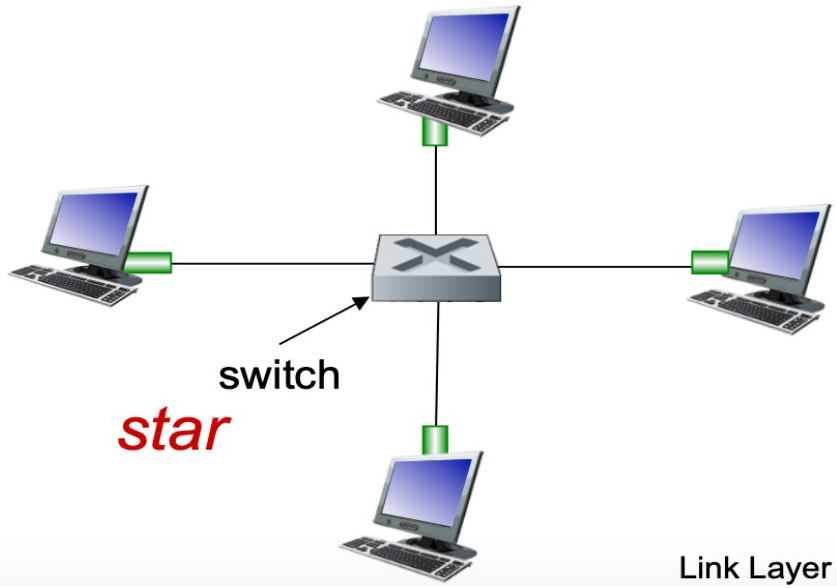
Link Layer and LANs (Ethernet)

# Ethernet: physical topology

- *bus:* popular through mid 90s
  - all nodes in same collision domain (can collide with each other)
- *star:* prevails today
  - active *switch* in center
  - each "spoke" runs a (separate) Ethernet protocol (nodes do not collide with each other)



*bus:* coaxial cable



Link Layer and LANs (Ethernet)

# Ethernet frame structure

sending adapter encapsulates IP datagram (or other network layer protocol packet) in Ethernet frame


*preamble:*

- 7 bytes with pattern 10101010 followed by one byte with pattern 10101011
- used to synchronize receiver, sender clock rates

Link Layer and LANs (Ethernet)

# Ethernet frame structure (more)

- *addresses:* 6 byte source, destination MAC addresses
  - if adapter receives frame with matching destination address, or with broadcast address (e.g. ARP packet), it passes data in frame to network layer protocol
  - otherwise, adapter discards frame
- *type:* indicates higher layer protocol (mostly IP but others possible, e.g., Novell IPX, AppleTalk)
- *CRC:* cyclic redundancy check at receiver
  - error detected: frame is dropped

Link Layer and LANs (Ethernet)

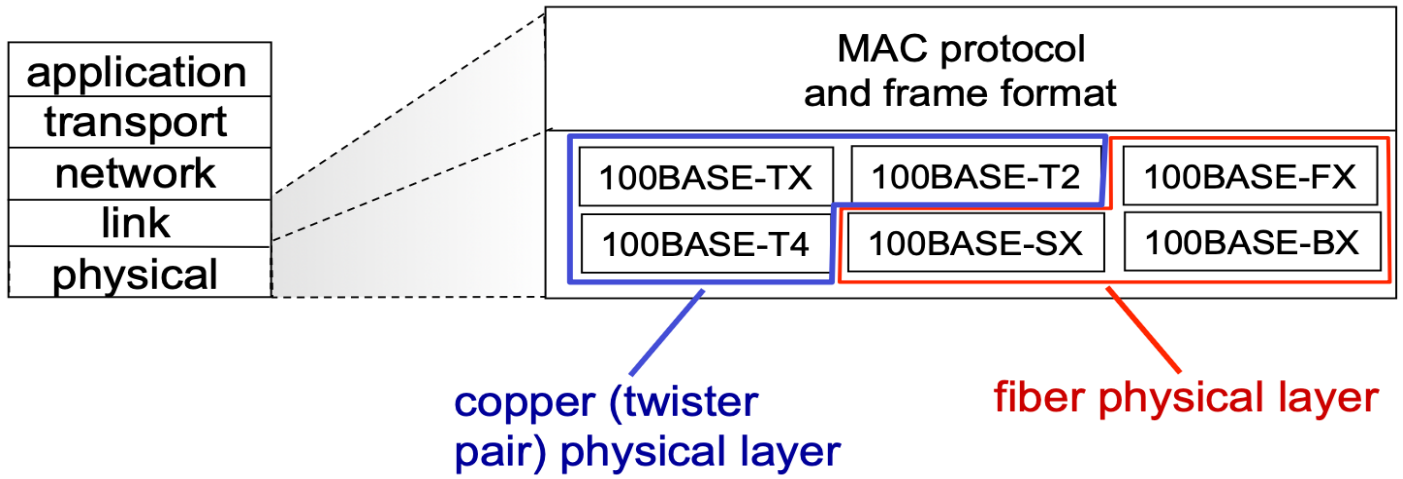
# Ethernet: unreliable, connectionless

- *connectionless:* no handshaking between sending and receiving NICs
- *unreliable:* receiving NIC doesnt send acks or nacks to sending NIC
  - data in dropped frames recovered only if initial sender uses higher layer rdt (e.g., TCP), otherwise dropped data lost
- Ethernet's MAC protocol: unslotted *CSMA/CD wth binary backoff*

Link Layer and LANs
(Ethernet)

# 802.3 Ethernet standards: link & physical layers

- *many* different Ethernet standards
  - common MAC protocol and frame format
  - different speeds: 2 Mbps, 10 Mbps, 100 Mbps, 1Gbps, 10G bps
  - different physical layer media: fiber, cable



Link Layer and LANs
(Switches & VLANs)

# Link layer, LANs: outline

5.1 introduction, services

5.2 error detection, correction

5.3 multiple access protocols

5.4 LANs

- addressing, ARP
- Ethernet
- switches
- VLANS

5.5 link virtualization: MPLS

5.6 data center networking

5.7 a day in the life of a web request

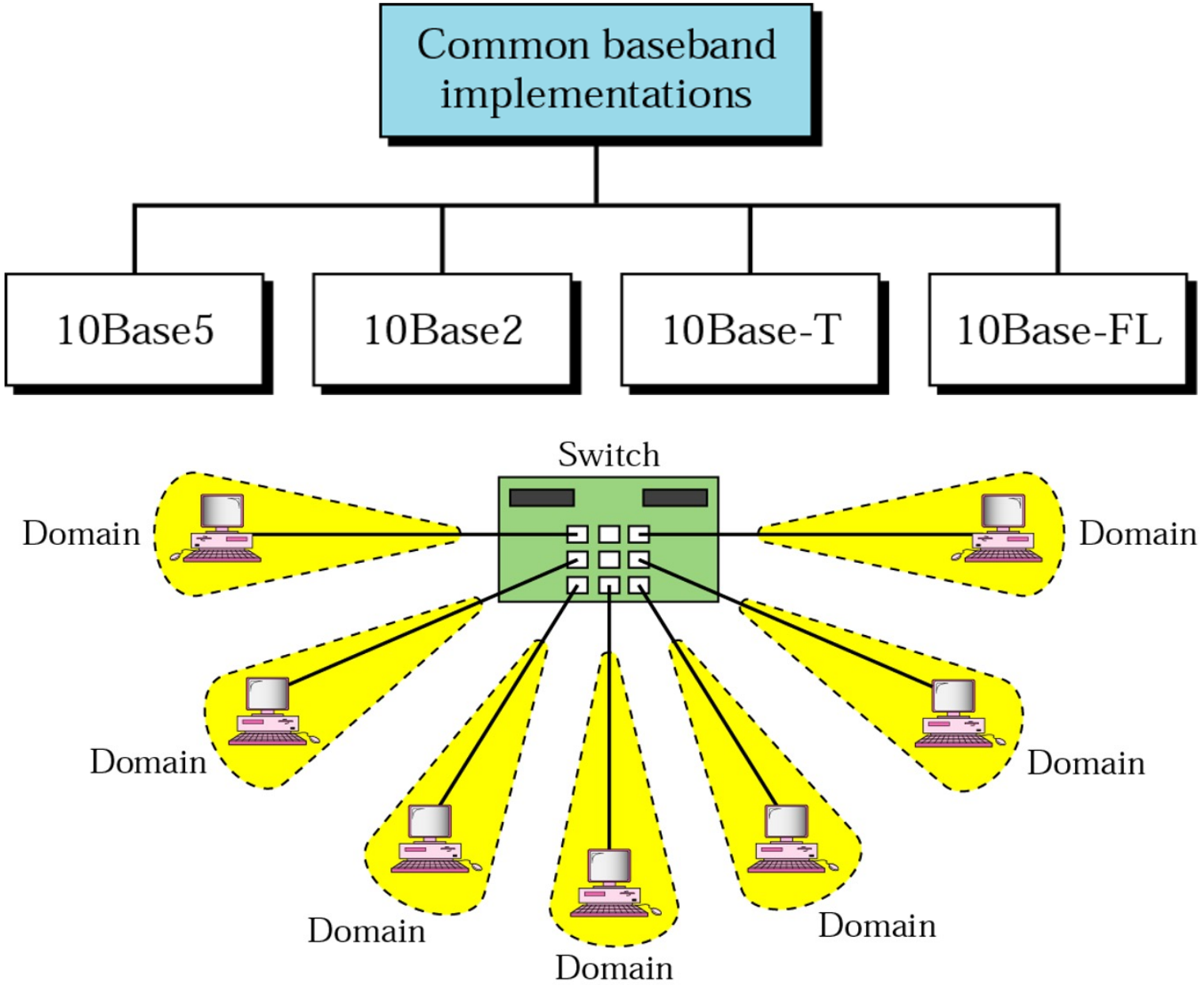
Link Layer and LANs
(Switches & VLANs)

# 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

# Categories of Ethernet

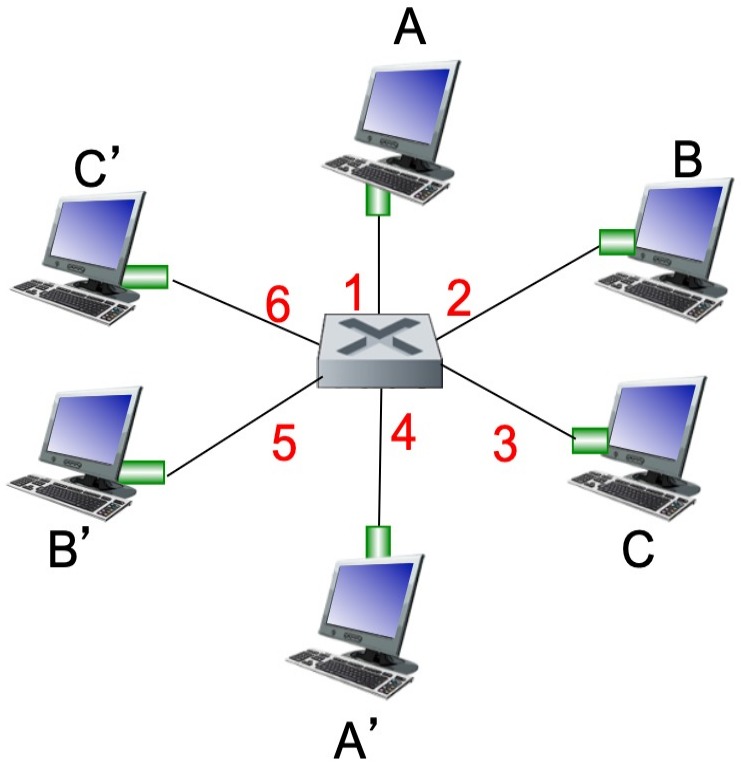
Link Layer and LANs (Switches & VLANs)



Link Layer and LANs (Switches & VLANs)

# 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)*

Link Layer and LANs (Switches & VLANs)

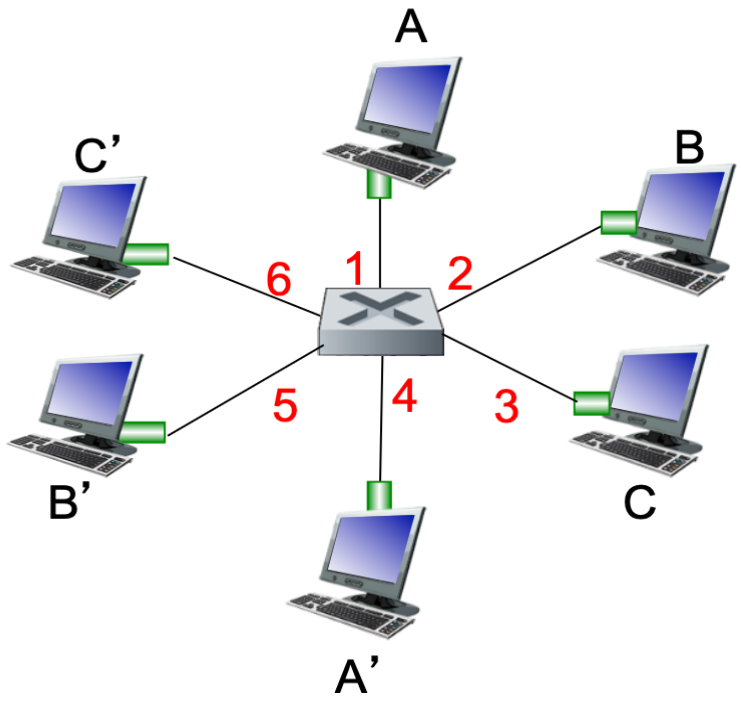
# Switch forwarding table

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

Link Layer and LANs (Switches & VLANs)

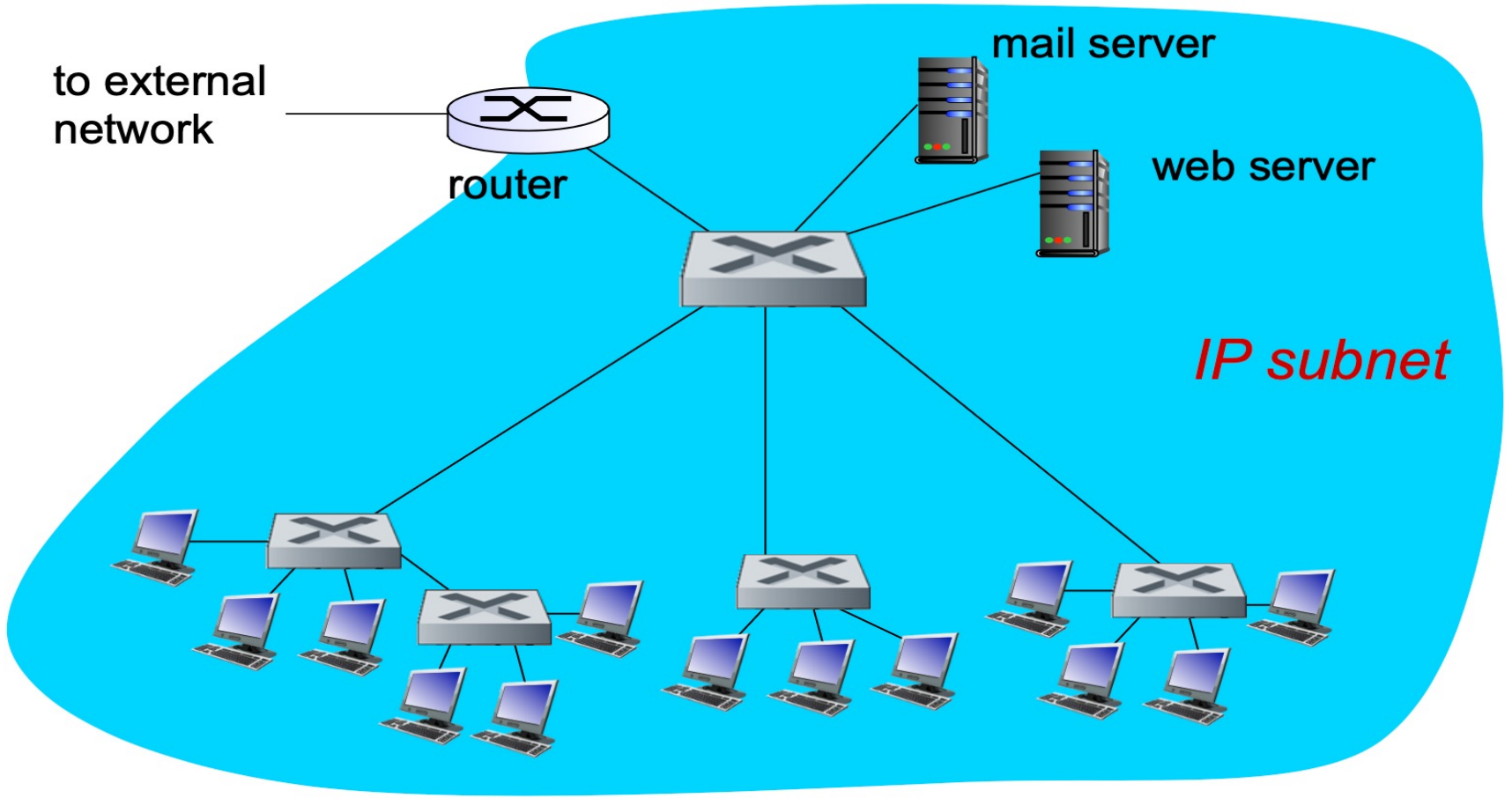
# Switch: frame filtering/forwarding

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 */

Link Layer and LANs
(Switches & VLANs)

# Institutional network



Link Layer and LANs (Switches & VLANs)

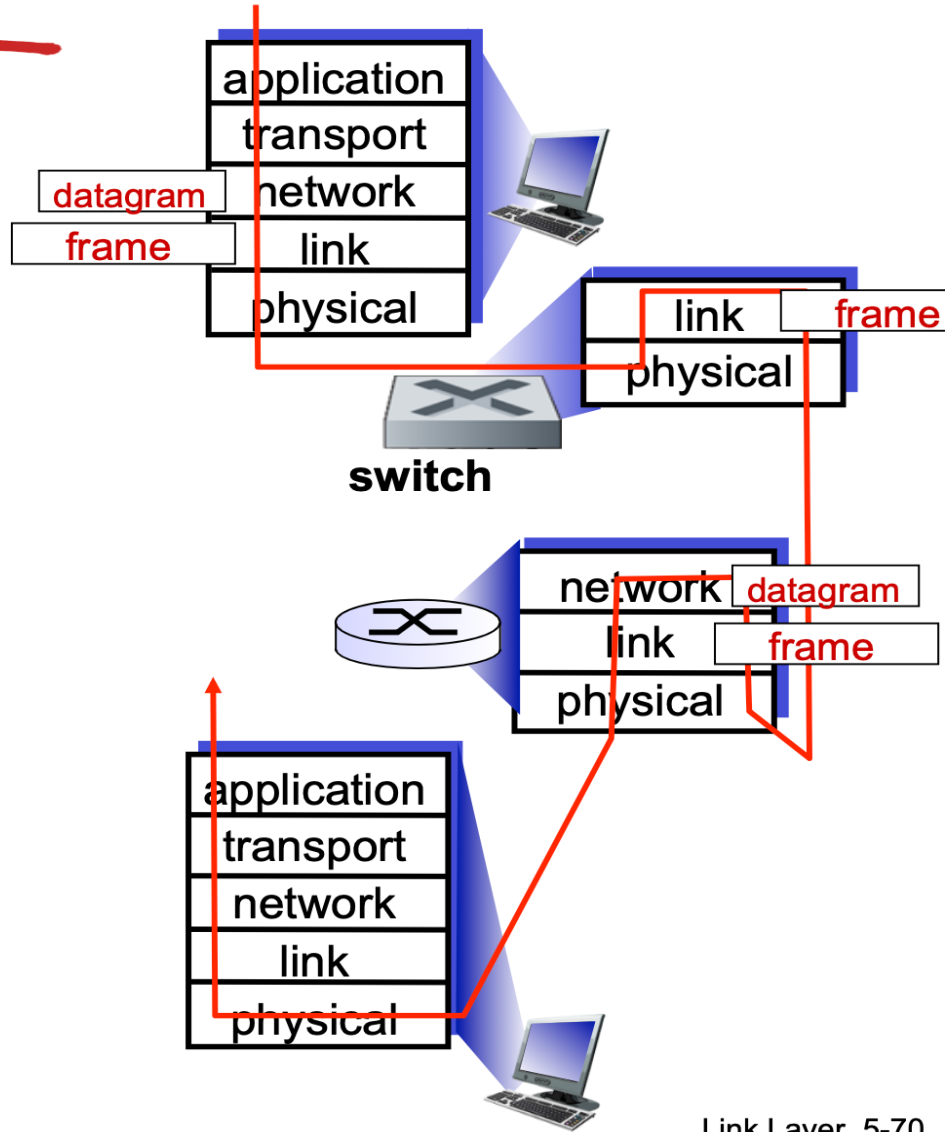
# Switches vs. routers

**both are store-and-forward:**

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

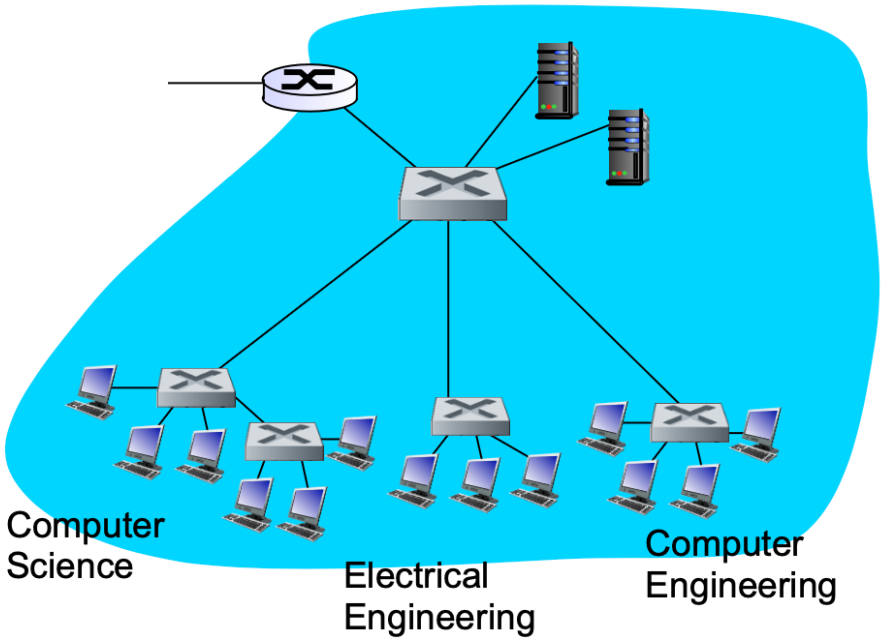
**both have forwarding tables:**

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



Link Layer and LANs
(Switches & VLANs)

# VLANs: motivation




*consider*:

- CS user moves office to EE, but wants connect to CS switch?
- single broadcast domain:
  - all layer-2 broadcast traffic (ARP, DHCP, unknown location of destination MAC address) must cross entire LAN
  - security/privacy, efficiency issues

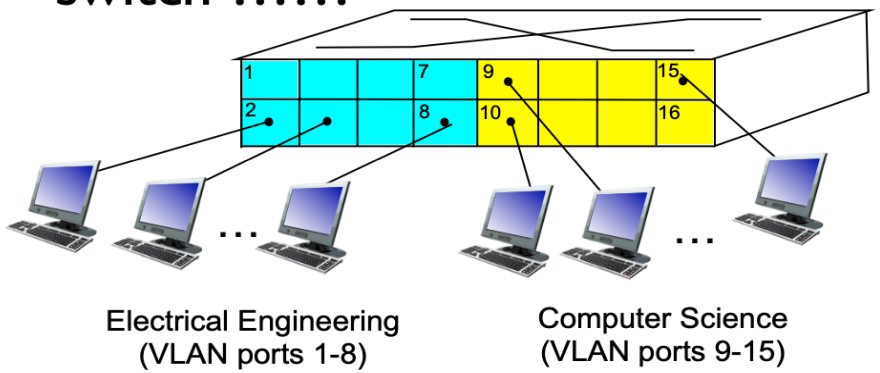
# Link Layer and LANs (Switches & VLANs)

# VLANs

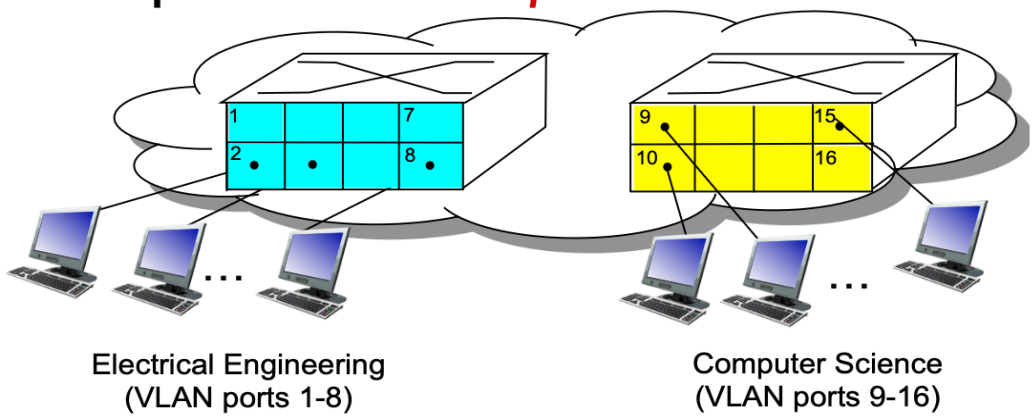
***Virtual Local Area Network***

switch(es) supporting VLAN capabilities can be configured to define multiple ***virtual*** LANS over single physical LAN infrastructure.

port-based VLAN: switch ports grouped (by switch management software) so that *single* physical switch ……



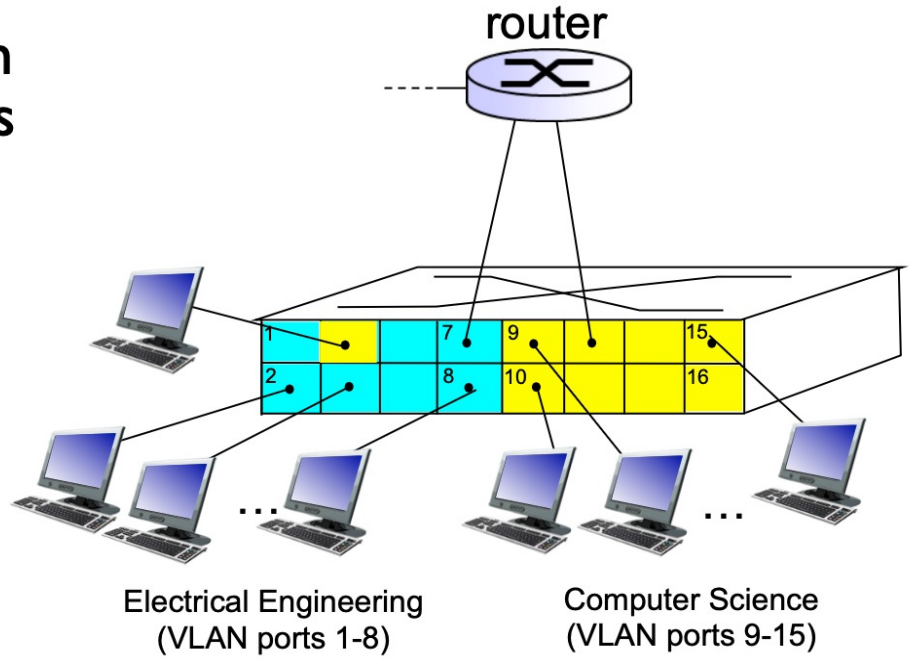

… operates as *multiple* virtual switches



Link Layer and LANs (Switches & VLANs)

# Port-based VLAN

- *traffic isolation:* frames to/from ports 1-8 can *only* reach ports 1-8
  - can also define VLAN based on MAC addresses of endpoints, rather than switch port
- *dynamic membership:* ports can be dynamically assigned among VLANs
- *forwarding between VLANS:* done via routing (just as with separate switches)
  - in practice vendors sell combined switches plus routers



Link Layer and LANs (HDLC)

# High-Level Data Link Control

***Configurations and Transfer Modes***

***Frames***

***Frame Format***

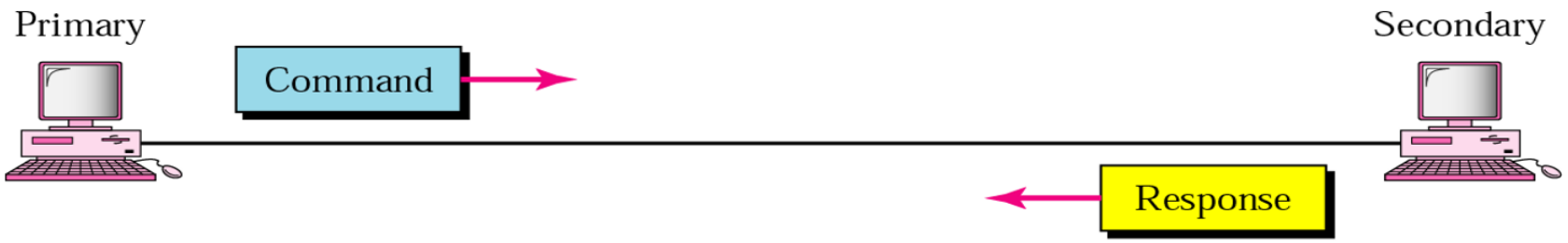
***Examples***

***Data Transparency***

# Link Layer and LANs (HDLC)

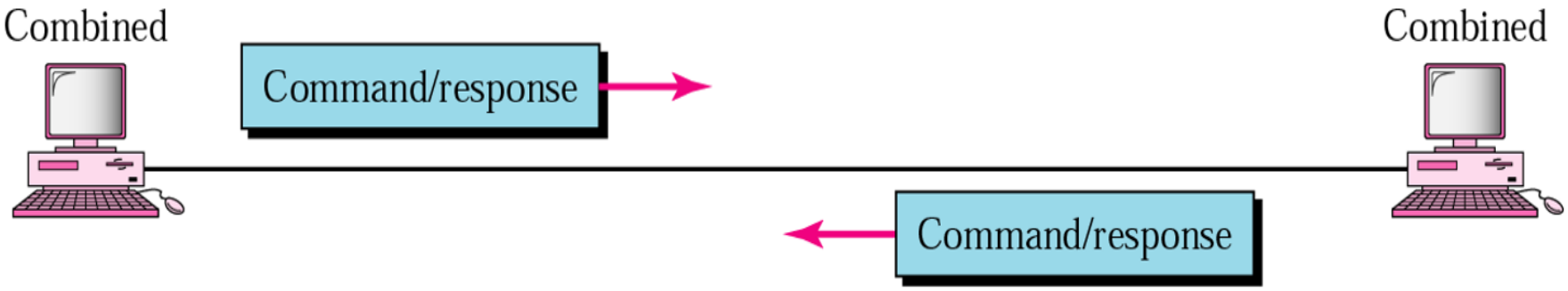
(NRM, ARM)

## NRM- Normal Response Mode




a. Point-to-point


b. Multipoint

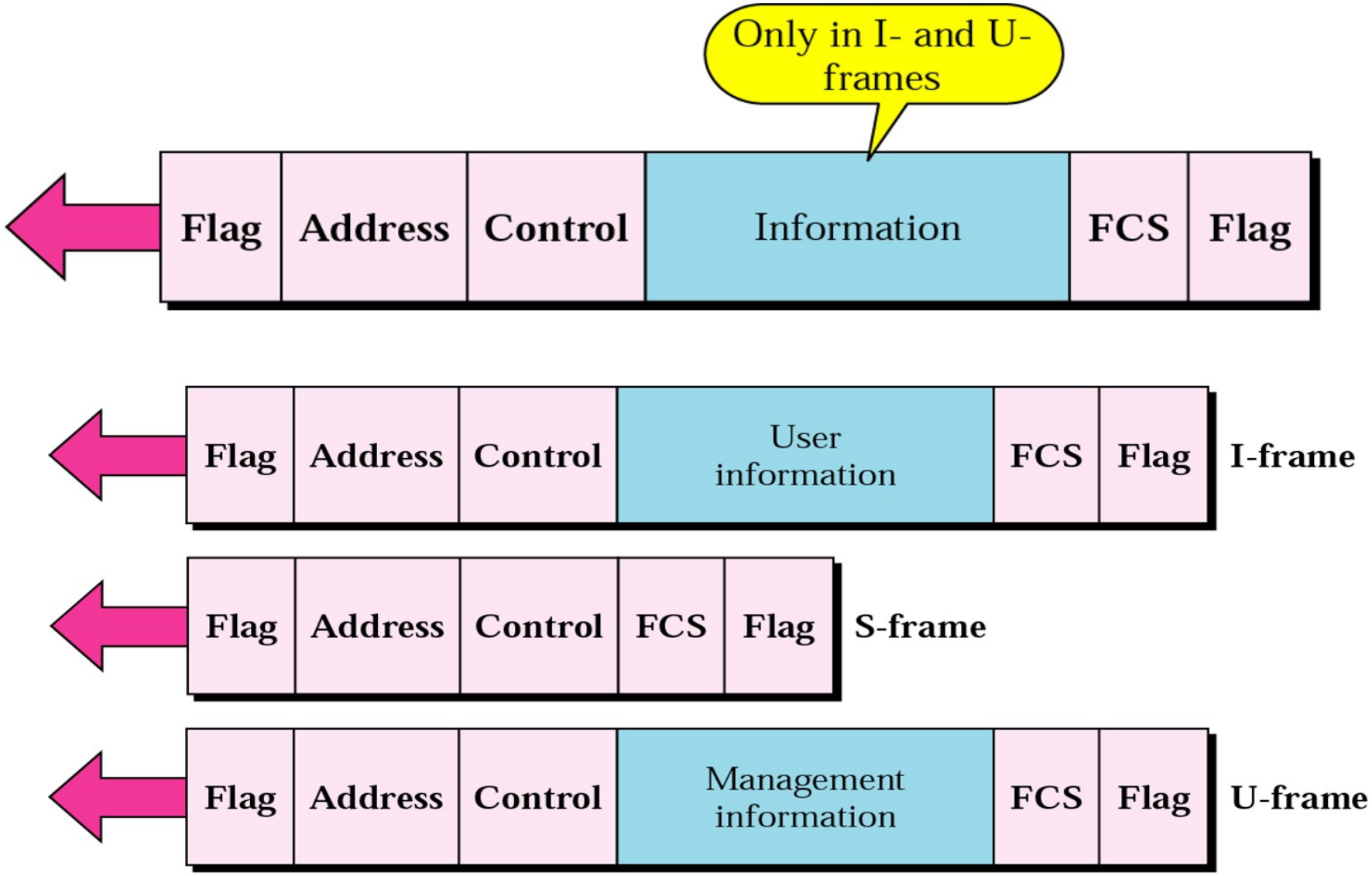



## ARM, ABM- Asynchronous Response, Balanced Mode

# HDLC Frame and Frame Type

Link Layer and LANs
(HDLC)

(Frame and Type)

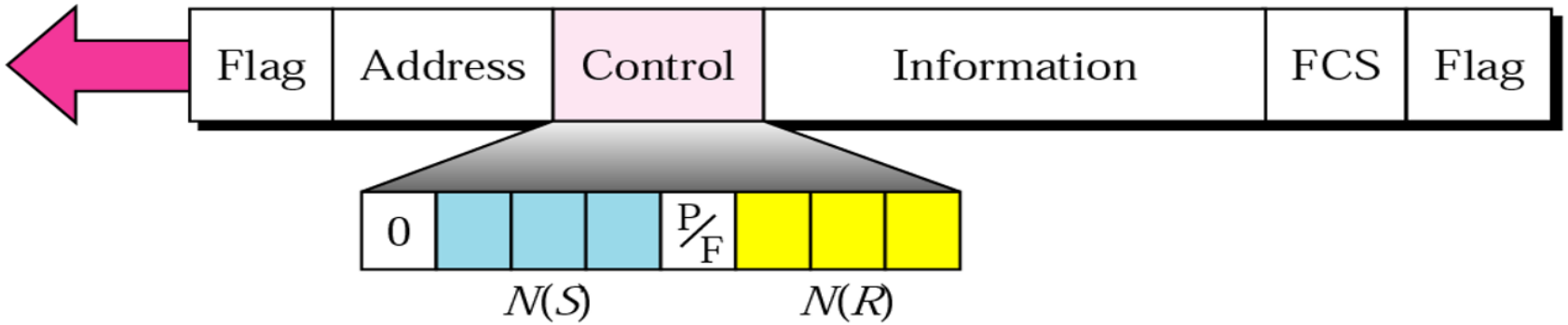


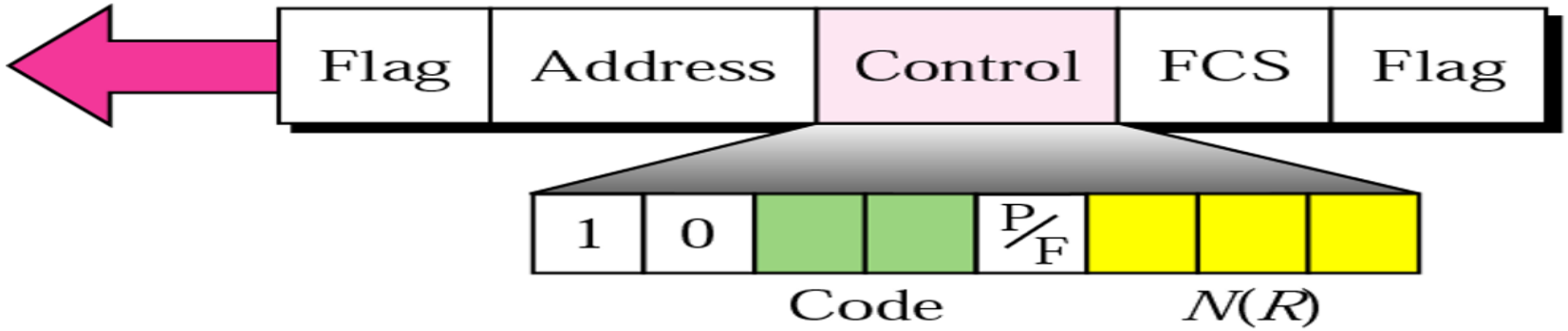

Information, Supervisory, Unnumbered

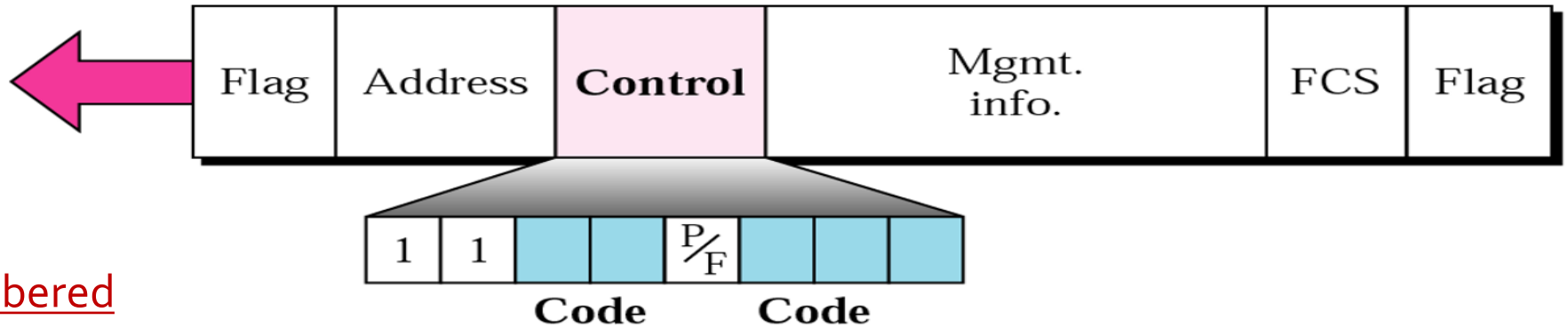
# HDLC Frame and Frame Type

Link Layer and LANs
(HDLC)

(Frame and Type)




Information, Supervisory, Unnumbered

Link Layer and LANs (HDLC)

## *Table 11.1 U-frame control command and response*

| Command/response | Meaning |
|---|---|
| SNRM | Set normal response mode |
| SNRME | Set normal response mode (extended) |
| SABM | Set asynchronous balanced mode |
| SABME | Set asynchronous balanced mode (extended) |
| UP | Unnumbered poll |
| UI | Unnumbered information |
| UA | Unnumbered acknowledgment |
| RD | Request disconnect |
| DISC | Disconnect |
| DM | Disconnect mode |
| RIM | Request information mode |
| SIM | Set initialization mode |
| RSET | Reset |
| XID | Exchange ID |
| FRMR | Frame reject |

Link Layer and LANs
(Link Virtualization: MPLS)

# Link layer, LANs: outline

5.1 introduction, services

5.2 error detection, correction

5.3 multiple access protocols

5.4 LANs

- addressing, ARP
- Ethernet
- switches
- VLANS

5.5 link virtualization: MPLS

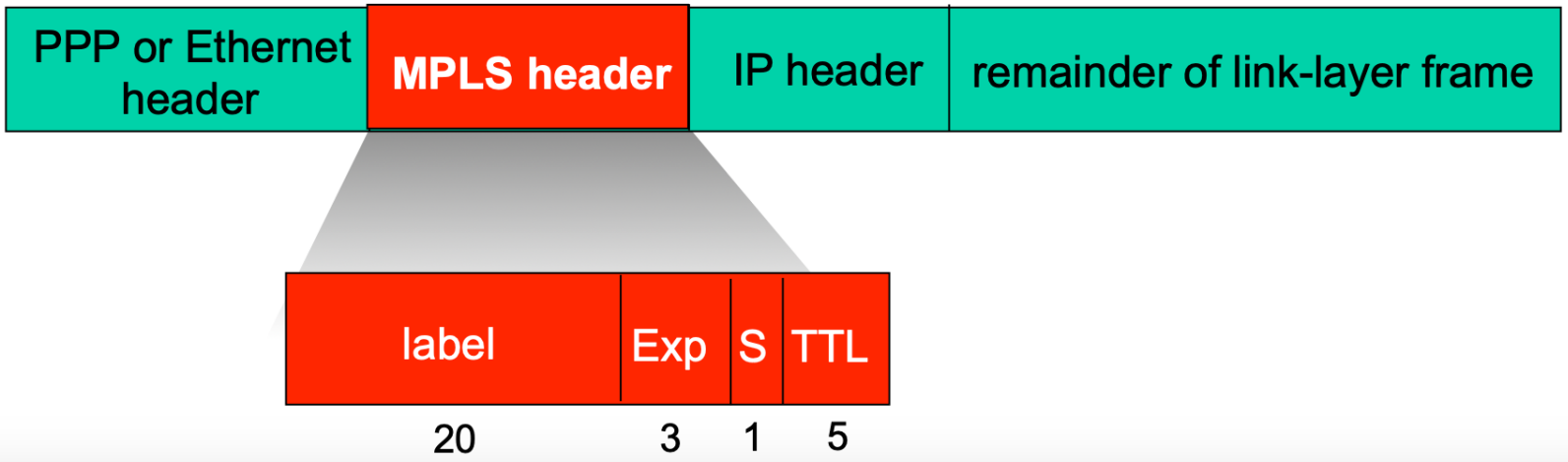
5.6 data center networking

5.7 a day in the life of a web request

Link Layer and LANs
(Link Virtualization: MPLS)

# Multiprotocol label switching (MPLS)

- initial goal: high-speed IP forwarding using fixed length label (instead of IP address)
  - fast lookup using fixed length identifier (rather than shortest prefix matching)
  - borrowing ideas from Virtual Circuit (VC) approach
  - but IP datagram still keeps IP address!



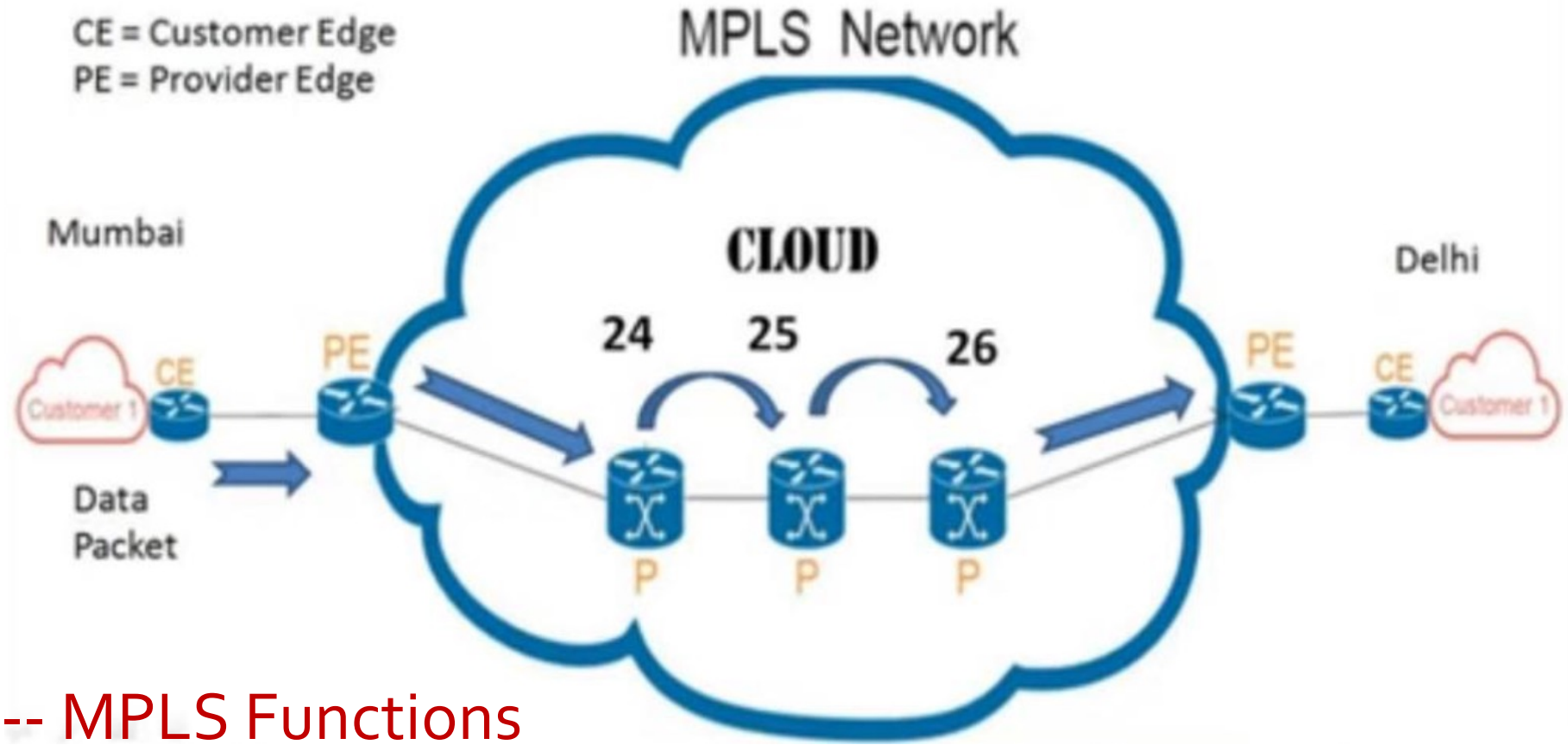


2.5 Layer Protocol/ Technique

Link Layer and LANs
(Link Virtualization: MPLS)

# MPLS Network

## 2) How MPLS Work.




PUSH, SWAP, POP------ MPLS Functions

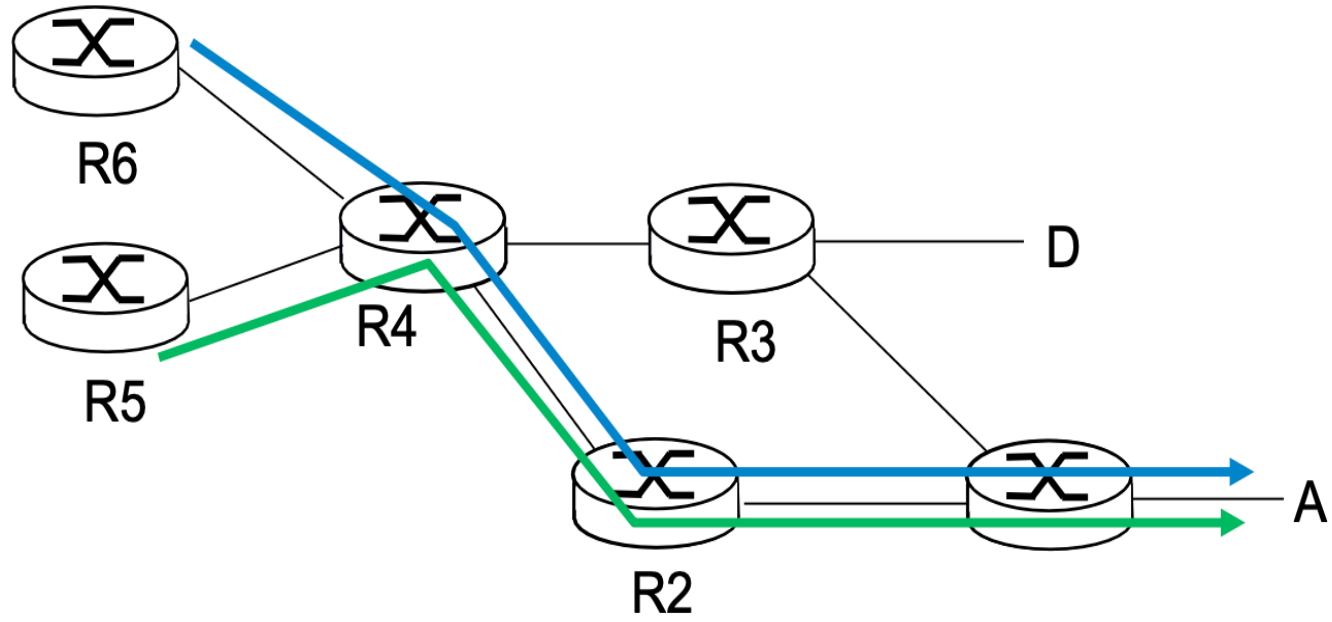
Link Layer and LANs
(Link Virtualization: MPLS)

# MPLS capable routers

- a.k.a. label-switched router
- forward packets to outgoing interface based only on label value (*don't inspect IP address*)
  - MPLS forwarding table distinct from IP forwarding tables
- *flexibility:* MPLS forwarding decisions can *differ* from those of IP
  - use destination *and* source addresses to route flows to same destination differently (traffic engineering)
  - re-route flows quickly if link fails: pre-computed backup paths (useful for VoIP)

Link Layer and LANs
(Link Virtualization: MPLS)

# MPLS versus IP paths

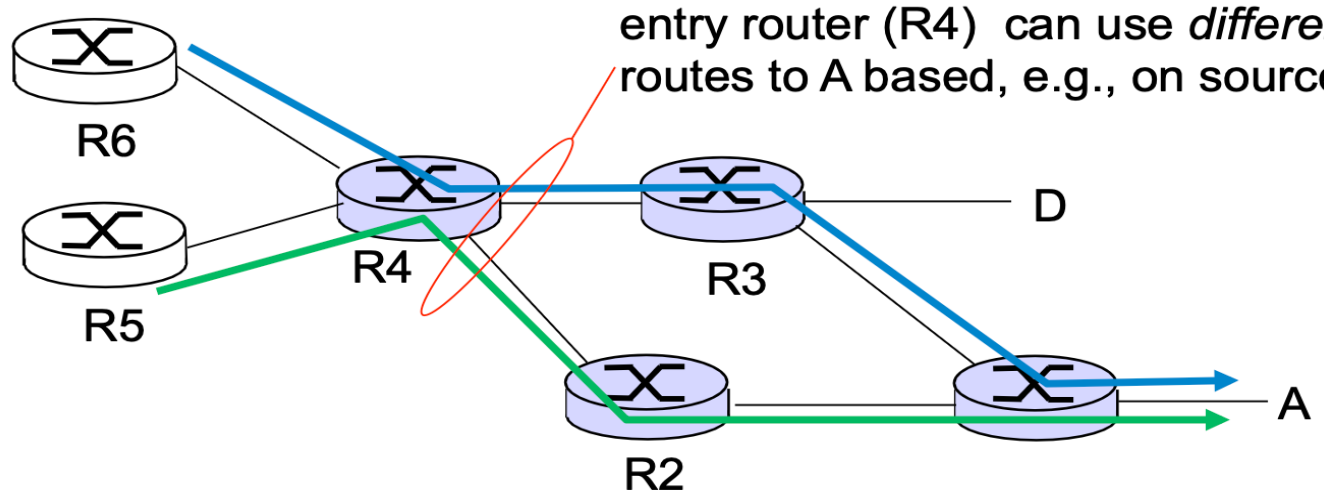



- *IP routing: path to destination determined by destination address alone*

*IP router*

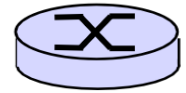
Link Layer and LANs
(Link Virtualization: MPLS)

# MPLS versus IP paths




- *IP routing:* *path to destination determined by destination address alone*
- *MPLS routing:* path to destination can be based on source *and* dest. address
  - *fast reroute:* precompute backup routes in case of link failure

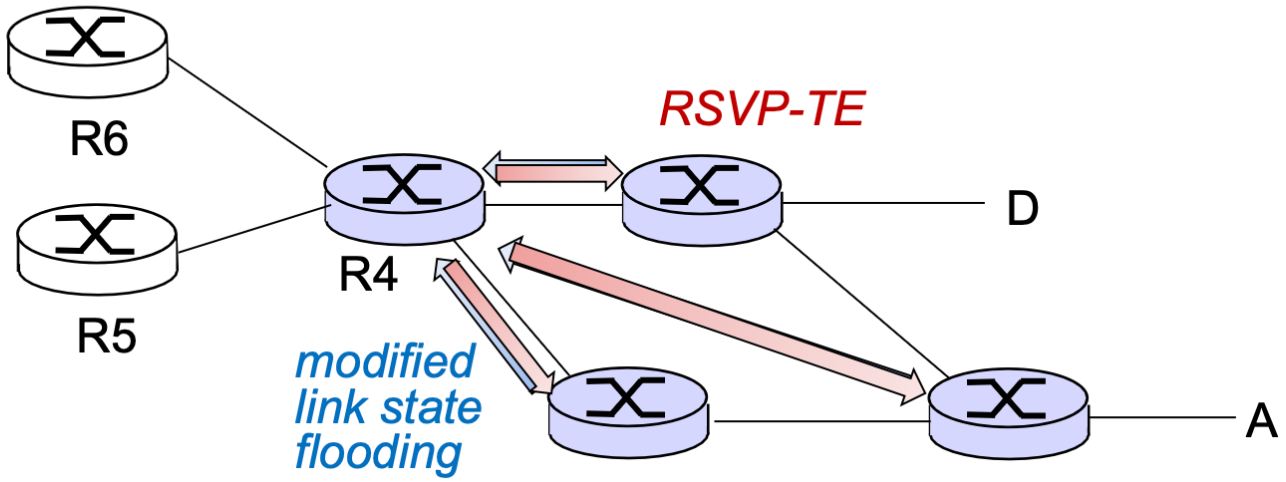

*IP-only router*


*MPLS and IP router*

Link Layer and LANs
(Link Virtualization: MPLS)

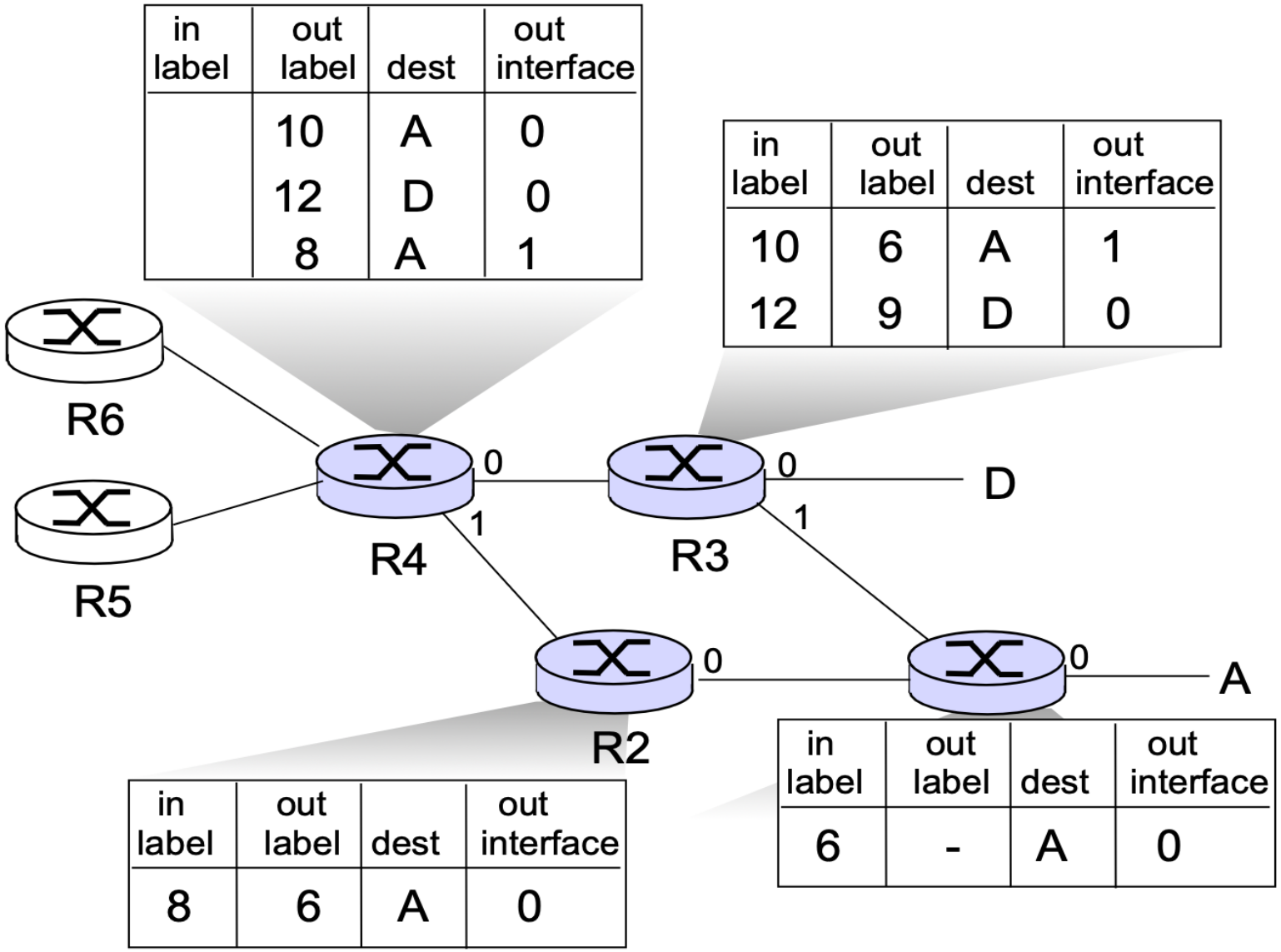
# MPLS signaling

- modify OSPF, IS-IS link-state flooding protocols to carry info used by MPLS routing,
  - e.g., link bandwidth, amount of "reserved" link bandwidth
- *entry MPLS router uses RSVP-TE signaling protocol to set up MPLS forwarding at downstream routers*

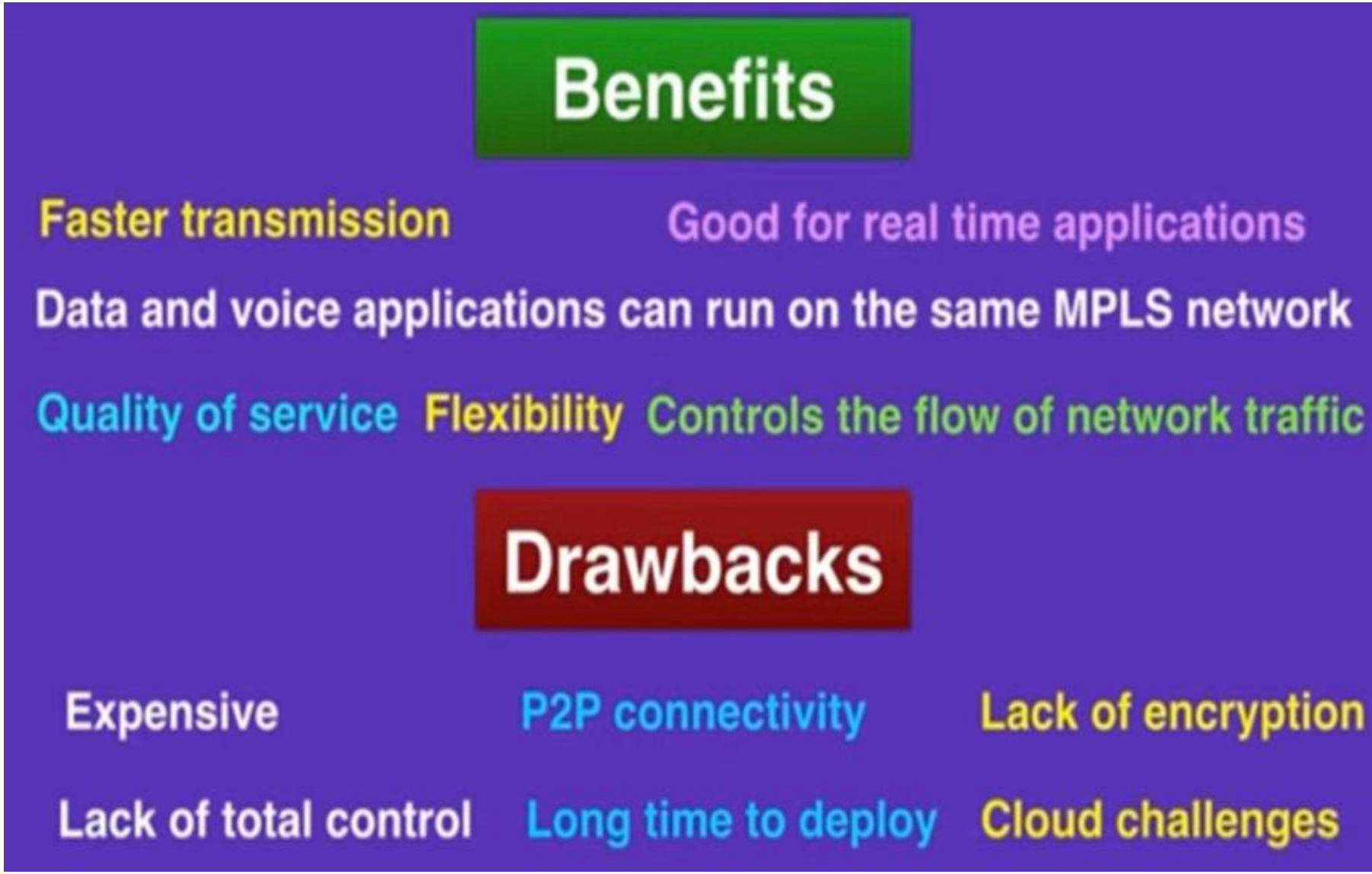


Link Layer and LANs
(Link Virtualization: MPLS)

# MPLS forwarding tables



# Link Layer and LANs (Link Virtualization: MPLS)



Link Layer and LANs
(Data Center Networking)

# Link layer, LANs: outline

5.1 introduction, services

5.2 error detection, correction

5.3 multiple access protocols

5.4 LANs

- addressing, ARP
- Ethernet
- switches
- VLANS

5.5 link virtualization: MPLS

5.6 data center networking

5.7 a day in the life of a web request

Link Layer and LANs
(Data Center Networking)

# Data center networks

- 10's to 100's of thousands of hosts, often closely coupled, in close proximity:
  - e-business (e.g. Amazon)
  - content-servers (e.g., YouTube, Akamai, Apple, Microsoft)
  - search engines, data mining (e.g., Google)
- challenges:
  - multiple applications, each serving massive numbers of clients
  - managing/balancing load, avoiding processing, networking, data bottlenecks

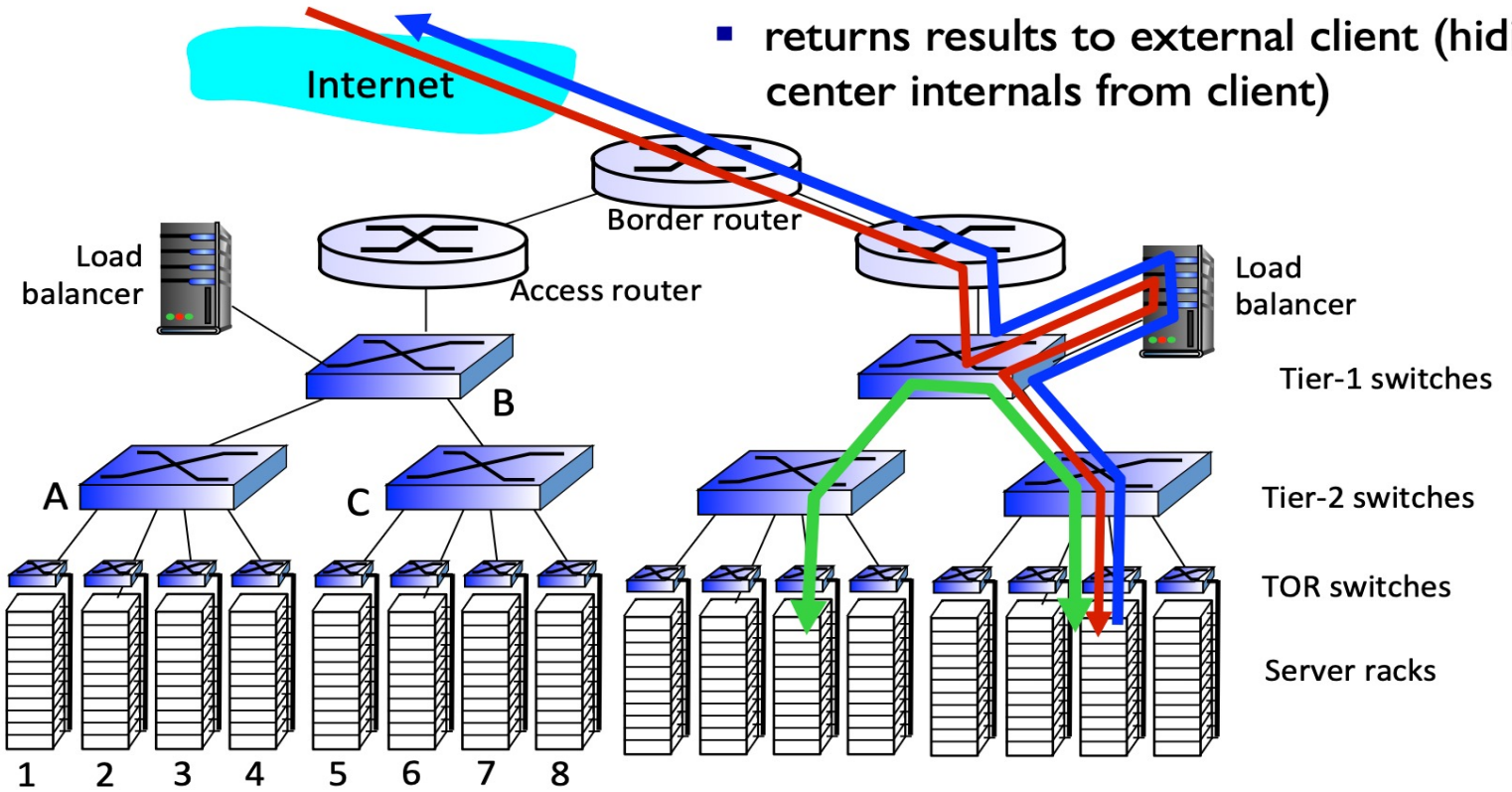
Inside a 40-ft Microsoft container, Chicago data center

Link Layer and LANs
(Data Center Networking)

# Data center networks

*load balancer: application-layer routing*

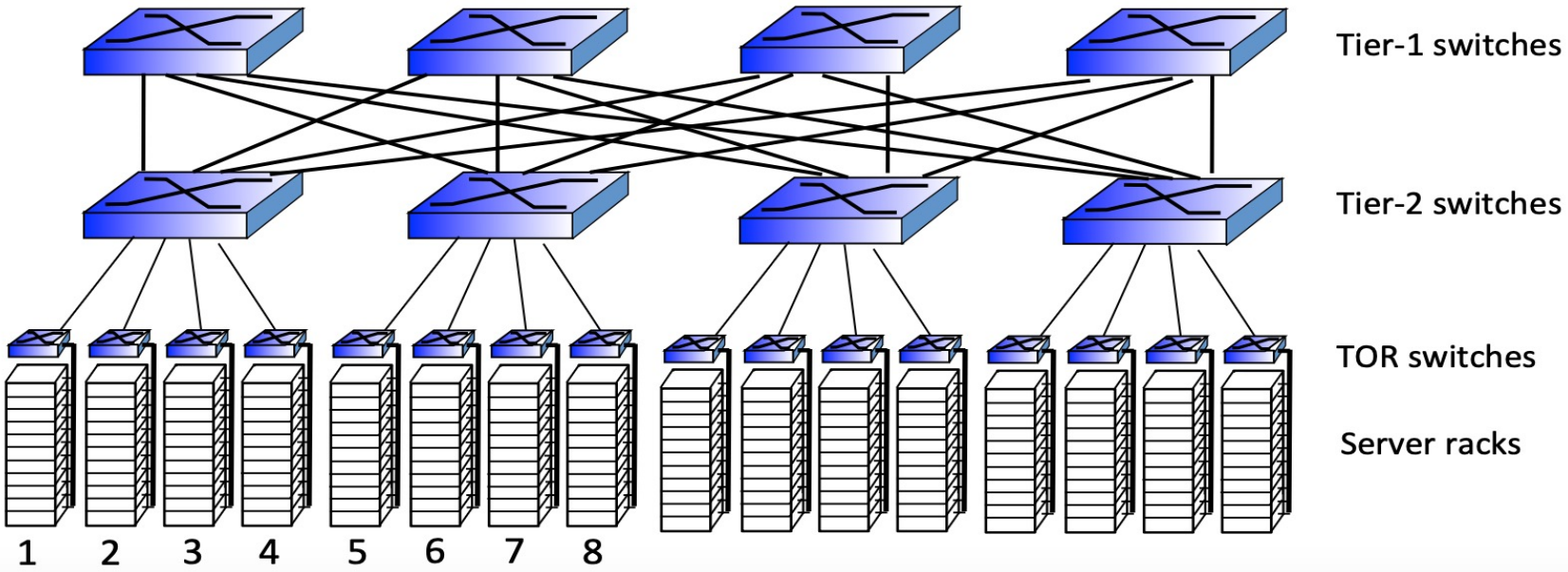
- receives external client requests
- directs workload within data center
- returns results to external client (hiding data center internals from client)



Link Layer and LANs
(Data Center Networking)

# Data center networks

- rich interconnection among switches, racks:
  - increased throughput between racks (multiple routing paths possible)
  - increased reliability via redundancy



Link Layer and LANs
(Web Request)

# Link layer, LANs: outline

5.1 introduction, services

5.2 error detection, correction

5.3 multiple access protocols

5.4 LANs

- addressing, ARP
- Ethernet
- switches
- VLANS

5.5 link virtualization: MPLS

5.6 data center networking

5.7 a day in the life of a web request

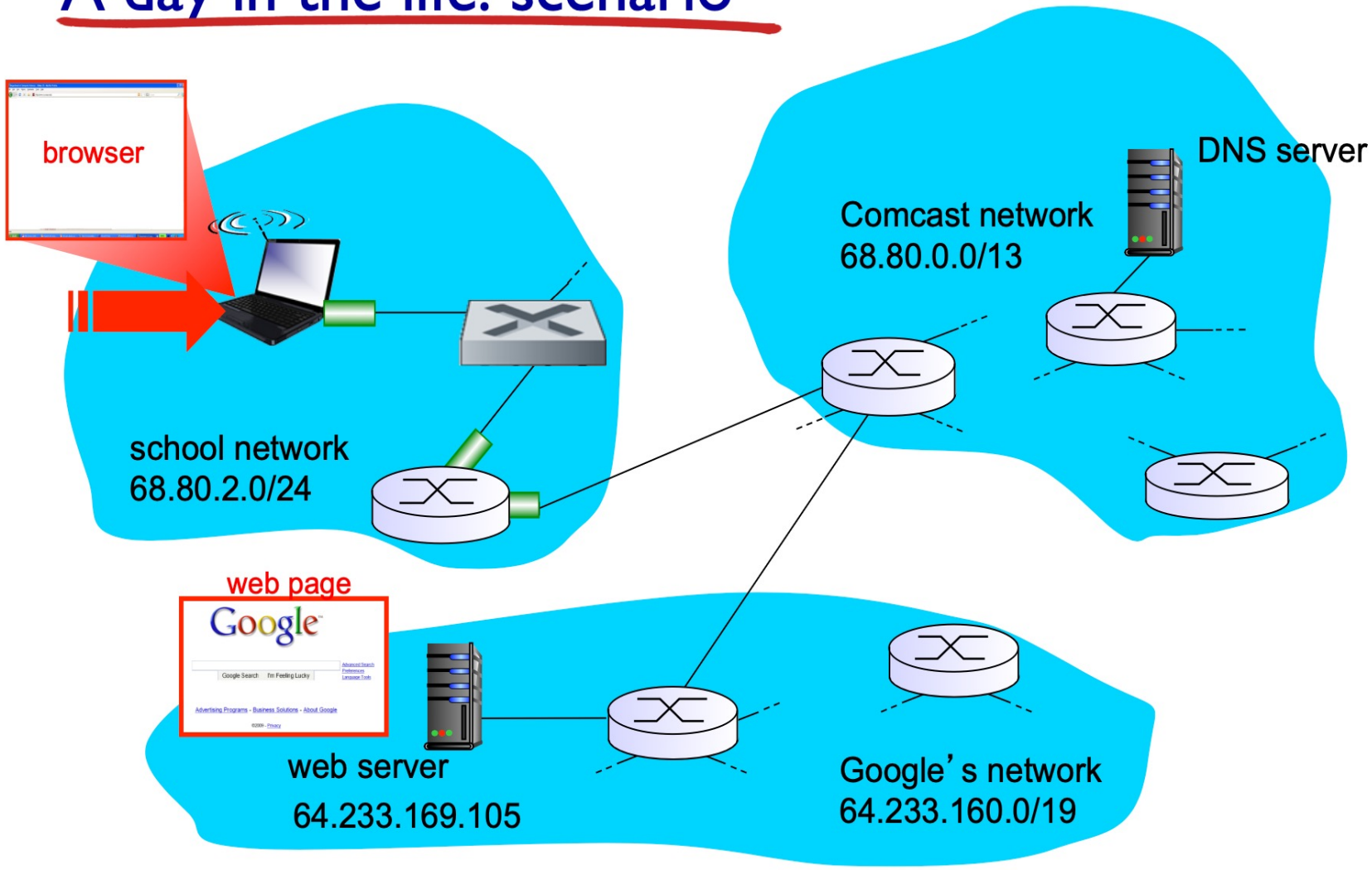
Link Layer and LANs
(Web Request)

## *Synthesis:* a day in the life of a web request

- journey down protocol stack complete!
  - application, transport, network, link
- putting-it-all-together: synthesis!
  - *goal:* identify, review, understand protocols (at all layers) involved in seemingly simple scenario: requesting www page
  - *scenario:* student attaches laptop to campus network, requests/receives www.google.com

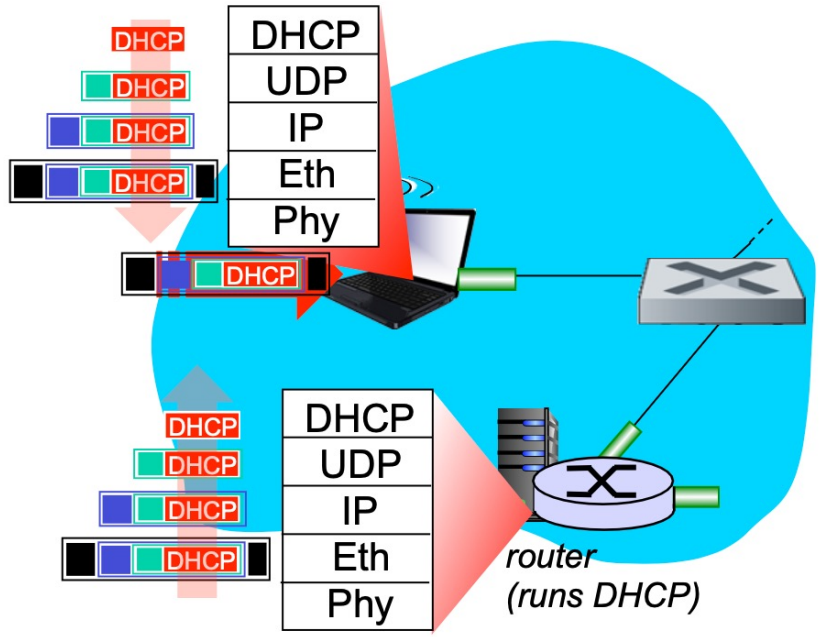
# Link Layer and LANs (Web Request)

## A day in the life: scenario



Link Layer and LANs
(Web Request)

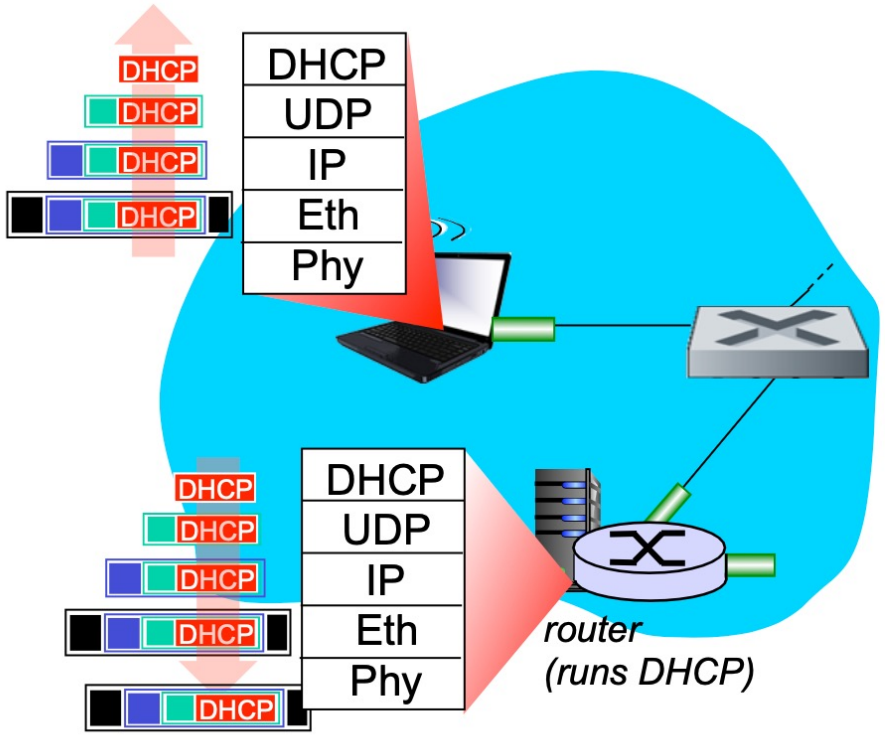
# A day in the life… connecting to the Internet




- connecting laptop needs to get its own IP address, addr of first-hop router, addr of DNS server: use *DHCP*
- DHCP request *encapsulated* in *UDP*, encapsulated in *IP*, encapsulated in *802.3* Ethernet
- Ethernet frame *broadcast* (dest: FFFFFFFFFFFF) on LAN, received at router running *DHCP* server
- Ethernet *demuxed* to IP demuxed, UDP demuxed to DHCP

Link Layer and LANs
(Web Request)

# A day in the life… connecting to the Internet

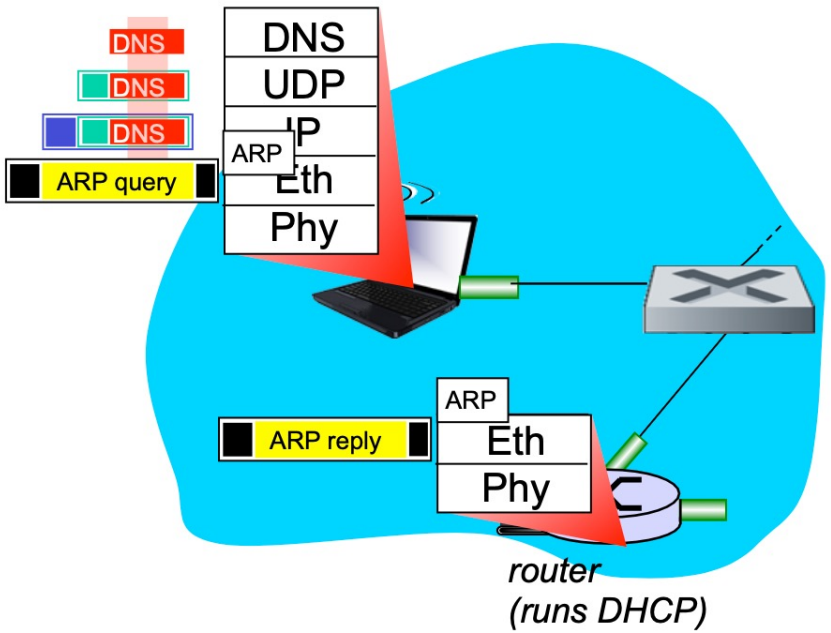



- DHCP server formulates *DHCP ACK* containing client's IP address, IP address of first-hop router for client, name & IP address of DNS server
- encapsulation at DHCP server, frame forwarded (*switch learning*) through LAN, demultiplexing at client
- DHCP client receives DHCP ACK reply

*Client now has IP address, knows name & addr of DNS server, IP address of its first-hop router*

Link Layer and LANs
(Web Request)

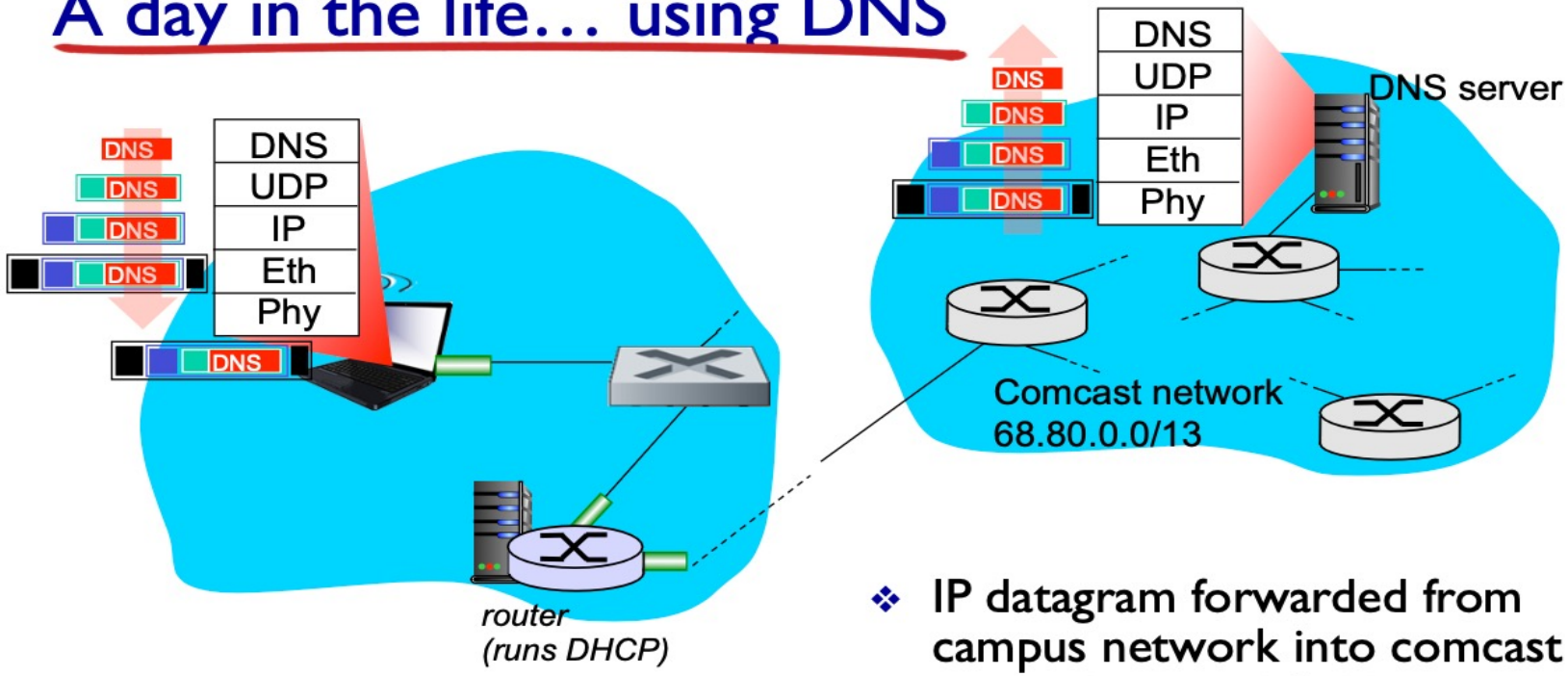
# A day in the life… ARP (before DNS, before HTTP)




- before sending *HTTP* request, need IP address of www.google.com: *DNS*
- DNS query created, encapsulated in UDP, encapsulated in IP, encapsulated in Eth. To send frame to router, need MAC address of router interface: *ARP*
- *ARP query* broadcast, received by router, which replies with *ARP reply* giving MAC address of router interface
- client now knows MAC address of first hop router, so can now send frame containing DNS query

# Link Layer and LANs (Web Request)

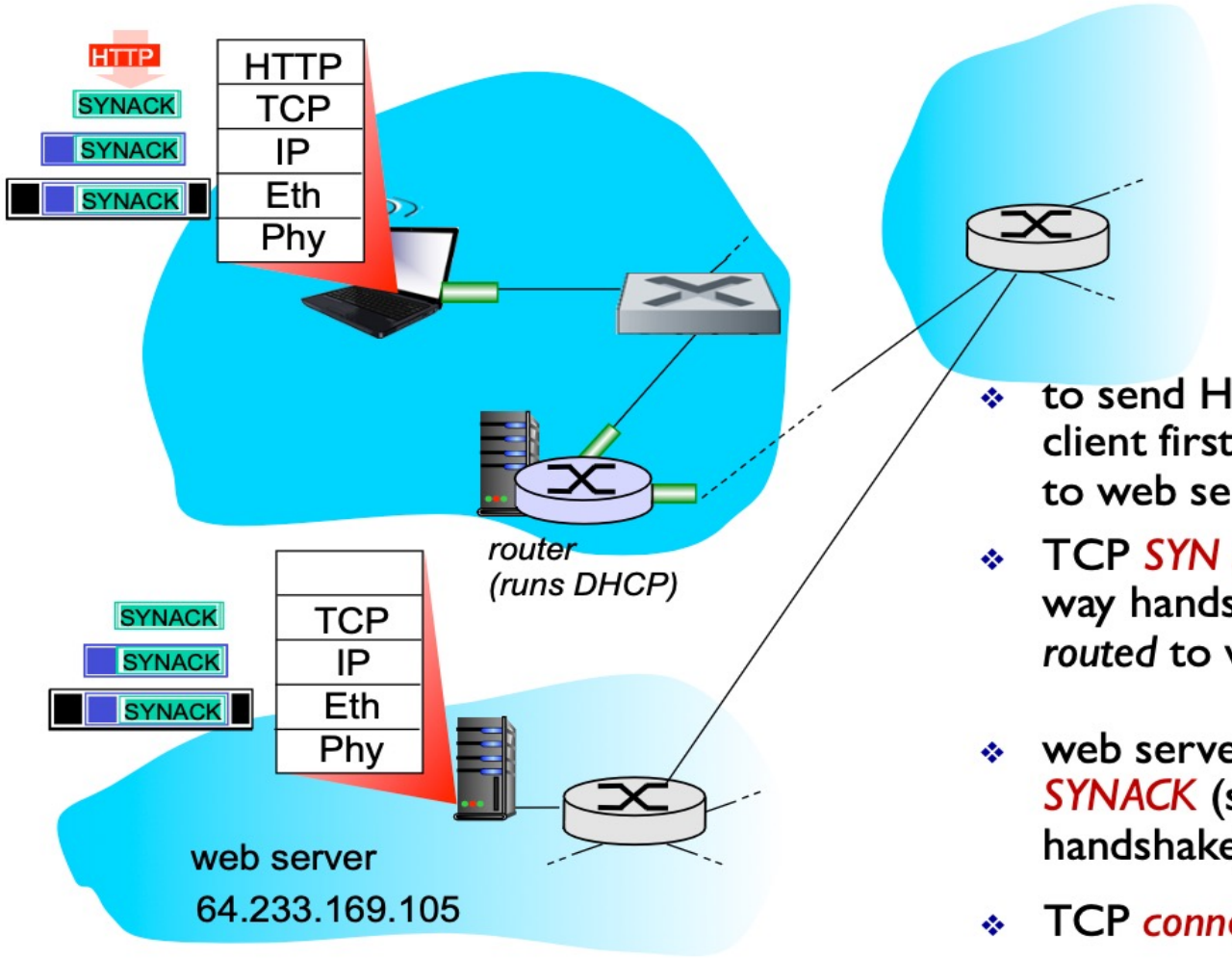
## A day in the life… using DNS




- IP datagram containing DNS query forwarded via LAN switch from client to 1st hop router
- IP datagram forwarded from campus network into comcast network, routed (tables created by *RIP, OSPF, IS-IS* and/or *BGP* routing protocols) to DNS server
- demux'ed to DNS server
- DNS server replies to client with IP address of www.google.com

# Link Layer and LANs (Web Request)

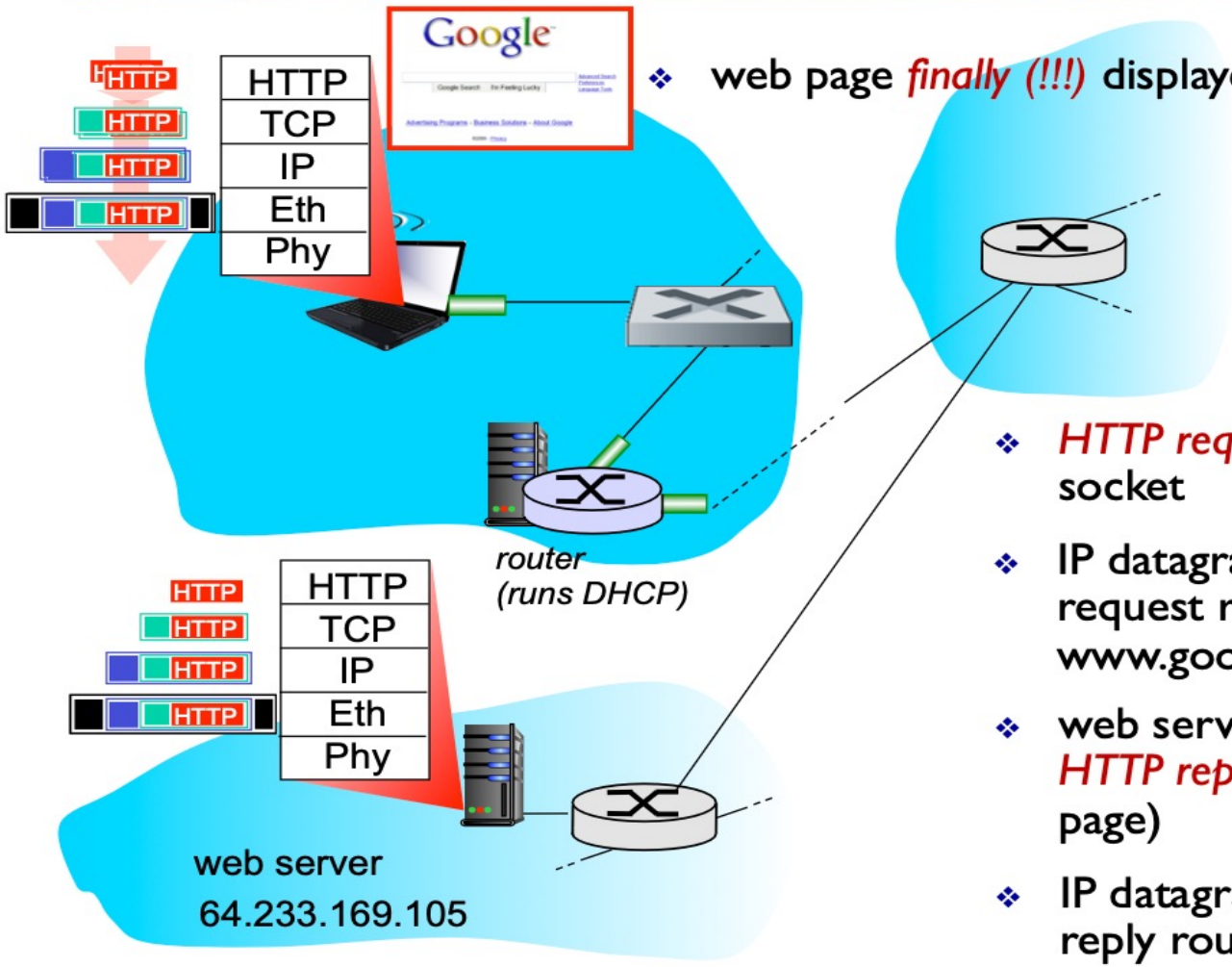
## A day in the life…TCP connection carrying HTTP




- to send HTTP request, client first opens *TCP socket* to web server
- TCP *SYN segment* (step 1 in 3-way handshake) *inter-domain routed* to web server
- web server responds with *TCP SYNACK* (step 2 in 3-way handshake)
- TCP *connection established!*

# Link Layer and LANs (Web Request)

## A day in the life… HTTP request/reply




- web page *finally (!!!)* displayed
- *HTTP request* sent into TCP socket
- IP datagram containing HTTP request routed to www.google.com
- web server responds with *HTTP reply* (containing web page)
- IP datagram containing HTTP reply routed back to client

Link Layer and LANs
(Web Request)

# Chapter 5: Summary

- principles behind data link layer services:
  - error detection, correction
  - sharing a broadcast channel: multiple access
  - link layer addressing
- instantiation and implementation of various link layer technologies
  - Ethernet
  - switched LANS, VLANs
  - virtualized networks as a link layer: MPLS
- synthesis: a day in the life of a web request