

ELEN 4017

Network Fundamentals

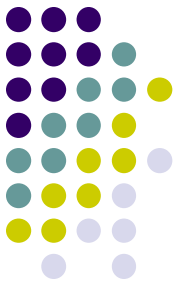
Lecture 29



Purpose of lecture



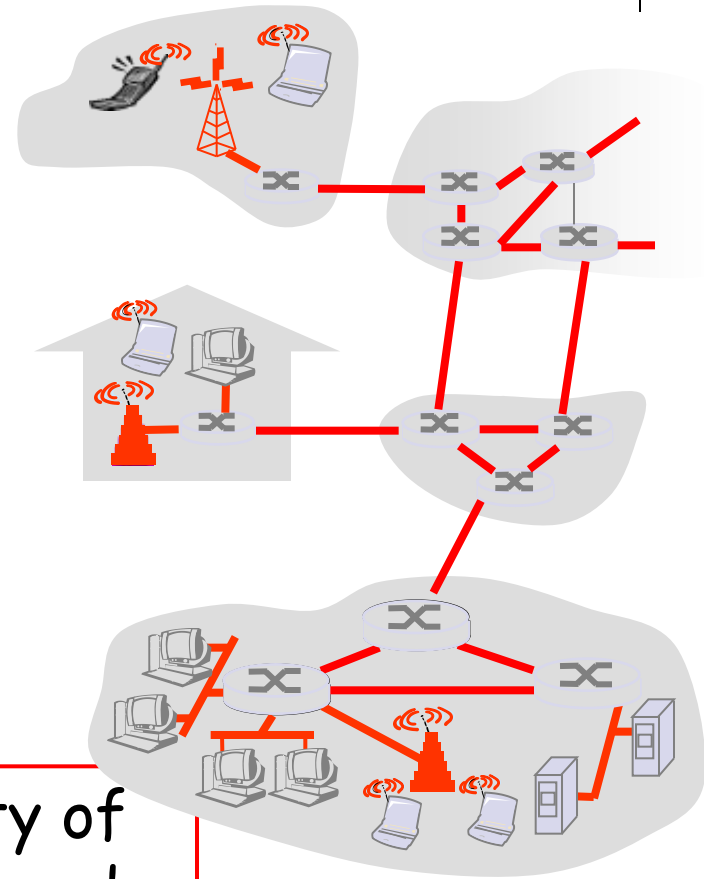
- Data Link Layer
 - Introduction
 - Multiple access protocols



Link Layer: Introduction

Some terminology:

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



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

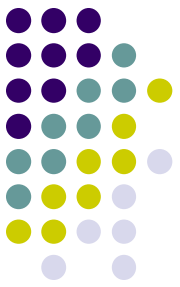
Link layer: context



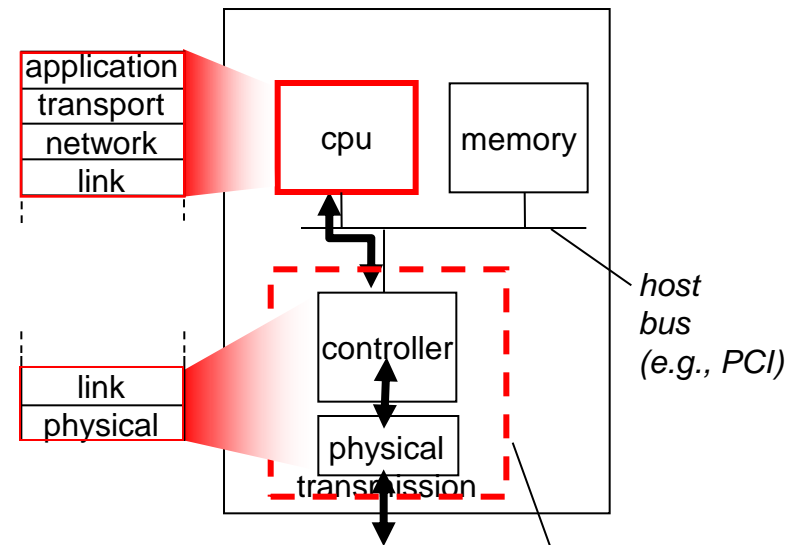
- datagram transferred by different link protocols over different links:
 - e.g., Ethernet on first link, frame relay on intermediate links, 802.11 on last link
- each link protocol provides different services
 - e.g., may or may not provide rdt over link

Where is the link layer implemented?

- in each and every host
- link layer implemented in “adaptor” (aka *network interface card* NIC)
 - Ethernet card, PCMCIA card, 802.11 card
 - implements link, physical layer
- attaches into host’s system buses
- combination of hardware, software, firmware



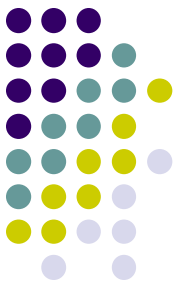
host schematic



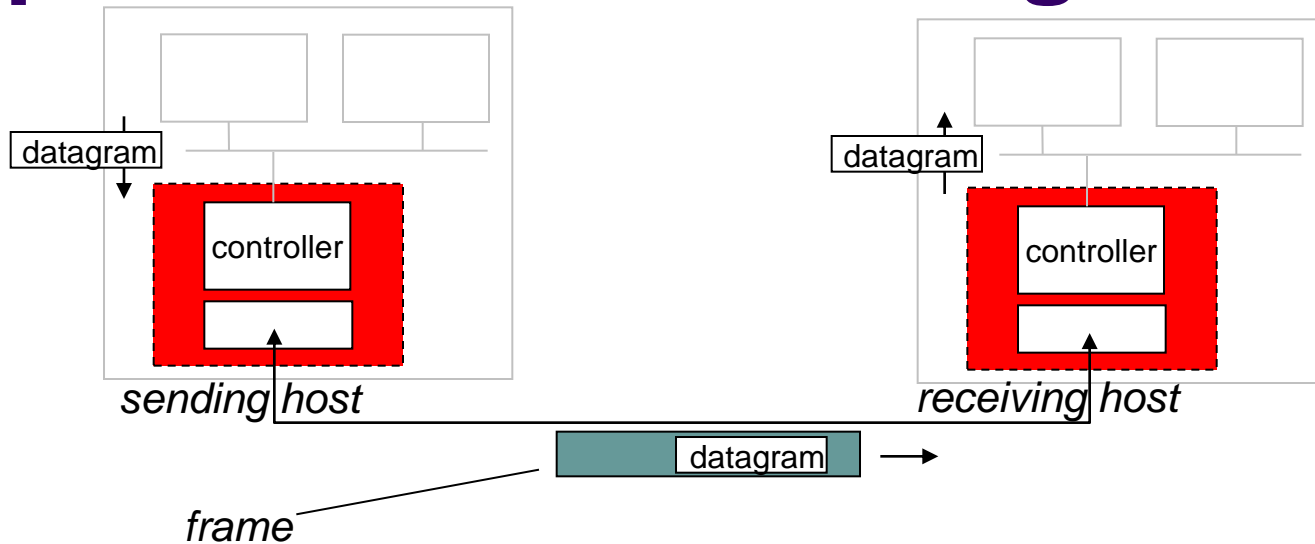
host bus (e.g., PCI)

network adapter card





Adaptors Communicating



- sending side:

- encapsulates datagram in frame
- adds error checking bits, rdt, flow control, etc.

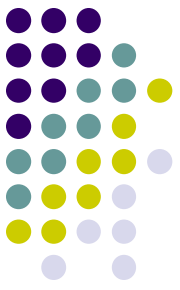
- receiving side

- looks for errors, rdt, flow control, etc
- extracts datagram, passes to upper layer at receiving side



Purpose of lecture

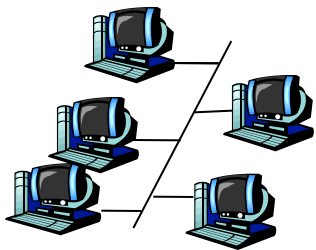
- Data Link Layer
 - Introduction
 - Multiple access protocols



Multiple Access Links and Protocols

Two types of “links”:

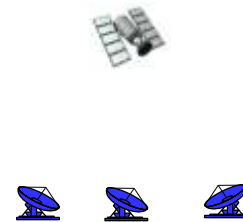
- point-to-point
 - PPP for dial-up access
 - point-to-point link between Ethernet switch and host
- **broadcast** (shared wire or medium)
 - old-fashioned Ethernet



shared wire (e.g.,
cabled Ethernet)



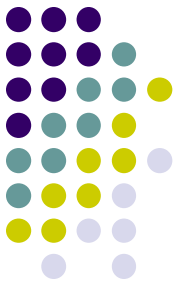
shared RF
(e.g., 802.11 WiFi)



shared RF
(satellite)



humans at a
cocktail party
(shared air, acoustical)



Multiple Access protocols

- single shared broadcast channel
- two or more simultaneous transmissions by nodes: interference
 - **collision** if node receives two or more signals at the same time

multiple access protocol

- distributed algorithm that determines how nodes share channel, i.e., determine when node can transmit
- communication about channel sharing must use channel itself!
 - no out-of-band channel for coordination

Ideal Multiple Access Protocol



Broadcast channel of rate R bps

1. when one node wants to transmit, it can send at rate R .
2. when M nodes want to transmit, each can send at average rate R/M
3. fully decentralized:
 - no special node to coordinate transmissions
 - no synchronization of clocks, slots
4. simple



MAC Protocols: a taxonomy

Three broad classes:

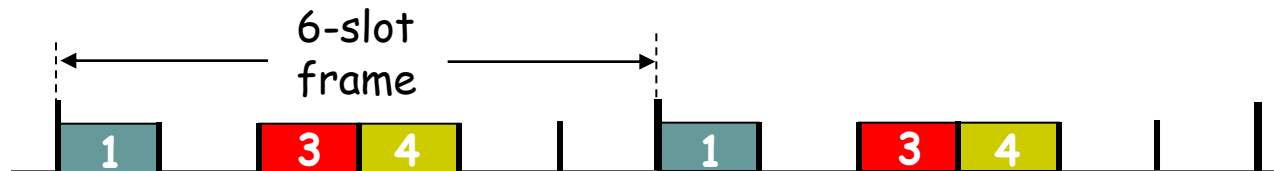
- **Channel Partitioning**
 - divide channel into smaller “pieces” (time slots, frequency, code)
 - allocate piece to node for exclusive use
- **Random Access**
 - channel not divided, allow collisions
 - “recover” from collisions
- **“Taking turns”**
 - nodes take turns, but nodes with more to send can take longer turns



Channel Partitioning MAC protocols: TDMA

TDMA: time division multiple access

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

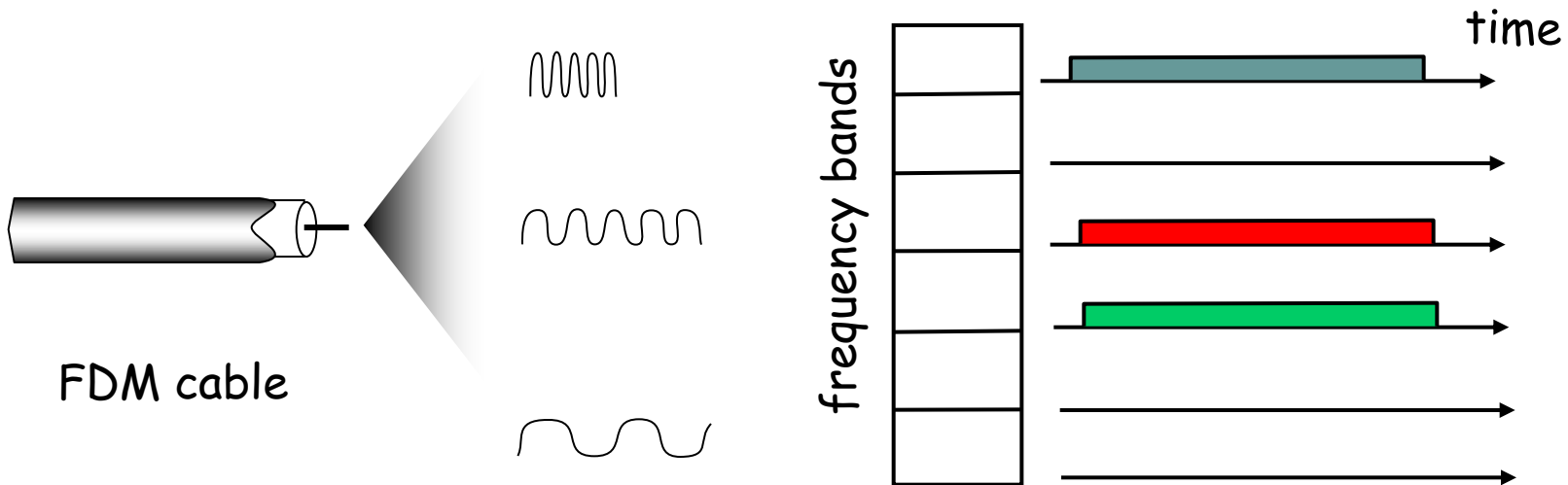


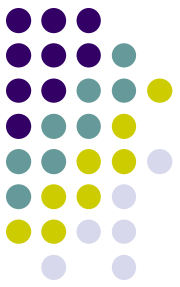


Channel Partitioning MAC protocols: FDMA

FDMA: frequency division multiple access

- channel spectrum divided into frequency bands
- each station assigned fixed frequency band
- unused transmission time in frequency bands go idle .





Random Access Protocols

- When node has packet to send
 - transmit at full channel data rate R .
 - no *a priori* coordination among nodes
- two or more transmitting nodes → “collision”,
- **random access MAC protocol** specifies:
 - how to detect collisions
 - how to recover from collisions (e.g., via delayed retransmissions)
- Examples of random access MAC protocols:
 - slotted ALOHA
 - ALOHA
 - CSMA, CSMA/CD, CSMA/CA

Slotted ALOHA



Assumptions:

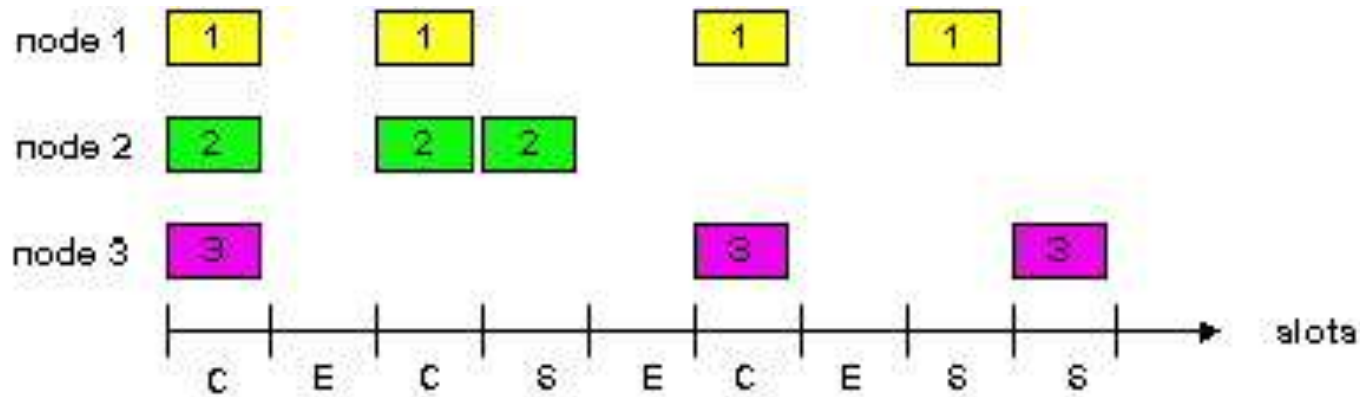
- all frames same size
- time divided into equal size slots (time to transmit 1 frame)
- nodes start to transmit only slot beginning
- nodes are synchronized
- if 2 or more nodes transmit in slot, all nodes detect collision

Operation:

- when node obtains fresh frame, transmits in next slot
 - *if no collision:* node can send new frame in next slot
 - *if collision:* node retransmits frame in each subsequent slot with prob. p until success



Slotted ALOHA



Pros

- single active node can continuously transmit at full rate of channel
- highly decentralized: only slots in nodes need to be in sync
- simple

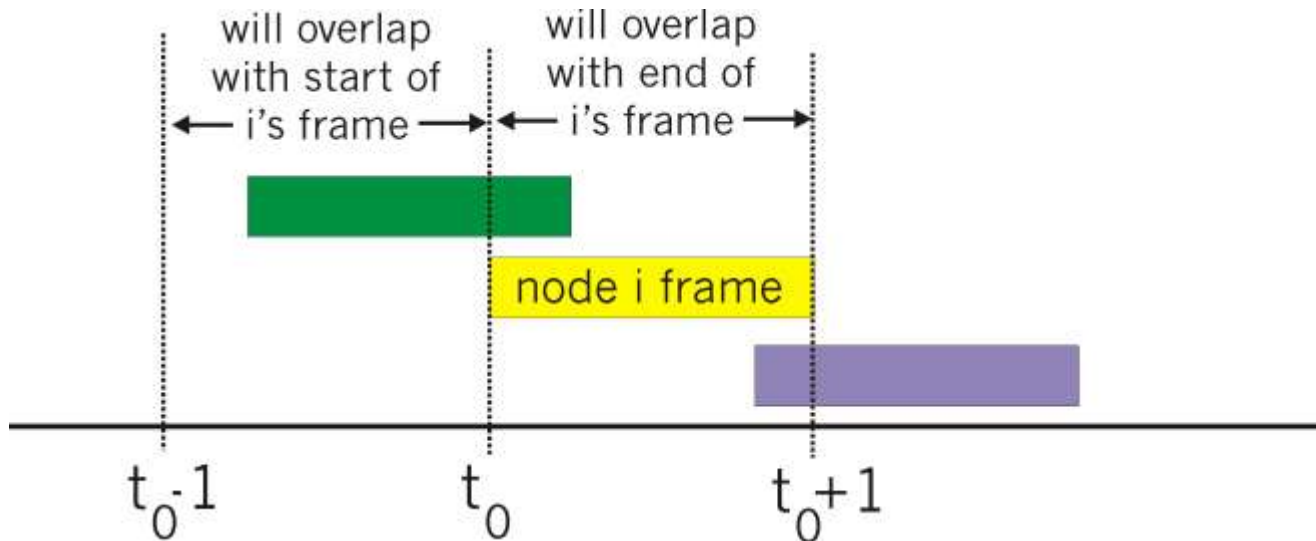
Cons

- collisions, wasting slots
- idle slots
- clock synchronization



Pure (unslotted) ALOHA

- unslotted Aloha: simpler, no synchronization
- when frame first arrives
 - transmit immediately
- collision probability increases:
 - frame sent at t_0 collides with other frames sent in $[t_0-1, t_0+1]$



CSMA (Carrier Sense Multiple Access)



CSMA: listen before transmit:

If channel sensed idle: transmit entire frame

- If channel sensed busy, defer transmission

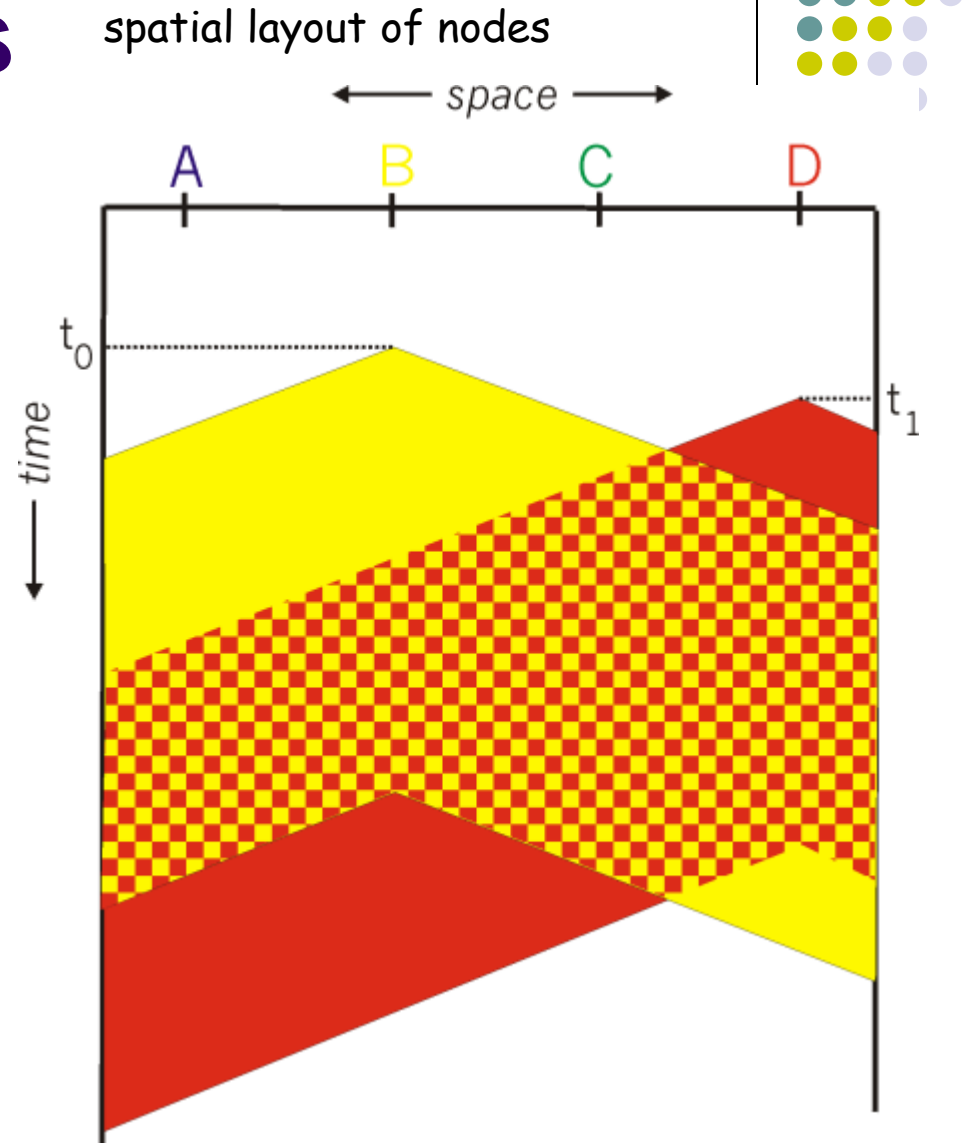
- human analogy: don't interrupt others!

CSMA collisions

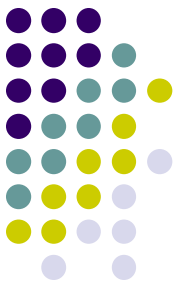
collisions *can still occur*:
propagation delay means
two nodes may not hear
each other's transmission

collision:
entire packet transmission
time wasted

note:
role of distance & propagation
delay in determining collision
probability



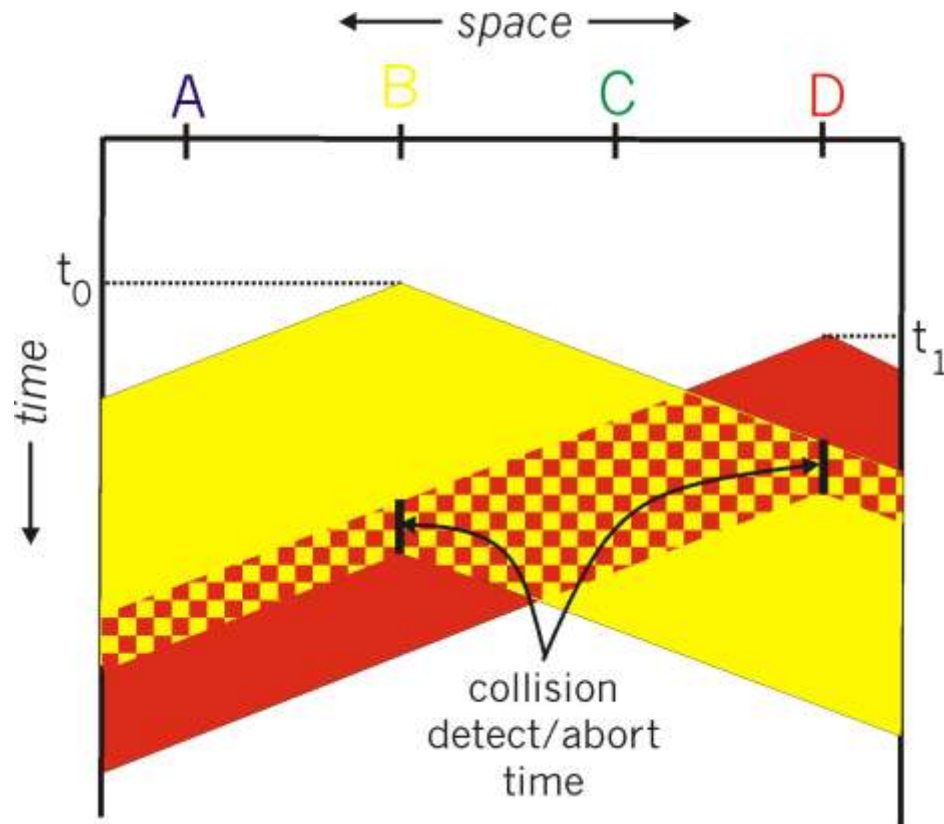
CSMA/CD (Collision Detection)



CSMA/CD: carrier sensing, deferral as in CSMA

- collisions *detected* within short time
- colliding transmissions aborted, reducing channel wastage
- collision detection:
 - easy in wired LANs: measure signal strengths, compare transmitted, received signals
 - difficult in wireless LANs: received signal strength overwhelmed by local transmission strength
- human analogy: the polite conversationalist

CSMA/CD collision detection



“Taking Turns” MAC protocols



channel partitioning MAC protocols:

- share channel *efficiently* and *fairly* at high load
- inefficient at low load: delay in channel access, $1/N$ bandwidth allocated even if only 1 active node!

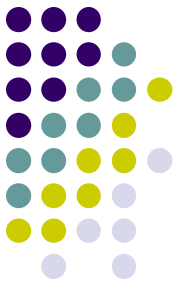
Random access MAC protocols

- efficient at low load: single node can fully utilize channel
- high load: collision overhead

“taking turns” protocols

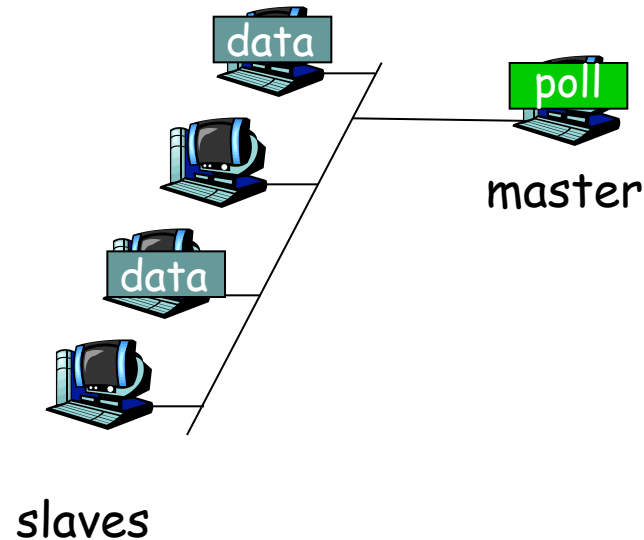
look for best of both worlds!

“Taking Turns” MAC protocols



Polling:

- master node
“invites” slave nodes to transmit in turn
- typically used with “dumb” slave devices
- concerns:
 - polling overhead
 - latency
 - single point of failure (master)

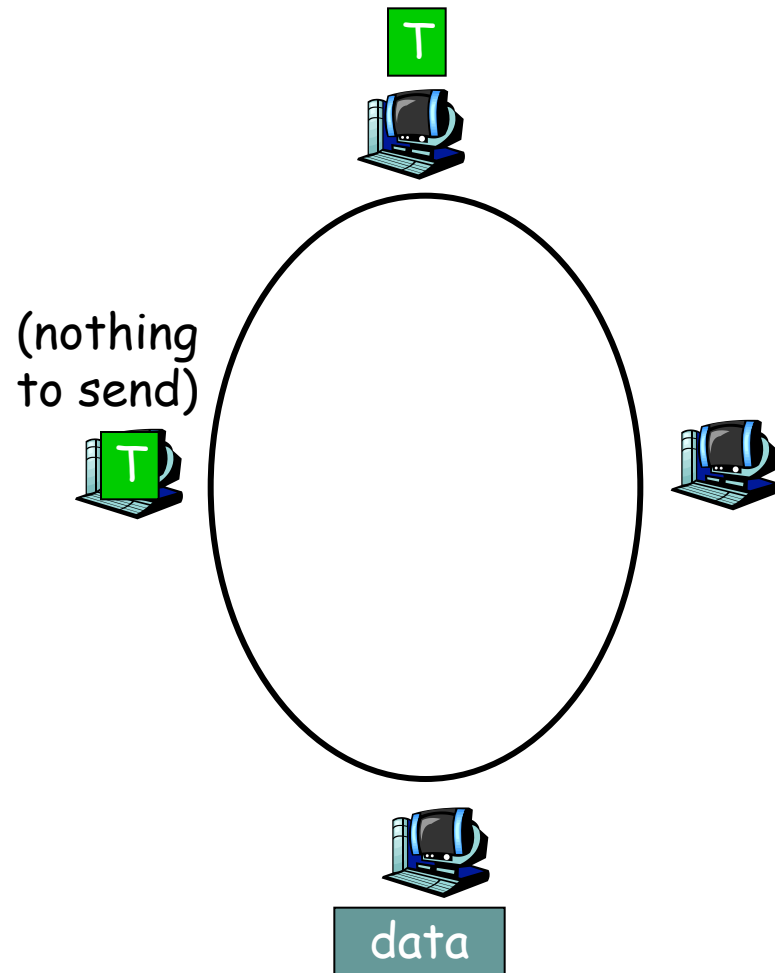


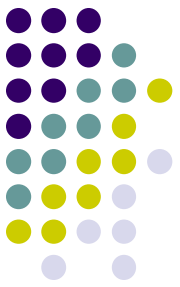


“Taking Turns” MAC protocols

Token passing:

- control **token** passed from one node to next sequentially.
- token message
- concerns:
 - token overhead
 - latency
 - single point of failure (token)





Summary of MAC protocols

- *channel partitioning*, by time, frequency or code
 - Time Division, Frequency Division
- *random access* (dynamic),
 - ALOHA, S-ALOHA, CSMA, CSMA/CD
 - carrier sensing: easy in some technologies (wire), hard in others (wireless)
 - CSMA/CD used in Ethernet
 - CSMA/CA used in 802.11
- *taking turns*
 - polling from central site, token passing
 - Bluetooth, FDDI, IBM Token Ring