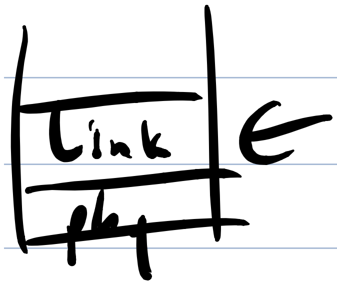
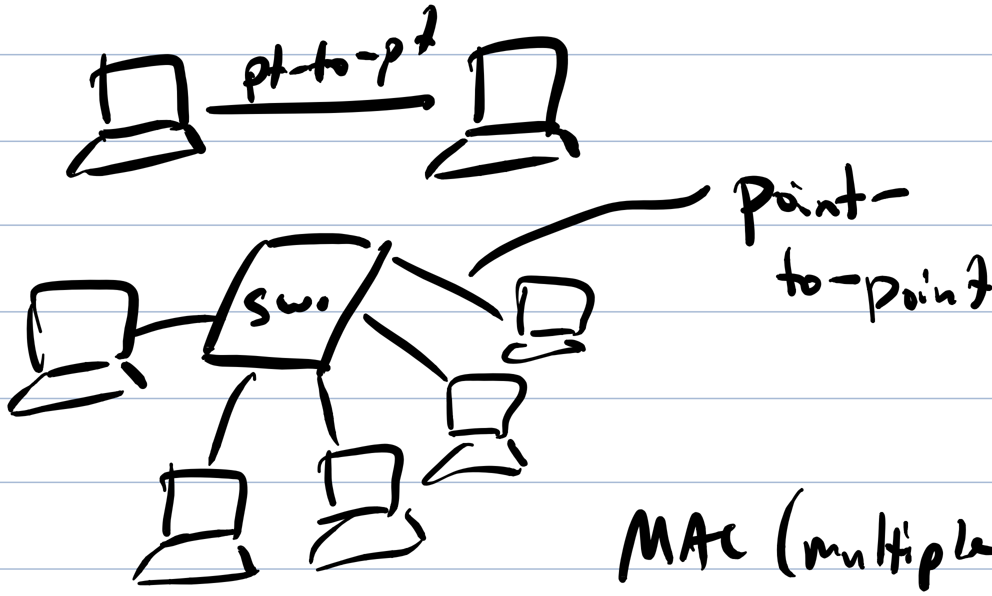
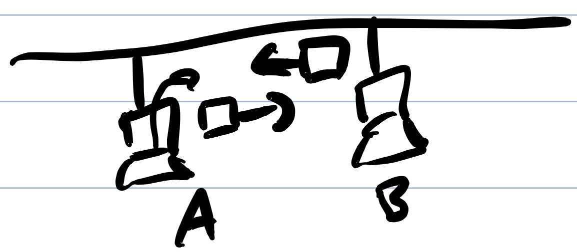



MAC (multiple access control)

shared (broadcast) link
WiFi Access point




old days: bus / channel for LAN

Scenario: single shared broadcast channel (link)
- interference: 2 or more simultaneous txs by node
  - problem: collision (node recv 2+ signals at the same time)

multiple access protocol - distributed algo. that determines when node can tx
- no out-of-band channel for coordination

ideal

given: broadcast channel of rate R bps

desiderata:

1. node can send at rate R when it wants
2. M nodes want to tx ⇒ each can send at avg $\frac{R}{M}$
3. fully decentralized
   → no sync (clocks)
   → no special node to coordinate tx
4. Simple

## MAC protocols: taxonomy

**Channel partitioning**
- partition channel into pieces
- node gets exclusive allocation

∴ Avoid collision

↳ TDMA

↳ FDMA

TDM
FDM

# of users known a priori

↳ # of users unknown a priori

**Random access**
- not div., allow collisions
- recover from collision

**"Taking turns"**
- nodes w/ more to send can take longer turns
- no partitioning

## TDMA

- timeslots in frames, limit to # of users per frame
- adv.: fair, collision free
- dis.: waste of bw (idle), unnecessary waiting

## FDMA

- same adv/disadv as TDMA but no waiting

## Random access

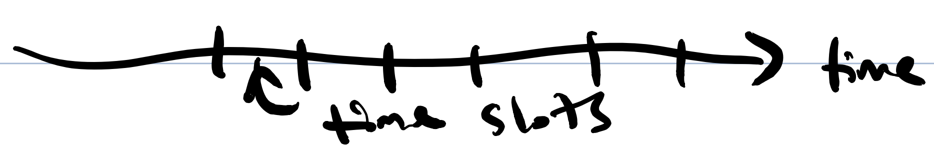
- transmit at full rate R
- how to detect collisions?
- how to recover from collisions?

} jobs

- e.g. ALOHA, CSMA, /CD, /CA (detect/avoid)
  (Hawaii Univ.) + Slotted ALOHA

## Slotted ALOHA

- assume all frames same size, time div. into equal time slots (time to submit 1 frame)
- nodes start to tx only at slot start
- nodes are synchronized
- if 2+ nodes tx in a slot, all nodes detect collision



Operation

- node obtains fresh frame → transmits at start of next slot
  - ↳ no collision: send new frame in next slot done. ✓
  - ↳ Collision: retx frame in each subsequent node with prob. p until success

<u>Pros:</u>

- single active node can tx at full channel rate (drawback of ch. p.)
- highly decentralized (only sync for slots)
- simple

Cons

- collisions can happen
- idle slots (waste b/w)
- nodes do not stop tx even if they can already detect collision
- clock synchronization - main disadvantage

Efficiency - long run fraction of successful slots (many nodes, all w/ many frames to send)

$$\frac{\#\ \text{successful tx}}{\text{total}\ \#\ \text{tx}} = P(\text{successful tx})$$

Suppose $N$ nodes w/ many frames to send
each tx in slot w/ probability $p$

$P(\text{given node succeeds in slot}) = p(1-p)^{N-1}$

$P(\text{any nodes succeed}) = Np(1-p)^{N-1}$

Max efficiency - find $p^*$ maxing $Np(1-p)^{N-1}$
for many nodes, take $\lim_{N \to \infty} Np^*(1-p^*)^{N-1}$

gives max eff $= 1/e \doteq 0.37$
$\Rightarrow$ at best, channel used for useful tx
37% of the time

<u>Pure (unslotted) ALOHA</u>

- Simpler, no sync
- when frame first arrives, tx immediately
- collision probability increases:
  - $\rightarrow$ frame sent at $t_0$ collides w/ other frames sent in $[t_0 - 1, t_0 + 1]$

$P(\text{any node success}) = Np(1-p)^{2(N-1)}$

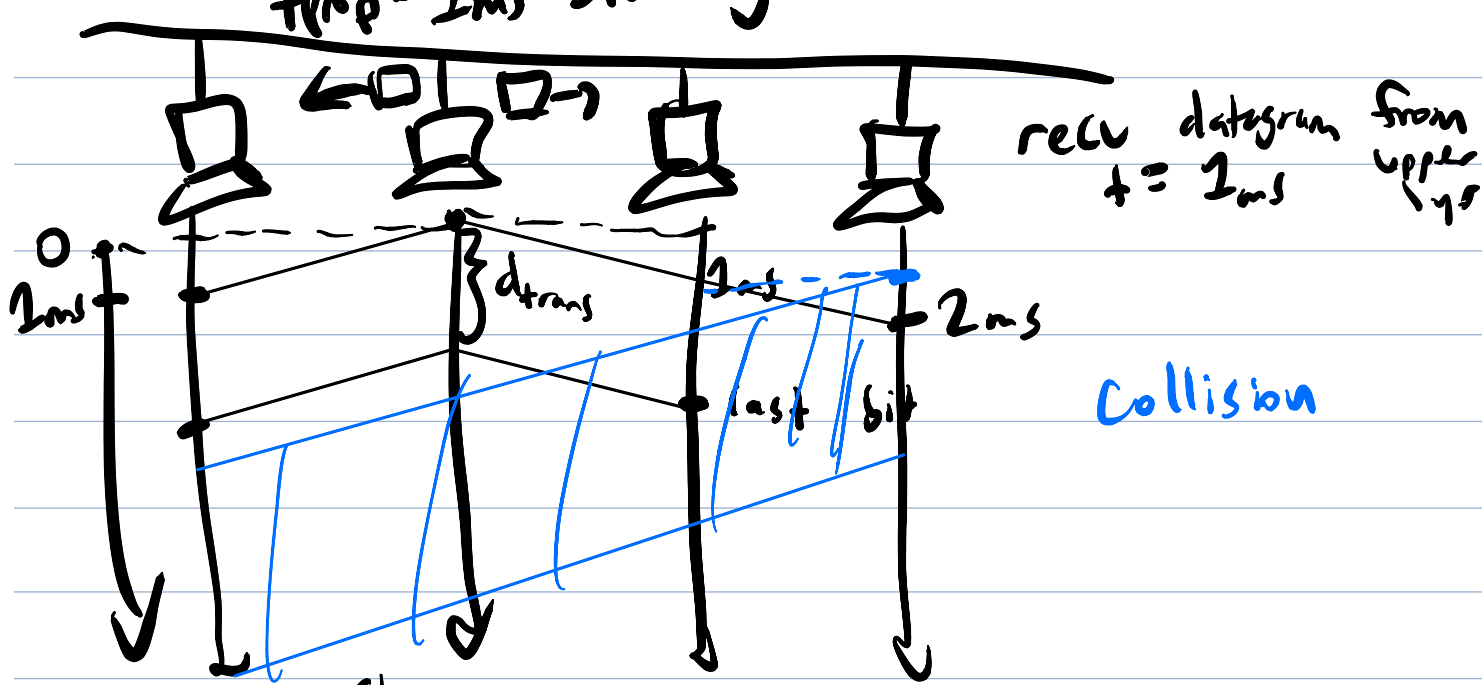
opt. $p$ and $N \to \infty$

$= \frac{1}{2e} \doteq 0.18 \Rightarrow$ worse than slotted Aloha

# Carrier Sense Multiple Access (CSMA)

- listen before tx
- idle: tx entire frame
- busy: defer
- concept: "don't interrupt others"
- collisions can occur due to prop. delay
  - ↳ entire packet tx time wasted
  - ↳ dist. & prop. delay play role in determining P(collision)

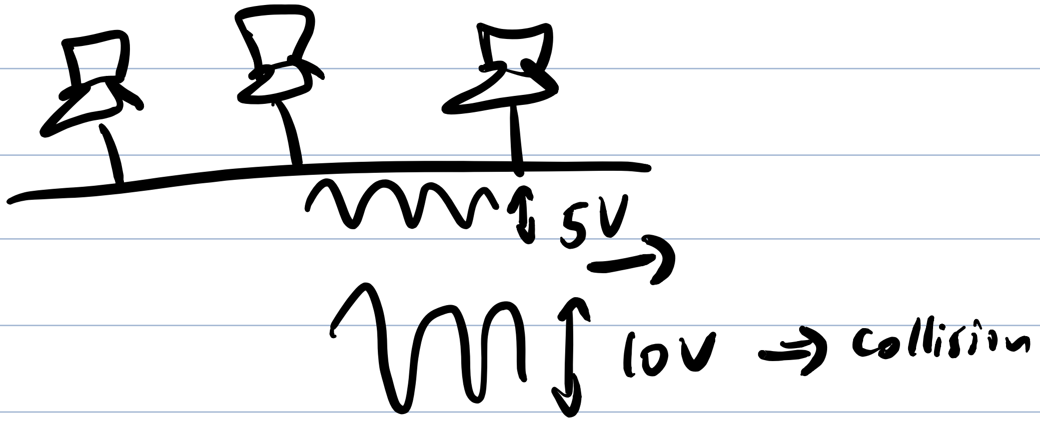
$t_{prop}$ = 1ms b/w adj nodes




- vanilla / 1st version of CSMA

## CSMA/CD

- avoid tx after collision detection
  - ↳ abort tx
- easy for wired media, hard for WLANs (recv strength overwhelmed by local tx)




<u>Ethernet CSMA/CD Algo</u>

1. NIC encapsulates datagram into frame
2. idle ⇒ tx
   busy ⇒ keep sensing until free
3. NIC done w/ frame if no coll. detected
4. abort tx if coll. detected, send jam signal
5. after aborting, NIC enters binary (exponential) backoff
   ↳ $m^{th}$ collision ⇒ NIC randomly selects
   $K \in \{0, 1, \ldots, 2^m - 1\}$
   NIC waits $K \cdot 512$ bit times, then → Step 2
   ⇒ longer backoff interval w/ more collisions

Step 2 senses medium for a period of 96 bit time to allow nodes to process previous packets and do some housekeeping

jamming signal - 48 bits

∴ saturates at 10

$T_p$ = 512 bit times

$$\text{efficiency} = \frac{1}{1 + \frac{5 t_{prop}}{t_{tx}}}$$

$t_{prop}$ → max between 2 nodes

$t_{tx}$ → time for max size frame

efficiency → $1^-$ when $t_{prop} \to 0$

or $t_{trans} \to \infty$

better than aloha: simple, cheap, decentralized