Computer Networks I

Medium Access Control Protocols

Amitangshu Pal
Computer Science and Engineering
IIT Kanpur

Multiple access links, protocols

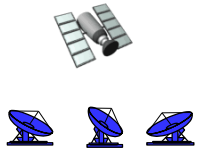
Two types of “links”:

- Point-to-point
  - Point-to-point link between Ethernet switch, host
  - PPP for dial-up access
- Broadcast (shared wire or medium)
  - Old-school Ethernet
  - Upstream HFC in cable-based access network
  - 802.11 wireless LAN, 4G/4G. satellite

shared wire (e.g., cabled Ethernet)

shared radio: 4G/5G

shared radio: WiFi



shared radio: 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

An ideal multiple access protocol

Given: multiple access channel (MAC) of rate *R* bps

Desired rate:

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: taxonomy

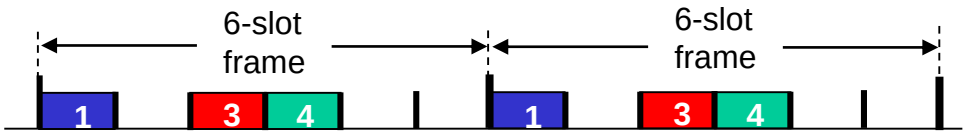
Three broad classes:

- 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

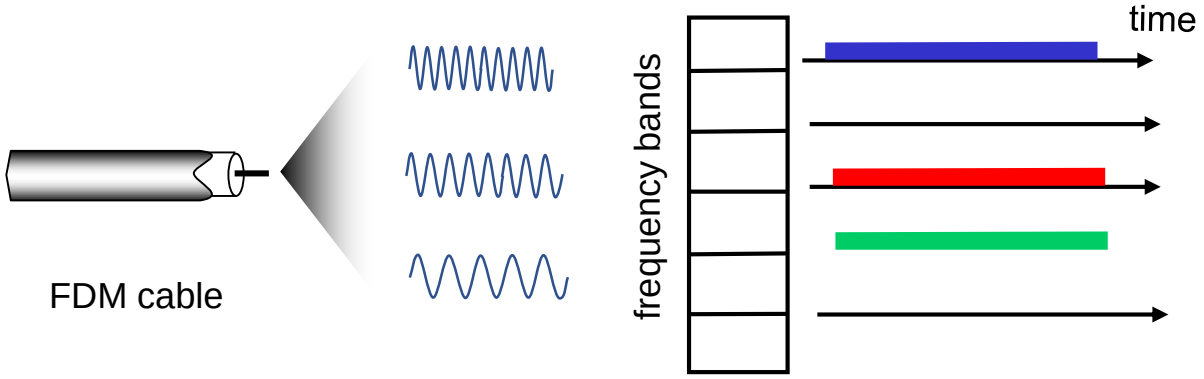
- Access to channel in “rounds”
- Each station gets fixed length slot (length = packet transmission time) in each round
- Unused slots go idle
- Example: 6-station LAN, 1,3,4 have packets to send, slots 2,5,6 idle



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
- Example: 6-station LAN, 1,3,4 have packet to send, frequency bands 2,5,6 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 protocol specifies:
  - How to detect collisions
  - How to recover from collisions (e.g., via delayed retransmissions)
- Examples of random access MAC protocols:
  - ALOHA, slotted ALOHA
  - CSMA, CSMA/CD, CSMA/CA

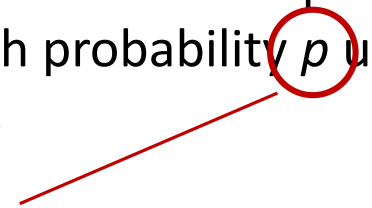
Slotted ALOHA

Assumptions:

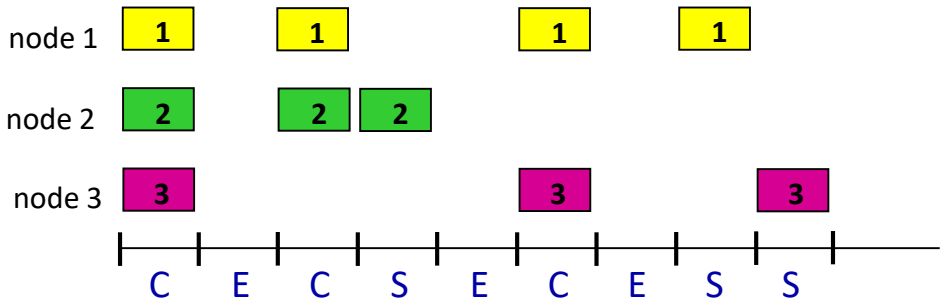
- All frames same size
- Time divided into equal size slots (time to transmit 1 frame)
- Nodes start to transmit only slot beginning
- Nodes are synchronized
- If 2 or more nodes transmit in slot, all nodes detect collision

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 probability *p* until success



Slotted ALOHA



C: Collision
S: Success
E: Empty

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
- Nodes may be able to detect collision in less than time to transmit packet
- Clock synchronization

Slotted ALOHA: efficiency

Efficiency: Long-run fraction of successful slots (many nodes, all with many frames to send)

- *Suppose: N* nodes with many frames to send, each transmits in slot with probability *p*
  - Prob that given node has success in a slot = $p(1-p)^{N-1}$
  - Prob that *any* node has a success = $Np(1-p)^{N-1}$
  - Max efficiency: find p^* that maximizes $Np(1-p)^{N-1}$
  - For many nodes, take limit of $Np^*(1-p^*)^{N-1}$ as *N* goes to infinity, gives:

  Max efficiency = 1/e = .37

- *At best:* channel used for useful transmissions 37% of time!

Pure ALOHA efficiency

P(success by given node) = P(node transmits) *

P(no other node transmits in $[t_0\text{-}1,t_0]$ * *

P(no other node transmits in $[t_0\text{-}1,t_0]$

$= p \cdot (1\text{-}p)^{N-1} \cdot (1\text{-}p)^{N-1}$

$= p \cdot (1\text{-}p)^{2(N-1)}$

… choosing optimum p and then letting $n \to \infty$

$= 1/(2e) = .18$

even worse than slotted Aloha!

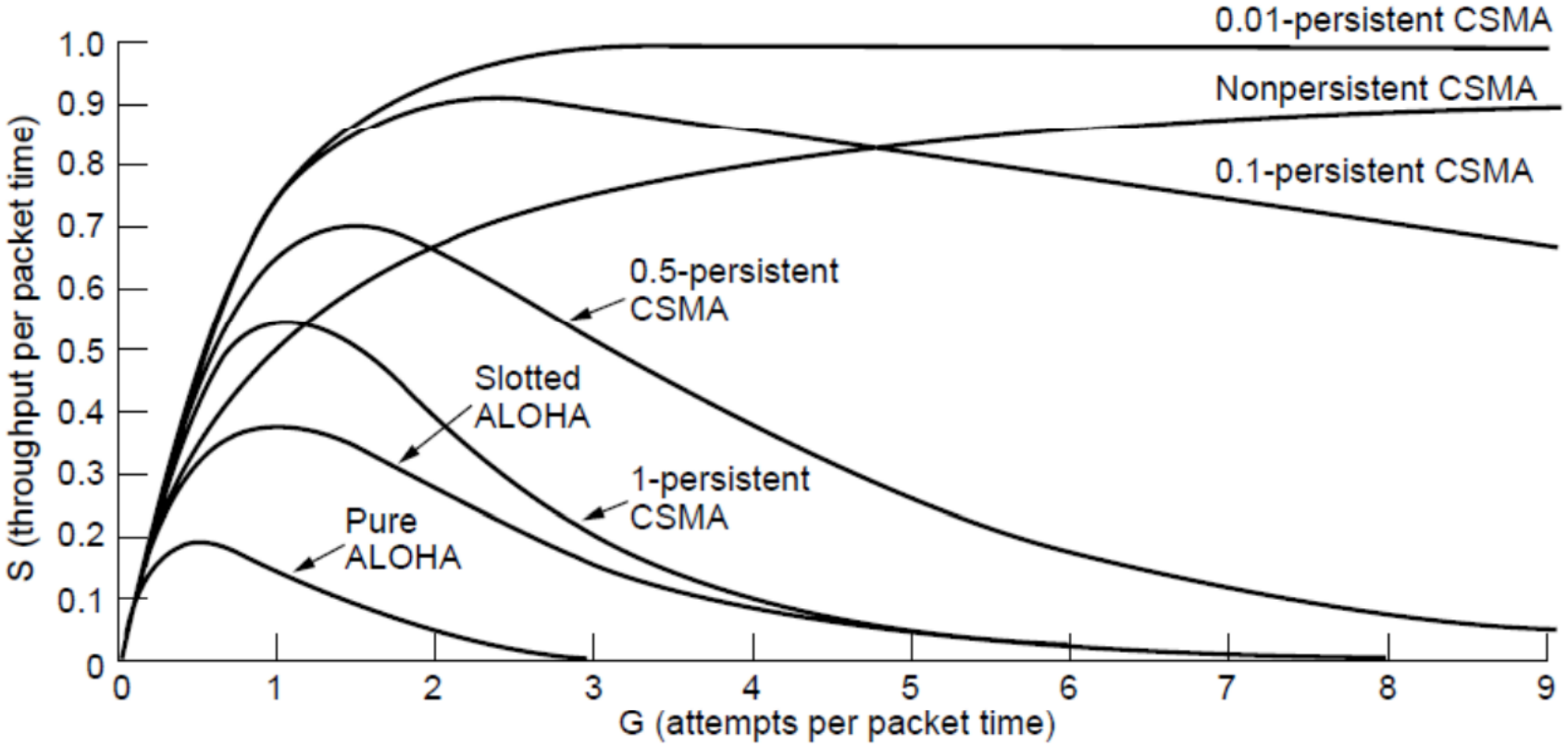
CSMA (carrier sense multiple access)

Simple CSMA: listen before transmit:

- If channel sensed idle: transmit entire frame
- If channel sensed busy: defer transmission

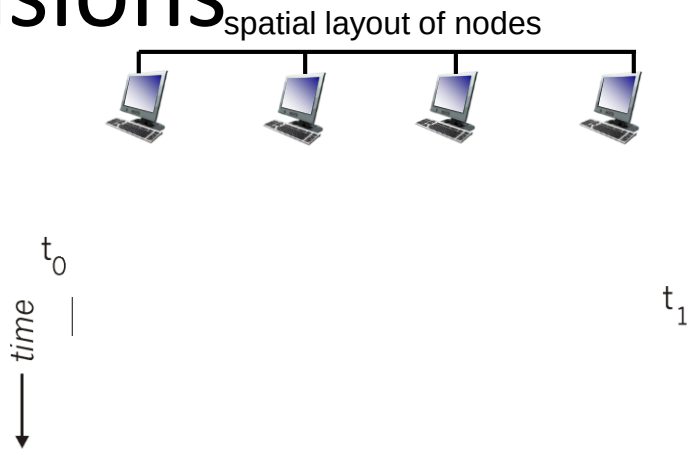
- Human analogy: don't interrupt others!

CSMA (carrier sense multiple access)



CSMA: collisions

- Collisions can *still* occur with carrier sensing:
  - Propagation delay means two nodes may not hear each other's just-started transmission
- Collision: entire packet transmission time wasted
  - Distance & propagation delay play role in in determining collision probability



CSMA (carrier sense multiple access)

Simple CSMA: listen before transmit:

- If channel sensed idle: transmit entire frame
- If channel sensed busy: defer transmission

- Human analogy: don't interrupt others!

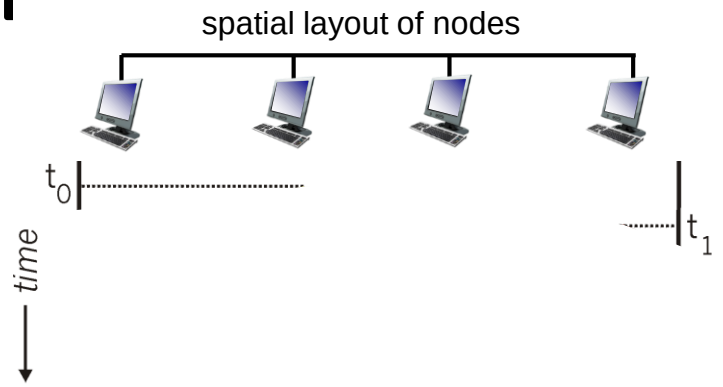
CSMA/CD: CSMA with *collision detection*

- Collisions *detected* within short time
- Colliding transmissions aborted, reducing channel wastage
- Collision detection easy in wired, difficult with wireless

- Human analogy: the polite conversation

CSMA/CD

- CSMA/CD reduces the amount of time wasted in collisions
  - Transmission aborted on collision detection



Ethernet CSMA/CD algorithm

1. Ethernet receives datagram from network layer, creates frame
2. If Ethernet senses channel:
   If idle: start frame transmission.
   If busy: wait until channel idle, then transmit
3. If entire frame transmitted without collision - done!
4. If another transmission detected while sending: abort, send jam signal
5. After aborting, enter *binary (exponential) backoff:*
   - After *m*th collision, chooses *K* at random from $\{0,1,2, \ldots, 2^m-1\}$. Ethernet waits $K \cdot 512$ bit times, returns to Step 2
   - More collisions: longer backoff interval

“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

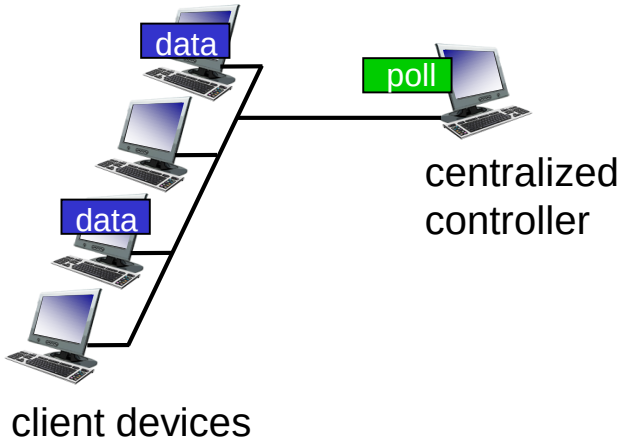
“Taking turns” protocols

- Look for best of both worlds!

“Taking turns” MAC protocols

Polling:

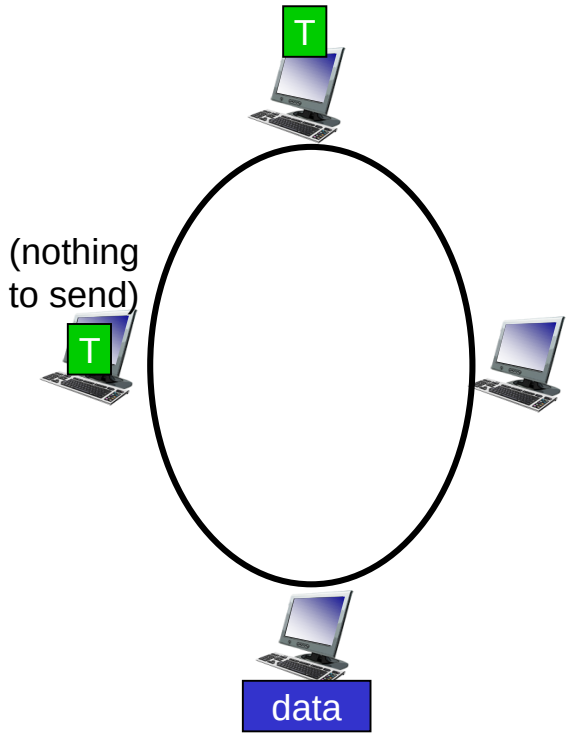
- Centralized controller “invites” other nodes to transmit in turn
- Typically used with “dumb” devices
- Concerns:
  - Polling overhead
  - Latency
  - Single point of failure (master)
- Bluetooth uses polling



“Taking turns” MAC protocols

Token passing:

- Control *token* message explicitly passed from one node to next, sequentially
  - Transmit while holding token
- Concerns:
  - Token overhead
  - Latency
  - Single point of failure (token)



Summary of MAC protocols

- Channel partitioning, by time, frequency or code
  - TDMA, FDMA
- Random access (dynamic),
  - ALOHA, S-ALOHA, CSMA, CSMA/CD
  - Carrier sensing: easy in some technologies (wire), hard in others (wireless)
  - Used in satellite, Ethernet, WiFi
- Taking turns
  - Polling from central site, token passing
  - Bluetooth (token ring)