

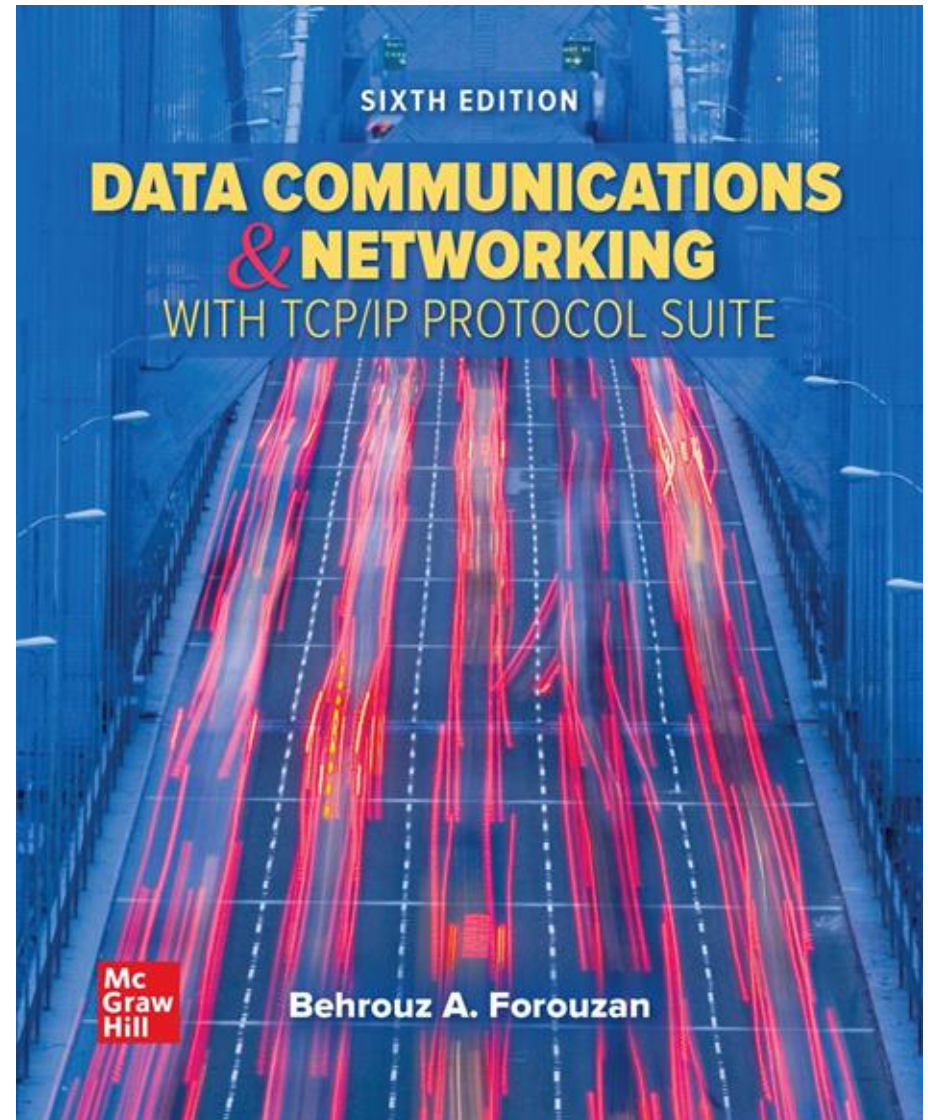
Sun 25 Jan

## Chapter 03

### Data-Link Layer

---

- Data Communications and Networking, With TCP/IP protocol suite Sixth Edition
- Behrouz A. Forouzan





# Chapter Outline

---

*RANDOM ACCESS*

*CONTROLLED ACCESS*

*CHANNELIZATION*

# Multiple Access

- **Broadcast (shared) link** consists of multiple sending and receiving nodes connected to or use a single shared link

## Shared links Examples



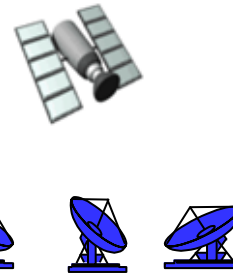
shared wire (e.g.,  
cabled Ethernet)

medium: wire  
line: crossover  
wiring: straight-over



shared RF  
(e.g., 802.11 WiFi)

air



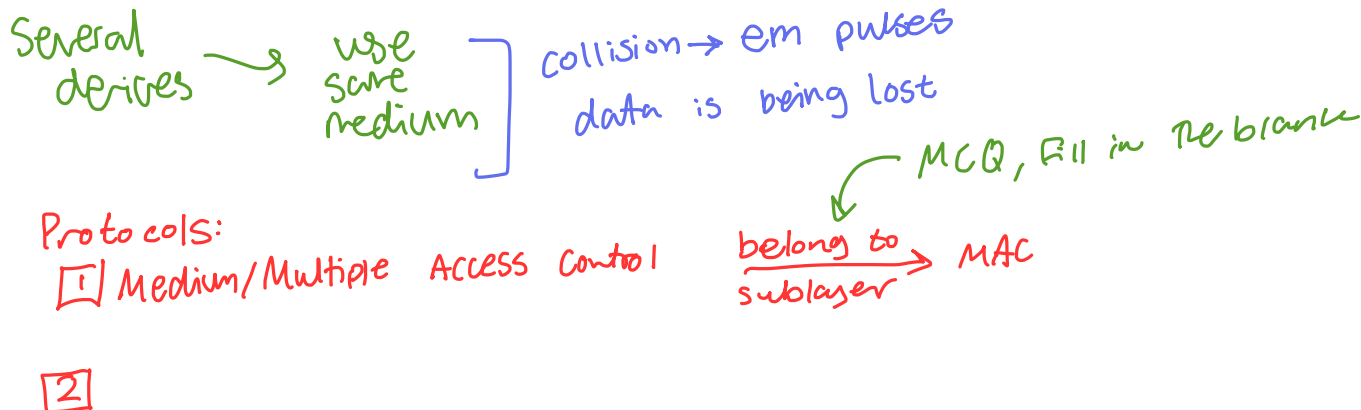
shared RF  
(satellite)



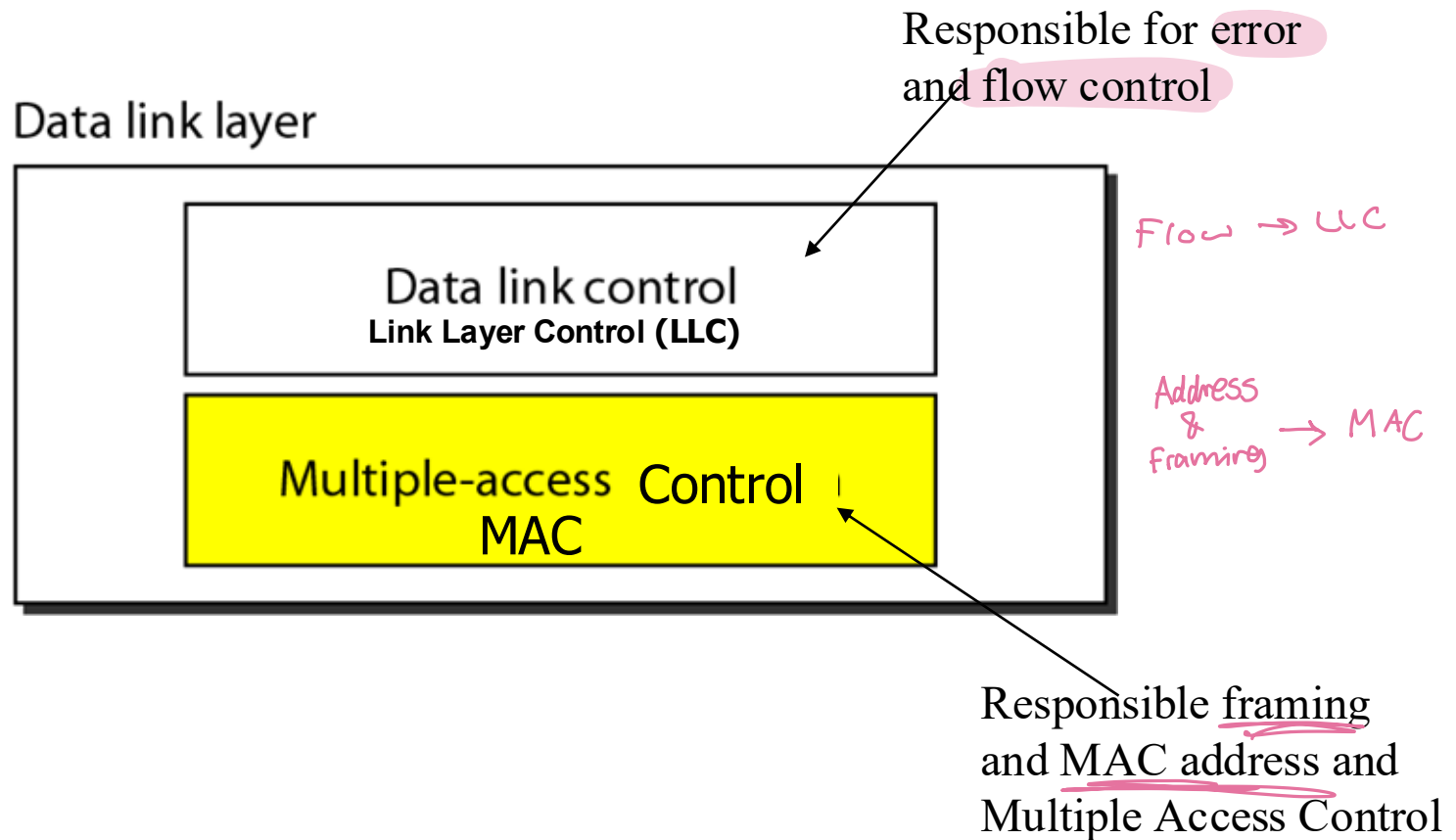
shared radio: 4G/5G

# Multiple Access

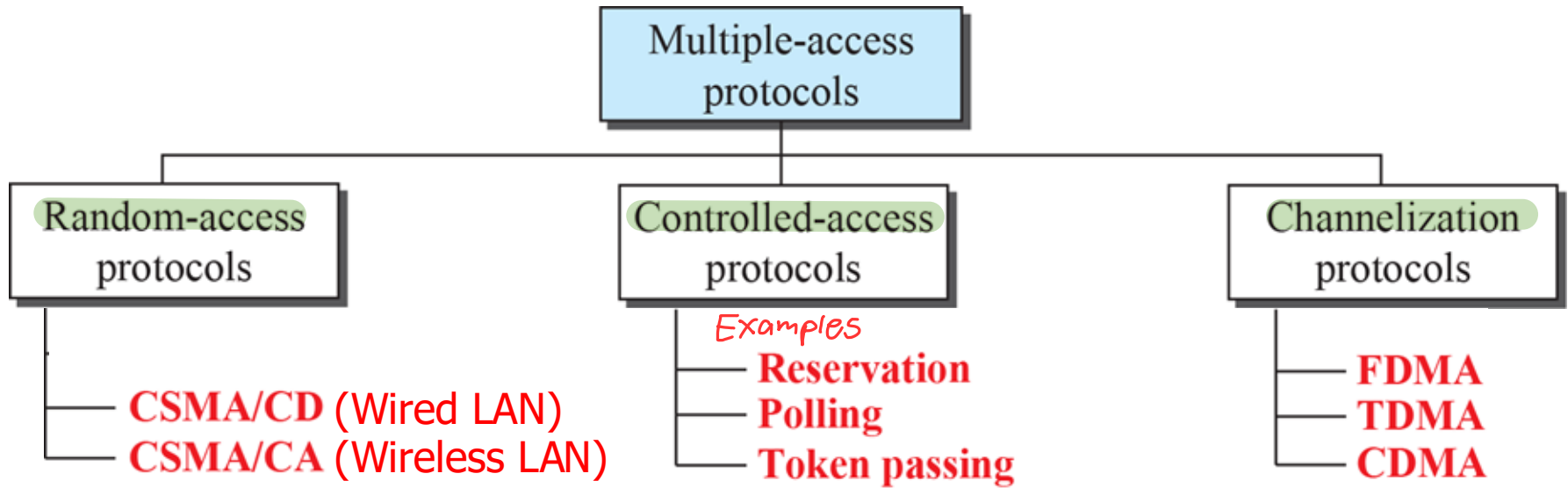
- **Problem:** When two or more nodes transmit at the same time, their frames will collide and the link bandwidth is **wasted** during collision
  - How to coordinate the access of multiple sending/receiving nodes to the shared link???
- **Solution:** We need a protocol to coordinate the transmission of the active nodes
- These protocols are called **Medium or Multiple Access Control (MAC) Protocols** belong to a **sublayer** of the data link layer called **MAC (Medium Access Control)**



**Figure** *Data link layer divided into two functionality-oriented sublayers*



**Figure :** *Taxonomy of multiple-access protocols*



# Random Access

All devices have medium  
all try to access at same time  
Info will collide

## ■ Random Access Protocols:

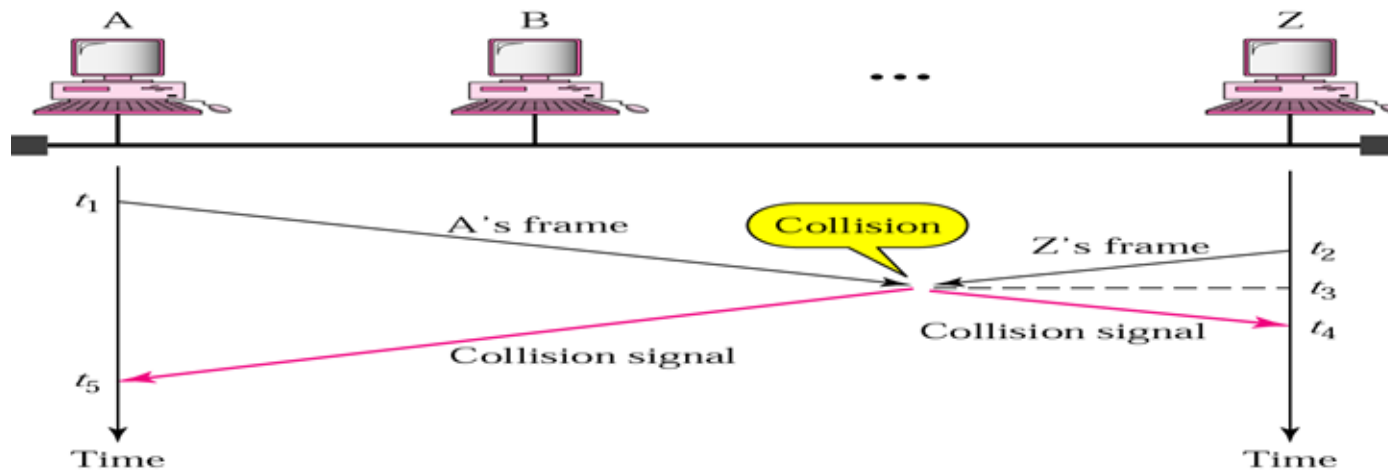
- No station is superior to another station and none is assigned the control over another.
- A station with a frame to be transmitted **can use the link directly based** on a procedure defined by the protocol to make a decision on whether or not to send.

All devices know who will send  
Check if medium, idle or busy

Devices check at same time, both find empty, both send  
Then problem happens

## Random Access – Carrier Sense Multiple Access (CSMA)

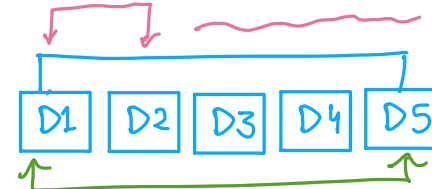
- Based on the fact that in LAN propagation time is **very small**
- □ If a frame was sent by a station, All stations **knows immediately** so they can **wait before start sending**
  - □ A station with frames to be sent, should **sense the medium** for the presence of another transmission (carrier) before it starts its own transmission
- This can **reduce** the possibility of collision but it cannot eliminate it.
  - Collision can only happen when more than one station begin transmitting within a short time (the **propagation time period**)



# Random Access – Carrier Sense Multiple Access (CSMA)

- Vulnerable (Unsafe) time for CSMA is the **maximum propagation time**

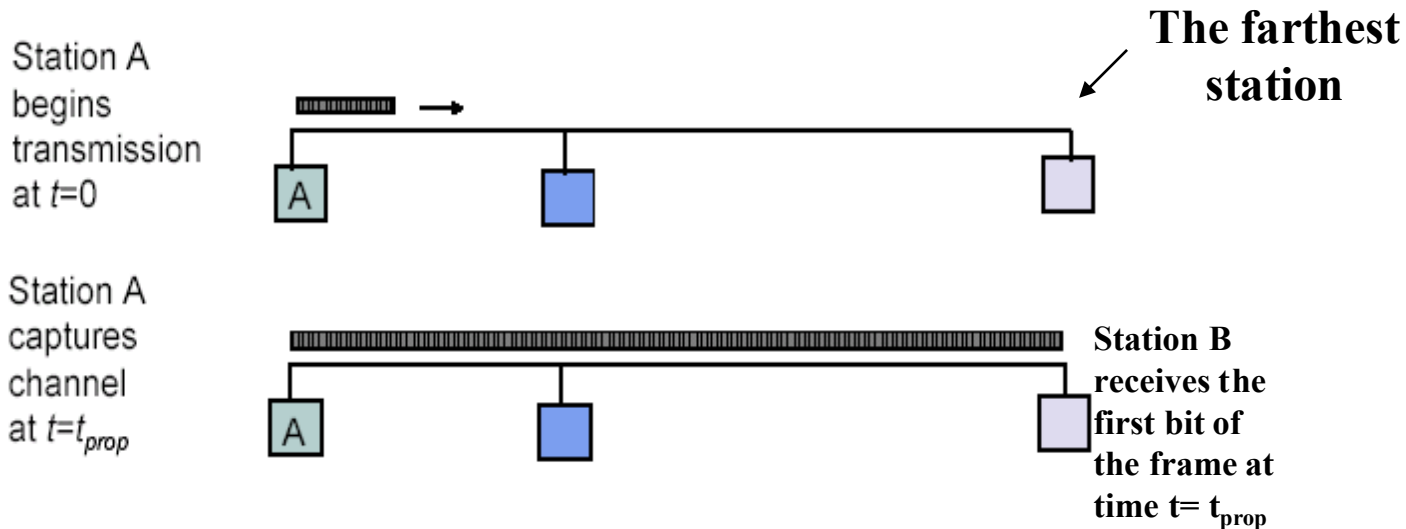
*inefficient so  
must find solution  
TBC....*



*max prop time: time it takes  
to send to  
2 furthest points*

## **What is Maximum Propagation Time?**

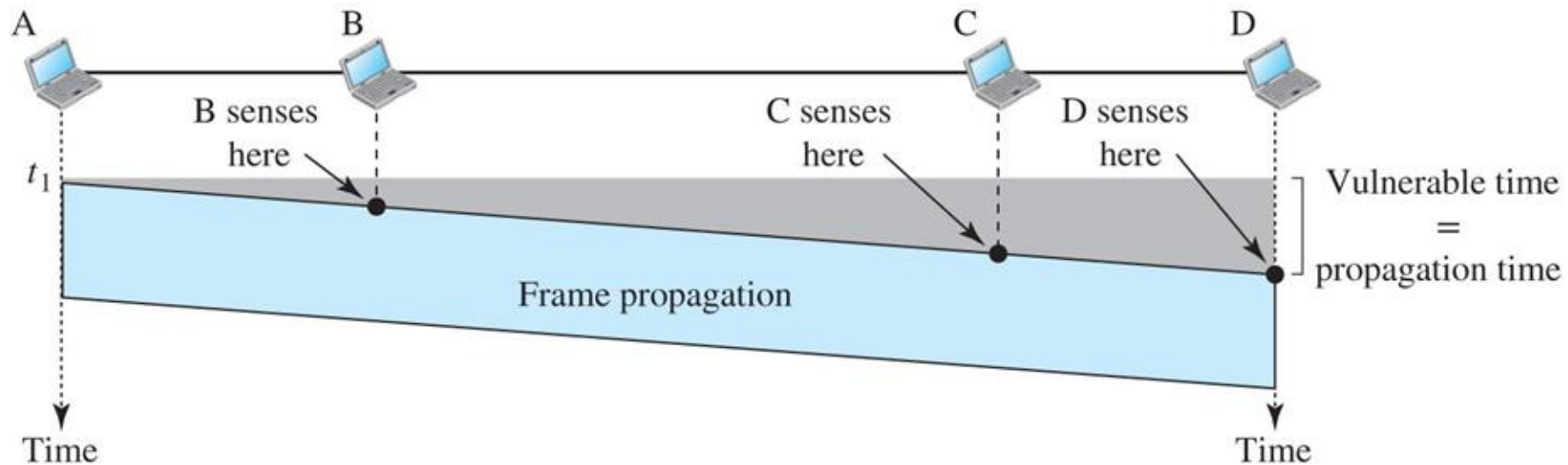
- **Maximum propagation delay ( $t_{prop}$ ):** time it takes for a bit of a frame to travel between the **two most widely** separated stations.

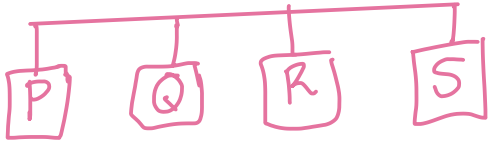


# Random Access – Carrier Sense Multiple Access (CSMA)

*Should know this, since could ask*

- The longer the propagation delay, the worse the performance of the protocol because of the above case.





$$\begin{aligned}
 m & \times 10^{-3} \\
 \mu & \times 10^{-6} \\
 n & \times 10^9 \\
 p & \times 10^{-12}
 \end{aligned}$$

time

A local network uses CSMA with four computers (P, Q, R, S) connected in a shared medium.

- one-way propagation delay between any two computers is  $3\mu s$

- At time  $t = 0ms$ , computer P starts transmitting a data frame

- At time  $t = 1ms$ , computer Q wants to <sup>transmit</sup>

• determine if the medium is busy or idle

force

Will there be a collision <sup>when</sup> and how to avoid it if there is?

propagation delay =  $3\mu s$

Busy since P is still sending

There will be no collision since Q hasn't sent yet.

avoided by checking if its idle

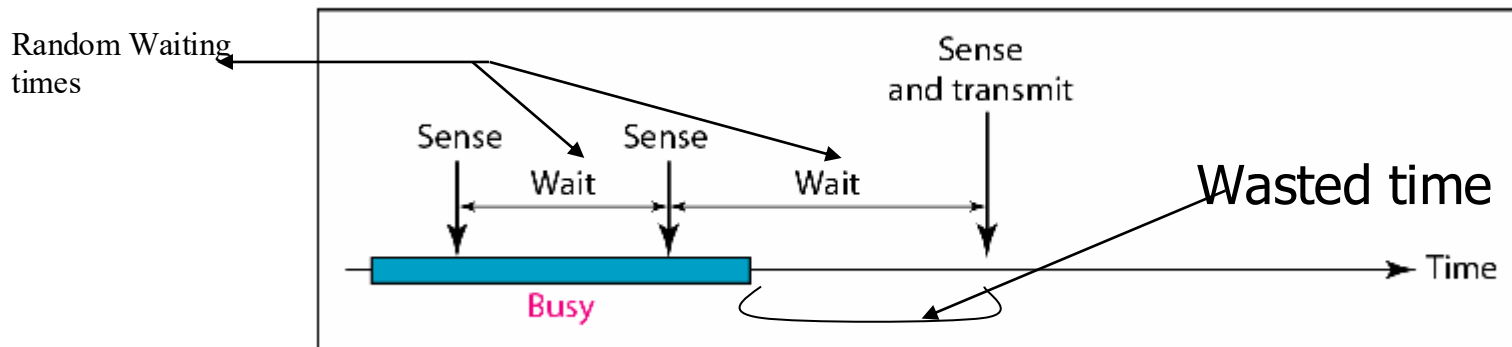
# Types of CSMA Protocols

Different CSMA protocols that determine:

- What a station should do when the medium is **idle**?
  - What a station should do when the medium is **busy**?
1. Non-Persistent CSMA
  2. 1-Persistent CSMA
  3. p-Persistent CSMA

# Nonpersistent CSMA

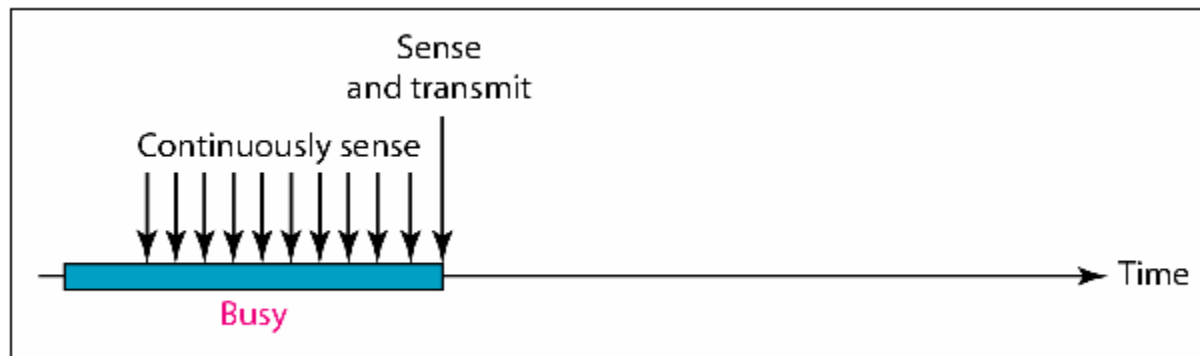
- A station with frames to be sent, should sense the medium
  1. If medium is idle, **transmit**; otherwise, go to 2
  2. If medium is busy, (**backoff**) wait a **random amount of time** and repeat 1
- Non-persistent Stations are **deferential (respect others)**
- Performance:
  - Random delays reduces probability of collisions because two stations with data to be transmitted will wait for different amount of times.
  - Bandwidth is **wasted** if waiting time (backoff) is large because medium will remain idle following end of transmission even if one or more stations have frames to send



b. Nonpersistent

# 1-persistent CSMA

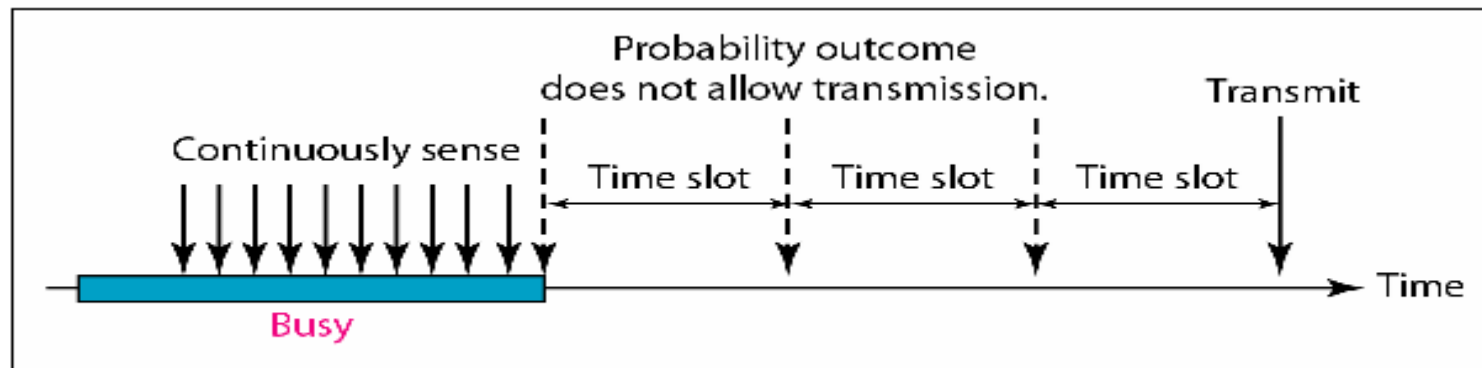
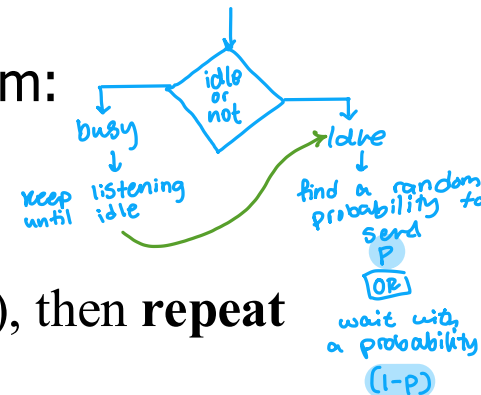
- To avoid idle channel time, 1-persistent protocol used
- Station wishing to transmit listens to the medium:
  1. If medium idle, **transmit** immediately;
  2. If medium busy, **continuously listen** until medium becomes idle; then transmit immediately with probability 1
- Performance
  - 1-persistent stations are selfish *"I want to go" and fight*
  - If two or more stations becomes ready at the same time, **collision guaranteed**



a. 1-persistent

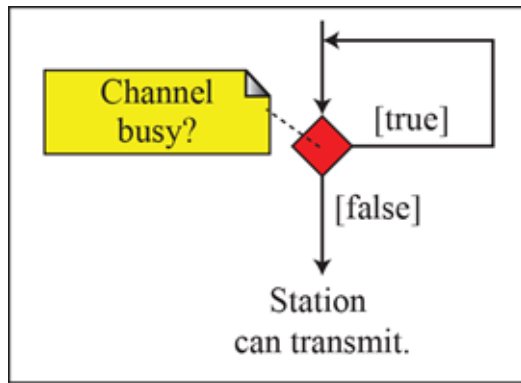
# P-persistent CSMA

- Time is divided to slots where each Time unit (slot) typically equals **maximum propagation delay**
- Station wishing to transmit listens to the medium:
  - If medium **idle**,
    - transmit with probability ( $p$ ), **OR**
    - wait **one time unit (slot)** with probability ( $1 - p$ ), then **repeat listening to the medium**.
  - If medium **busy**, **continuously listen until idle** and repeat step 1
  - Performance
    - Reduces the possibility of collisions like **nonpersistent**
    - Reduces channel idle time like **1-persistent**



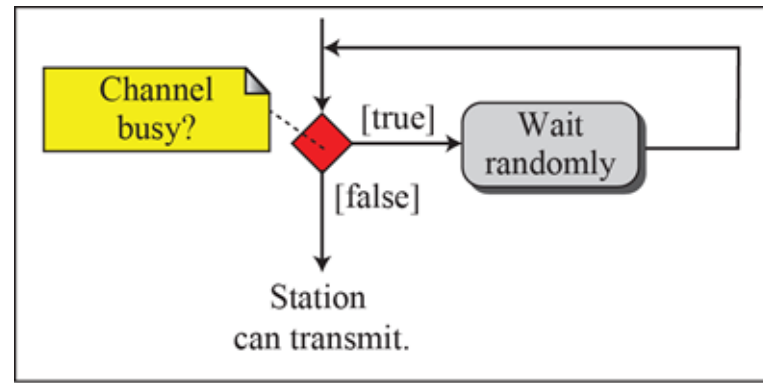
c. p-persistent

**Figure :** Flow diagram for three persistence methods

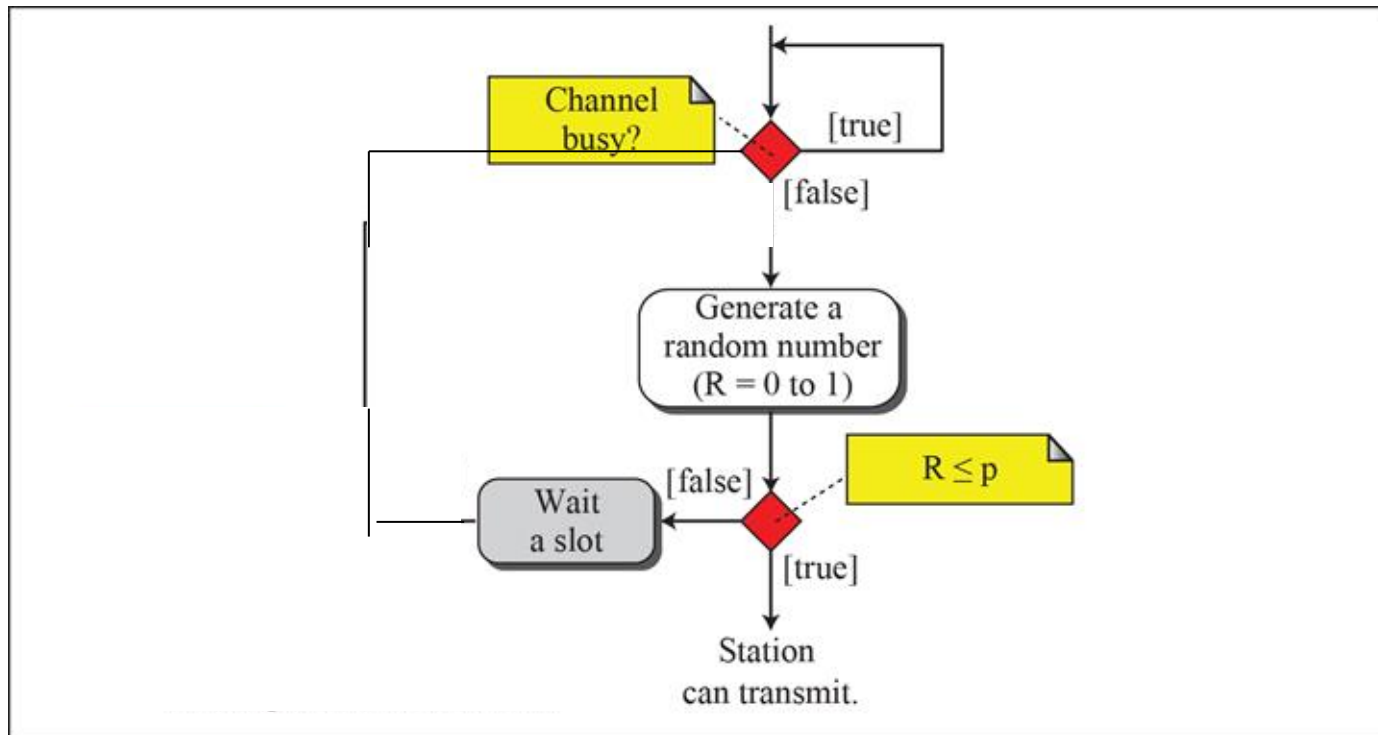


a. 1-persistent

*I think it means*



b. Nonpersistent

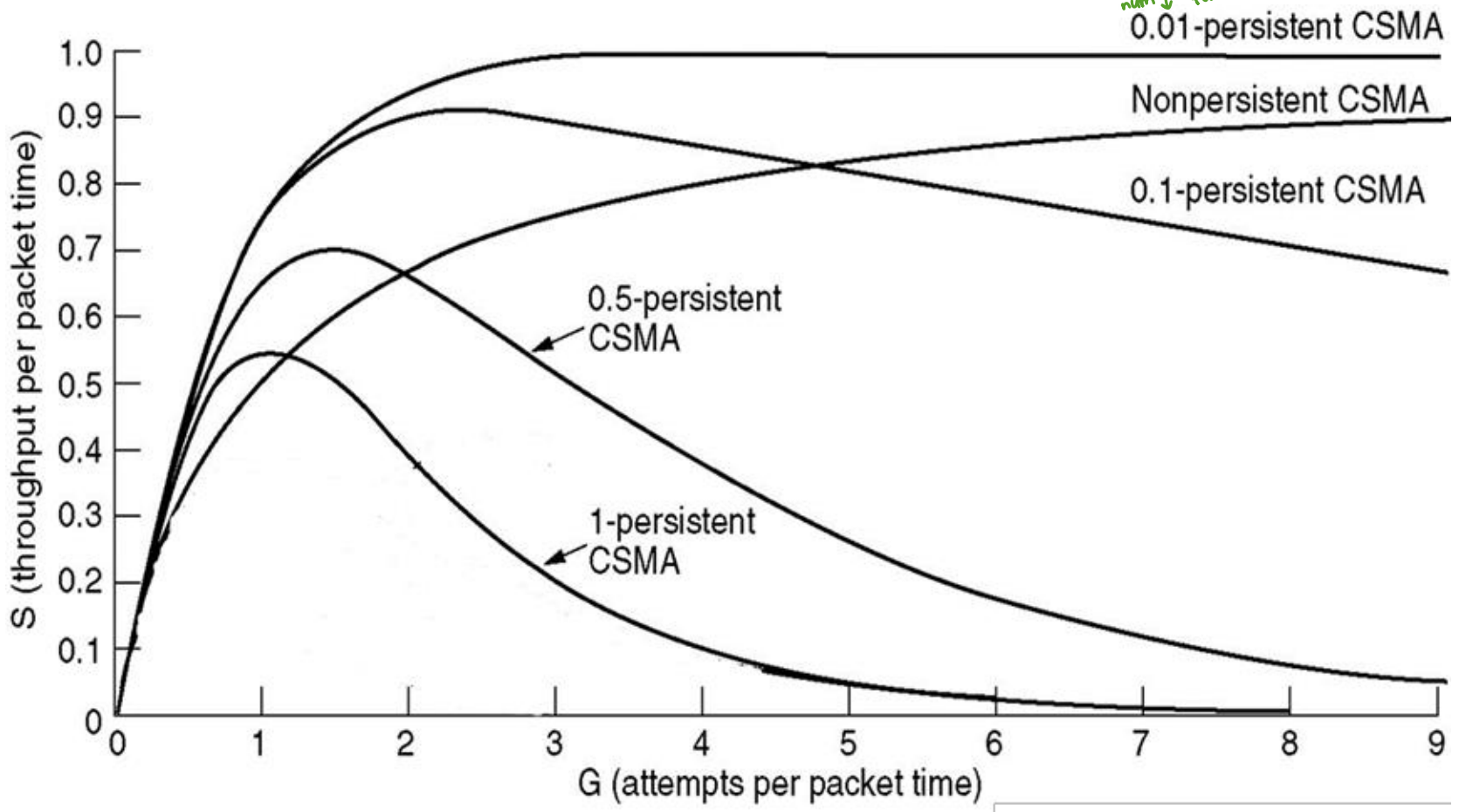


c. p-persistent

MUST know → higher probability what's behavior

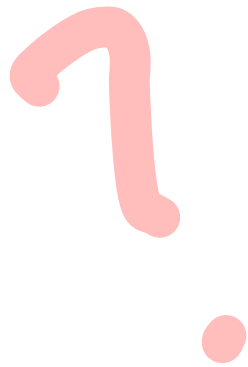
P-persistent  
can be diff  
num ↓ function become smoother

# Persistent and Nonpersistent CSMA



Comparison of the channel utilization versus load for various random access protocols.

	How does a device start the communication	For the second device accessing the network medium, what does it do? when $\begin{cases} \text{idle} \\ \text{busy} \end{cases}$ is found
Non persistent		Idle: Transmit Busy: Backoff, wait random time, go back to idle
1-persistent		
P-persistent		



In all three protocols,

- 1) Non-persistent
- 2) 1-persistent
- 3) p-persistent

- How does <sup>device</sup> it start the communication
- For the second device accessing the network medium, what does it do? when  $\begin{cases} \text{idle} \\ \text{busy} \end{cases}$  is found

# CSMA/CD (Collision Detection)

- *CSMA (all previous methods) has an inefficiency:*
  - If a collision has occurred, the channel is **unstable** until colliding packets have **been fully transmitted**
- *CSMA/CD (Carrier Sense Multiple Access with Collision Detection) overcomes this as follows:*
  - While transmitting, the sender is **listening to medium** for collisions.
  - Sender **stops transmission** if collision has occurred **reducing channel wastage** .

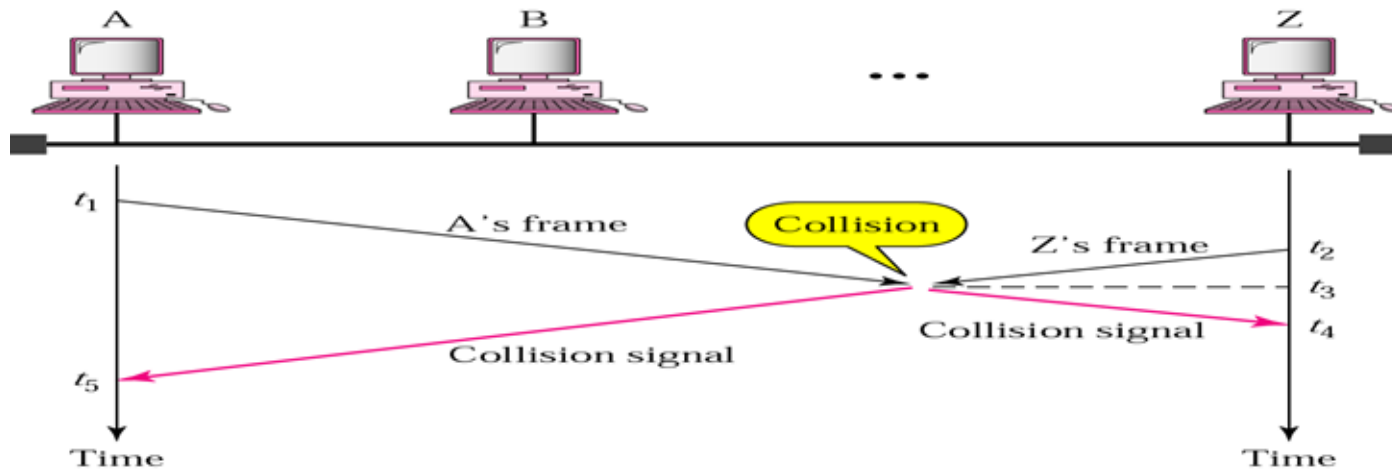
CSMA/CD is Widely used for in star topology if **Hub is used** (IEEE 802.3, **Ethernet**).

*irregularity*

*Stop when somethings wrong*

# How does a node detect a collision?

Transceiver: A node monitors the media while transmitting. If the observed power is more than transmitted power of its own signal, it means collision occurred



# CSMA/CD Protocol

↳ set of instructions

- Use one of the CSMA persistence algorithm (*non-persistent, 1-persistent, p-persistent*) for transmission
- If a collision is detected by a station during its transmission then it should do the following:
  - **Abort transmission** and
  - **Transmit a *jam signal* (48 bit)** to notify other stations of collision so that they will **discard the transmitted frame** also to make sure that the collision signal will stay until detected by the furthest station
  - After sending the *jam signal*, **backoff (wait) for a *random*** amount of time, then
  - Reapply the used CSMA persistent algorithm

Protocol basic idea:  
① Stop  
② cleaning  
③ removing

①

# Performance of Random Access Protocols

- Simple and easy to implement
- Decentralized (no central device that can fail and bring down the entire system)
- In low-traffic, packet transfer has low-delay
- However, in heavier traffic, packet delay has no limit.
- In some cases, a station may never have a chance to transfer its packet. (**unfair protocol**)
- A node that has frames to be transmitted can **transmit continuously** at the **full rate of channel (R)** if it is the **only node with frames**
- If (M) nodes want to transmit, many collisions can occur and the **maximum average throughput** for each node will **not be R/M**

Random access  
↳ will all have same time?  
No, some will wait a long time. even leading to some might not using the medium

1 device using medium, would have full capacity

Several devices, capacity will be divided

rate of channel  
number of nodes

could come as scenario and you extract

②

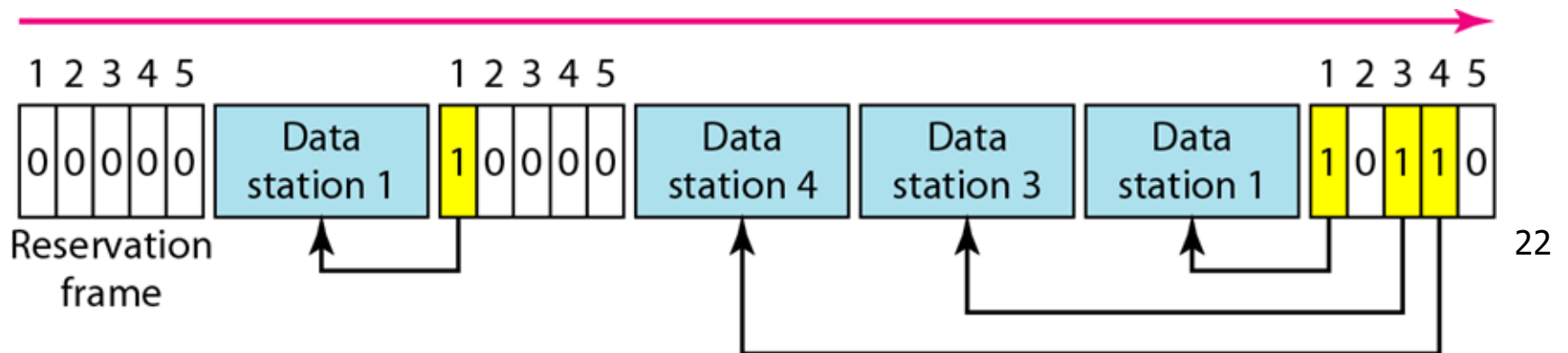
## 13.2 Controlled Access or Scheduling

Introduce rule, how each device access medium

- Provides **in order access** to shared medium so that every station has chance to transfer (**fair protocol**)
- It is fair protocol. A node is limited to throughput equal **R/M**
- *Eliminates* collision completely
- **Three methods** for controlled access:
  - Reservation
  - Polling
  - Token Passing

# 1-Reservation access method

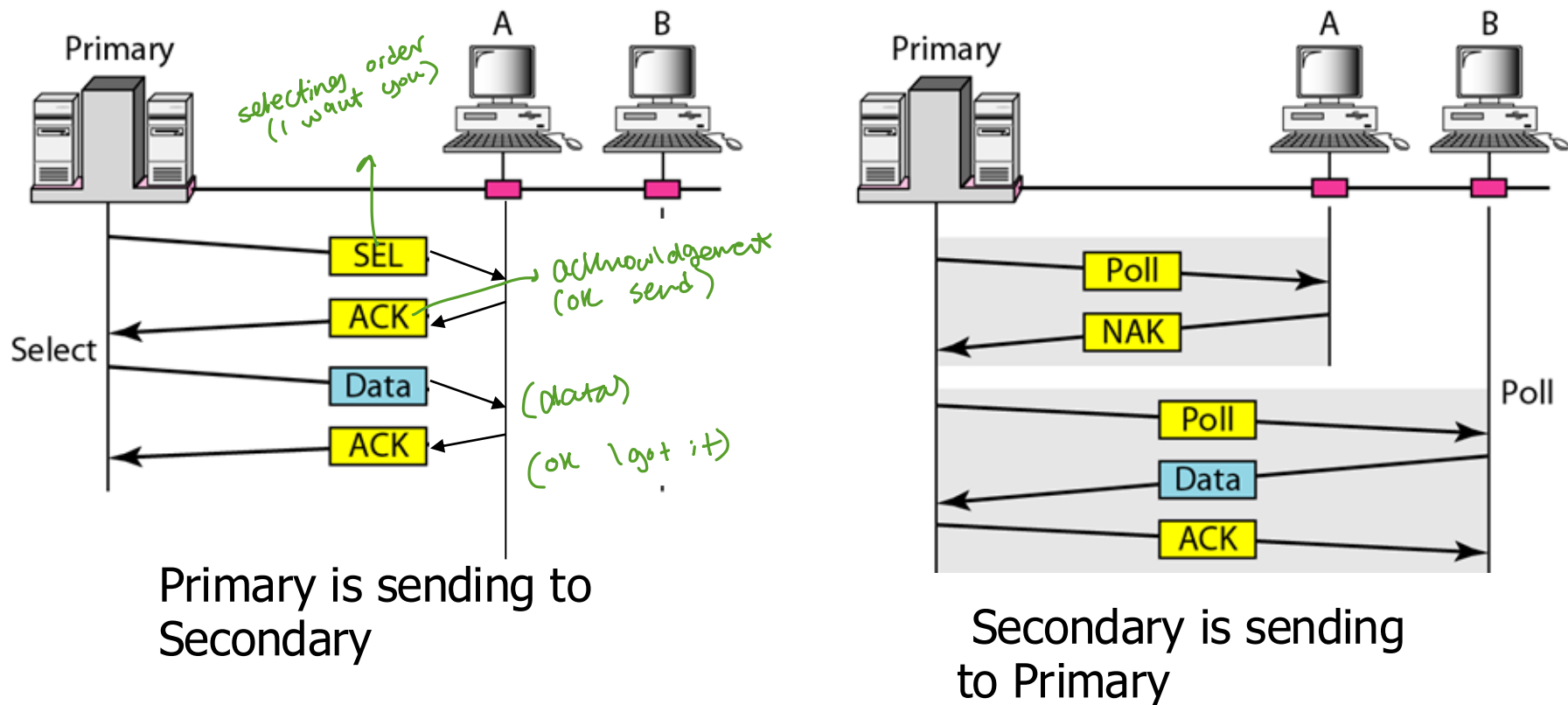
- Stations **take turns transmitting a single frame** at a **full rate ( R ) bps**
- Transmissions are organized into **variable length** cycles
- Each cycle **begins** with a reservation frame that consists of (N) minislots. One minislot for each of the N stations
- When a station needs to send a data frame, it makes a **reservation** in its own minislot.
- By listening to the reservation interval, every station knows which stations will transfer frames, and in **which order**.
- The stations that made reservations can send their data frames after the reservation frame.



## 2- Polling

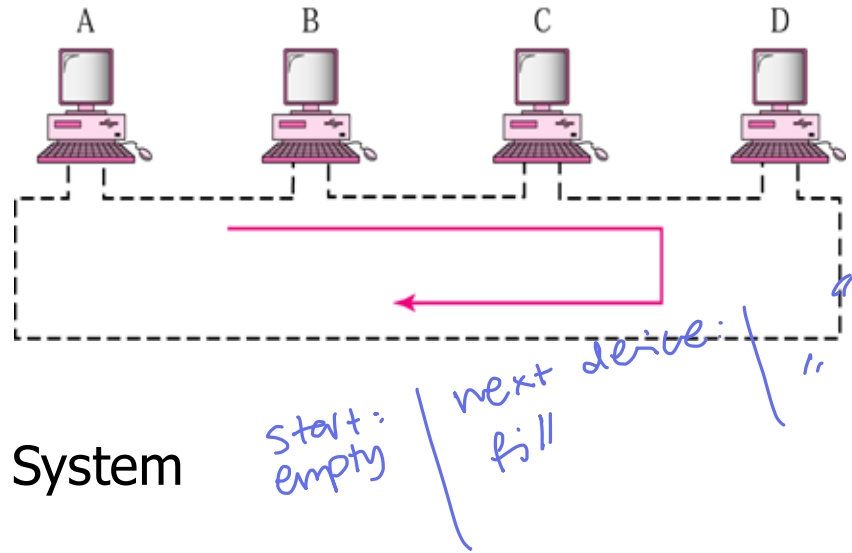
- Stations take turns accessing the medium
- Two models: **Centralized** and **distributed** polling
- **Centralized polling**
  - One device is assigned as **primary station** and the others as **secondary stations**
  - All data exchanges are done through the **primary**
  - When the **primary has a frame to send**, it sends a **select** frame that includes the address of the intended secondary
  - When the **primary is ready to receive** data it sends a **Poll** frame for each device to ask if it has data to send or not. If yes, **data** will be transmitted otherwise **NAK (Negative Acknowledgement)** is sent.
  - Polling can be done in order (Round-Robin) or based on predetermined order
- **Distributed polling**
  - **No primary and secondary**
  - Stations have a **known polling order** list which is made based on some protocol
  - **station with the highest priority** will have the access right first, then it passes the access right to the **next station (it will send a pulling message to the next station in the pulling list)**, which will pass the access right to the following next station, ... (**Reservation and Token Passing** are examples of **Distributed Polling**)

**Figure 12.19** *Select and poll functions in polling access method*



might or draw asked

### 3- Token-Passing network



Implements **Distributed Polling** System

There is a special frame rotating in the ring called **token**.

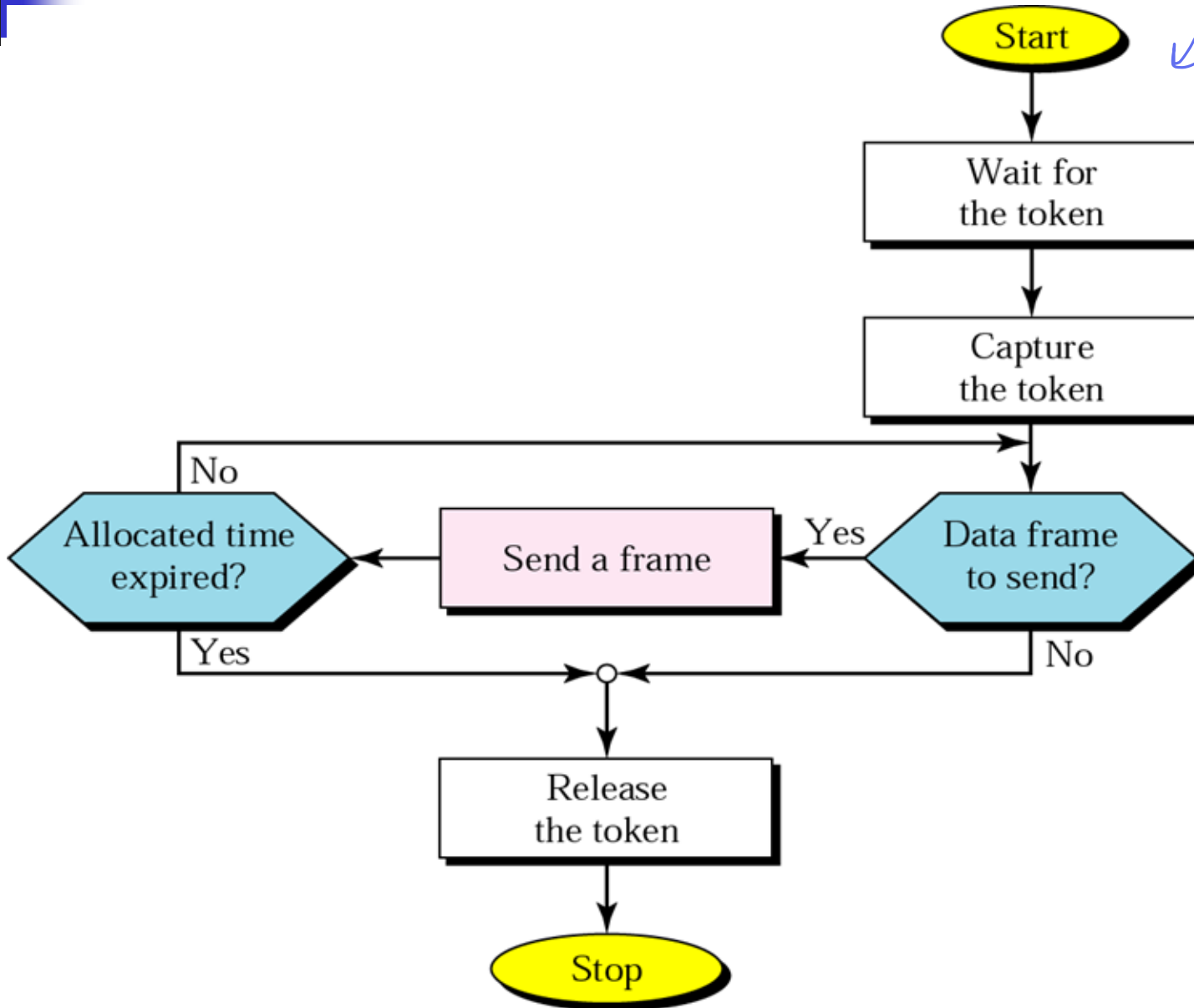
Any station that wants to send data has to wait for free token to arrive.

The station makes the token busy by inserting its data in the token and puts the destination address.

The destination station receive the data and change the token to free

Figure 13.13 Token-passing procedure

*when empty follows this*



## 12-3 CHANNELIZATION

***Channelization*** is a multiple-access method in which the available bandwidth of a link is shared in time, frequency, or through code, between different stations. In this section, we discuss three channelization protocols.

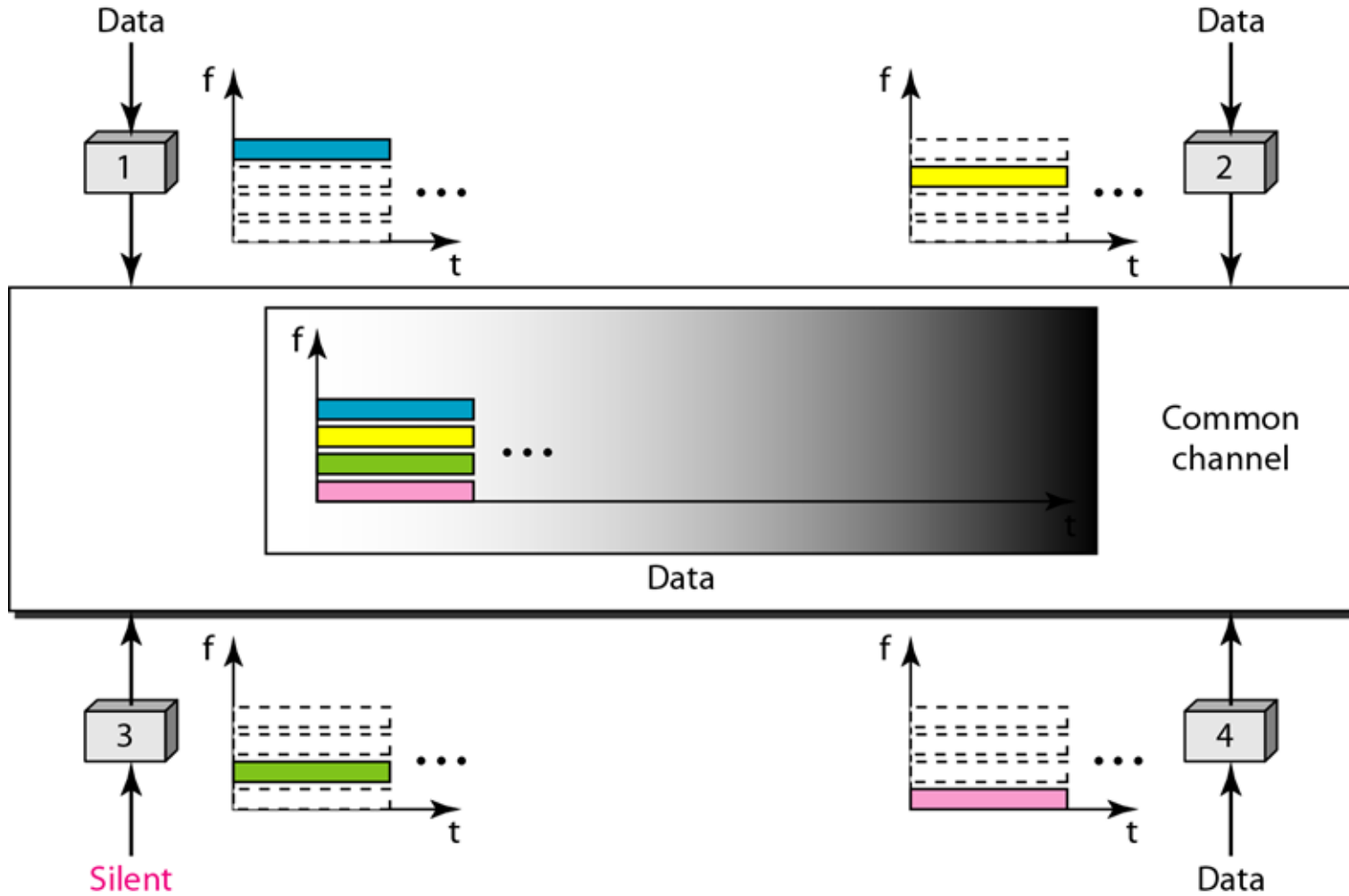
**Topics discussed in this section:**

**Frequency-Division Multiple Access (FDMA)**

**Time-Division Multiple Access (TDMA)**

**Code-Division Multiple Access (CDMA)**

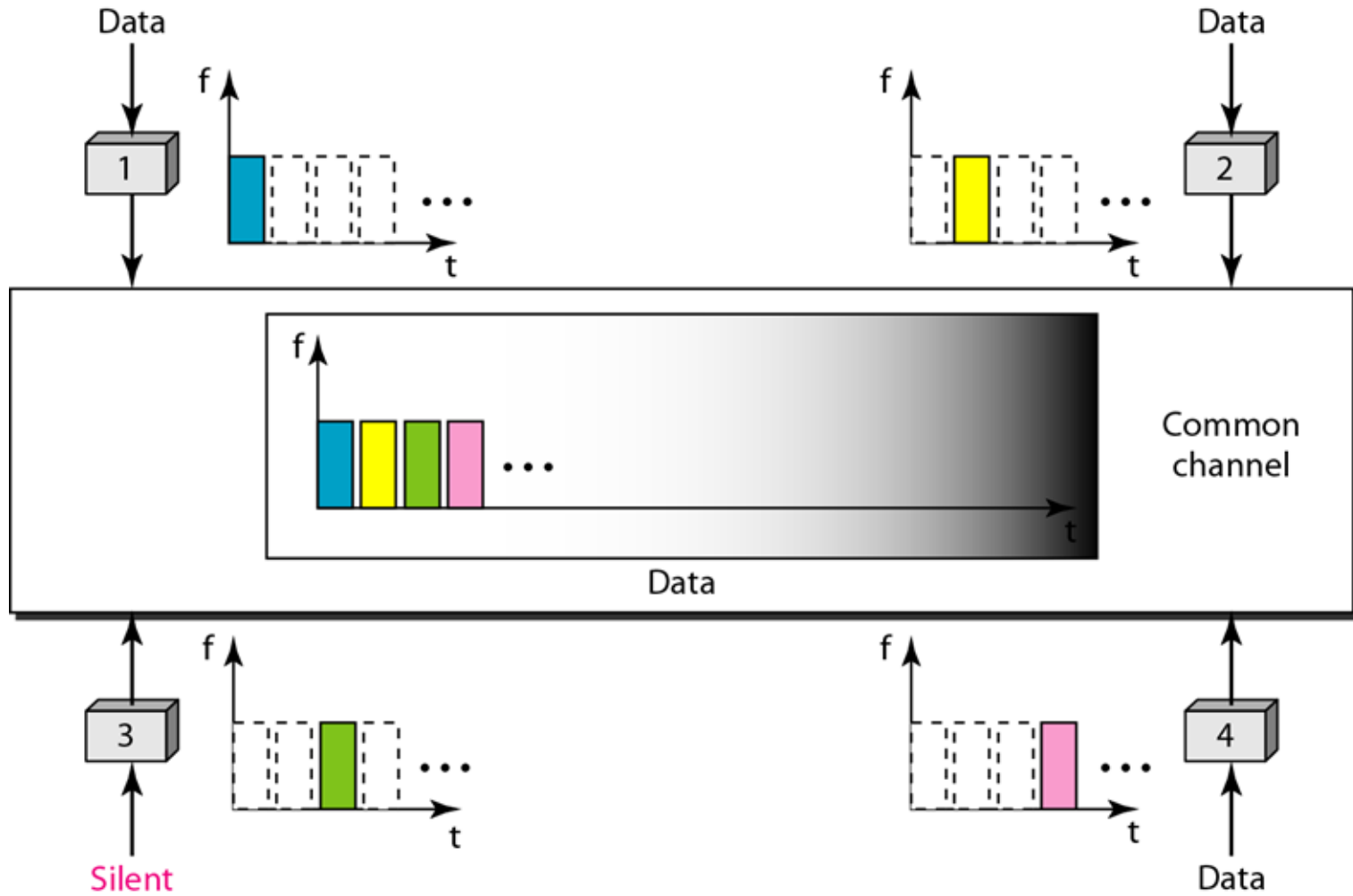
**Figure 12.21** *Frequency-division multiple access (FDMA)*



# 12-3 CHANNELIZATION - FDMA

- **FDMA:** Frequency Division Multiple Access:
  - Transmission medium is divided into **M** separate frequency bands
  - Each station transmits **continuously** on the assigned band at an average rate of **R/M**
  - A node is **limited** to an average throughput equal **R/M** (where M is number of nodes) even when it is **the only node with frame** to be sent

**Figure 12.22** *Time-division multiple access (TDMA)*



# 12-3 CHANNELIZATION - TDMA

- **TDMA: Time Division Multiple Access**
  - The entire bandwidth capacity is a **single channel** with its capacity shared **in time** between **M** stations
  - A node must **always wait for its turn** until its slot time arrives even when it is the **only node** with frames to send
  - A node is limited to an average throughput equal  **$R/M$**  (where M is number of nodes) even when it is the only node with frame to be sent

# 12-3 CHANNELIZATION - CDMA

- **CDMA: Code Division Multiple Access**
  - In CDMA, one channel carries all transmissions **simultaneously**
  - Each station codes its data signal by a specific codes before transmission
  - The stations receivers use these codes to recover the data for the desired station

Figure 12.23 Simple idea of communication with code

