University of Pittsburgh

# ECE 1150: Computer Networks

# Data Link Layer – Introduction & Medium Access Control


**Mai Abdelhakim, PhD**

ECE Department

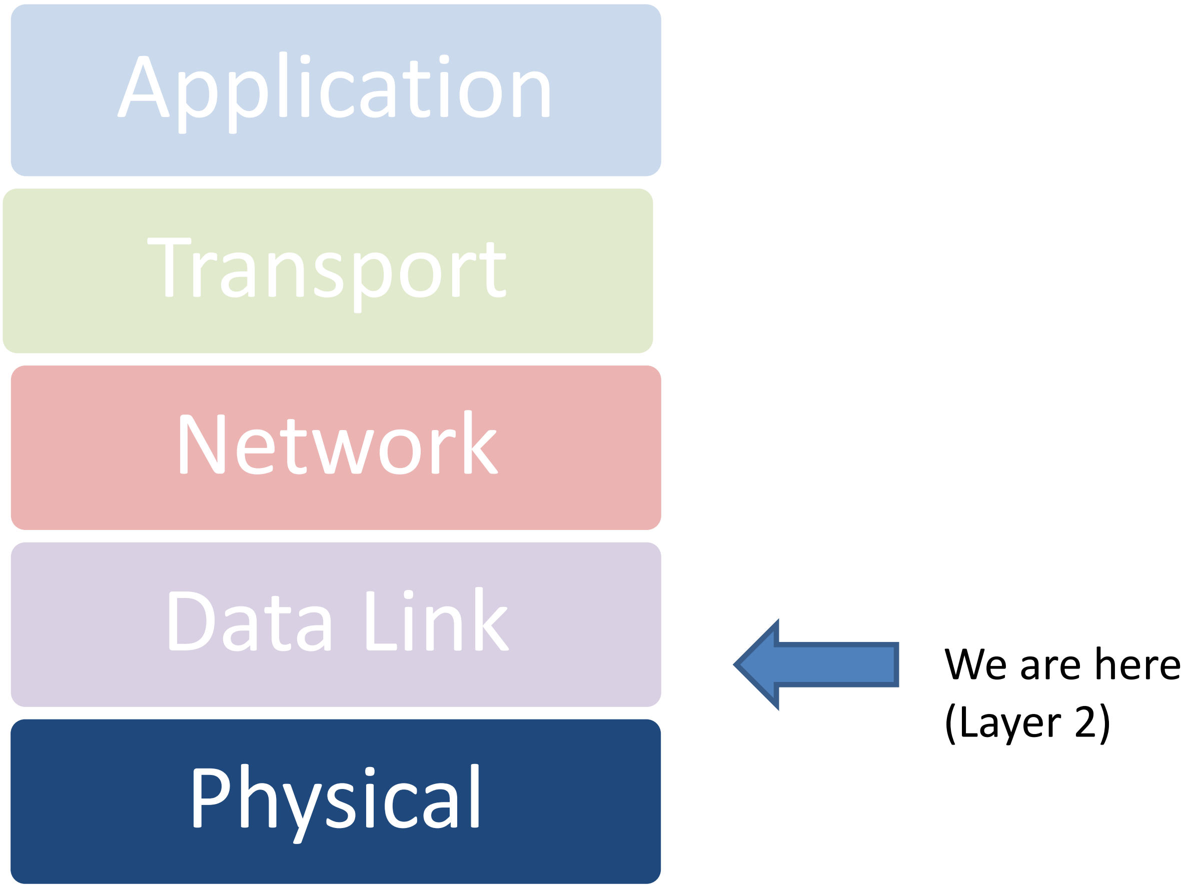
Swanson School of Engineering

University of Pittsburgh

# Previous Units

- Networks basics
- Layering and Protocols
- Communication basics (power, attenuation, capacity,..)
- Physical layer processing
- Multiplexing

# Context


Application
Transport
Network
Data Link
Physical
We are here
(Layer 2)

# Objectives of This Unit

- Explain the functions at the data link layer
- Describe the different medium access control protocols

- **Reading: Chapter 4, Section 4.2, Computer Networks, Tanenbaum et al.**

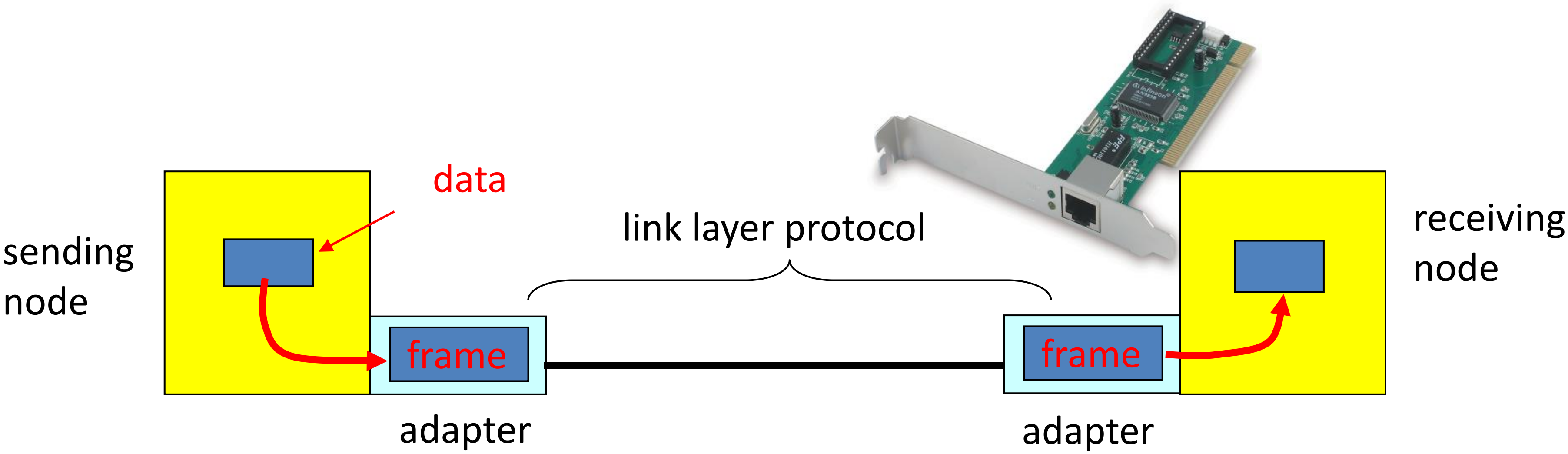
# Data-Link layer

- **Transfer data between individual links**

- **Ethernet** is the most common wired end-user **implementation of the data-link layer**

- **WiFi** is the most **common wireless link layer**

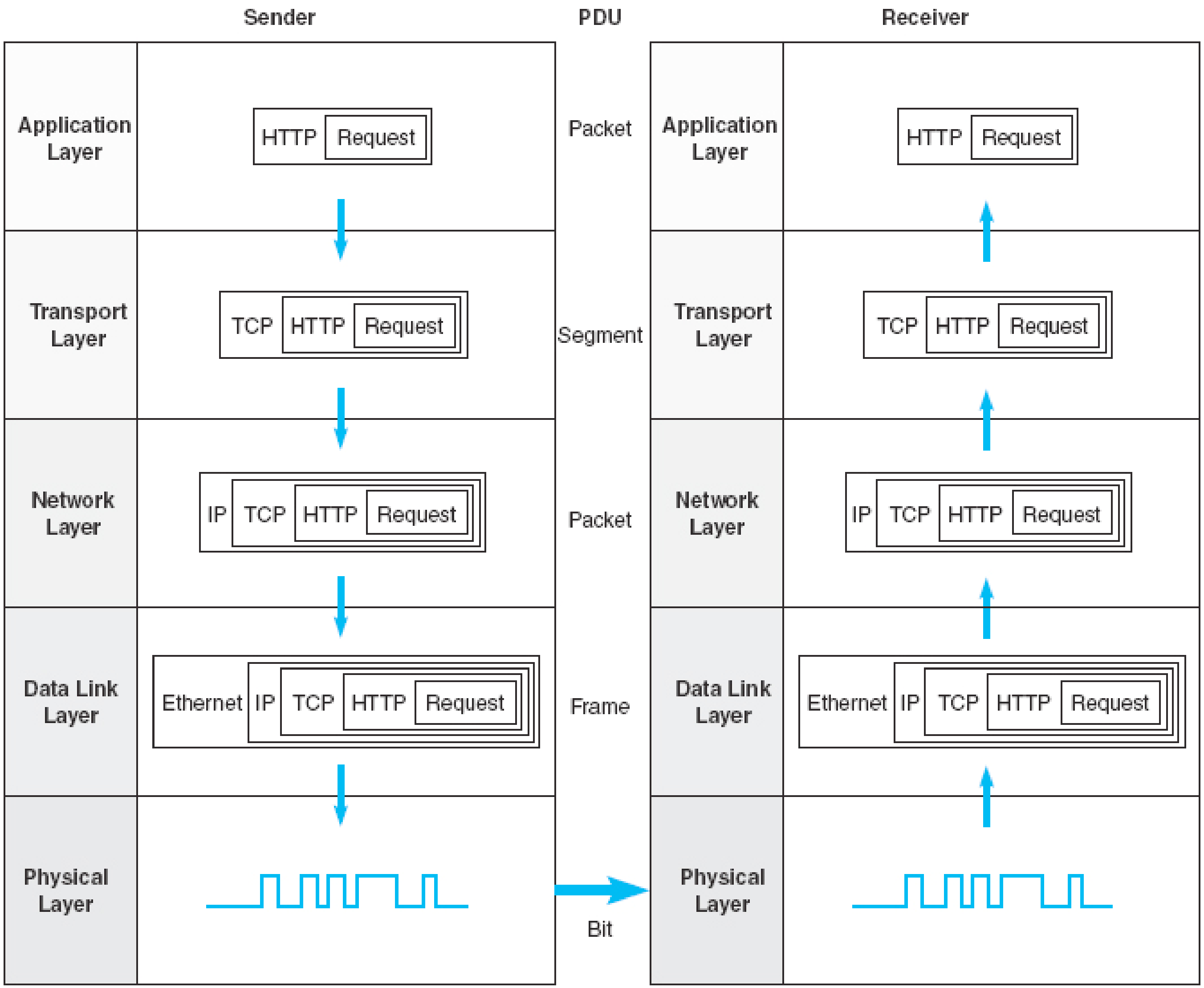
# Where is it implemented?




- Most link layer functions are implemented in hardware "adaptor" (aka Network Interface Card [NIC])
  - Ethernet card, 802.11 card

# Recall Encapsulation

Transport and application layers are implemented at end devices (end-to-end)



# Data Link Layer Functions

Also at the transport layer – "end-to-end"

- **Medium Access Control (MAC), if shared medium**
  - Control who uses channel
- **Addressing**
  - **MAC** address used **locally** on a link
- **Framing**
  - At transmitter, link layer encapsulate datagram into frame, **adding header** (e.g. address), **and trailer**
- **Reliable delivery between adjacent nodes**
  - Error Detection
    - Receiver detects presence of errors
  - Error Correction
    - Receiver asks for **retransmission** if error occurs
  - Flow Control
    - Pacing between adjacent sending and receiving nodes

# Types of links

- Point-to-point
  - Single sender at one side of the link and single receiver at the other end
- **Broadcast link**
  - Broadcast channel connecting multiple senders and receivers
  - E.g. WiFi

# Multiple Access Protocols

- How to **coordinate access of multiple sending and receiving nodes** sharing a **broadcast** medium?
  - **Broadcast: all devices can send and receive**
    - **Who gets to transmit?**

- **Multiple access protocols** regulate the transmission of devices into a single shared broadcast link

- **Multiple access protocols == Medium Access Control (MAC) protocols**

# Medium Access Control (MAC)

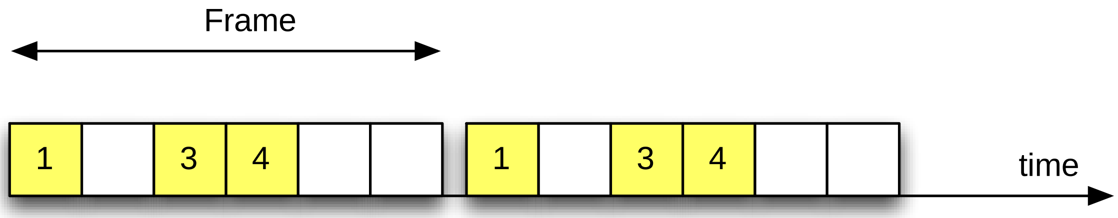
- MAC protocols needed when using a **single shared broadcast link**
- Two or more simultaneous transmissions by nodes can result in "interference"
  - **Collision** if node receives **two or more signals** at the **same time**
  - **All frames** involved in collisions **are lost**


# MAC Protocols Categories

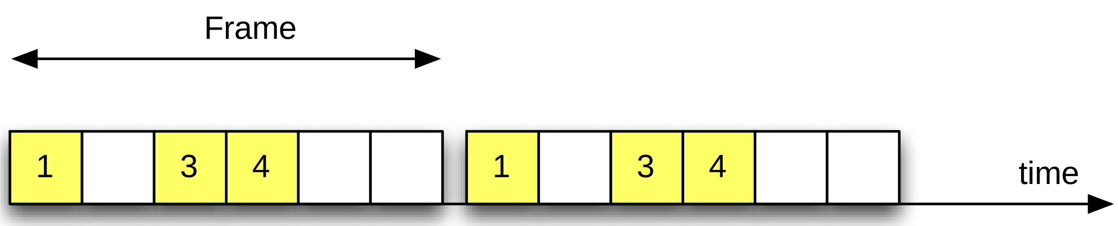
1. **Channel Partitioning**
   – Divide channel into smaller "pieces" (e.g. time slots, frequency)
   – Assign a piece to each node
   – TDMA, FDMA, CDMA, OFDMA
2. **Random Access**
   – No pre-allocation, i.e., channel not divided
   – Allow **collisions**, then "Recover" from collisions
3. **"Taking turns"**
   – Nodes take turns
   – Nodes that needs to send more, can take longer turns

# Channel Partitioning MAC protocols: TDMA

- TDMA: Time division multiple access
  - Each station gets fixed length time slot (e.g., length = maximum packet transmission time) in each round
- Assume N nodes share a broadcast medium:
  - Divide time frame into N time slots, each time slot is then assigned to one of the N nodes
- **Unused slots go idle**



- Example: 6-station LAN, only nodes 1,3,4 have packet, then time slots 2,5,6 are idle




- If transmission rate of the channel is **R**, then each device transmits at a rate **R/N**

# Channel Partitioning MAC protocols: FDMA

- FDMA: Frequency division multiple access
  - Each **device** is assigned to a different frequency

# Channel Partitioning MAC protocols: FDMA

- FDMA: Frequency division multiple access
  - Each **device** is assigned to a different frequency
  - N frequency bands, max rate of link R, then user rate is R/N

- In both TDMA and FDMA, the channel **bandwidth is divided evenly** among devices

# Channel Partitioning MAC protocols: OFDMA

- OFDMA: Orthogonal frequency division multiple access
  - Assign a subset of **orthogonal** subcarriers to each user
  - Robust over wireless fading channels
  - Used in cellular systems

# Channel Partitioning MAC protocols: CDMA

- CDMA: Code division multiple access
  - Assigns a different code to each device
  - With proper code design, nodes can send simultaneously, yet the receivers can correctly decode the information
  - Early used in **military systems** (due to the anti-jamming properties), now used in **civilian applications** as well (in cellular systems, 3G)

# Random Access Protocols

- **No prior coordination** or channel assignment
- When node has packet to send
  - **Transmit at full channel rate R**
- Two or more transmitting nodes → "collision"
  - Need to detect collisions and recover
- Examples of random access MAC protocols:
  - ALOHA, slotted ALOHA
  - CSMA, CSMA/CD, CSMA/CA

# Random Access: ALOHA

1970: ALOHAnet developed by University of Hawaii

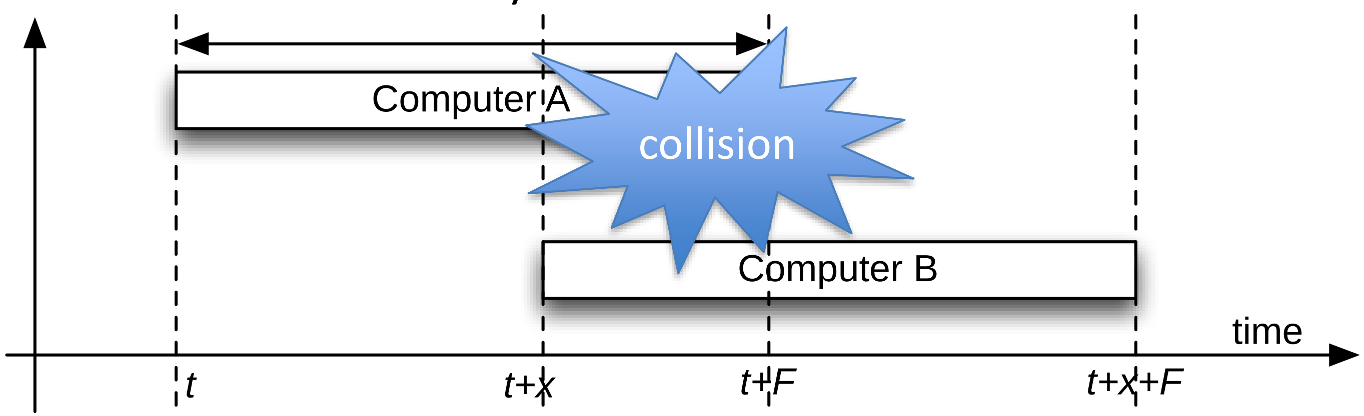
ALOHA: Additive Links on-line Hawaii Area

- **Transmit** a frame **whenever you want**!
  - **Wait for an acknowledgement (ACK)**

- If there is a **collision, no ACK** will be received
  - In this case, wait till a **random** amount of **time** and **retransmit**

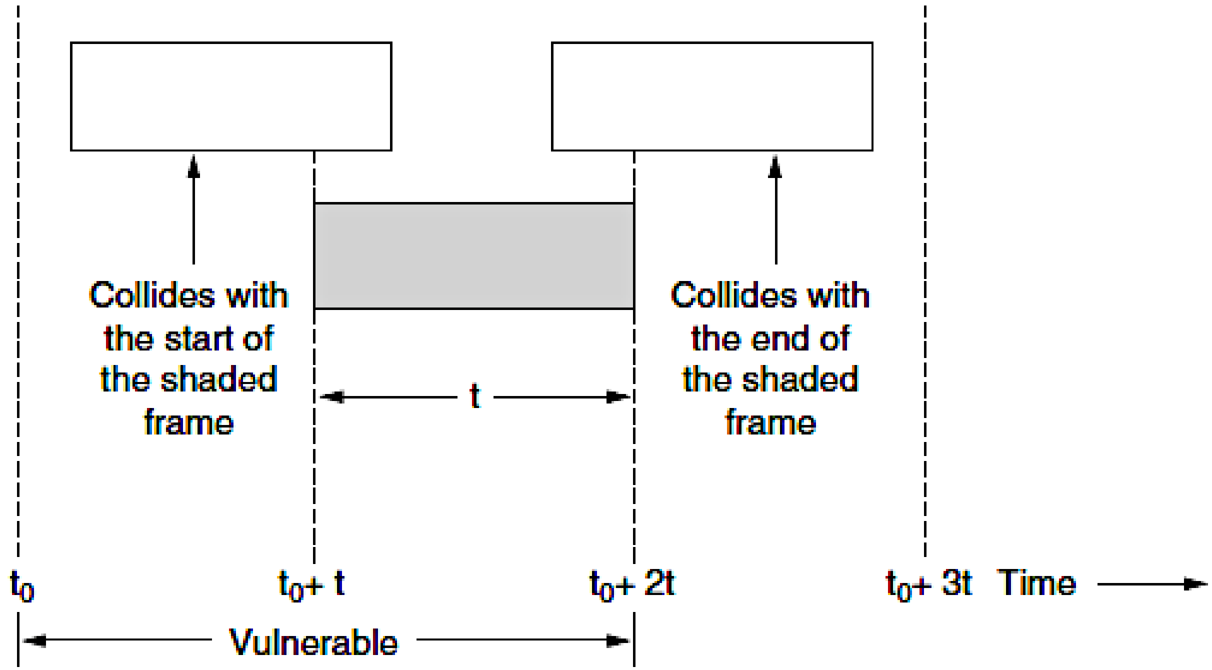
Collision => entire packet transmission time wasted for everyone

This is a link layer "Frame"



# Vulnerable time – only one device should transmit otherwise collision would occur

- Let t be the time required to send one frame




- Vulnerable time in Aloha is 2 t

Check chapter 4 in Tanenbaum's book

# Number of attempts to transmit a packet within same frame duration is Poisson random variable

- The number of **attempted packets per frame time (duration t)** is approximately a **Poisson** random variable of mean G

- The probability that **k frames** are generated during a given frame time

$$\Pr[k] = \frac{G^k e^{-G}}{k!}$$

- What is the probability of success?

# Aloha : Probability of success in transmitting one packet

- In Aloha we need one frame only one frame transmitted in vulnerable time

- Probability of Success S is to have only one frame transmitted in vulnerable time
  - Pr[success] = Pr[during one t, only one packet] x Pr[during the rest of the vulnerable time, no packet] = Pr[k=0] Pr[k=1] = *S*

$$S = Ge^{-2G}$$

# Aloha: what is the average G of the Poisson process to maximize the probability of success

- Taking derivative: dS/dG =0
  - The maximum throughput occurs at G = 0.5, with S = 1/2e=0.184.
    - The best we can hope for is 18% success rate
    - G=0.5, means in **two frame periods** only **one device** is transmitting

# Random Access: Slotted ALOHA

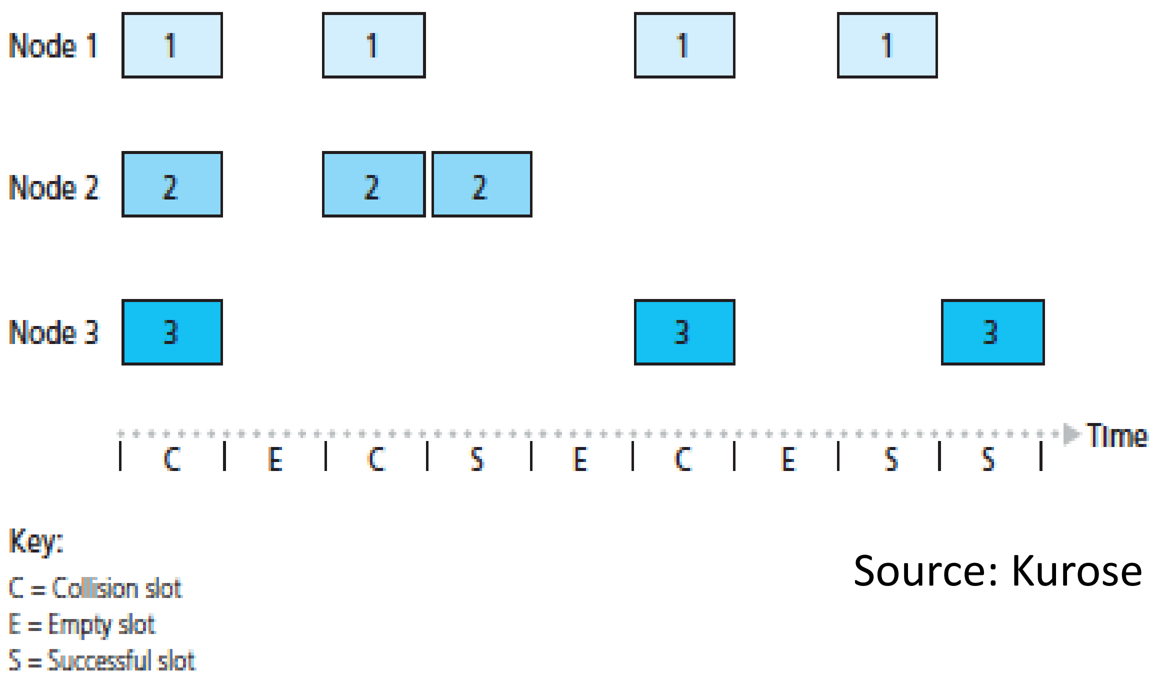
- Synchronizing senders to slots can reduce collisions

# Random Access: Slotted ALOHA

- Fix the starting/ending times to **specific values (time slots)**
  - All frames are exactly L bits
  - Slot is time to transmit L bits
- When a node has frame to send, it **waits till the beginning of the next slot** and transmits the frame
- If collision occurs (no acknowledgement is received), a node **transmits** the frame **in subsequent slot with certain probability (p)**
- **Repeat** attempt until frame is **successfully** received
- **Suitable for low traffic**

# Random Access: Slotted ALOHA -- Example

- Nodes transmit at the beginning of a time slot only
  - Example: nodes 1, 2, 3 collide in first slot, after several retransmission attempts node 2 succeed in 4th slot, node 1 succeed in 8th time slot, node 1 succeed in 9th time slot



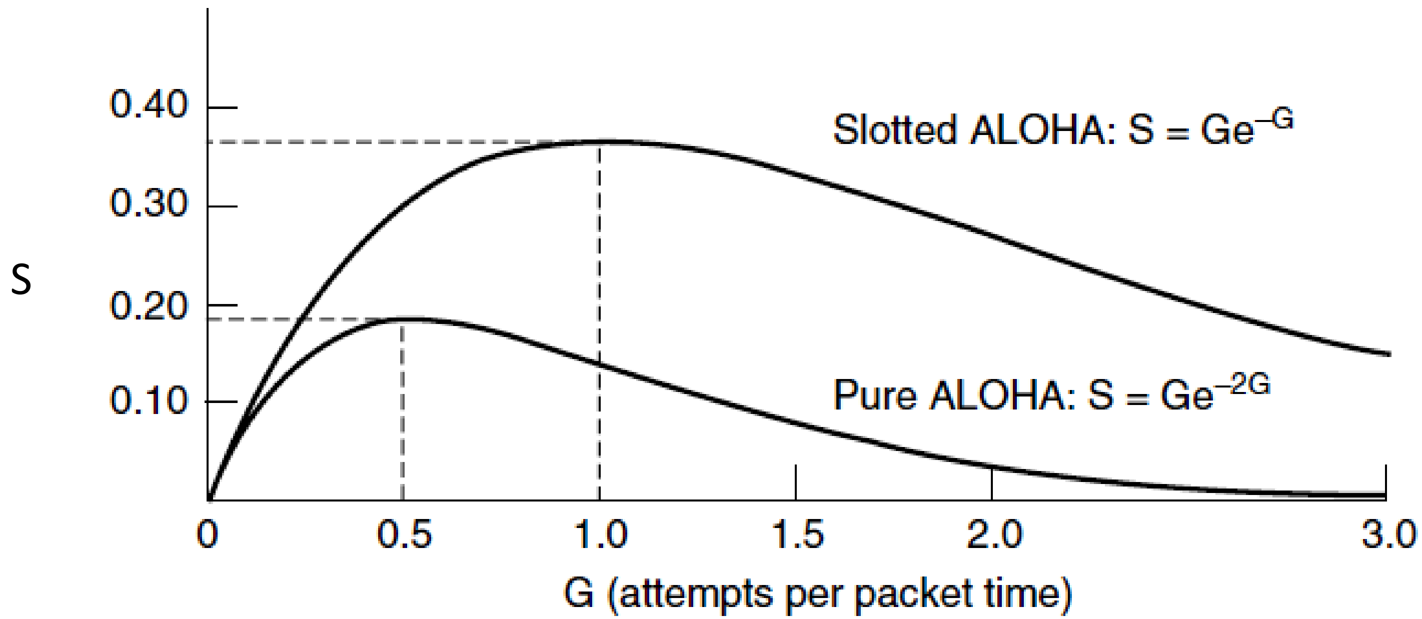

Source: Kurose et al.

# Random Access: Slotted ALOHA

- Probability of only one frame during slot time

$$S = Ge^{-G}$$

- Taking derivative: dS/dG =0
  - The maximum throughput occurs at G = 1,
  - Pr[success] = S = $e^{-1}$=0.368 (twice that of Aloha)


S
0.40
0.30
0.20
0.10
0
0.5
1.0
1.5
2.0
3.0
G (attempts per packet time)
Slotted ALOHA: S = Ge–G
Pure ALOHA: S = Ge–2G

# Question

- Suppose three stations trying to access a channel using slotted Aloha. The probability that any attempts transmission in a slot is X. What is the probability that Station 1 will have the first packet transmission succeed in the second slot?

# Question - Answer

- Suppose three stations trying to access a channel using slotted Aloha. The probability that any attempts transmission in a slot is X. What is the probability that Station 1 will have the first packet transmission succeed in the second slot?
  – Pr[success in a slot]=Ps = X(1-X)(1-X)
  – Pr[first success in second slot]= (1-Ps) Ps

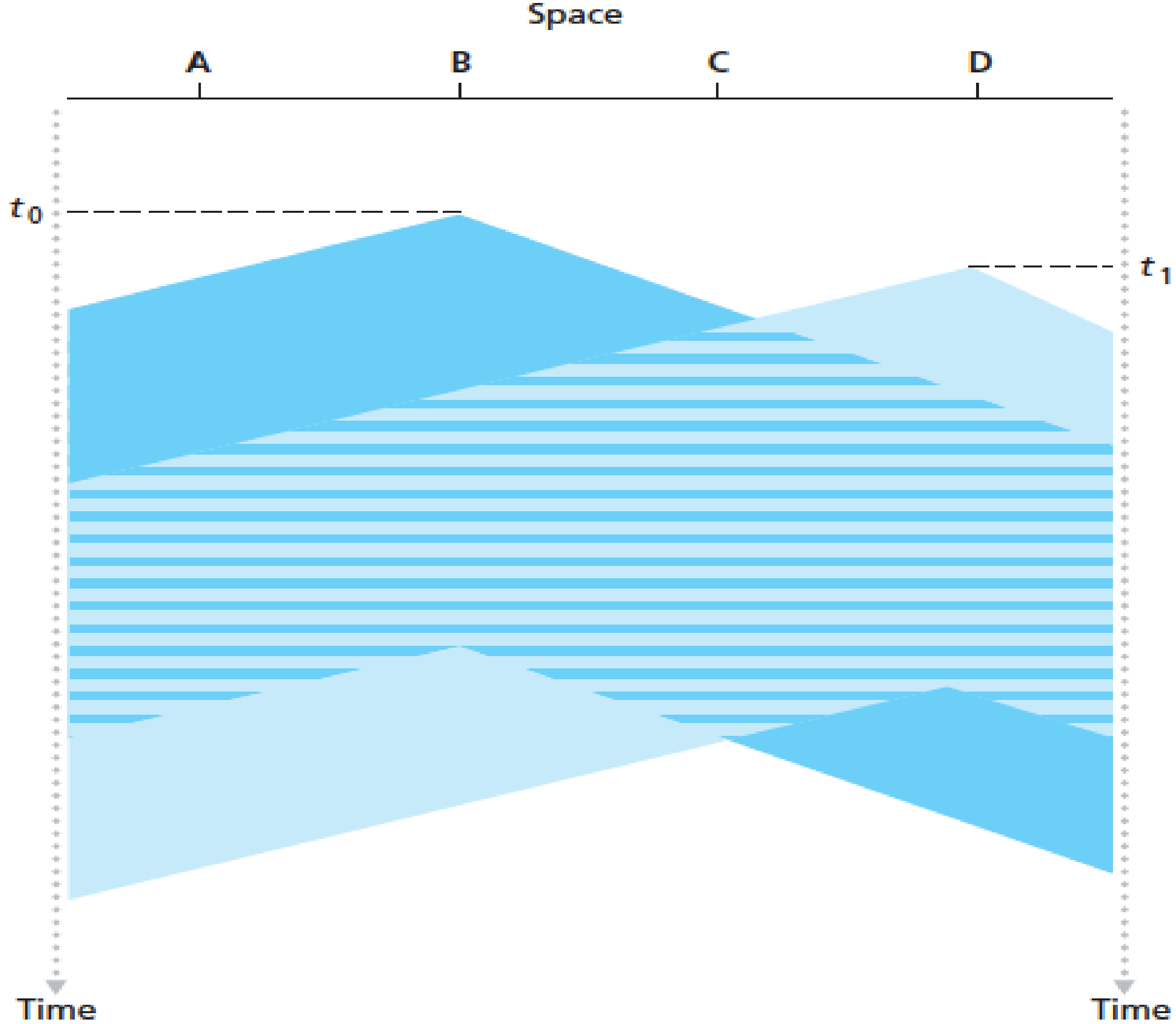
# Random Access: CSMA

- In **ALOHA**, **node decides to transmit independent of other nodes' activity**
- Carrier sense multiple access (CSMA)
  - "**Sense**" the medium to see if it is occupied before transmitting
  - If sensed channel is idle: **transmit** entire frame
  - If sensed channel is **busy**, **defer** transmission
- Analogy: listen before speaking(sensing)
- Can collisions occur?

# Collisions in CSMA

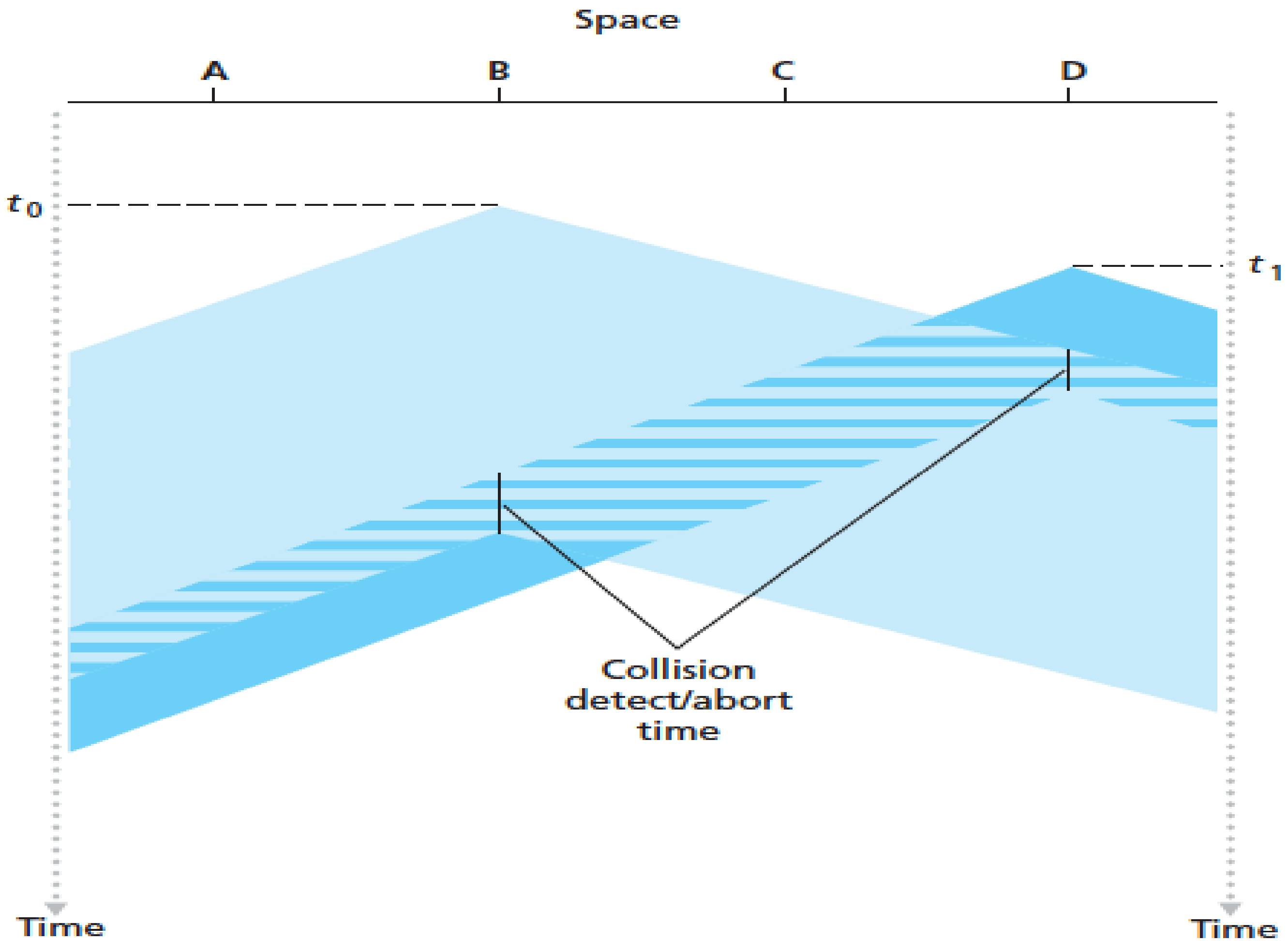
Space-time diagram shows collision can still occur due to propagation delay

**Note that nodes continue transmission even after collision is detected ➔ channel waste**



# Random Access: CSMA/CD

- CSMA with **collision detection**, if collision is detected after transmission, devices **ceases transmission**



# CSMA/CD

- Collisions can occur
- Nodes involved in collision retransmit their data frame
- Each node involved in collision waits for a **random delay** before retransmitting the frame
  - If delay is not random, then all nodes can wait the same amount of time - > collisions will occur again

# CSMA/CD efficiency

- The efficiency of CSMA/CD is the fraction of time when frames are transmitted over channel without collision

- Efficiency decreases when the propagation time increases

# MAC Protocol: Taking-turns Protocol

- Eliminate collisions

- Approaches:
  - Polling
  - Token-passing protocol

# Taking Turns MAC protocols: Polling Protocol

- One node acts as a master node
- Master node polls other nodes in a round robin fashion (in order) to check whether they have data to transmit
- Node may or may not have data to send
- **Bluetooth** uses polling

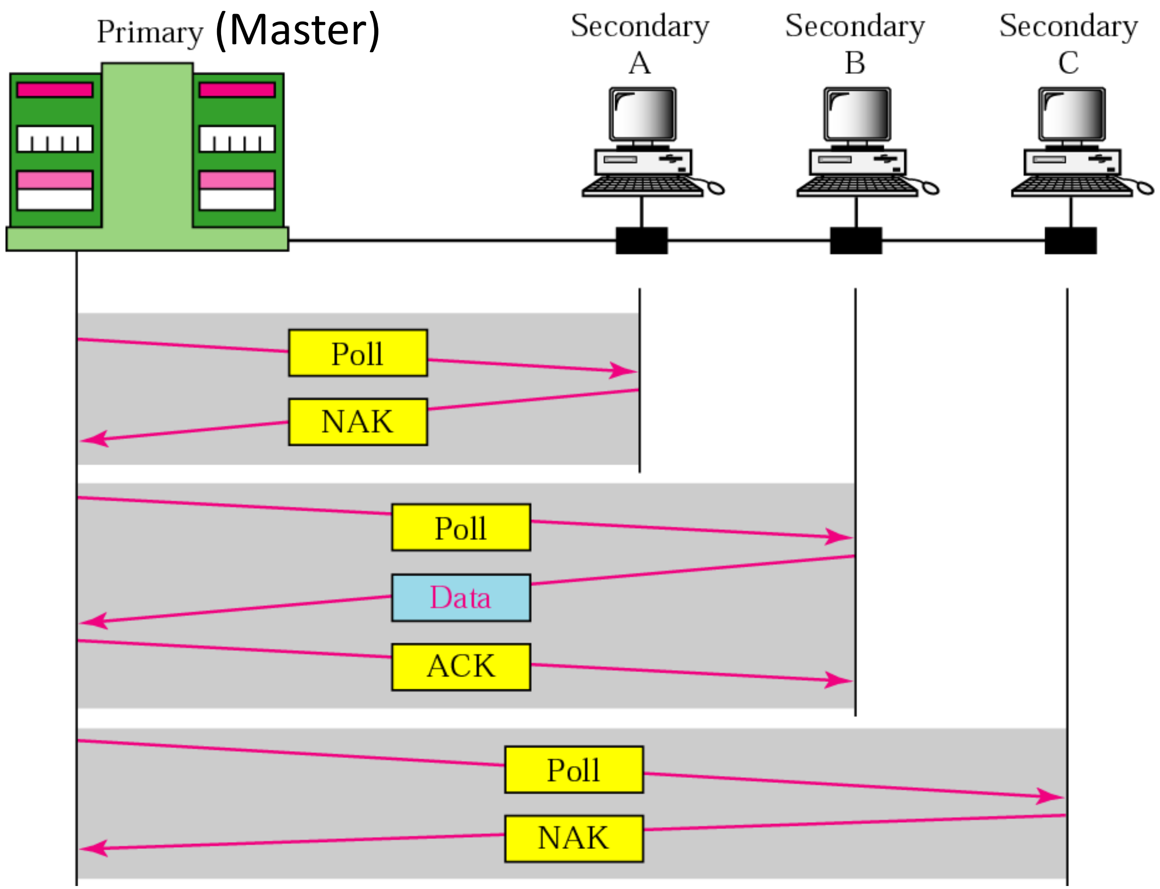
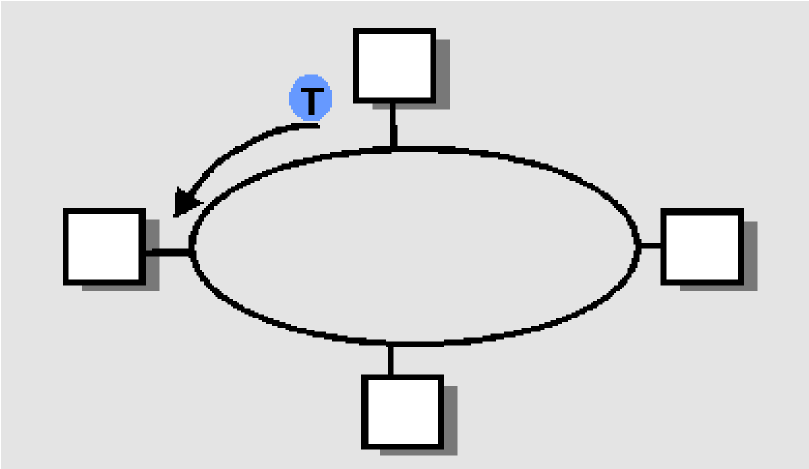


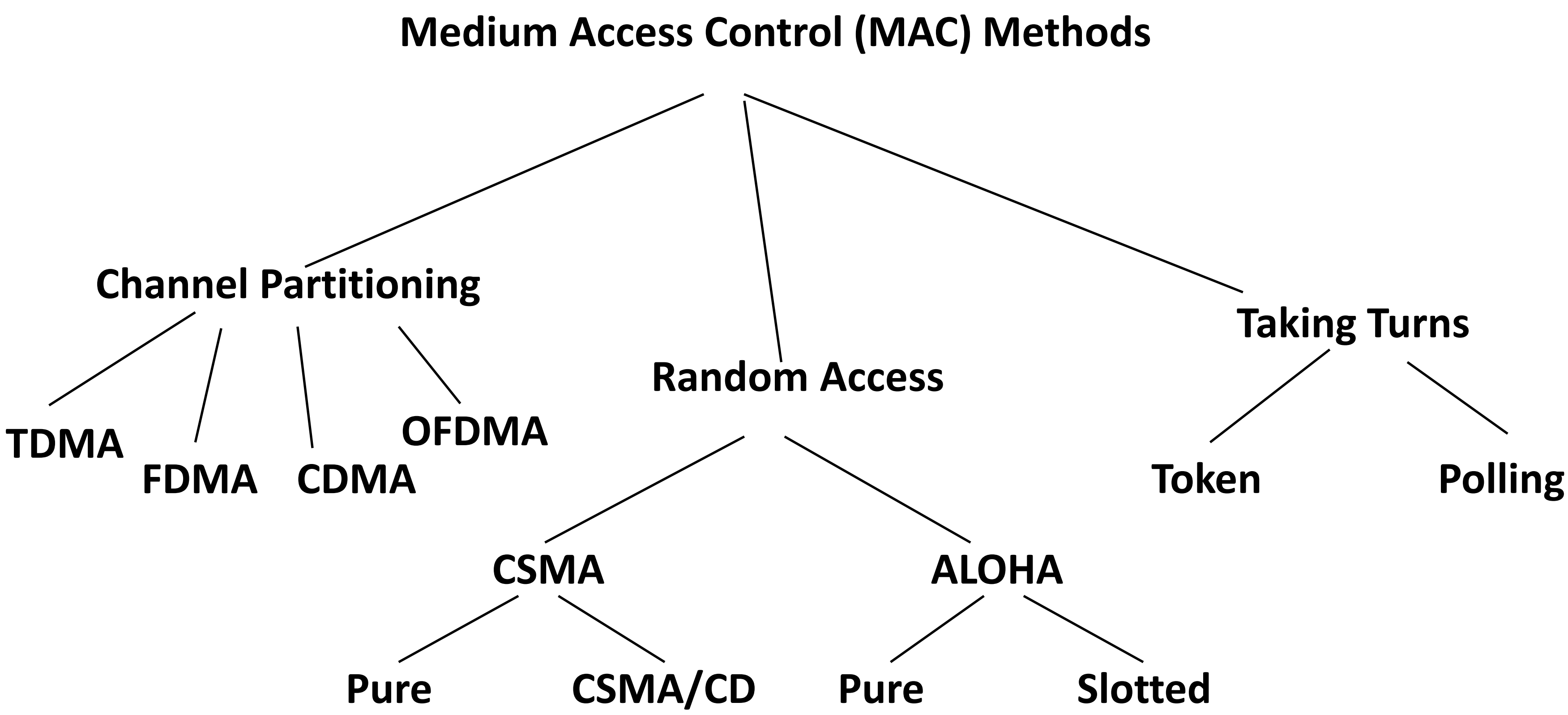

*Figure Source: Forouzan*

# Taking Turns MAC protocols: Token-Passing

- No master node
- Token messages are exchanged between nodes in a fixed order
  - Control token passed from one node to next sequentially.
- Concerns:
  - Token overhead
  - Latency



# Multiple Access Protocols: Summary




Book: “Computer Networking - A Top-Down Approach”, Jim Kurose and Keith W. Ross, Addison-Wesley

# Tophat

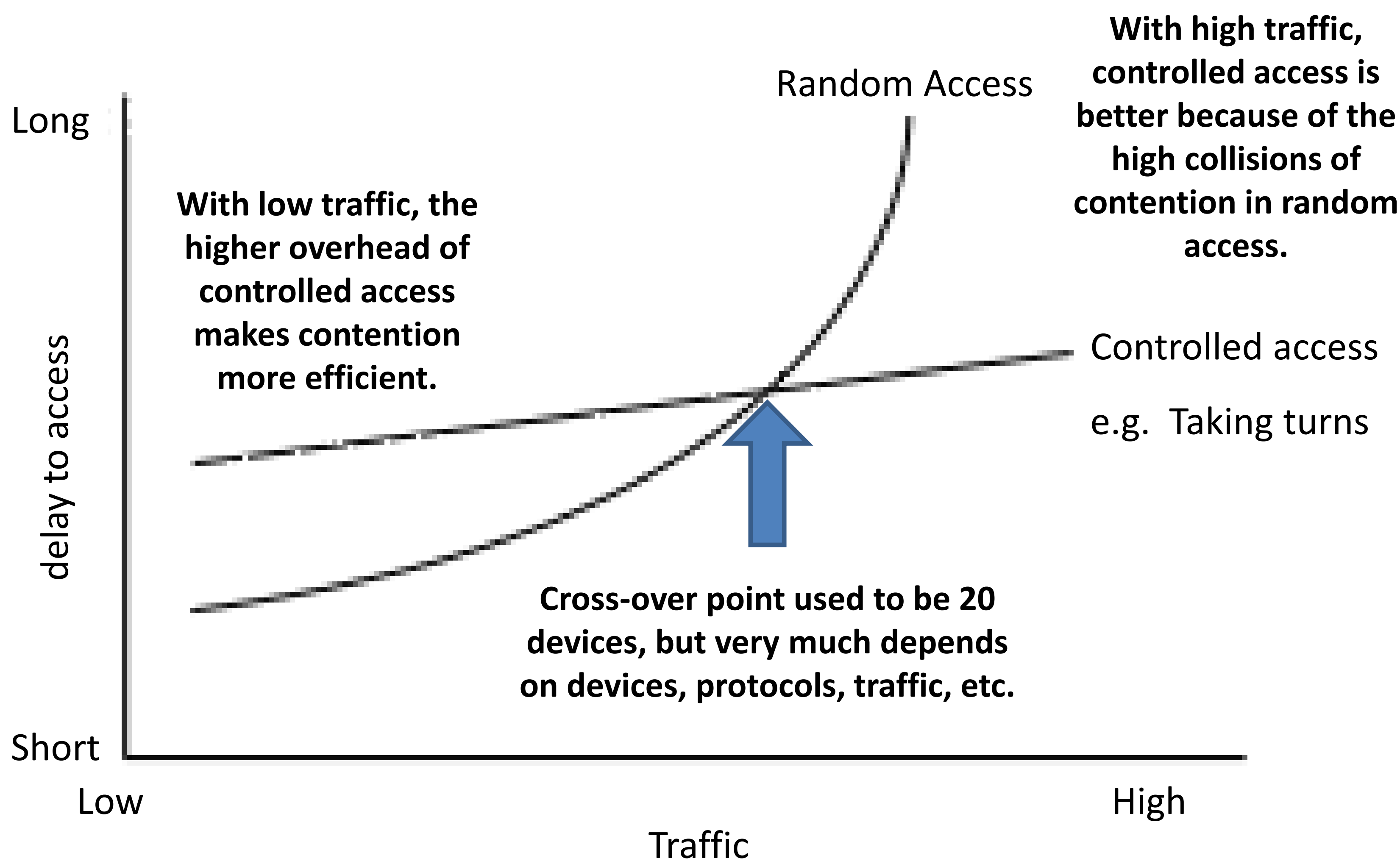
**Q_Multiple access control**

What is the impact of random multiple access on a user throughput?

| | |
|---|---|
| A | Has no impact |
| B | Improves the throughput |
| C | lowers the throughput |

- Which is better: controlled access or random access?
  - Depends on the throughput (useful information/total delay)
    - Delay until successfully access the channel
    - Useful information does not include headers or control information (like Ack messages)

# Multiple Access Protocols Performance



# Throughput – Channel Access Delay

- Suppose it is desired to send 10Kbit data chunk, and the transmission and propagation delays are 1 sec. With measurements, we find out that it takes on an average 10 msec to access the channel. What is the throughput?

Total delay = Average wait time for channel access + transmission & propagation=1.01 sec

Throughput = useful data to send / total delay = 10Kbit/1.01=9.9Kbps

- At high load (many users send simultaneously), instead of 10msec, now it takes on an average 50msec to send 10Kbit data chunk. What is the throughput?

Throughput at peak time = 10Kbit/ 1.05sec ~ 9.5 Kbps

# Pros and Cons

- Channel partitioning MAC protocols:
  - **No collisions**
  - Share channel efficiently and **fairly** at high load
- Random access MAC protocols
  - **Efficient at low load:** single node can fully utilize channel
  - **Collisions** can occur; **High load -> high collision overhead**
- "Taking turns" protocols
  - **No collisions**
  - Delay until you get your turn, overhead