Medium Access Control

COS 598a: Wireless Networking and Sensing Systems

Kyle Jamieson

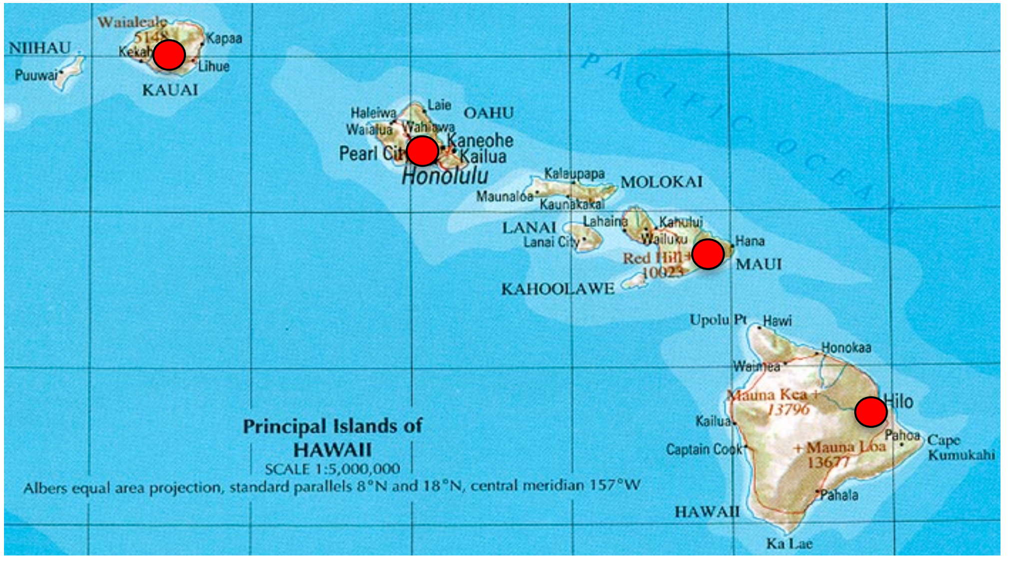
[Parts adapted from B. Karp, S. Shenker, P. Steenkiste]

Medium access: Timeline

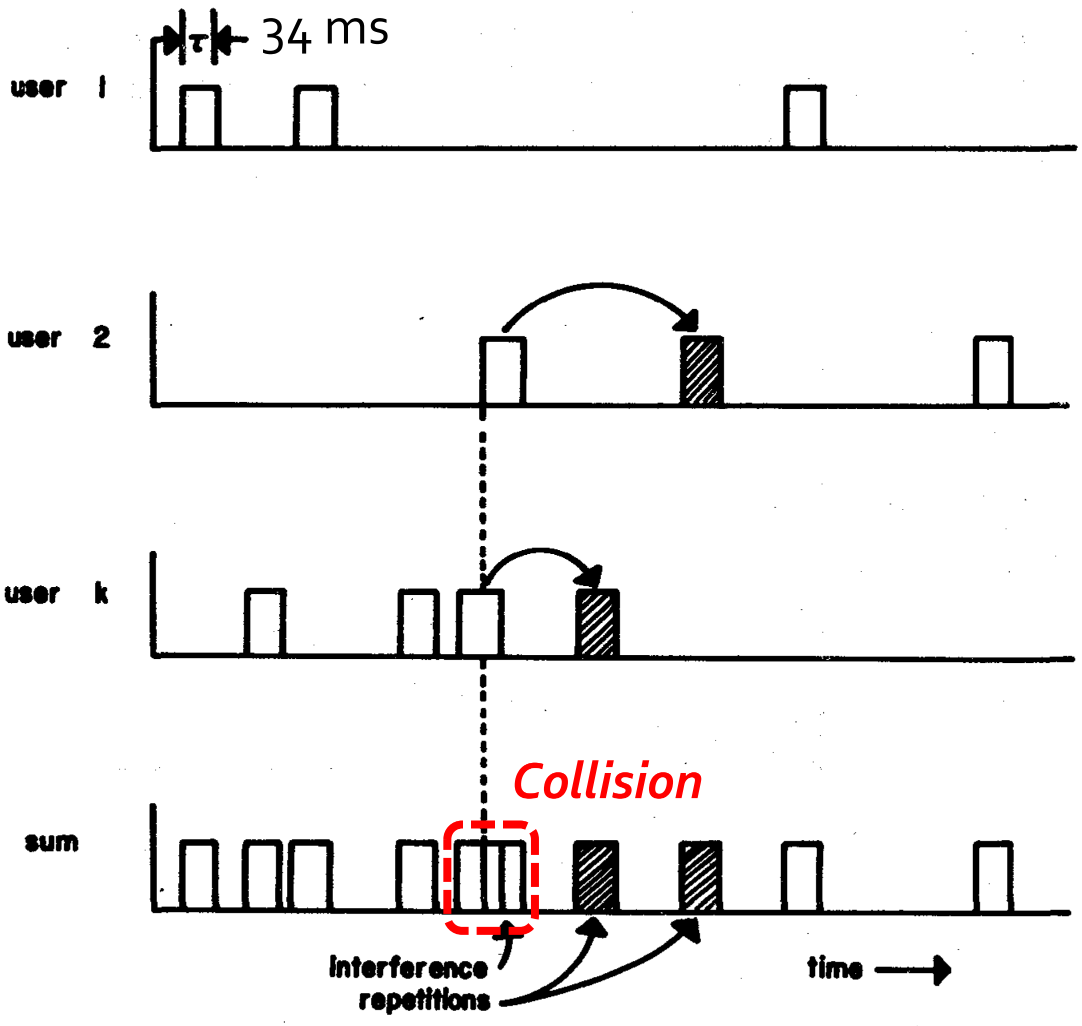
Packet radio	Wireless LAN	Wired LAN	
ALOHAnet			*1960s*
↓			
Amateur packet radio		Ethernet	*1970s*

ALOHAnet: Context

- Norm Abramson, **late 1960s** at the **University of Hawaii**
  - Seven campuses on four islands
  - Want to keep campus terminals in contact with mainframe
  - Telephone costs high, so build (the first) **packet radio** network

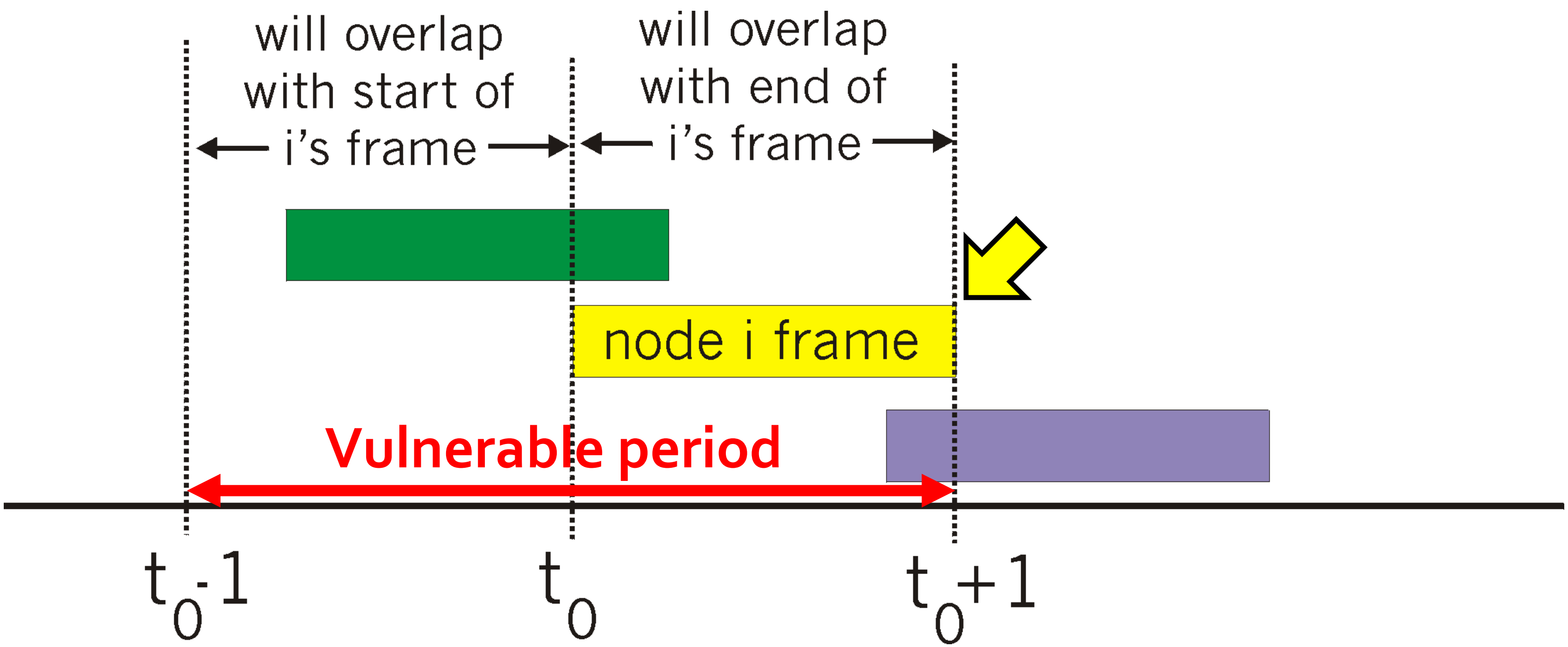


Unslotted ALOHA



Unslotted ALOHA: Performance

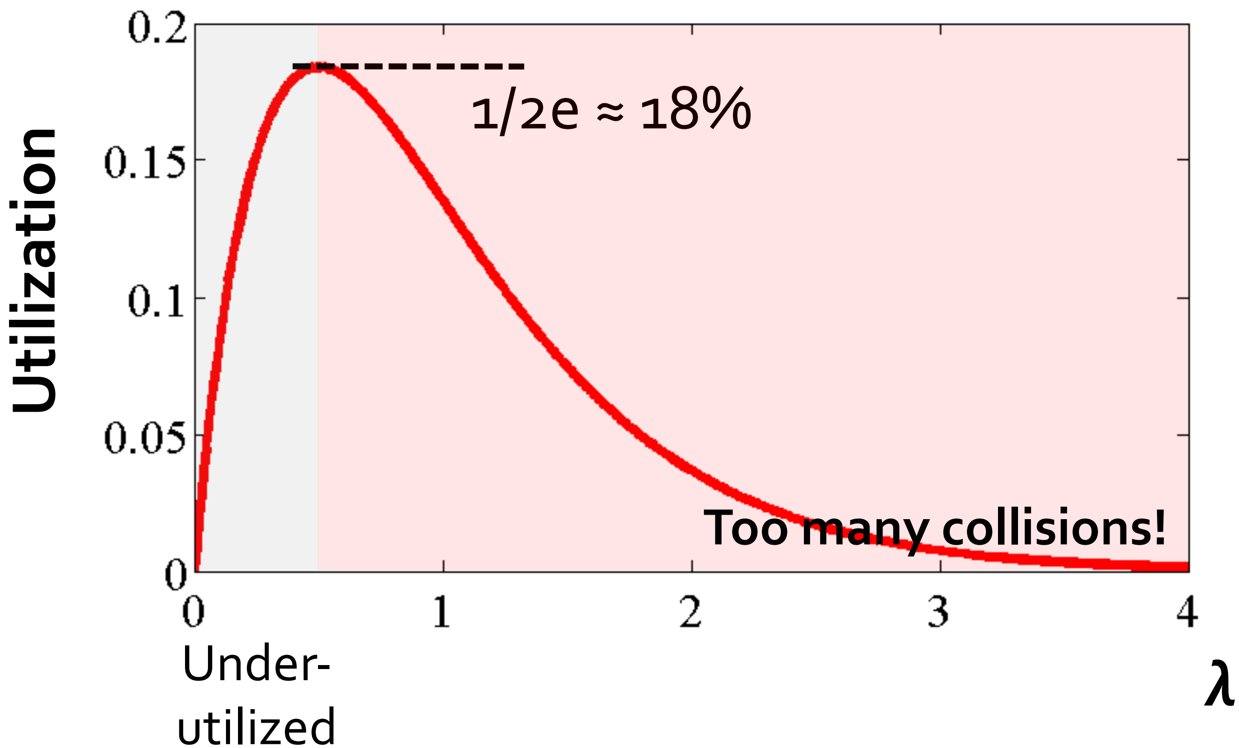
- Suppose some node ***i*** is transmitting; let's focus on **i's frame**



- ***Vulnerable period*** of (normalized) length 2 around i's frame

Unslotted ALOHA: Utilization

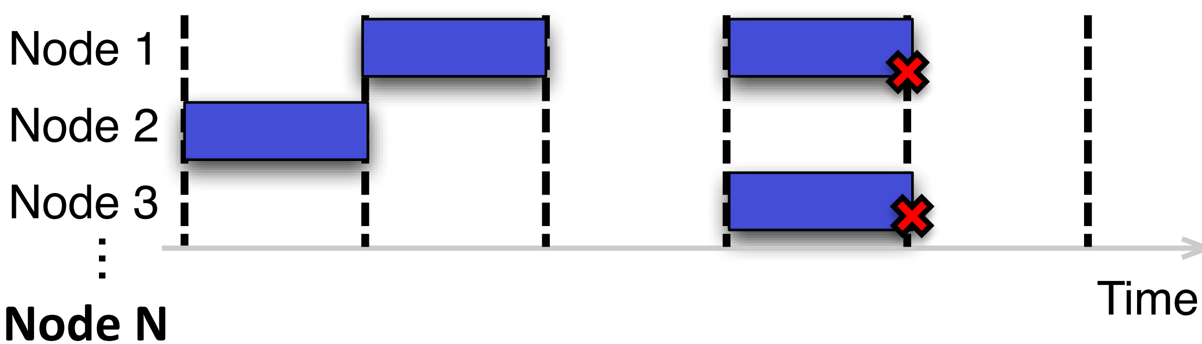
- What fraction of the time is there a non-colliding packet on the medium? This is called ***utilization***
- Utilization: $\lambda \times \Pr(\text{no other transmission in } 2) = \lambda e^{-2\lambda}$



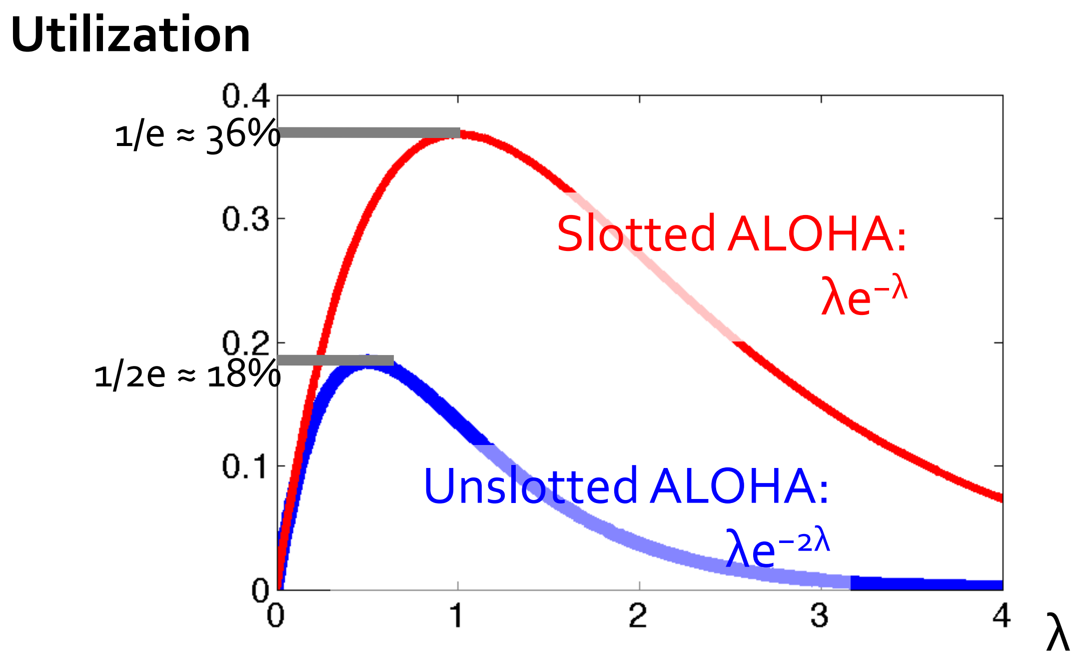
Slotted ALOHA

- **Divide time into slots** of duration 1, **synchronize** so that nodes transmit **only** in a slot
  - Each of N nodes transmits with probability p in each slot
  - So **aggregate transmission rate** $\lambda = N \times p$

- As before, if there is **exactly one** transmission in a slot, can receive; if **two or more** in a slot, no one can receive (**collision**)



ALOHA throughput: slotted versus unslotted



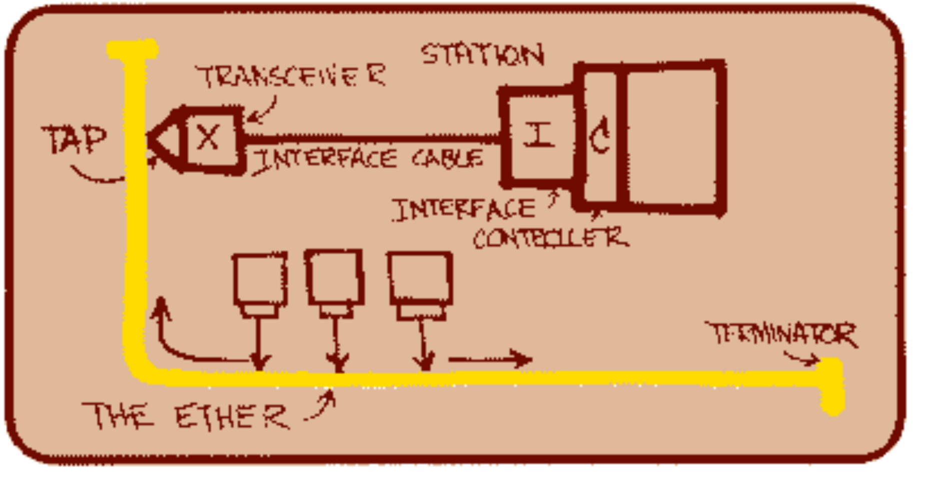
Forcing transmissions into slots → 2× peak medium utilization!

Medium access: Timeline

Packet radio	Wireless LAN	Wired LAN	
ALOHAnet			*1960s*
Amateur packet radio		**Ethernet**	*1970s*

How did the Ethernet get built?

- Bob Metcalfe, PhD student at Harvard in early 1970s
  - Working on protocols for the ARPAnet
  - Intern at Xerox Palo Alto Research Center (PARC), **1973**
  - Needed a way to network the ≈100 *Alto* workstations in-building
  - Adapt ALOHA packet radio
- Metcalfe later founds *3Com*, acquired by HP in April '10 for USD $2.7 bn

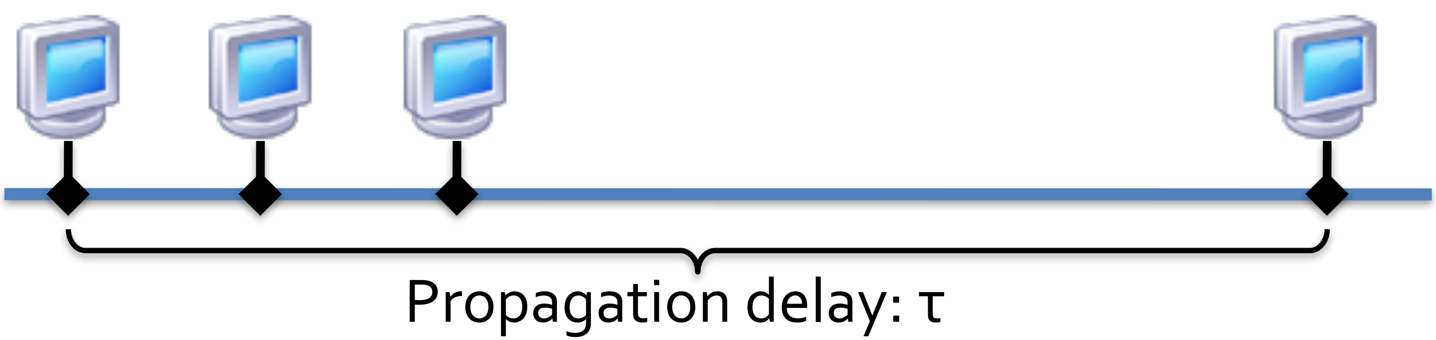


The Ethernet: Physical design

- Coaxial cable, ***propagation delay*** τ
  - Propagation speed: 3/5 × speed of light

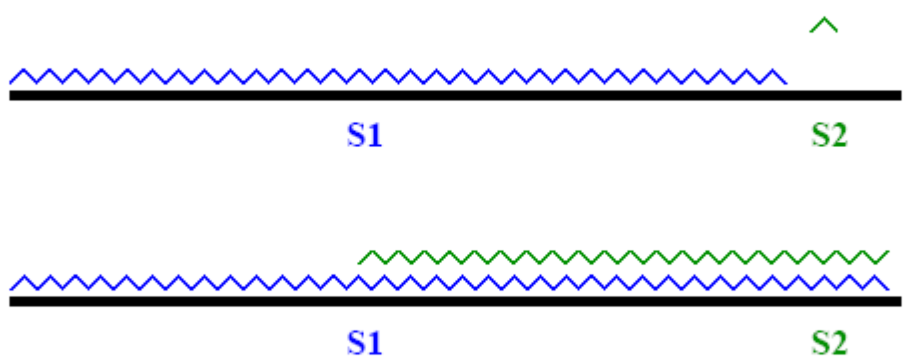
- Experimental Ethernet
  - Data rate: $B = 3$ Mbits/s
  - Maximum length: 1000 m

$$\tau = \frac{10^3 \text{ m}}{\frac{3}{5}\left(3 \times 10^8 \text{ m/s}\right)} \approx 5\ \mu s$$



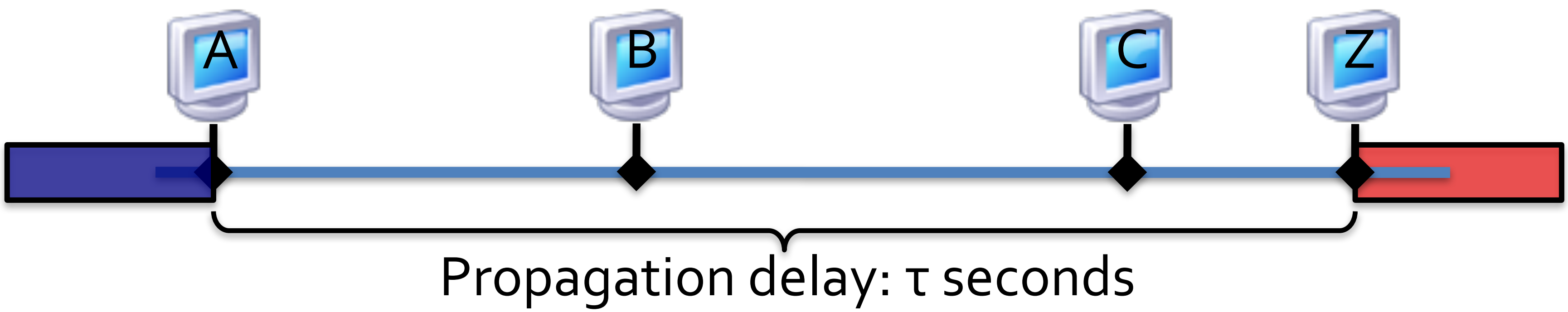
Review: Ethernet MAC

- **CS** (**Carrier Sense**): listen for others' transmissions before transmitting; **defer** to others you hear
- **CD** (**Collision Detection**): as you transmit, listen and **verify** you hear exactly what you send; if not, **abort** and try again later



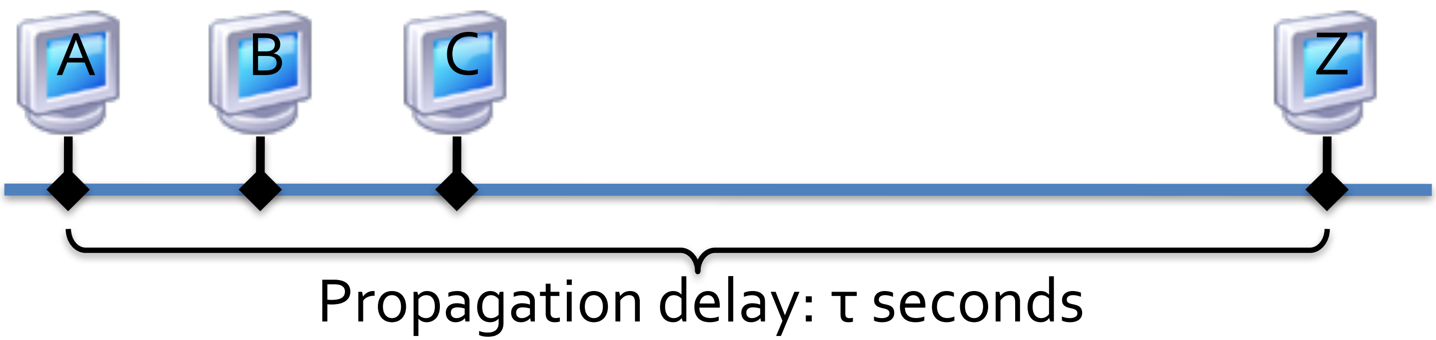
- Is CD possible on a wireless link? Why or why not?

Collisions



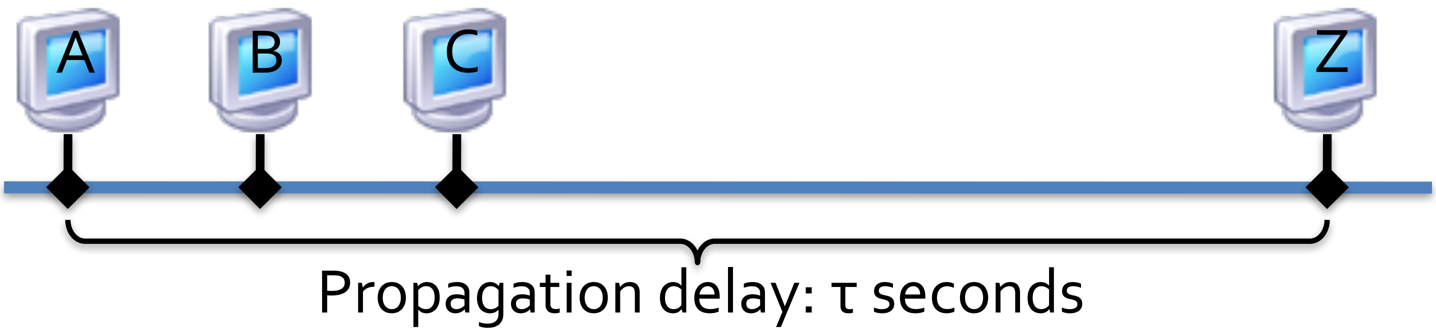
- Packet of *N* bits: *N*/*B* seconds on the wire
- From the perspective of a receiver (**B**):
  - Overlapping packets at **B** means signals sum
  - Not time-synchronized: result is bit errors at **B**
  - No fate-sharing among receivers: **C** receives okay in this example

Collision detection



- Paper isn't clear on this point (authors did have a patent in the filing process)
- Mechanism: monitor average voltage on cable
  - Manchester encoding means your transmission will have a predictable average voltage V_o; others will increase V_o
  - Abort transmission immediately if $V_{measured} > V_o$

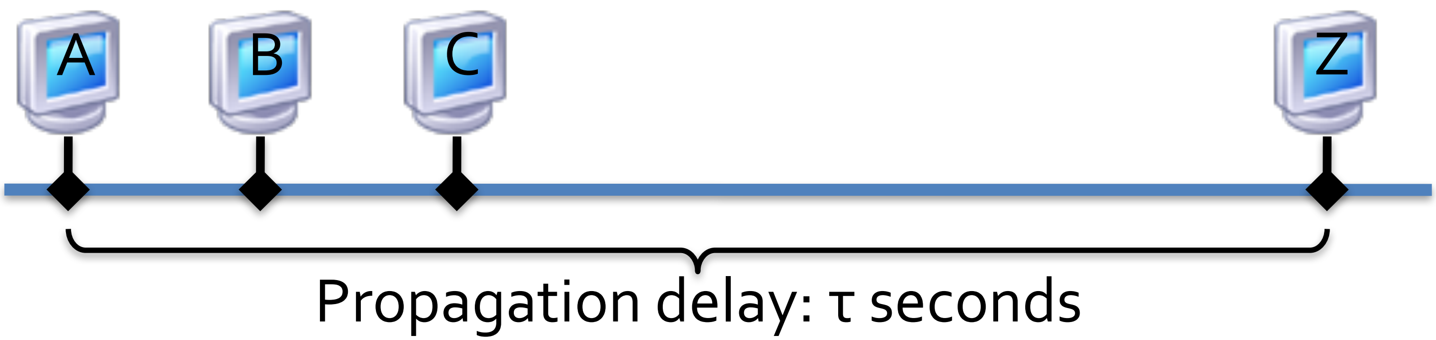
When does a collision happen?



- Suppose Station A begins transmitting at time 0
- Assume that the packet lasts much longer than τ
- All stations sense transmission and **defer** by time τ
  - Don't begin any new transmissions
- At time τ, will a packet be collision-free?

  Only if no other transmissions began before time τ

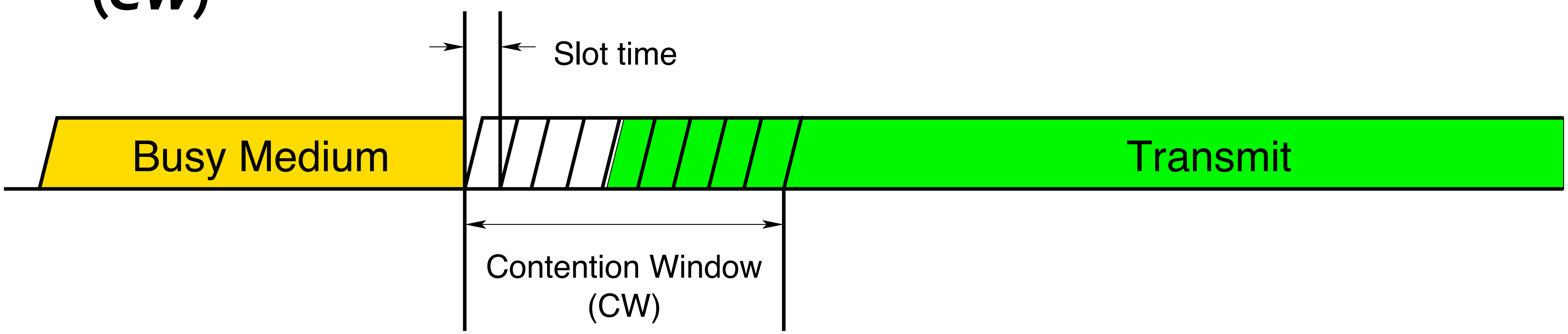
How long does a collision take to detect?



- Suppose Station A begins transmitting at time 0
- τ seconds after Z starts, A hears Z's transmission
- When does A know whether its packet collided or not? At time 2τ

Slotted Ethernet backoff

- Backoff time is **slotted** and **random**
  - Station's view of the where the first slot begins is at the end of the busy medium
  - Random choice of slots within a window, the **contention window (*CW*)**

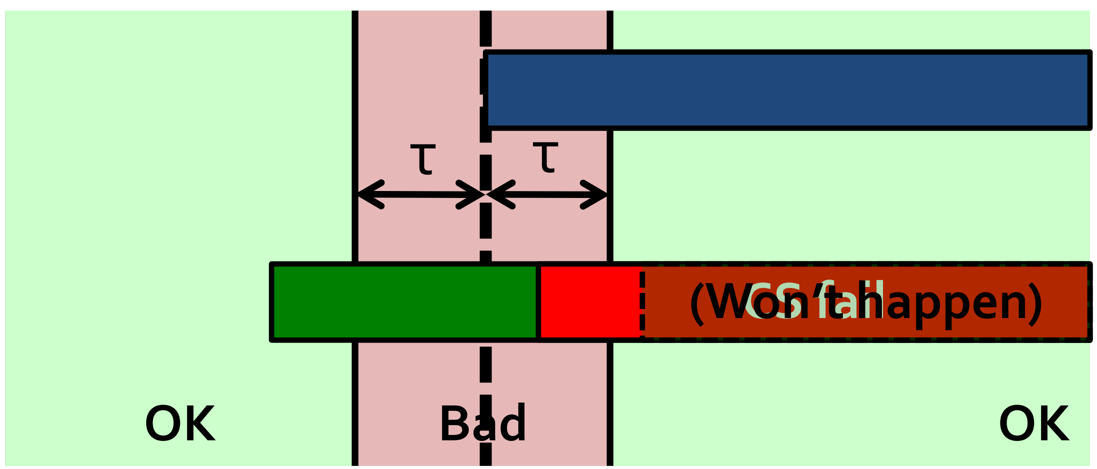


- **Goal:** Choose slot time so that **<u>different</u>** nodes picking **<u>different</u>** slots carrier sense and defer, thus **don't collide**

Picking the length of a backoff slot

- Consider from the perspective of one **packet**
  1. Transmissions beginning > τ before will cause **packet** **to defer**
  2. Transmissions beginning > τ after **will not happen** *(why not?)*
- **Transmissions beginning < time τ apart will collide with packet**

- ***So should we pick a backoff slot length of τ?***

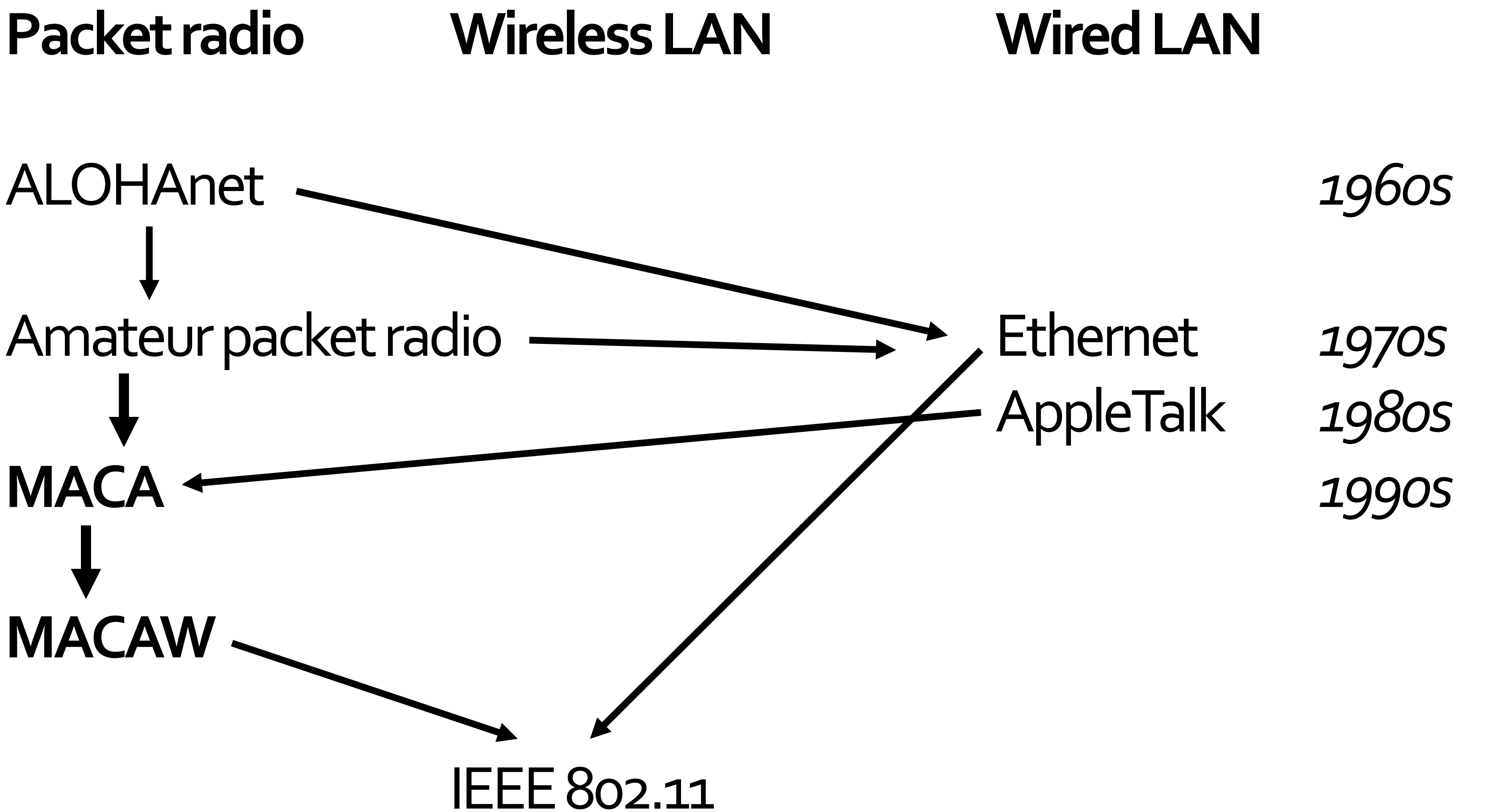


The problem of clock skew

- **No!** Slots are timed off the tail-end of the last packet
  - Therefore, **stations' clocks** differ **by at most τ**
  - This is called ***clock skew*** $\boldsymbol{\Delta}$ ($-\tau < \Delta < \tau$)

- So choose backoff slot length 2 **τ = 10 microseconds**

Medium access: Timeline



Multi-channel

- Suppose we have 100 MHz of spectrum to use for a wireless LAN
- **Strawman:** Subdivide into 50 channels of 2 MHz each: FDMA, narrow-band transmission
  - Radio hardware simple, channels don't mutually interfere, **but**
  - **Multi-path fading** (mutual cancellation of out-of-phase reflections)
  - Base station can allocate channels to users. How do you support **arbitrary communication patterns?**

Single, shared channel

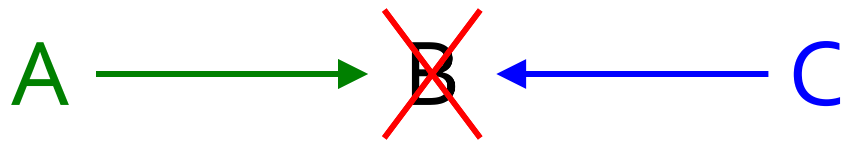
- Spread transmission across whole 100 MHz of spectrum
  - Remove constraint of one communication channel per user
  - Robust to multi-path fading (some frequencies arrive intact)
  - Supports peer-to-peer communication
- **Collisions:** A receiver must hear ≤1 strong transmission at a time

Assumptions and goals

- Assumptions
  - **Uniform, circular** radio propagation
  - Fixed transmit power
  - **Equal interference and transmit ranges**

- What are authors' stated goals?
  - Fairness in sharing of medium
  - Efficiency (total bandwidth achieved)
  - Reliability of data transfer at MAC layer

Hidden Terminal Problem



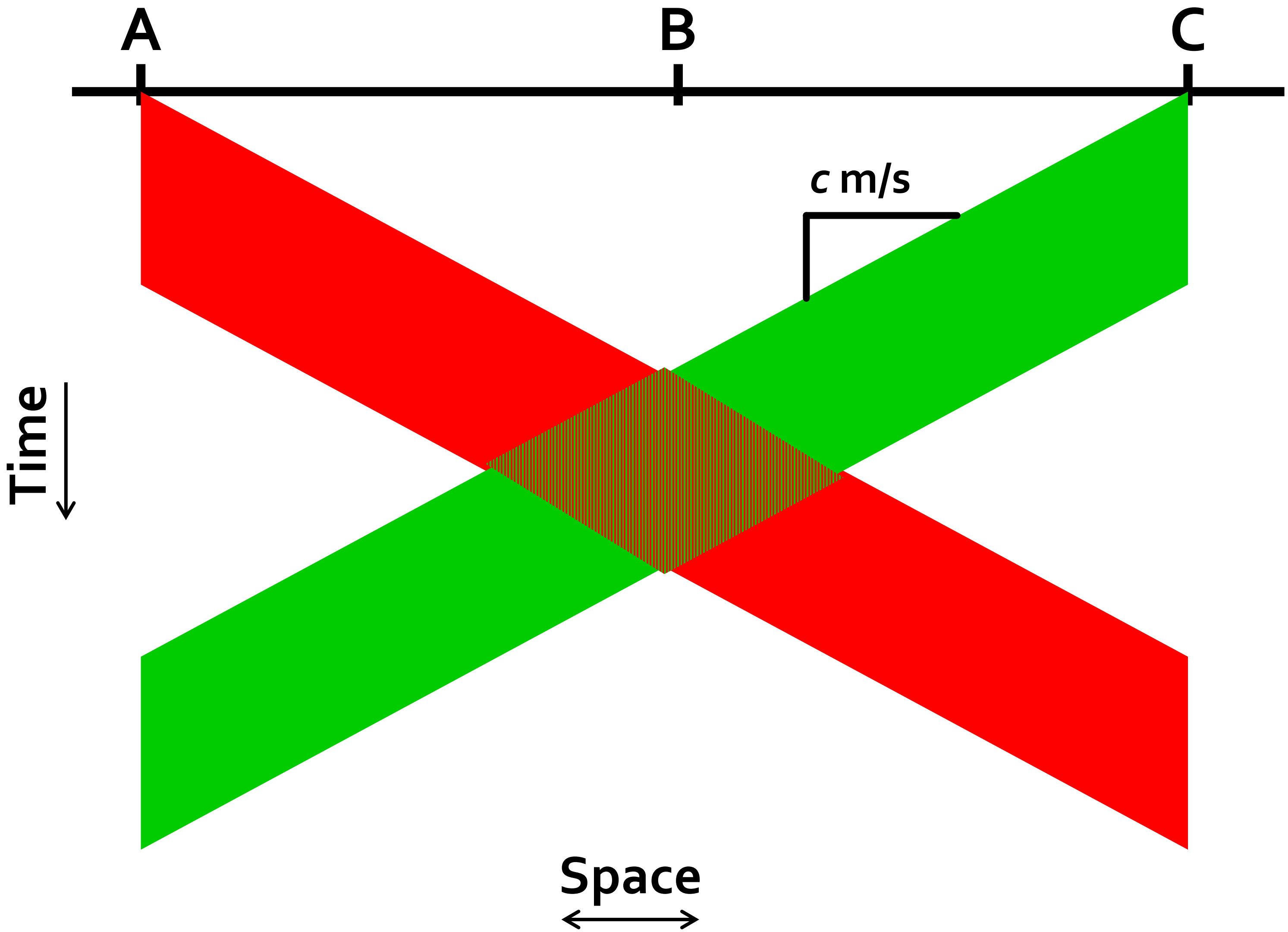
- **CS Multiple Access (CSMA):** nodes listen to determine channel idle before transmitting

- Nodes placed a little less than one radio range apart
  - C can't hear A, so will send while A sends; result: **collision at B**

- **CS insufficient** to detect all transmissions on wireless networks!

- **Key insight:** Spatially, collisions are **located at receiver**

Collisions



Exposed Terminal Problem

- B sends to A; C sends to a node other than B
- If C transmits, does it cause a collision at A?
  - Yet **C cannot transmit** while B transmits to A!

- Same insight: collisions are spatially located at receiver
- One possibility: directional antennas rather than omnidirectional
  Why does this help? Why is it hard?
- Simpler solution: use *receiver's* medium state to determine transmitter behavior

Key Points

- No concept of a "global collision:"
  - Different receivers **hear** different signals
  - Different senders **reach** different receivers

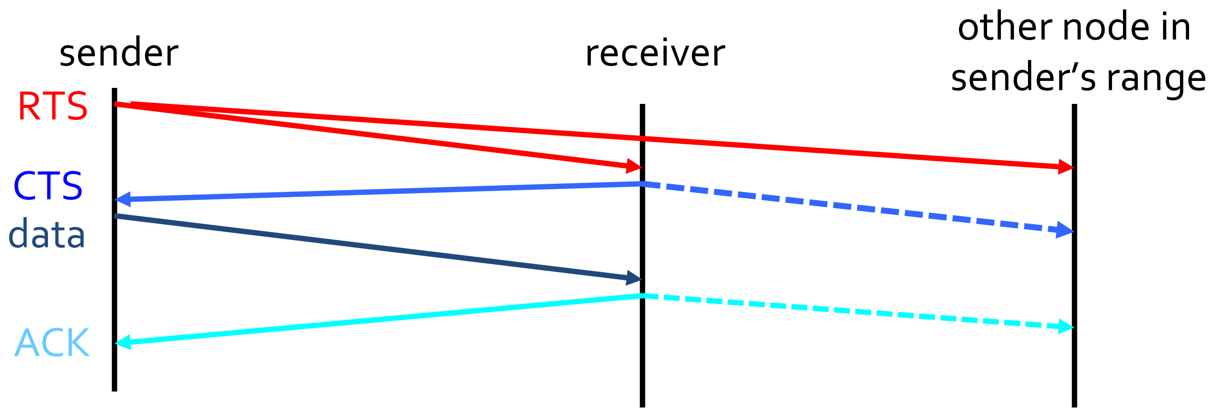
- Collisions are at the receiver, not sender
  - Only care if receiver can hear the sender clearly
    - It does not matter if sender can hear someone else
      - As long as that signal does not interfere with receiver

- Goal of protocol:
  - Detect if receiver can hear sender
  - Tell senders who might interfere with receiver to shut up

Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)

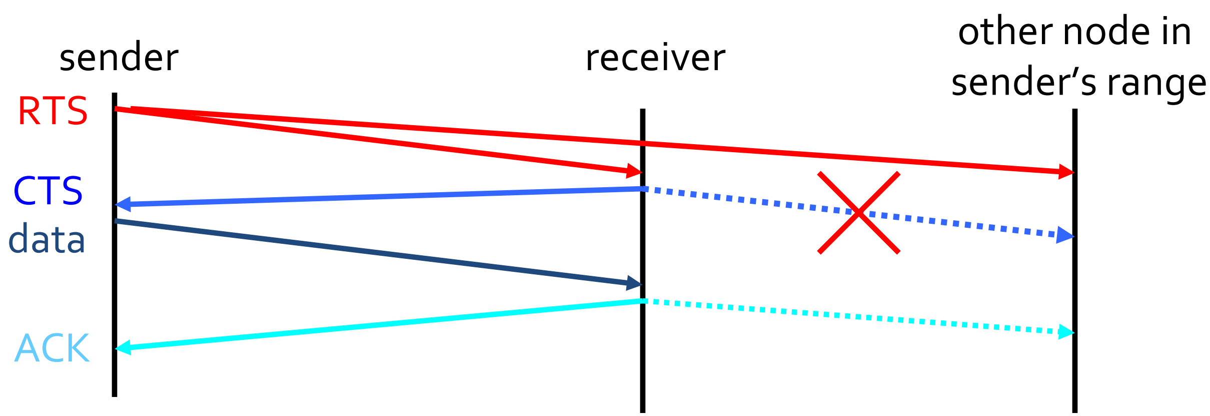
- One of the first uses was the ***AppleTalk*** wired LAN
- Since **can't detect** collisions, we try to ***avoid*** them:
- Before every data transmission
  - Sender sends a Request to Send (RTS) frame containing the length of the transmission
  - Receiver respond with a Clear to Send (CTS) frame
- Overhear **RTS <u>or</u> CTS packet** directed elsewhere?
  - **Defer** until the end of the data transmission

Multiple Access with Collision Avoidance (MACA)



- Before every data transmission
  - Sender sends RTS containing length of transmission
  - Receiver responds with CTS
  - Sender sends data
  - Receiver sends an ACK; now another sender can send data
- When sender doesn't get a CTS back, it assumes **collision**

MACA's collision avoidance

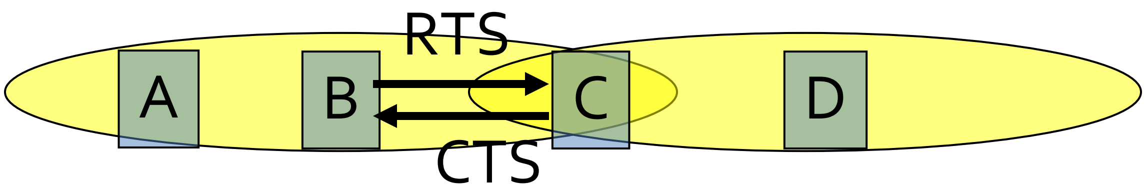


- When you hear **RTS, but not CTS,** **send** (after the CTS finishes)
  - Presumably, sender's **intended destination out of your range**
    - Can cause problems when a CTS is **lost**

- When you hear a CTS, you **keep quiet** until scheduled transmission is over (hear ACK)

RTS / CTS Protocols (MACA)

B sends to C



Overcomes hidden terminal problems with contention-free protocol

1. B sends to C Request To Send (RTS)
2. A hears RTS and defers (to allow C to answer)
3. C replies to B with Clear To Send (CTS)
4. D hears CTS and defers to allow the data
5. B sends to C

RTS/CTS in MACA and MACAW

- RTS/CTS solves hidden terminal problem!

- What happens if CTSs get lost? RTSs collide *themselves*?

- No CTS reply
- The sender **must persist:**
  - But at times of high load, "back off"
    - **Idea from Ethernet:** Sender backs off exponentially (**BEB**) before retrying

BEB in MACA

- Maintain a current **backoff window** of duration B
  - Maximum size B_M; minimum size B_o

- MACA sender:
  - $B_o = 2$ and $B_M = 64$
  - Upon successful RTS/CTS, $B \leftarrow B_o$
  - Upon failed RTS/CTS, $B \leftarrow \min[2B, B_M]$

- Before retransmission, wait a uniform random number of RTS lengths (30 bytes) in [0,B]

- No carrier sense! (Karn concluded useless because of hidden terminals)

BEB leads to unfairness

- BEB can lead to **unfairness**

- Simple example: two senders sending to the same receiver, each sending at a rate that can alone saturate the network
  - After a collision the one with smaller B is more likely to win
    - Thus resetting its $B \leftarrow B_0$
      - Thus more likely to win next time

  - One with smaller *B* has decreasing chance to acquire medium

- **Result:** One sends at channel capacity, other zero throughput

BEB in MACAW: "Copy" mechanism

- MACAW proposal: senders write their *B* into packets; upon hearing a packet, "copy" its *B*
  - **Result:** dissemination of congestion level of "winning" transmitter to its competitors

- Is this a good idea?
- RTS failure rate at one node propagates far and wide
- Ambient noise? Regions with different loads?

Reliability: ACK

- MACA relies on transport layer for reliability
  - **Significant wait** for recovery of lost DATA packets

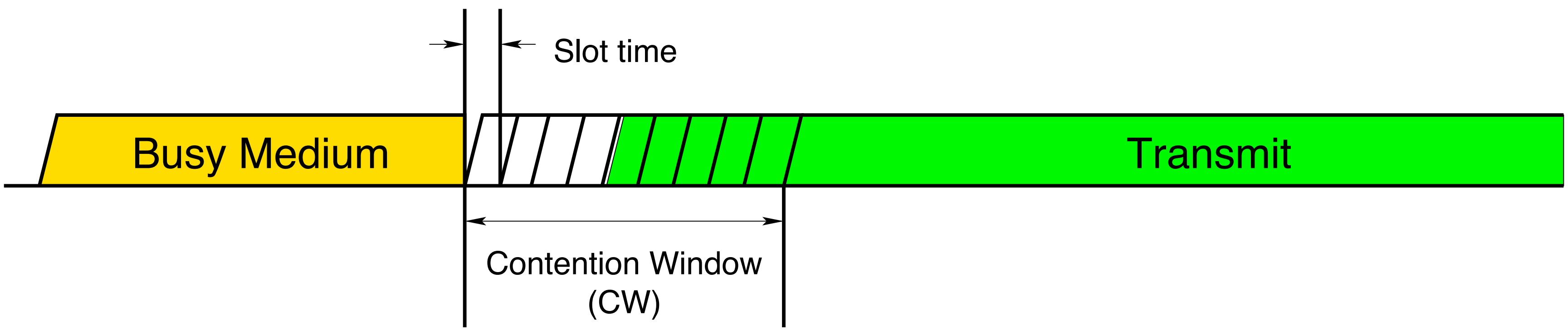
- MACAW introduces an **ACK** after DATA packets
  - Sender **retransmits** if RTS/CTS succeeds but no ACK returns; doesn't back off

  - Avoids TCP window reductions when interference present

  - Are ACKs useful for broadcast packets?
    - Consequences for, *e.g.*, ARP?

802.11 backoff with physical carrier sense

- Backoff time is **slotted** and **random**
  - Station's view of the where the first slot begins is at the end of the busy medium
  - Random choice of slots within a window, the **contention window (*CW*)**

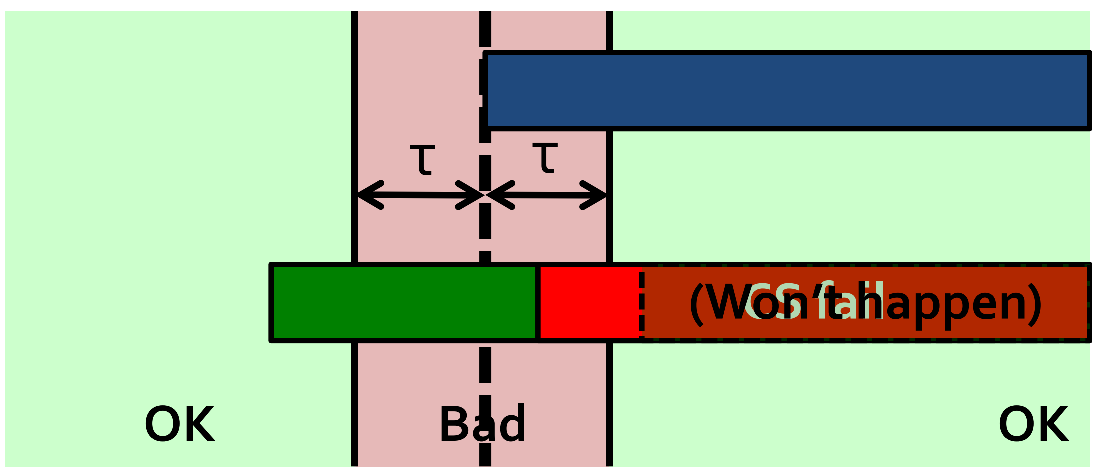


- **Goal:** Choose slot time so that **<u>different</u>** nodes picking **<u>different</u>** slots carrier sense and defer, thus **don't collide**

Picking the length of a backoff slot

- Consider from the perspective of one **packet**
  1. Transmissions beginning > τ before will cause **packet to defer**
  2. Transmissions beginning > τ after **will not happen** *(why not?)*
- **Transmissions beginning < time τ apart will collide with packet**

- ***So should we pick a backoff slot length of τ?***



MACAW and 802.11 Differences

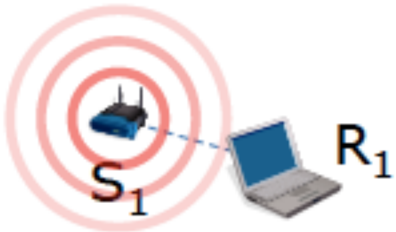
- 802.11 uses physical CS before transmissions and defers a uniform random number of slots, in [0, *B*]
  - Sets timer to count down random period
    - Timer **pauses** when carrier sensed, **continues** when channel idle
  - Packet transmitted when timer reaches zero

- 802.11 combines physical CS with virtual CS from RTS/CTS packets in the Network Allocation Vector (NAV)

- 802.11 uses BEB when an ACK doesn't return

Two regimes: Concurrency v. taking turns

- Far-apart links should send concurrently:



- Near links should time-multiplex:

Carrier sense attempts to distinguish these cases

Uses energy threshold to determine if medium occupied

What about cases in between these extremes?

[Figures: Micah Brodsky]

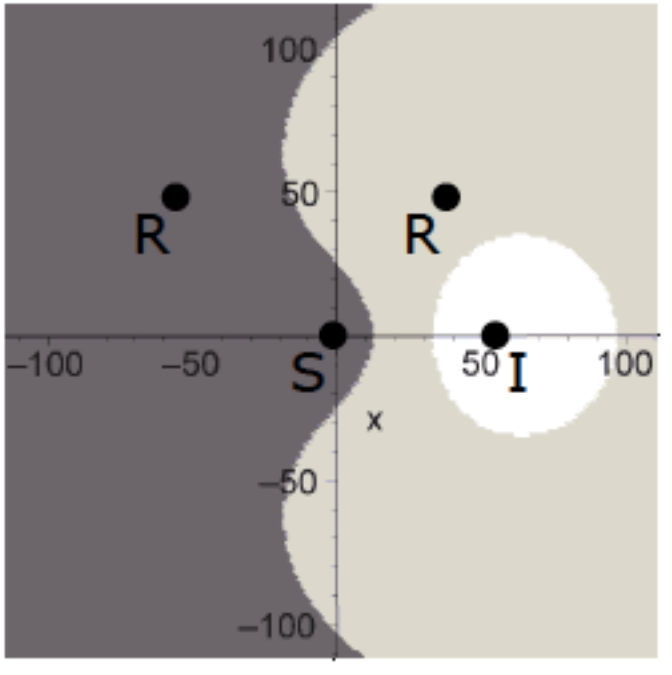
When does carrier sense work well?

- Agreement:
  - If two senders and two receivers, and both receivers achieve highest throughput when both use concurrency or both use multiplexing, the links agree
- Far-apart links agree on concurrency
- Near links agree on time-multiplexing
- In between, risk links don't agree; CS may not work well

Simulation study of carrier sense
[Brodsky and Morris, SIGCOMM '09]

- Place sender S and interferer I at fixed locations
- Place receiver from S uniformly at random within some radius of S
- Compare throughputs at receiver over all locations
- Vary distance between sender and interferer

Individual receivers

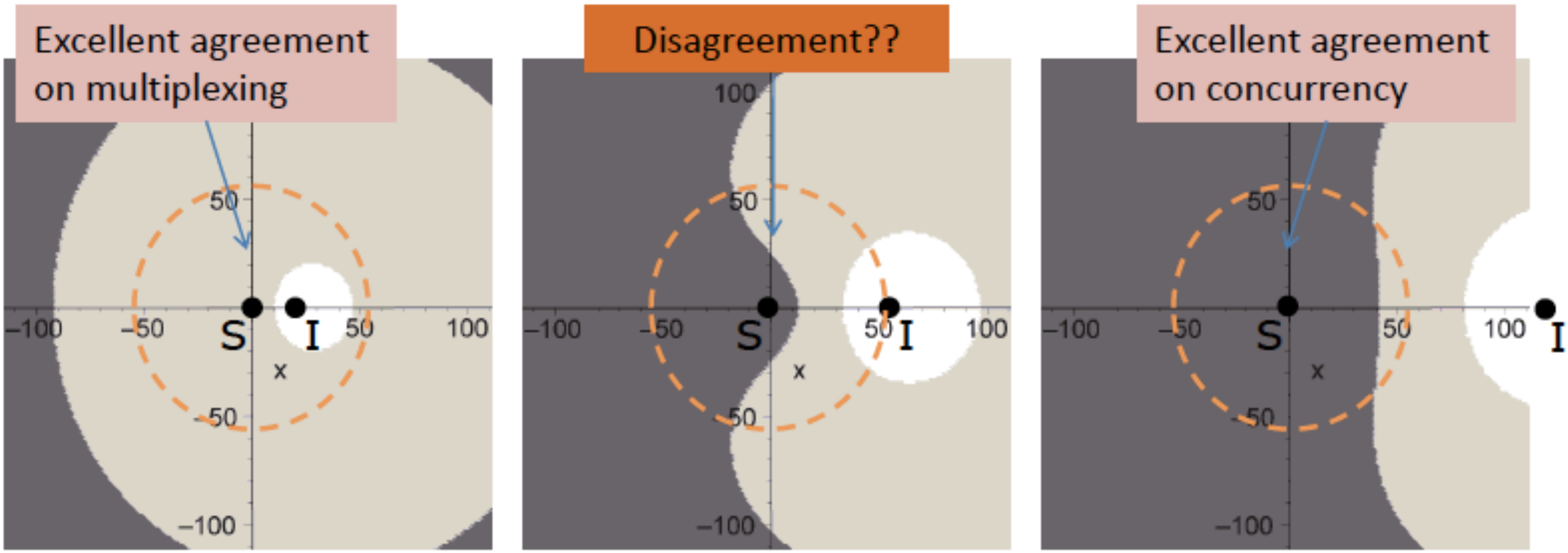


S to I distance $D = 55$

- ■ Prefers concurrency
- □ Prefers multiplexing
- □ Starved w/o multiplexing

Receiver location only matters in the "transitional" case

Receiver preference vs. position:



D = 20 | D = 55 | D = 120

■ Prefers concurrency

□ Prefers multiplexing

□ Starved w/o multiplexing

802.11: A Dose of Reality

- The canonical wireless link in the research community. Why?
  - Hardware commoditized, cheap
  - First robust wireless network with LAN-like bitrate

- In practice no one uses RTS/CTS!
  - Take-away from prior slides: CS works pretty well
  - Have I been wasting your time?

- Why? Are MACAW and the hidden terminal problem irrelevant?

802.11, Base Stations, and Hidden Terminals

- To first order, everyone uses base stations, not peer-to-peer 802.11 networks
  - When base station transmits, there can be no hidden terminals within one LAN. Why?
- Clients can be hidden from one another. But what's the usual packet output stream of a wireless client (e.g., laptop)? Packet sizes? TCP ACKs; short packets.
- What's the cost of RTS/CTS? How big are RTS and CTS packets? Greatest cost when RTS/CTS same size as data

Topic for next time:

Bit rate adaptation

Mesh networking

Your task:

Read papers, file HotCRP reviews