Lecture 04-02: Link Layer Multiple Access Control

CS 356R

Intro to Wireless Networks

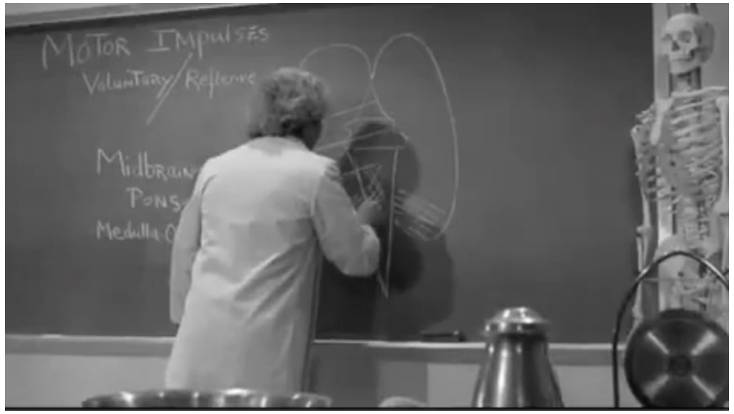
Mikyung Han

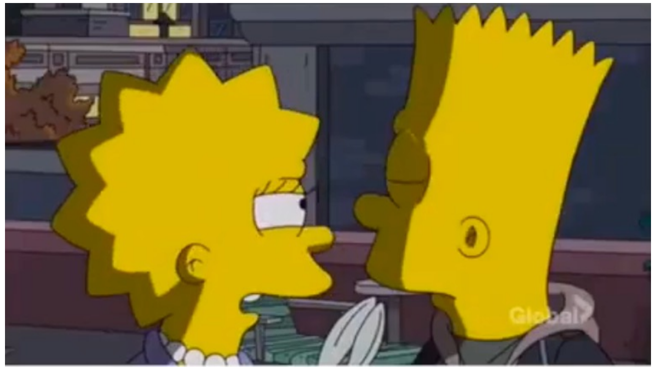
Outline

1. Multiple Access Procotol Recap

MAC protocols has 3 broad classes

- channel partitioning
- random access
- "taking turns"





Which class does each real-life example fall into?

Which class does TDMA and FDMA falls into?

- Random access
- ”taking turns”
- Channel partitioning

Pros and cons?

T/F Random access needs coordination among nodes

- False!

Pros and cons?

Avoiding collision or recovering from collision is the key in random access protocols

Which one works better in high load?
Which one works better in low load?
Which one suffers from potential single point of failure?

- Random access
- ”taking turns”
- Channel partitioning

Outline

1. Multiple Access Protocol Recap
2. Random Access Examples
    1. Slotted ALOHA

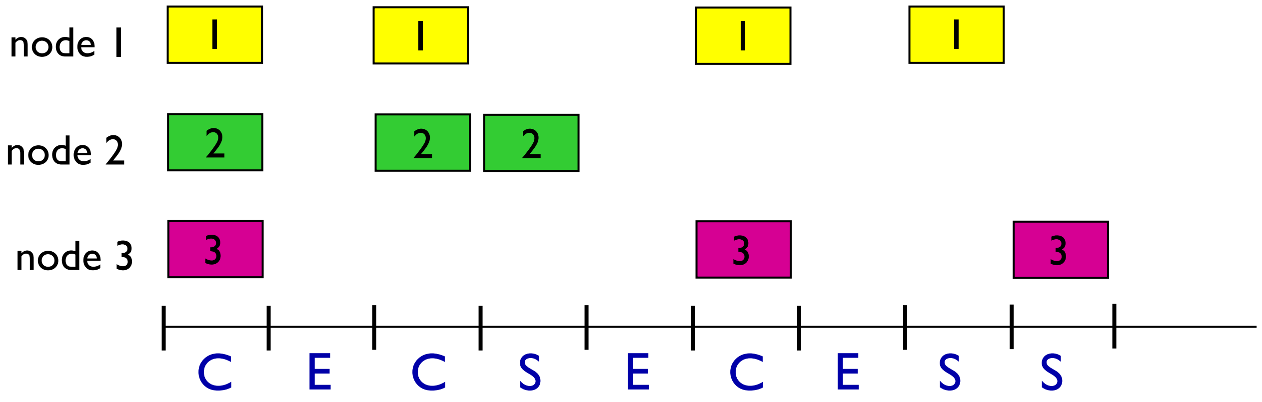
Slotted ALOPHA

- Each node just sends whenever it has a frame to transmit
  - Let collisions to happen and then deal with them later
- Who detects collision?
  - Sender needs to detect them
- How to recover?
  - Recover by retransmission
- Is it a good idea everyone starts retransmitting at the next slot?
  - Use randomization
- Nodes retransmits frame in each subsequent slot with probability p

Assumptions in Slotted ALOHA

- 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

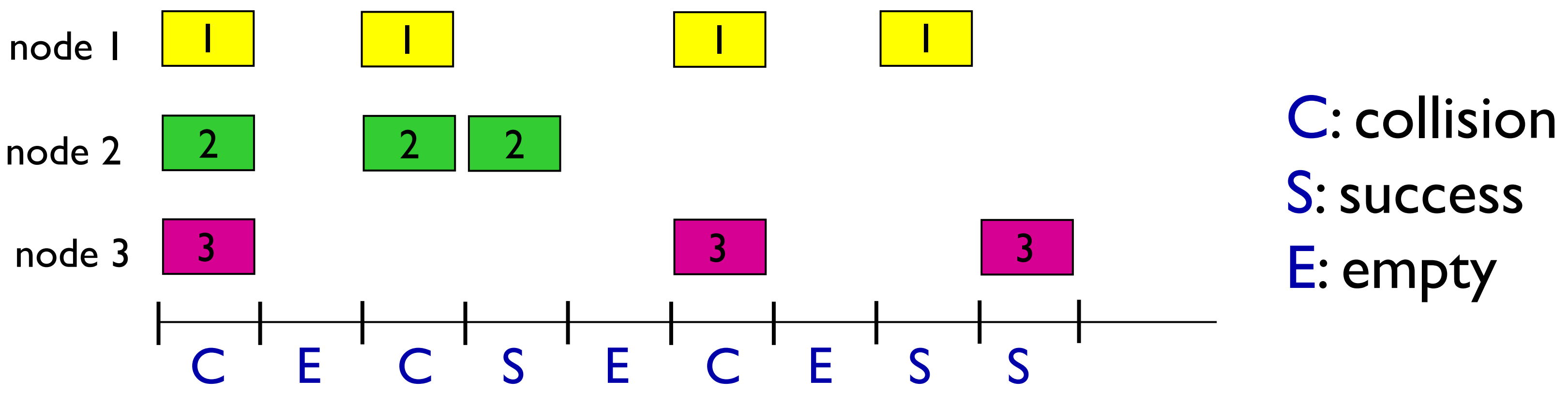
Slotted ALOHA Example



C: collision
S: success
E: empty

Pros and cons?

Slotted ALOHA Example



- Can a single node continuously transmit at full rate of channel?
- Do we need any coordination?
- Would it be fair?
- Are there idle slots?
- How to sync clock?

Slotted ALOHA: efficiency

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

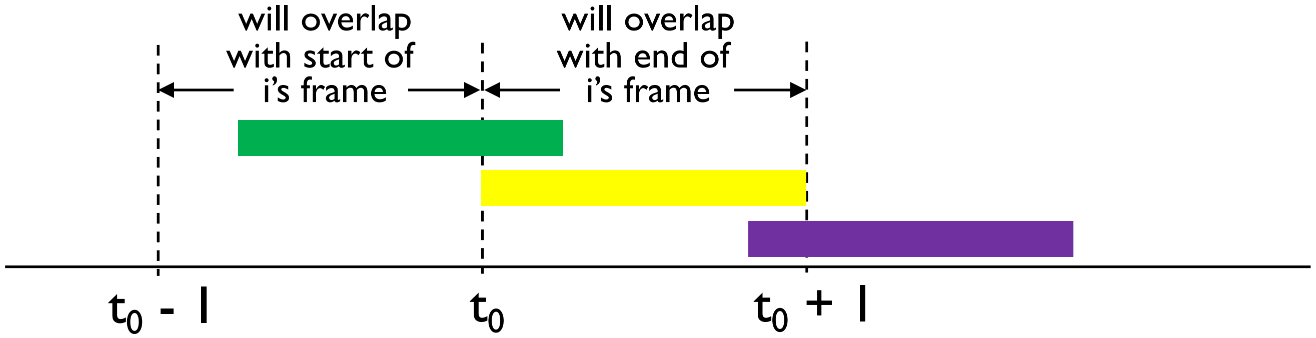
- *suppose: N* nodes with many frames to send, each transmits in slot with probability p
  - prob that given node has success in a slot $= p(1-p)^{N-1}$
  - prob that *any* node has a success $= Np(1-p)^{N-1} = efficiency$
  - max efficiency: find p^* that maximizes $Np(1-p)^{N-1}$
  - With large N, *max efficiency = 1/e = .37*

At best channel efficiency is only 37%

Only 37% of slot will carry a successful frame

FYI, Pure ALOHA with NO synchronization

- unslotted Aloha: simpler, no synchronization
  - when frame first arrives: transmit immediately (not at the "start" of the slot
- Would collision probability increase or decrease compared to Slotted ALOHA?
  - Without sync, frame sent at t_0 collides with other frames sent in $[t_0-1, t_0+1]$



Pure ALOHA's channel efficiency is 18%

What is the root cause of low efficiency?

- What is the efficiency with no collision?
- How to reduce collision from the first place?
- Can we detect someone is already sending before we transmit?

Outline

1. Multiple Access Protocol Recap
2. Random Access Examples
    1. Slotted ALOHA
    2. CSMA/CD

Listen before you send!

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

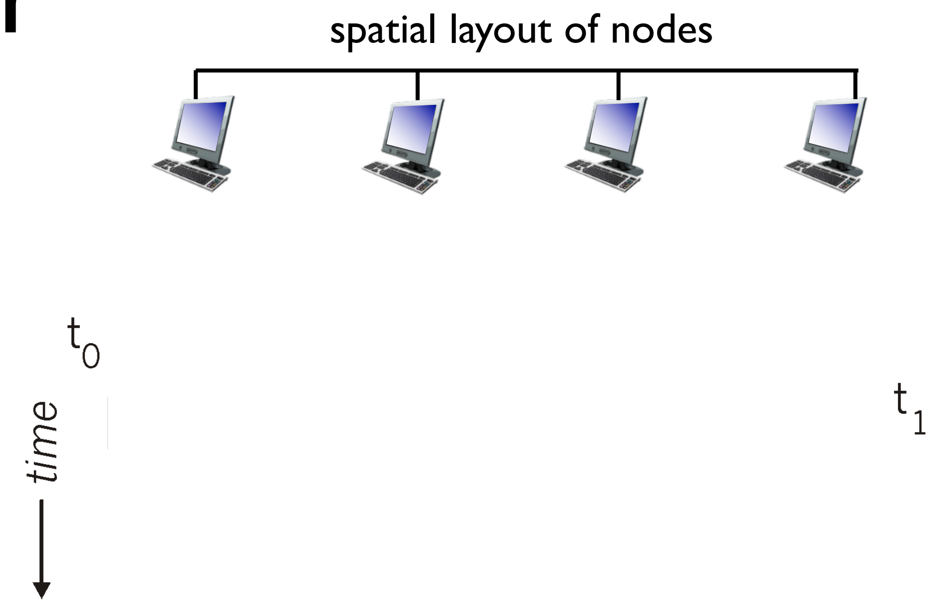
- collisions detected within short time
- colliding transmissions aborted, reducing channel wastage
- collision detection easy in wired, difficult with wireless

- human analogy: the polite conversationalist

How come we still have collision when everyone listens before sending?

CSMA: collisions still occur

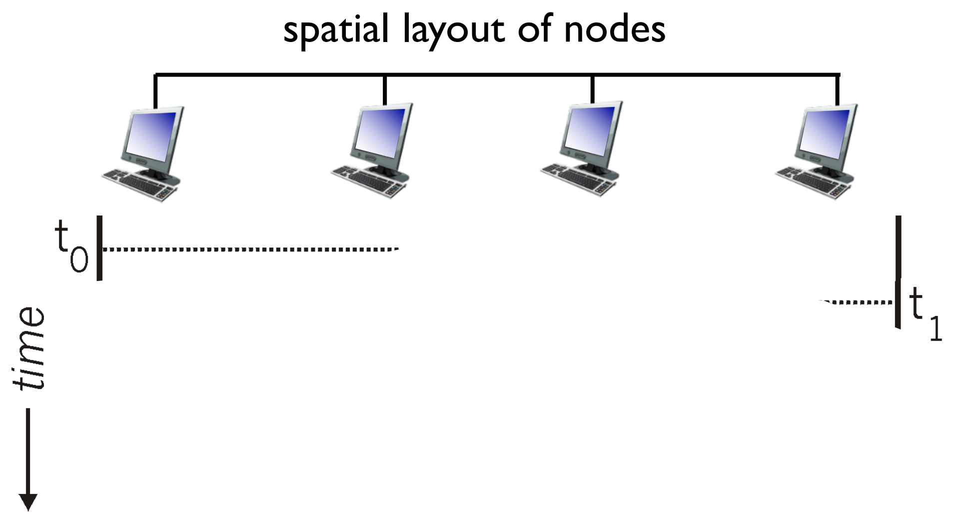
- Even with carrier sensing due to propagation delay
  - Two nodes may not hear each other's just-started transmission
- collision: entire packet transmission time wasted
  - distance & propagation delay play role in in determining collision probability



Collision detection is important to minimize this waste

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:

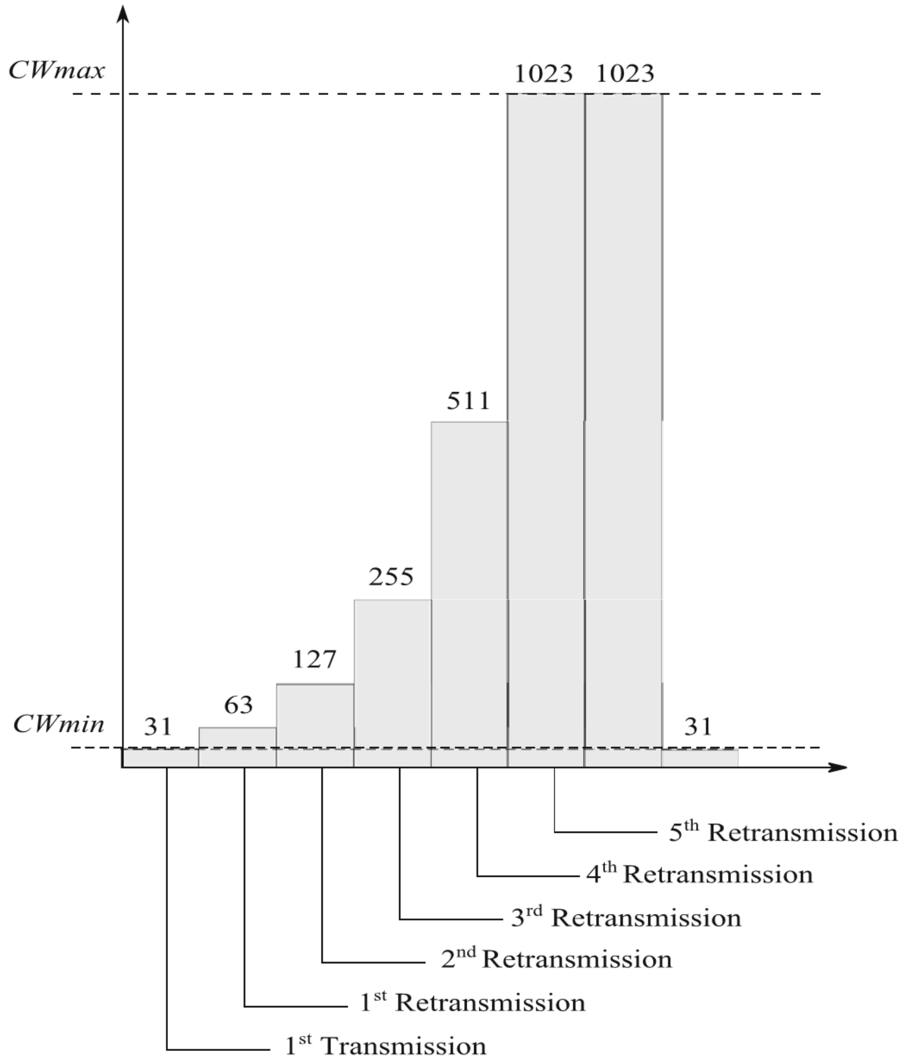
   if idle: start frame transmission.

   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·512 bit times, returns to Step 2

More collisions mean longer backoff interval

802.11 performs CSMA/CA without collision detection

- WLAN radios are half duplex:
  - Only talk or listen, but not both!
- Cannot detect as they send
  - Then how collision is detected?
- By not getting ACK for DATA packet sent it detects collision
- 802.11 uses carrier sensing with binary backoff



CSMA/CS vs CSMA/CD

Key	CSMA/CA	CSMA/CD
Effectiveness	Effective before a collision.	Effective before/after a collision.
Network Type	Generally used in wireless networks.	Generally used in wired networks.
Recovery Time	Minimizes the risk of collision but does not reduce the recovery time	Reduces the recovery time.
Conflict Management	Not receiving ACK triggers retransmission	Collision detection triggers retransmission
IEEE Standards	IEEE 802.11 standard.	IEEE 802.3 standard.