Multiple Access

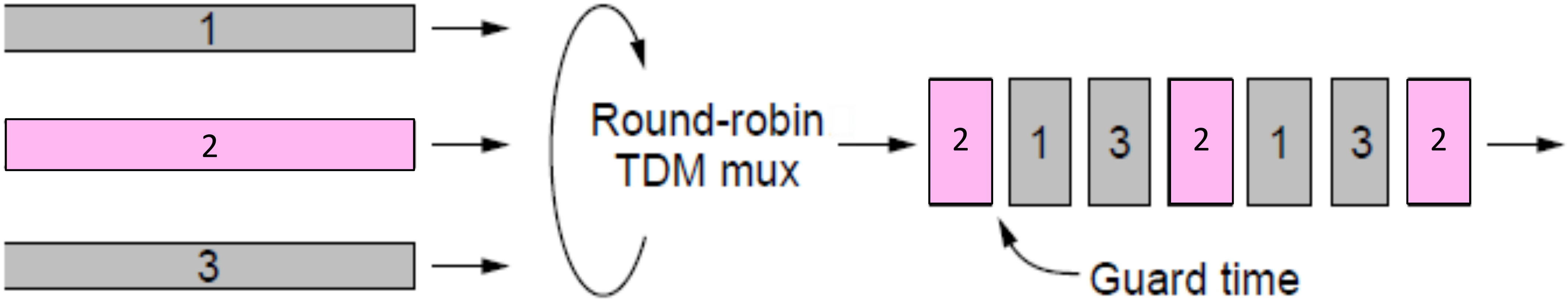
Topic

- Multiplexing is the network word for the sharing of a resource

- Classic scenario is sharing a link among different users
  - Time Division Multiplexing (TDM)
  - Frequency Division Multiplexing (FDM)

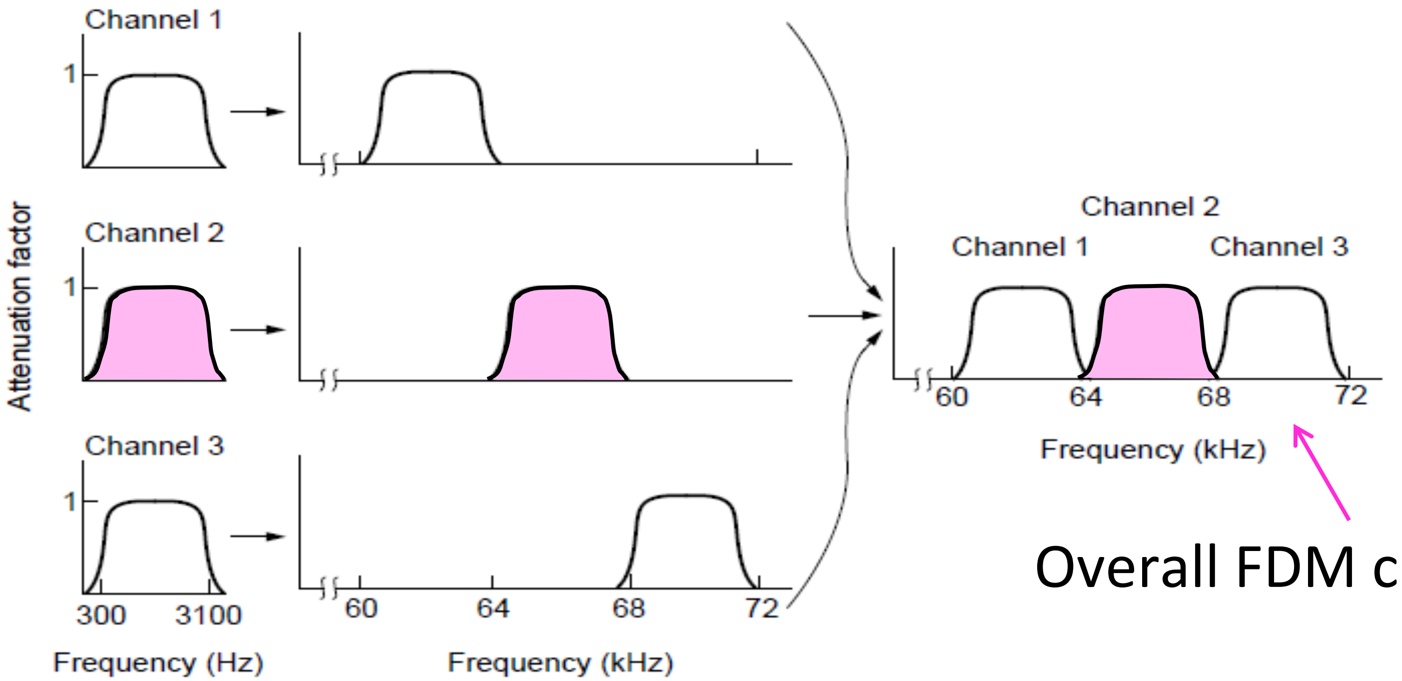
Time Division Multiplexing (TDM)

- Users take turns on a fixed schedule



Frequency Division Multiplexing (FDM)

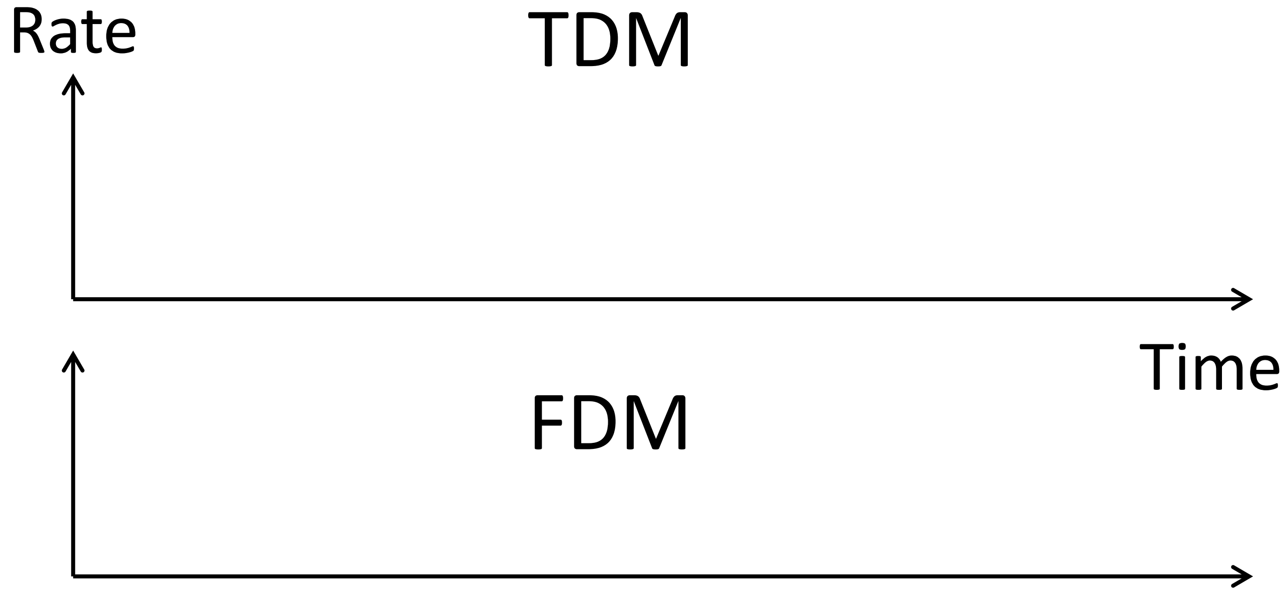
- Put different users on different frequency bands



Overall FDM channel

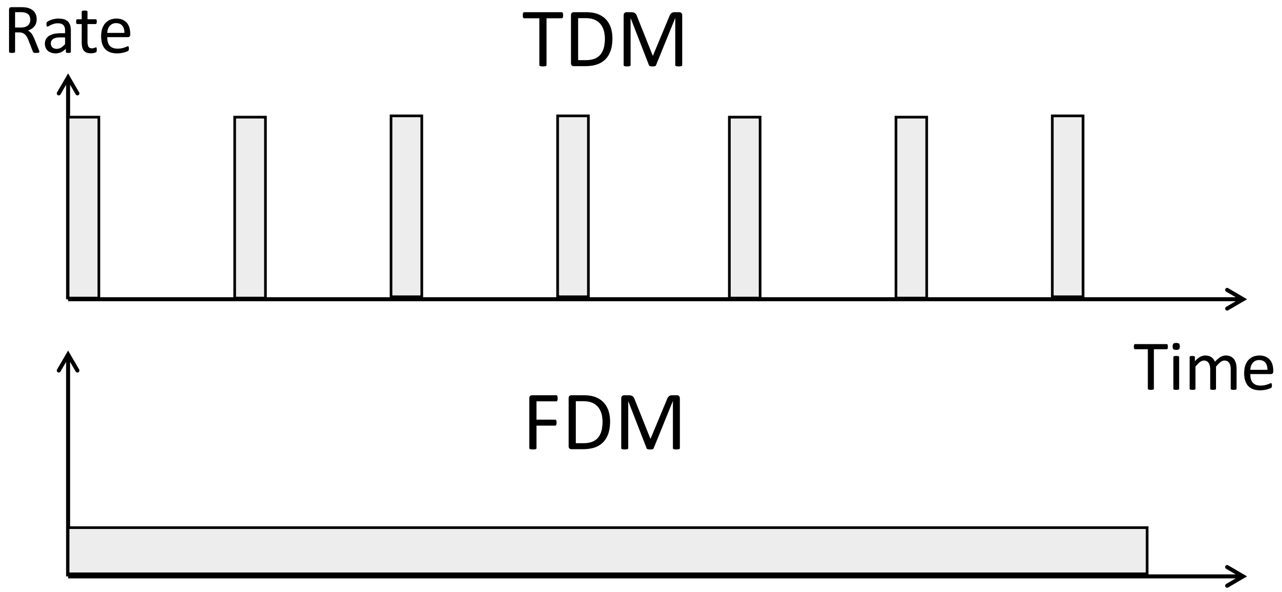
TDM versus FDM

- In TDM a user sends at a high rate a fraction of the time; in FDM, a user sends at a low rate all the time



TDM versus FDM (2)

- In TDM a user sends at a high rate a fraction of the time; in FDM, a user sends at a low rate all the time

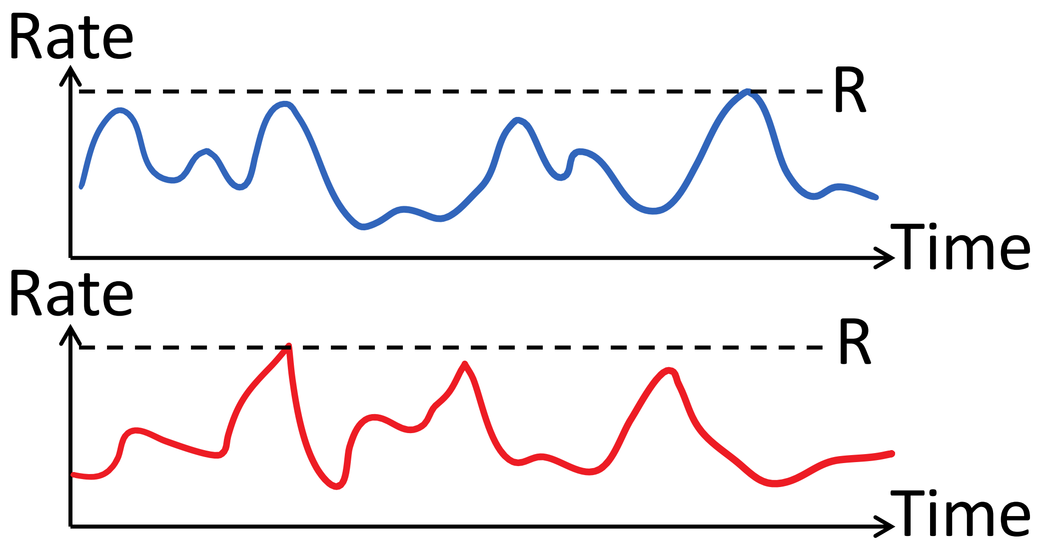


TDM/FDM Usage

- Statically divide a resource
  - Suited for continuous traffic, fixed number of users
- Widely used in telecommunications
  - TV and radio stations (FDM)
  - GSM (2G cellular) allocates calls using TDM within FDM

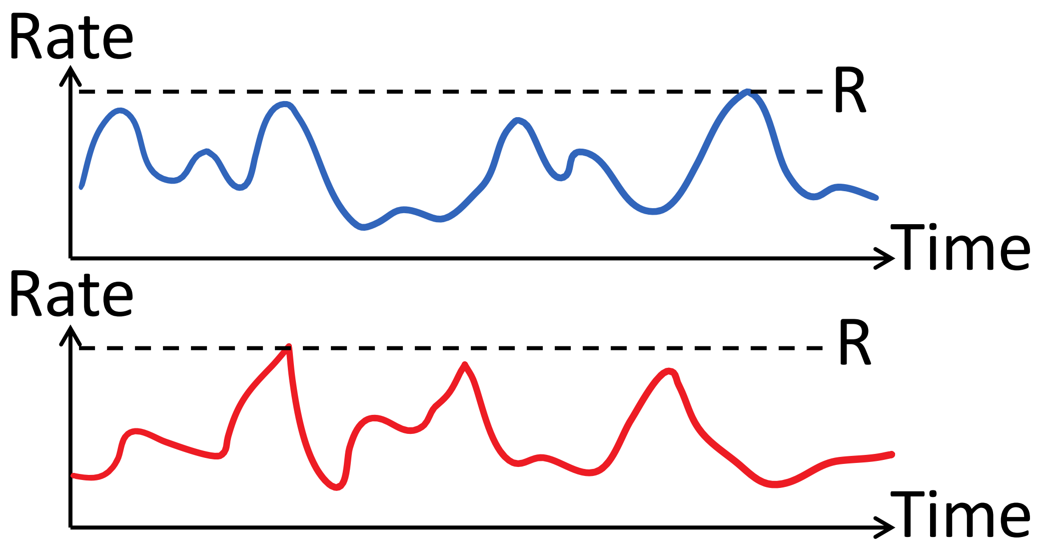
Multiplexing Network Traffic

- Network traffic is <u>bursty</u>
  - ON/OFF sources
  - Load varies greatly over time



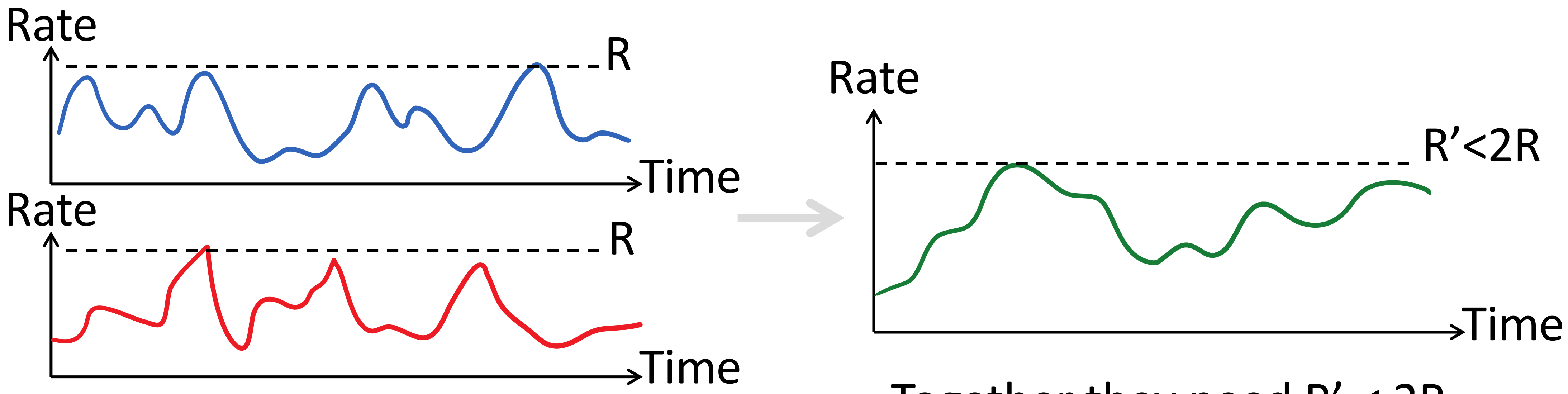
Multiplexing Network Traffic (2)

- Network traffic is <u>bursty</u>
  - Inefficient to always allocate user their ON needs with TDM/FDM



Multiplexing Network Traffic (3)

- Multiple access schemes multiplex users according to demands – for gains of statistical multiplexing



Two users, each need R

Together they need R' < 2R

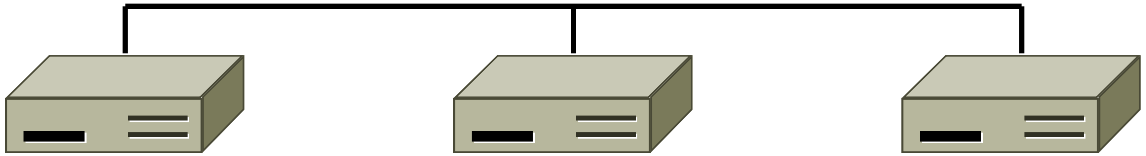
How to control?

Two classes of multiple access algorithms

- **Centralized**: Use a “Scheduler” to pick who transmits and when
  - Scales well and is usually efficient, but requires setup and management
  - Example: Cellular networks (tower coordinates)
- **Distributed**: Have participants “figure it out” via some mechanism
  - Operates well under low load and easy set up but scaling efficiently is hard
  - Example: WiFi networks

Distributed (random) Access

- How do nodes share a single link? Who sends when?
  - Explore with a simple model



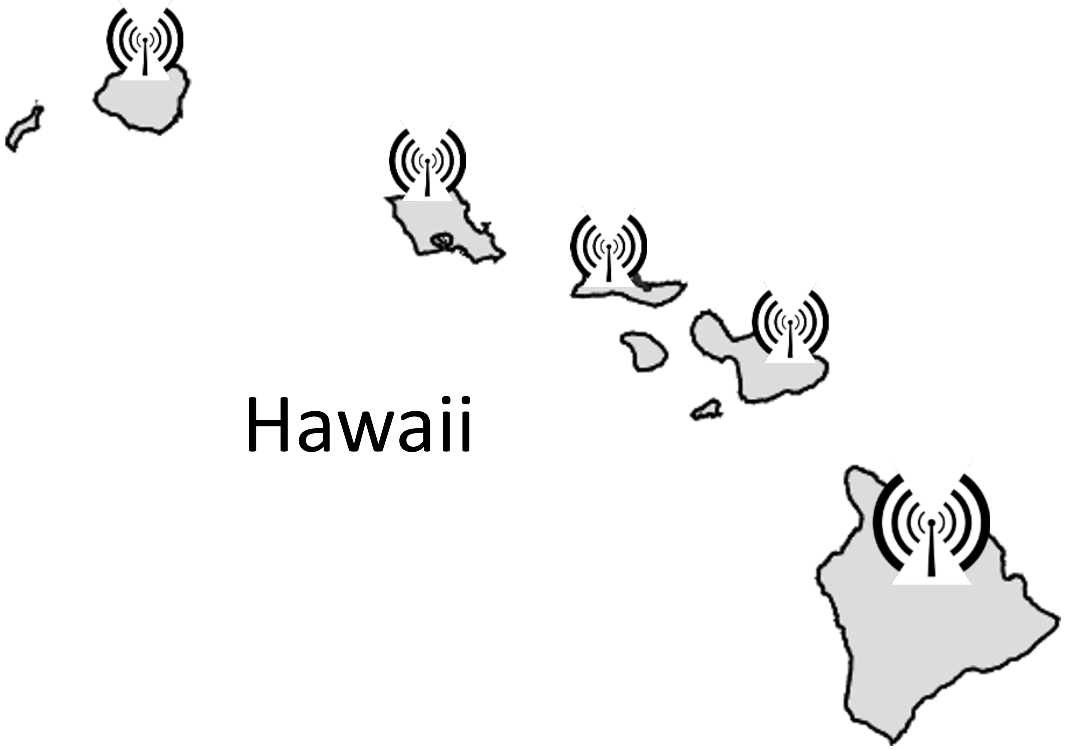
- Assume no-one is in charge
  - Distributed system

Distributed (random) Access (2)

- We will explore random multiple access control (MAC) protocols
  - This is the basis for classic Ethernet
  - Remember: data traffic is bursty

ALOHA Network

- Seminal computer network connecting the Hawaiian islands in the late 1960s
  - When should nodes send?
  - A new protocol was devised by Norm Abramson …



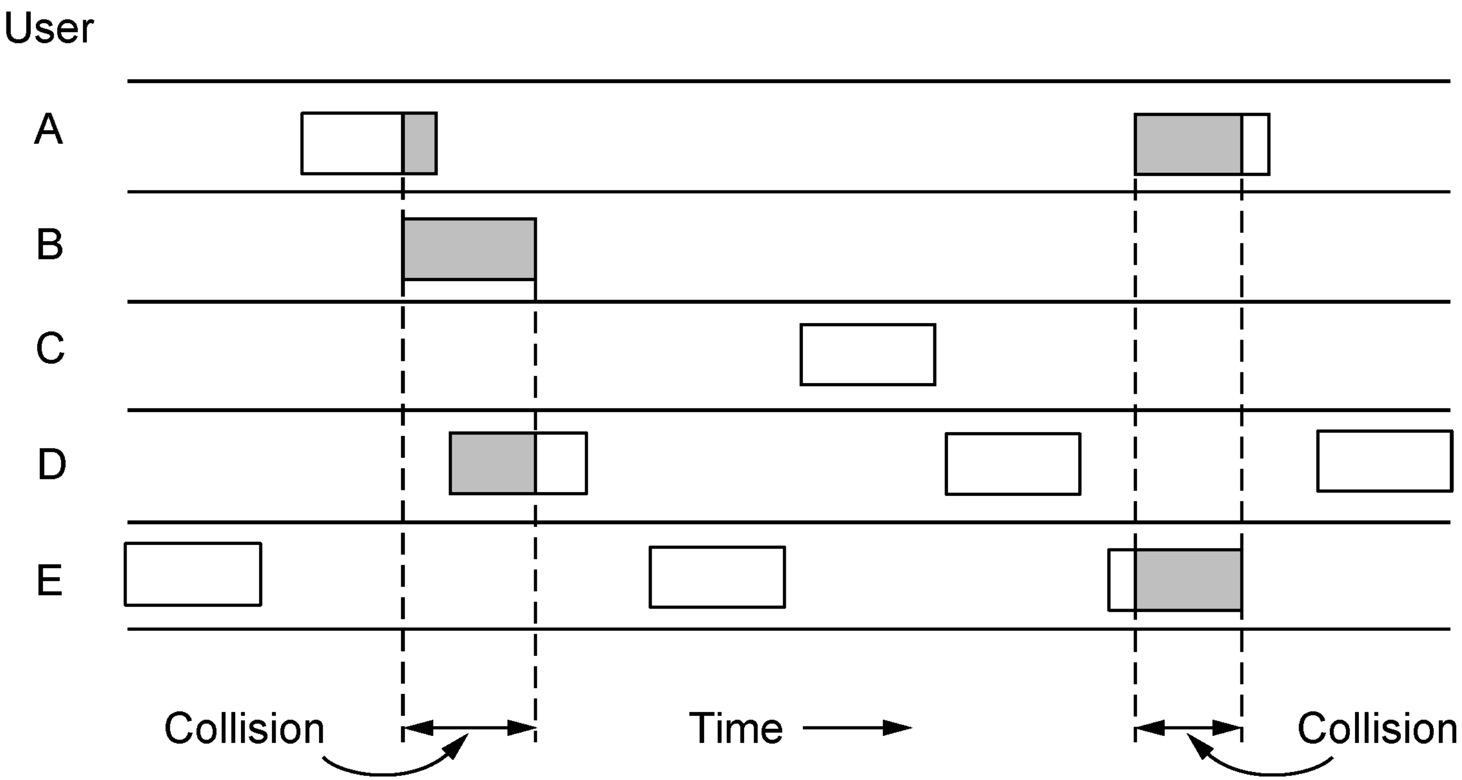
ALOHA Protocol

- Simple idea:
  - Node just sends when it has traffic.
  - If there was a collision (no ACK received) then wait a random time and resend
- That’s it!

ALOHA Protocol (2)

- Some frames will be lost, but many may get through…

- Limitations?



ALOHA Protocol (3)

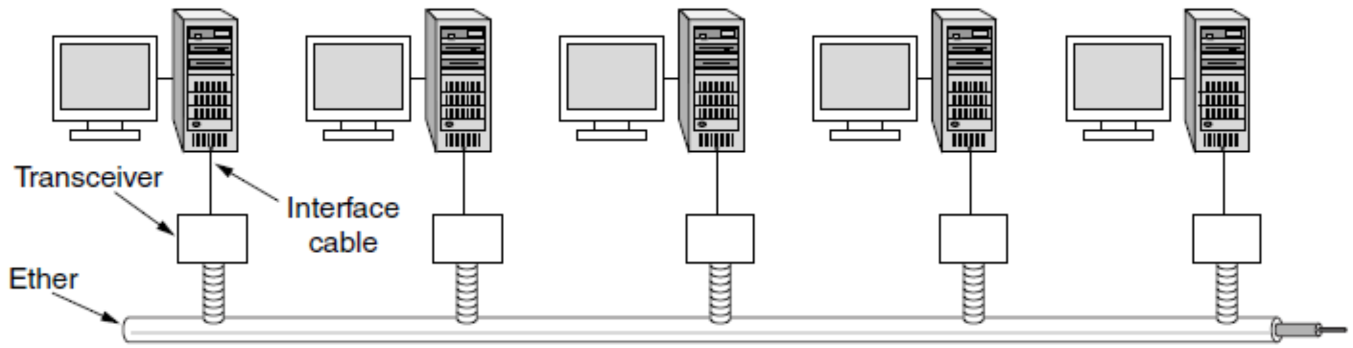
- Simple, decentralized protocol that works well under low load!

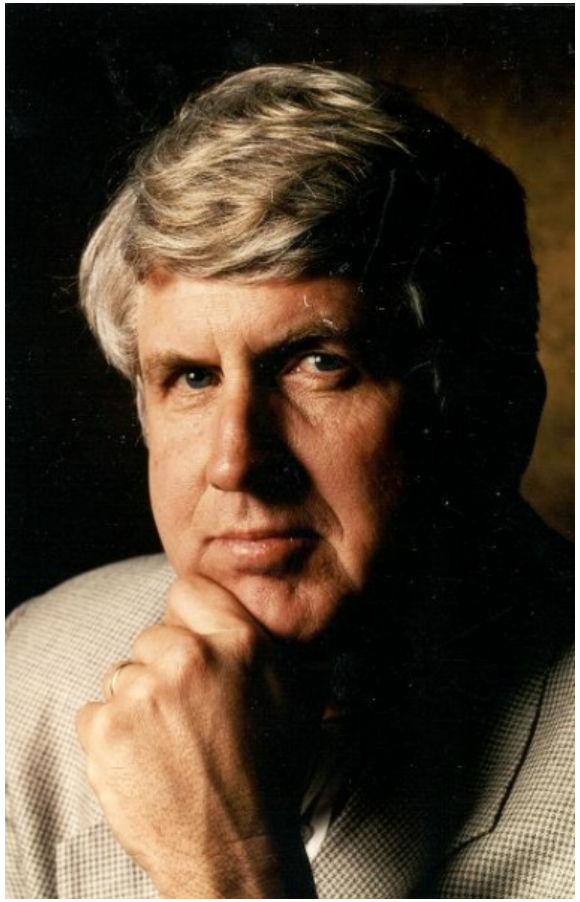
- Not efficient under high load
  - Analysis shows at most 18% efficiency
  - Improvement: divide time into slots and efficiency goes up to 36%

- We'll look at other improvements

Classic Ethernet

- ALOHA inspired Bob Metcalfe to invent Ethernet for LANs in 1973
  - Nodes share 10 Mbps coaxial cable
  - Hugely popular in 1980s, 1990s





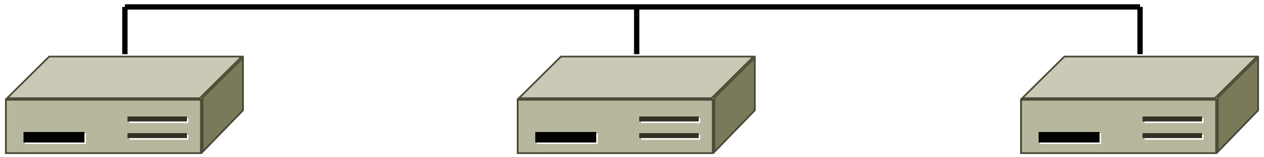
: © 2009 IEEE

CSMA (Carrier Sense Multiple Access)

- Improve ALOHA by listening for activity before we send (Doh!)
  - Can do easily with wires, not wireless
- So does this eliminate collisions?
  - Why or why not?

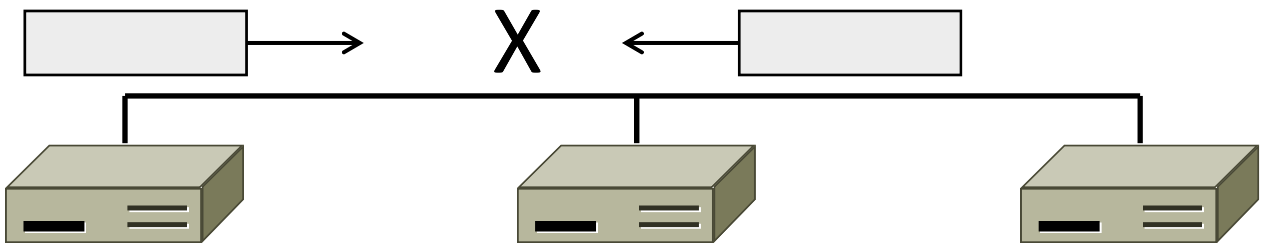
CSMA (2)

- Still possible to listen and hear nothing when another node is sending because of delay



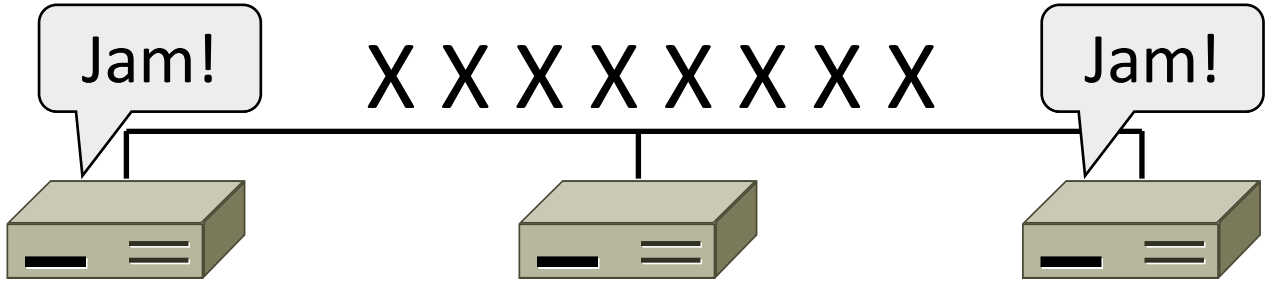
CSMA (3)

- CSMA is a good defense against collisions only when BD is small



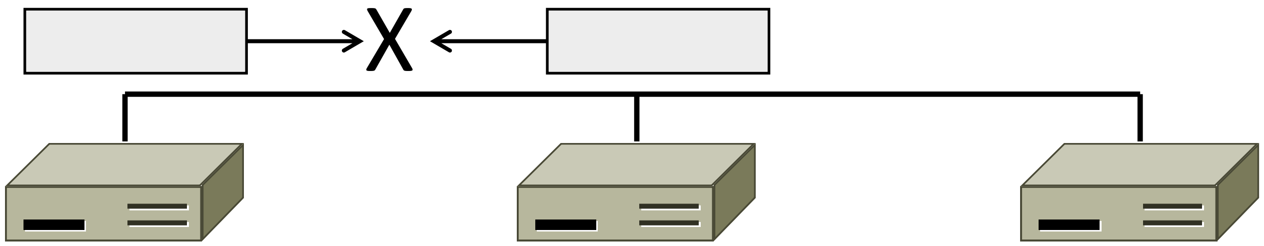
CSMA/CD (with Collision Detection)

- Can reduce the cost of collisions by detecting them and aborting (Jam) the rest of the frame time
  - Again, we can do this with wires



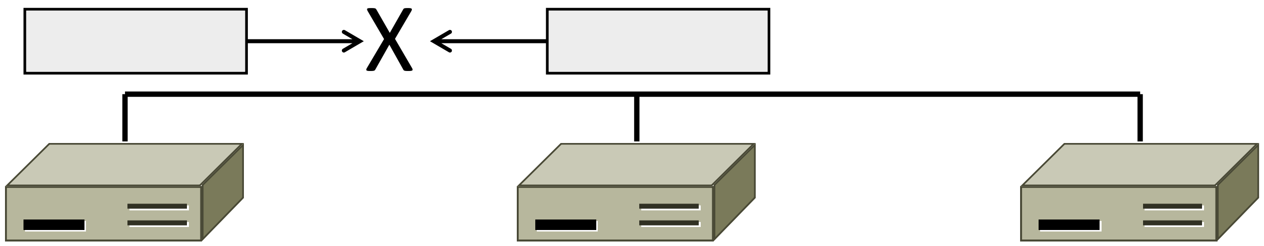
CSMA/CD Complications

- Everyone who collides needs to know it happened
  - How long do we need to wait to know there wasn't a JAM?



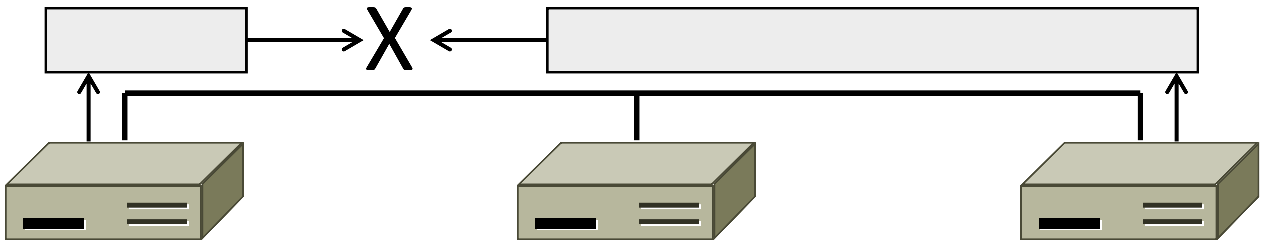
CSMA/CD Complications

- Everyone who collides needs to know it happened
  - How long do we need to wait to know there wasn't a JAM?
  - Time window in which a node may hear of a collision (transmission + jam) is 2D seconds



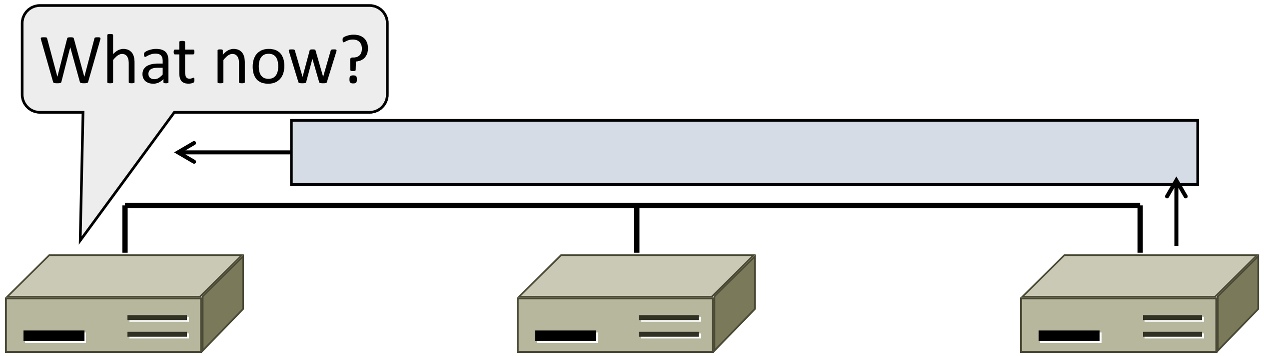
CSMA/CD Complications (2)

- Impose a minimum frame length of 2D seconds
  - So node can't finish before collision
  - Ethernet minimum frame is 64 bytes – Also sets maximum network length (500m w/ coax, 100m w/ Twisted Pair)



CSMA "Persistence"

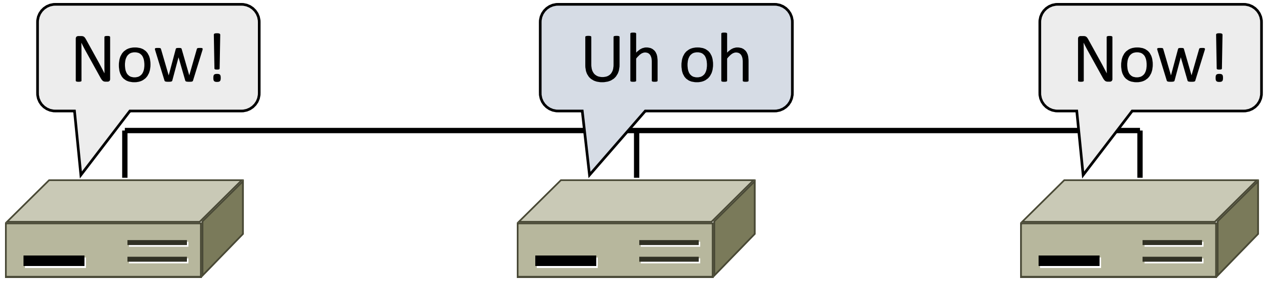
- What should a node do if another node is sending?



- Idea: Wait until it is done, and send

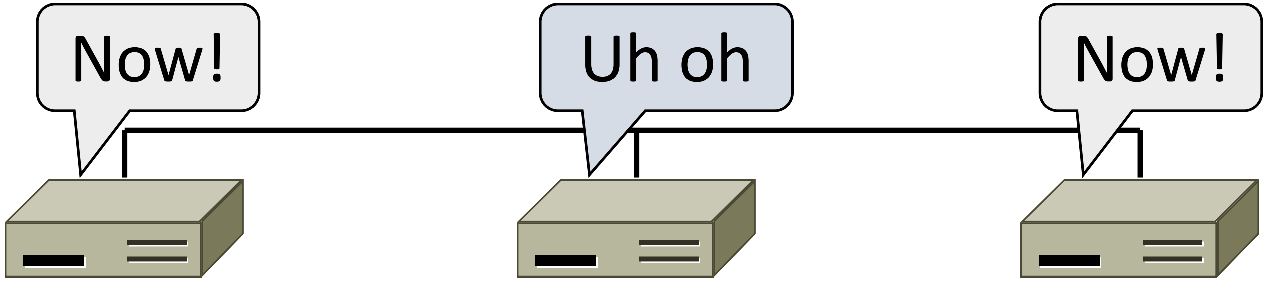
CSMA "Persistence" (2)

- Problem is that multiple waiting nodes will queue up then collide
  - More load, more of a problem



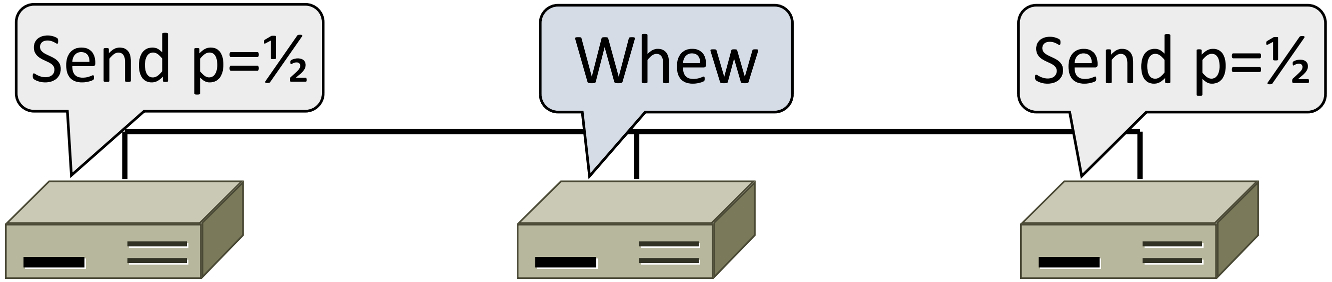
CSMA "Persistence" (2)

- Problem is that multiple waiting nodes will queue up then collide
  - Ideas?



CSMA "Persistence" (3)

- Intuition for a better solution
  - If there are N queued senders, we want each to send next with probability 1/N



Binary Exponential Backoff (BEB)

- Cleverly estimates the probability
  - 1st collision, wait 0 or 1 frame times
  - 2nd collision, wait from 0 to 3 times
  - 3rd collision, wait from 0 to 7 times …

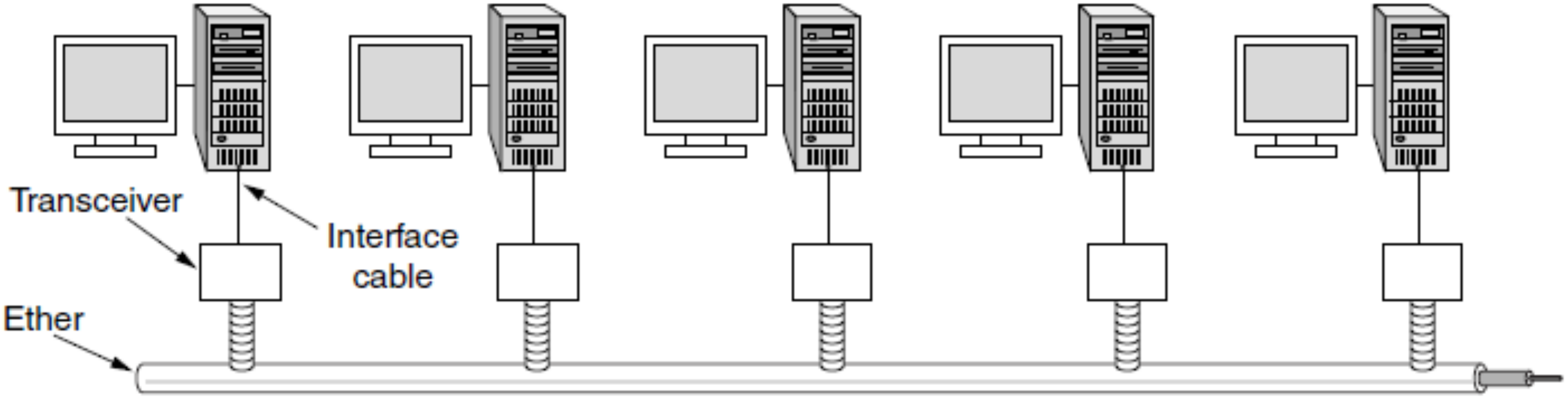
- BEB doubles interval for each successive collision
  - Quickly gets large enough to work
  - Very efficient in practice

Recap: MAC layer ideas

- Random wait times upon collisions
- Carrier sense
  - Persistence
- Collision detection
- Binary exponential backoff

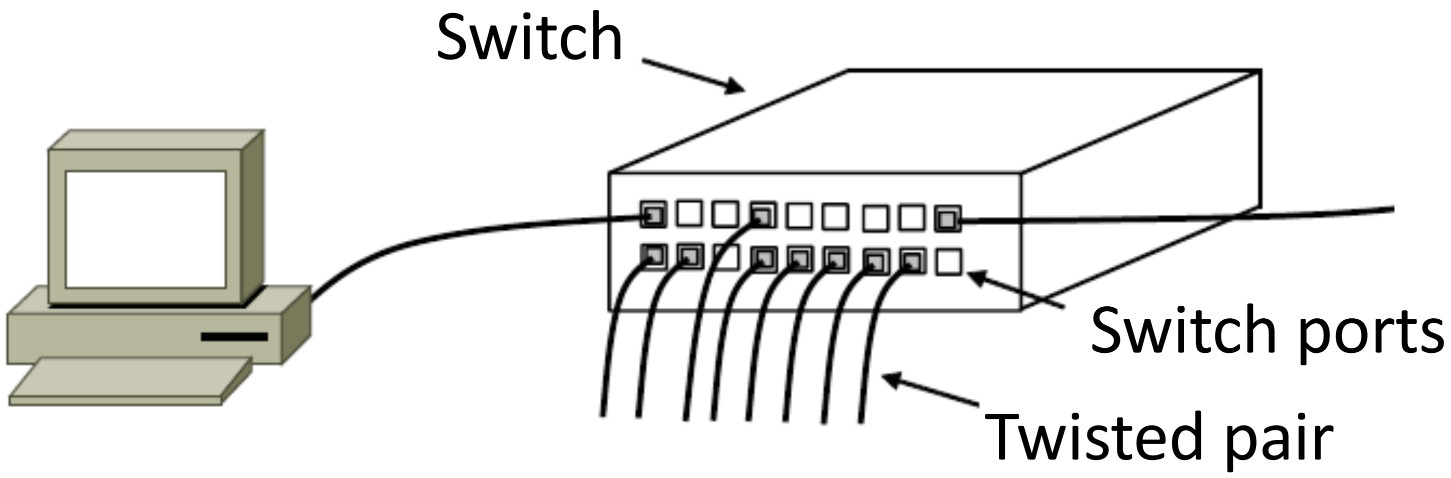
Classic Ethernet, or IEEE 802.3

- Most popular LAN of the 1980s, 1990s
  - 10 Mbps over shared coaxial cable
  - Multiple access with persistent CSMA/CD with BEB



Modern Ethernet

- Based on switches, not multiple access, but still called Ethernet
  - We'll get to it in a later segment



Ethernet Frame Format

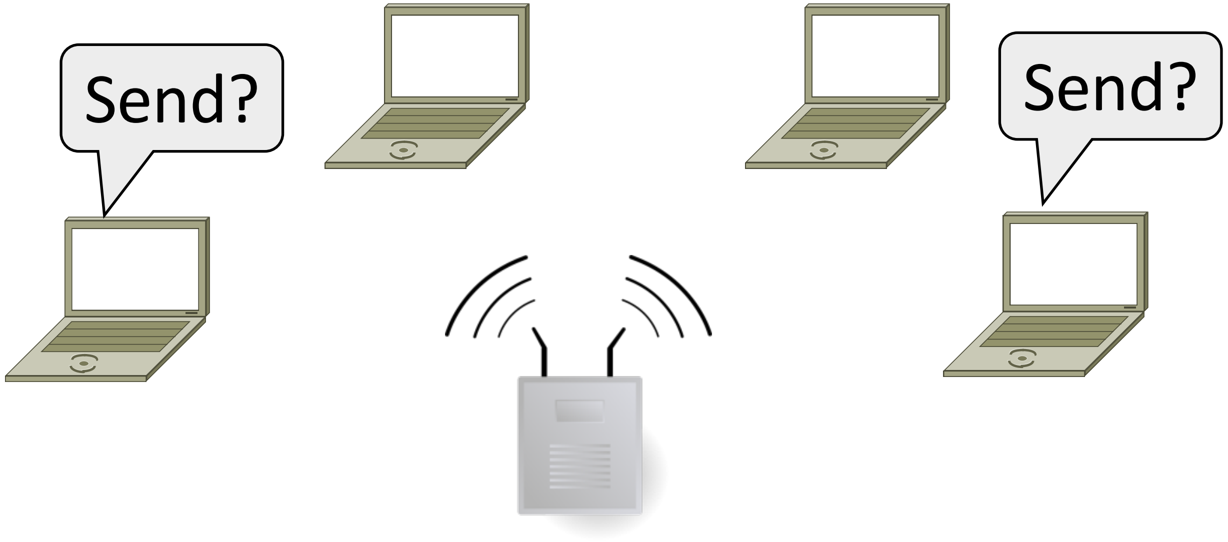
- Has addresses to identify the sender and receiver
- CRC-32 for error detection; no ACKs or retransmission
- Start of frame identified with physical layer preamble

Packet from Network layer (IP)

	Preamble	Destination address	Source address	Type	Data	Pad	Check-sum
Bytes	8	6	6	2	0-1500	0-46	4

Wireless MACs

- How do wireless nodes share a single link? (Yes, this is WiFi!)
  - Build on our simple, wired model



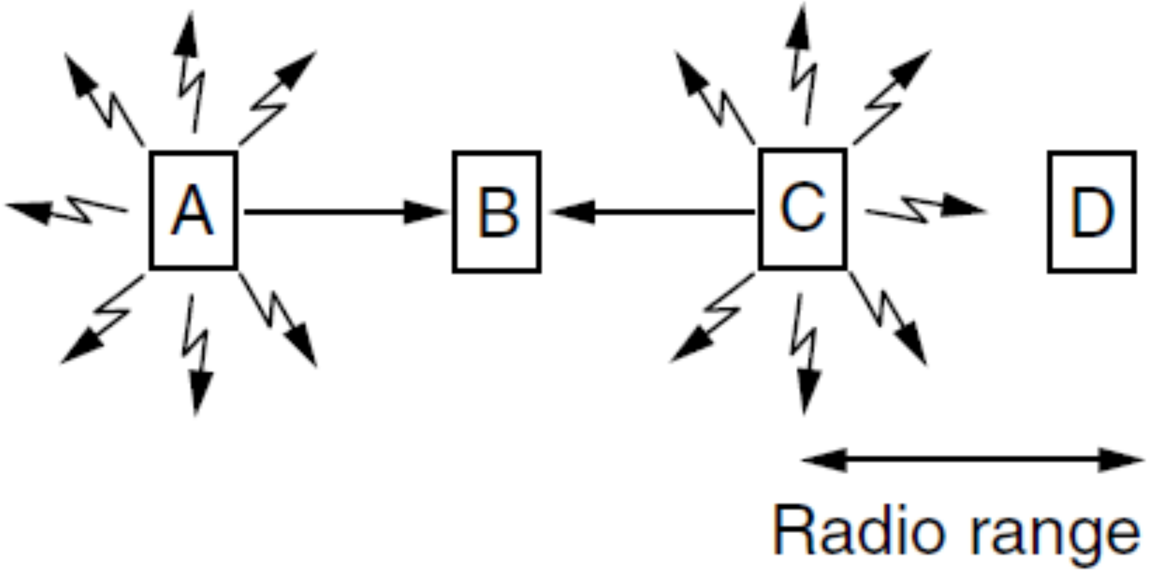
Wireless Complications

- Wireless is more complicated than wired (surprise!)
  1. Media is infinite – can't Carrier Sense
  2. Nodes can't hear while sending – can't Collision Detect

 ≠ CSMA/CD

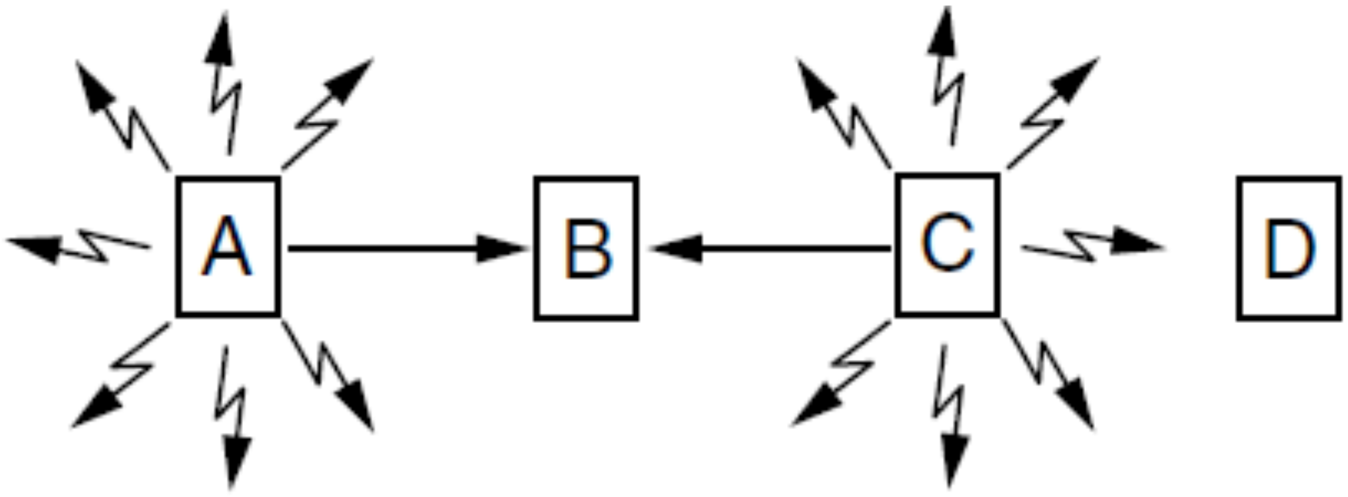
No CS: Different Coverage Areas

- Wireless signal is broadcast and received nearby, where there is sufficient SNR



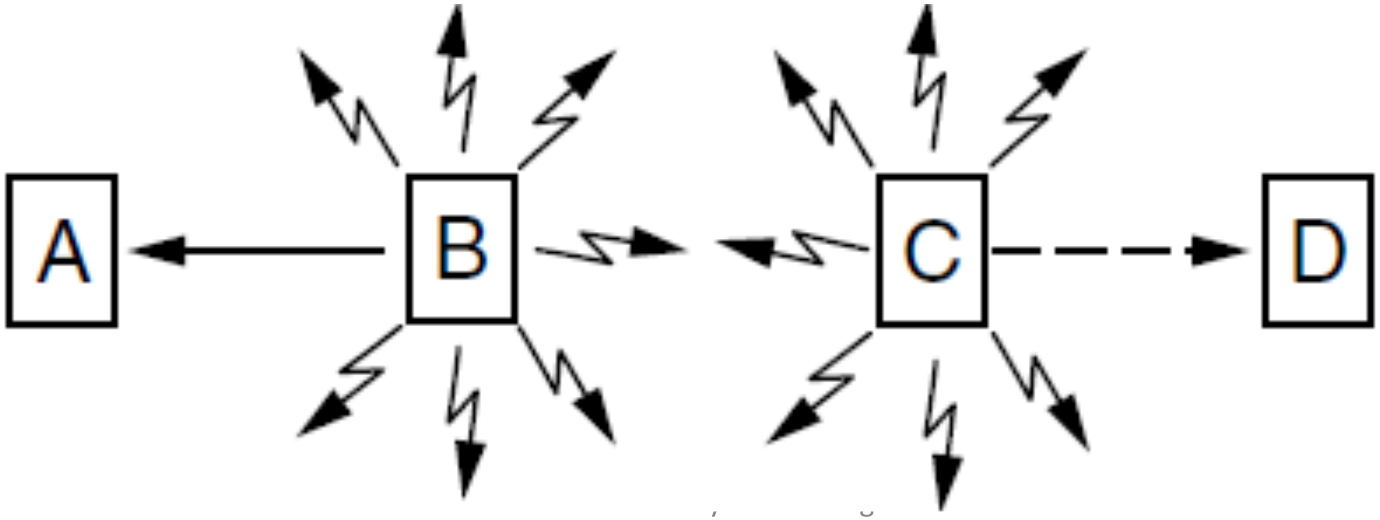
No CS: Hidden Terminals

- Node C is a <u>hidden terminal</u> when A sends to B
  - Similarly, A is a hidden terminal when C sends to B
  - A, C can’t hear each other (to coordinate) yet collide at B
  - We want to avoid the inefficiency of collisions



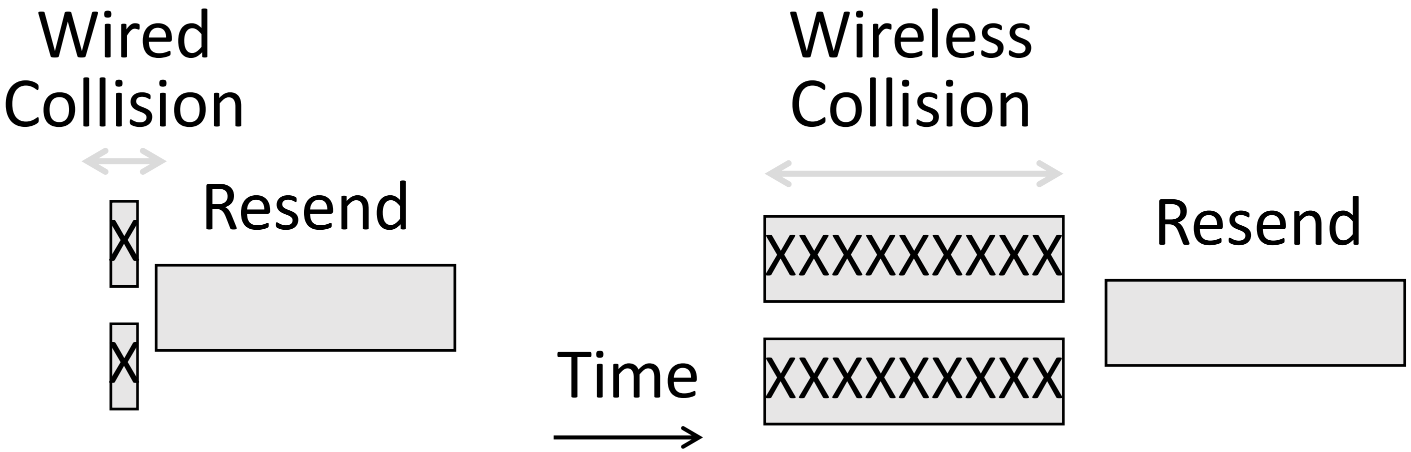
No CS: Exposed Terminals

- B, C are <u>exposed terminals</u> when sending to A, D
  - Can hear each other yet don't collide at receivers A and D
  - We want to send concurrently to increase performance



Nodes Can't Hear While Sending

- With wires, detecting collisions (and aborting) lowers their cost
- With wireless, more wasted time



Wireless Problems:

- Ideas?

MACA: Multiple Access w/ Collision Avoidance

- MACA uses a short handshake instead of CSMA (Karn, 1990)
  - 802.11 uses a refinement of MACA (later)

- Protocol rules:
  1. A sender node transmits a RTS (Request-To-Send, with frame length)
  2. The receiver replies with a CTS (Clear-To-Send, with frame length)
  3. Sender transmits the frame while nodes hearing the CTS stay silent

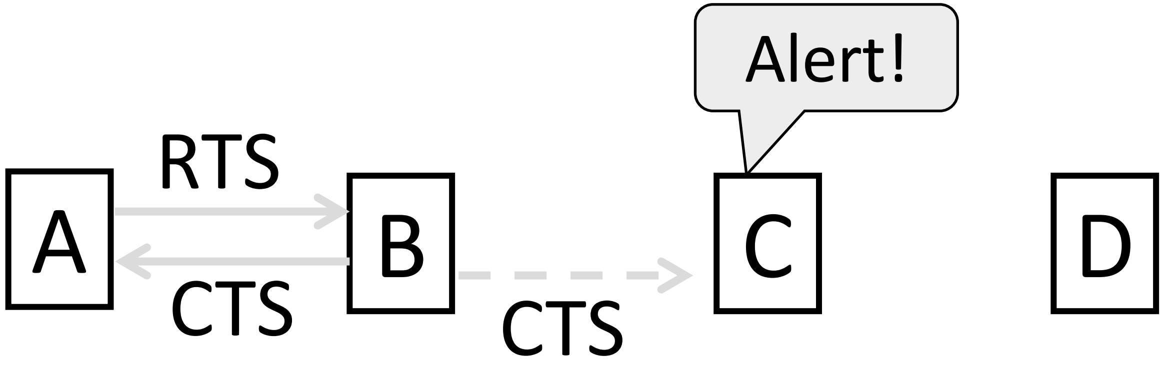
- Collisions on the RTS/CTS are still possible, but less likely

MACA – Hidden Terminals

- A→B with hidden terminal C
  1. A sends RTS, to B

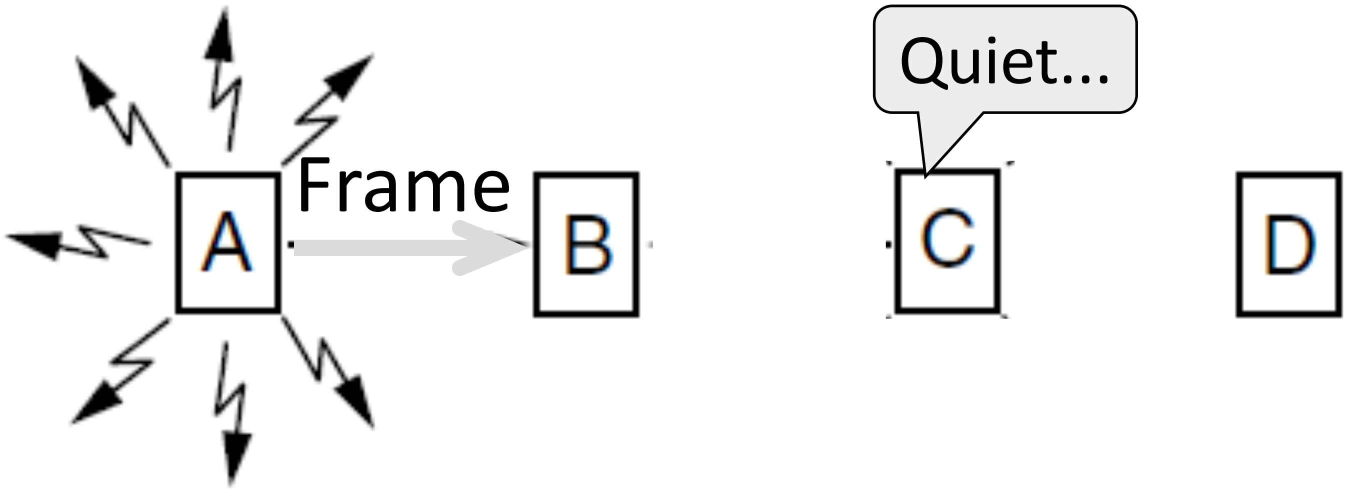
MACA – Hidden Terminals (2)

- A→B with hidden terminal C
    2. B sends CTS to A, and C overhears



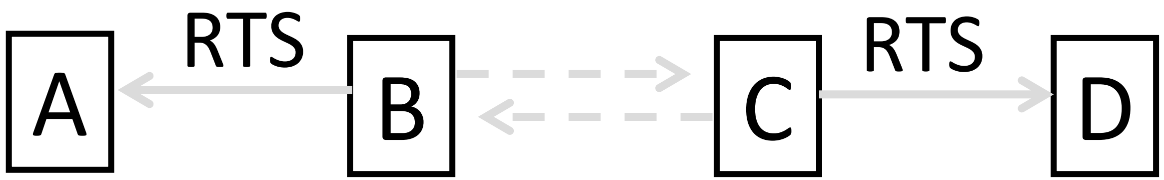
MACA – Hidden Terminals (3)

- A→B with hidden terminal C
    3. A sends frame while C defers



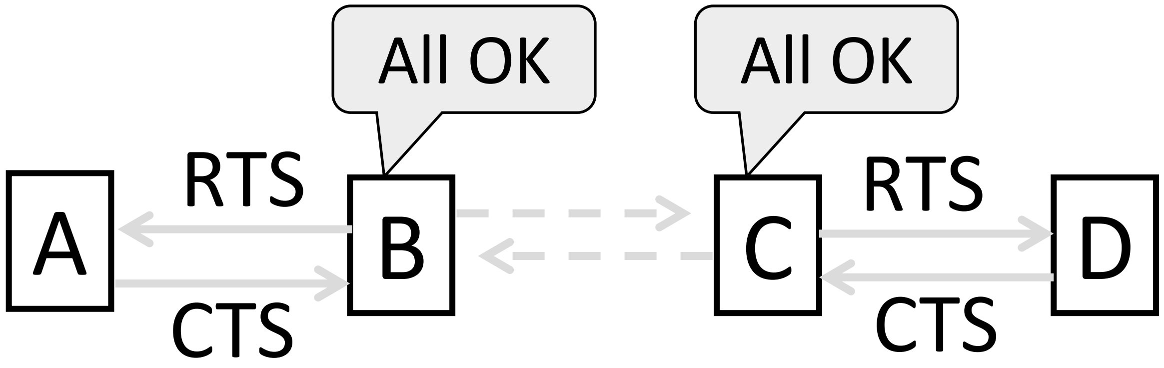
MACA – Exposed Terminals

- B→A, C→D as exposed terminals
  - B and C send RTS to A and D



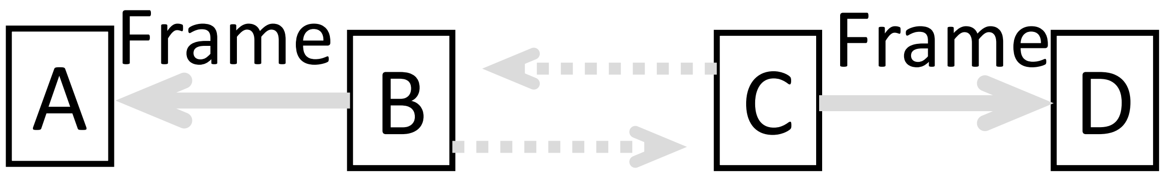
MACA – Exposed Terminals (2)

- B→A, C→D as exposed terminals
  - A and D send CTS to B and C



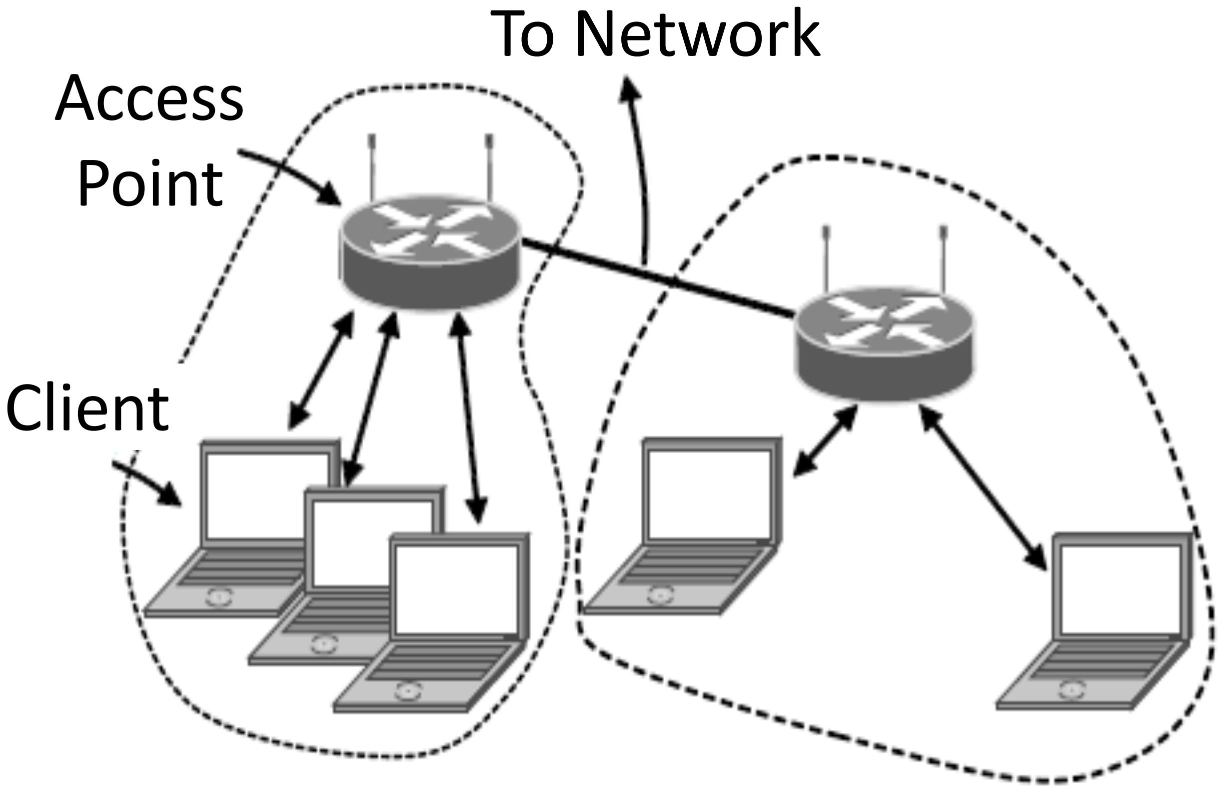
MACA – Exposed Terminals (3)

- B→A, C→D as exposed terminals
  - A and D send CTS to B and C



802.11, or WiFi

- Very popular wireless LAN started in the 1990s
- Clients get connectivity from a (wired) AP (Access Point)
- It’s a multi-access problem ☺
- Various flavors have been developed over time
  - Faster, more features



802.11 Physical Layer

- Uses 20/40 MHz channels on ISM (unlicensed) bands
  - 802.11b/g/n on 2.4 GHz
  - 802.11 a/n on 5 GHz
- OFDM modulation (except legacy 802.11b)
  - Different amplitudes/phases for varying SNRs
  - Rates from 6 to 54 Mbps plus error correction
  - 802.11n uses multiple antennas
    - Lots of fun tricks here

802.11 Link Layer

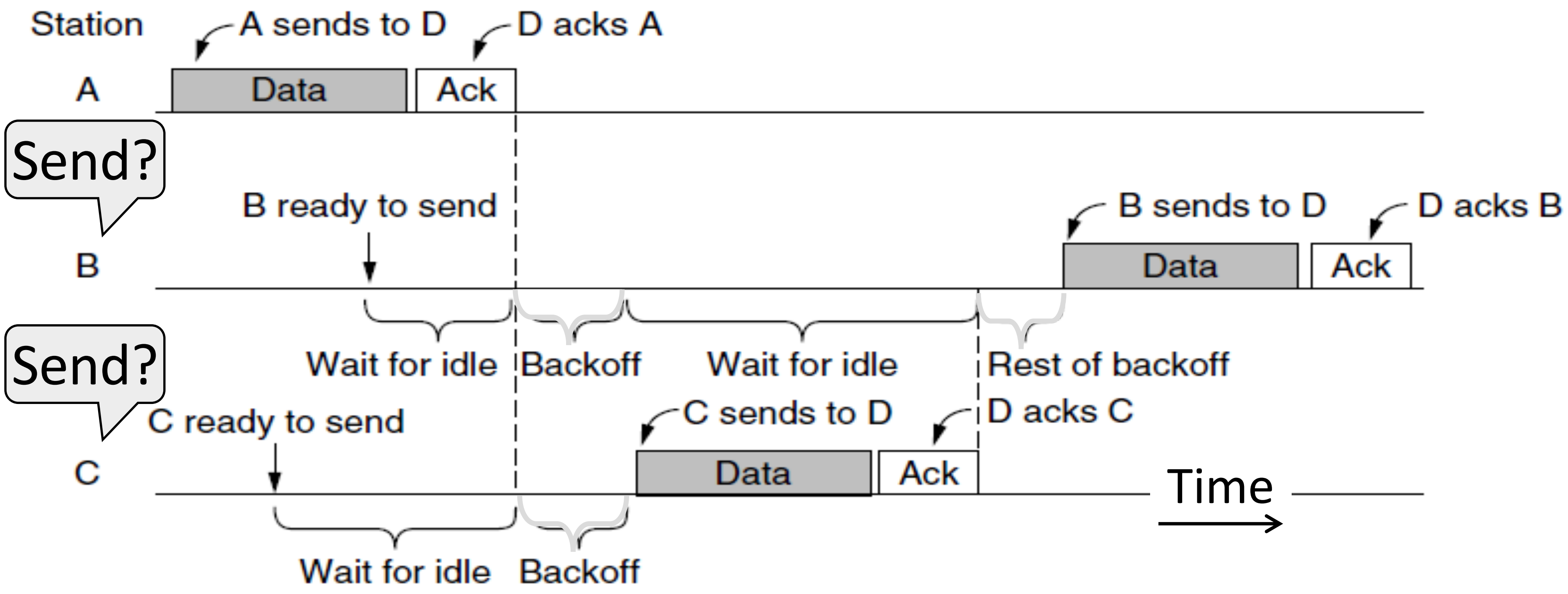
- Multiple access uses CSMA/CA (next); RTS/CTS optional
- Frames are ACKed and retransmitted with ARQ
- Funky addressing (three addresses!) due to AP
- Errors are detected with a 32-bit CRC
- Many, many features (e.g., encryption, power save)

Packet from Network layer (IP)

	Frame control	Duration	Address 1 (recipient)	Address 2 (transmitter)	Address 3	Sequence	Data	Check sequence
Bytes	2	2	6	6	6	2	0–2312	4

802.11 CSMA/CA for Multiple Access

- Still using BEB!



Centralized MAC: Cellular

- Spectrum suddenly very scarce
  - We can’t waste all of it sending JAMs
- We have QoS requirements
  - Can’t be as loose with expectations
  - Can’t have traffic fail
- We also have client/server
  - Centralized control
  - Not peer-to-peer/decentralized

GSM MAC

- FDMA/TDMA
- Use one channel for coordination – Random access w/BEB (no CSMA, can't detect)
- Use other channels for traffic
  - Dedicated channel for QoS