Lecture 6:
Media Access Control

CSE 123: Computer Networks
Stefan Savage

UCSDCSE

Today: Media access

- How to share a channel among different hosts?

- Channel partitioning
  - FDMA (frequency division multiple access)
  - TDMA (time division multiple access)
  - CDMA (code division multiple access)

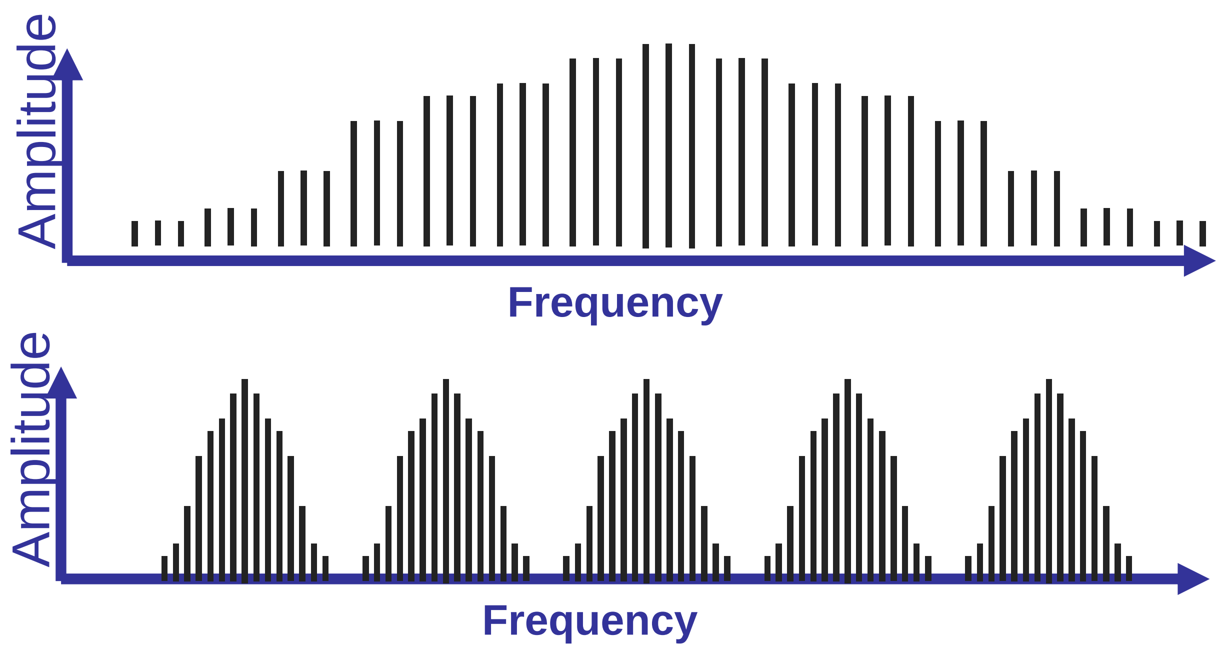
- Random access
  - Contention-based
    - Aloha
    - CSMA, CSMA/CD, CSMA/CA
    - Ethernet, 802.11
  - Contention-free
    - Token-ring, FDDI

Channel Partitioning

- Need to share media with multiple nodes (*n*)
  - Multiple *simultaneous* conversations
- A simple solution
  - Divide the channel into multiple, separate channels
- Channels are completely separate
  - Bitrate of the channel is split across channels
  - Nodes can only send/receive on their assigned channel
- Several different ways to do it…

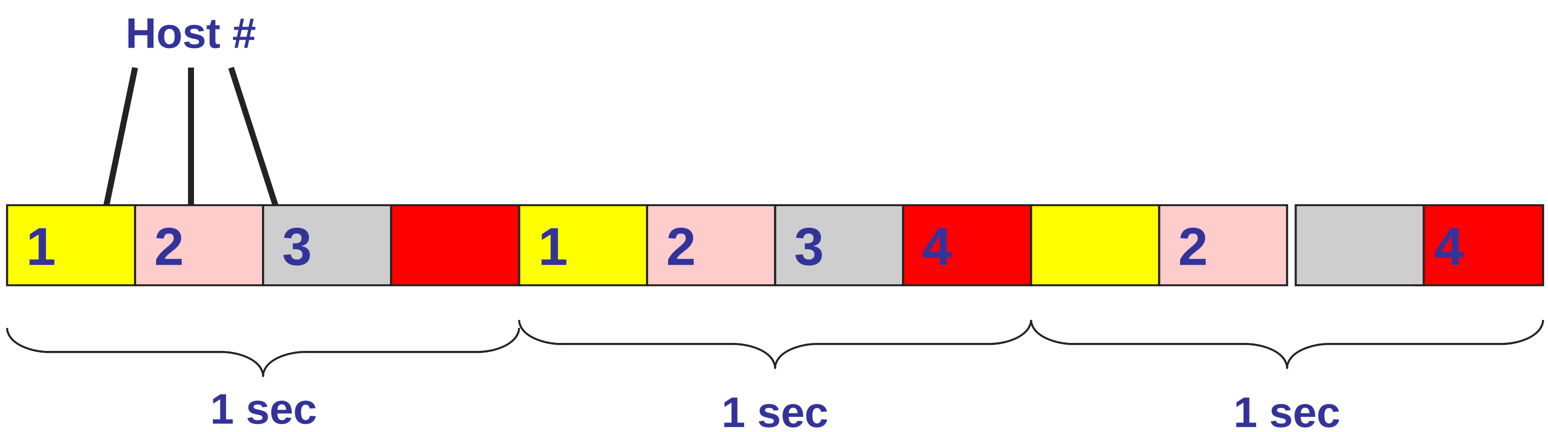
Frequency Division (FDMA)

- Divide bandwidth of *f* Hz into *n* channels each with bandwidth *f/n* Hz
  - Easy to implement, but unused subchannels go idle
    - Also need "guard bands" between channels to prevent interference
  - Used by traditional analog cell phone service, radio, TV



Time Division (TDMA)

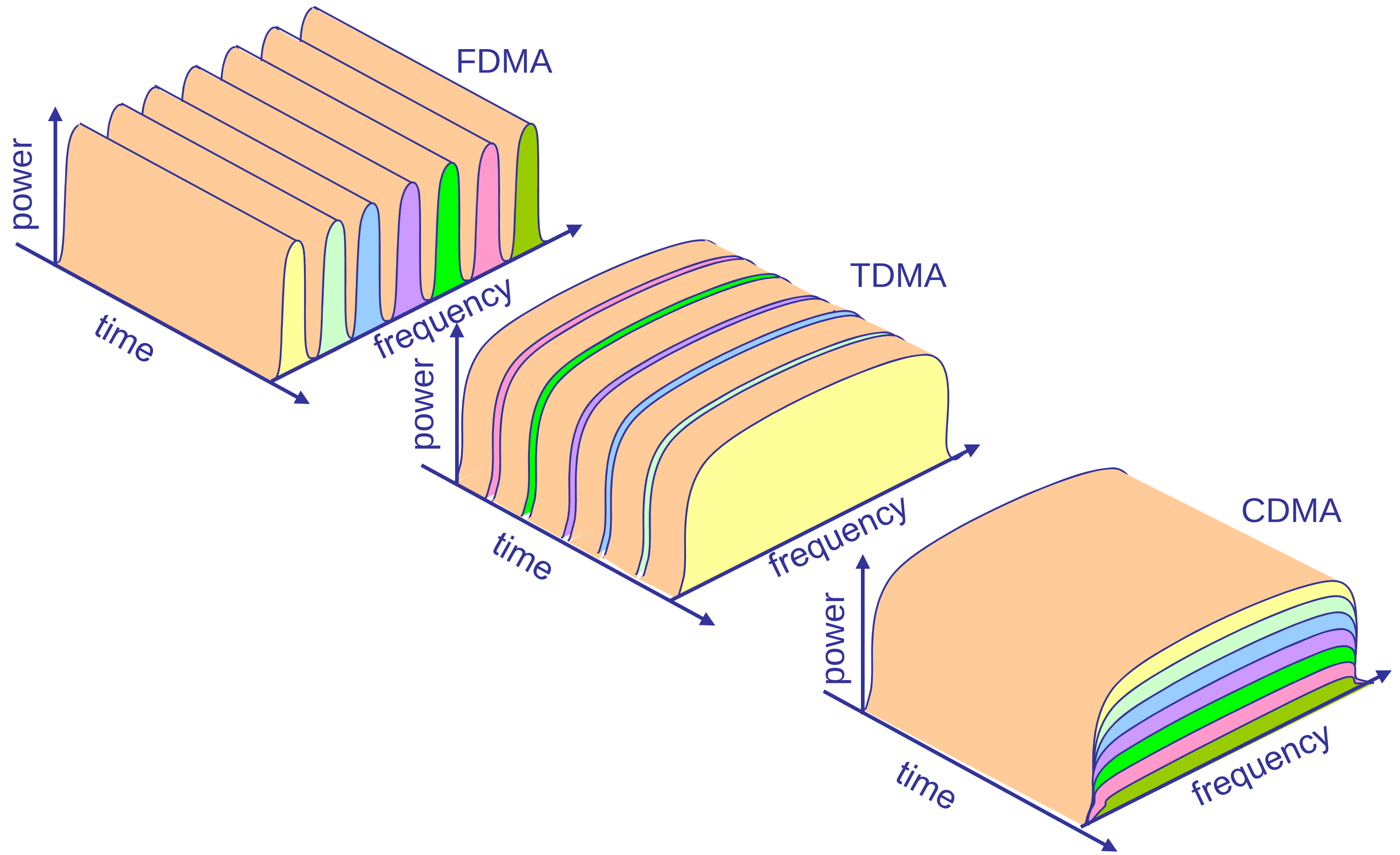
- Divide channel into rounds of *n* time slots each
  - Assign different hosts to different time slots within a round
  - Unused time slots are idle
  - Used in GSM cell phones & digital cordless phones
- Example with 1-second rounds
  - *n*=4 timeslots (250ms each) per round



Code Division (CDMA)

- Do nothing to physically separate the channels
  - All stations transmit at same time in same frequency bands
  - One of so-called spread-spectrum techniques
- Sender modulates their signal on top of unique code
  - Sort of like the way Manchester modulates on top of clock
  - The bit rate of resulting signal much lower than entire channel
- Receiver applies code filter to extract desired sender
  - All other senders seem like *noise* with respect to signal
- Used in some newer digital cellular technologies (Verizon/Sprint US Cellular)

Partitioning Visualization



Courtesy Takashi Inoue

Problem w/Channel partitioning

- Not terribly well suited for random access usage
  - Why?

- Instead, design schemes for more common situations
  - Not all nodes want to send all the time
  - Don’t have a fixed number of nodes

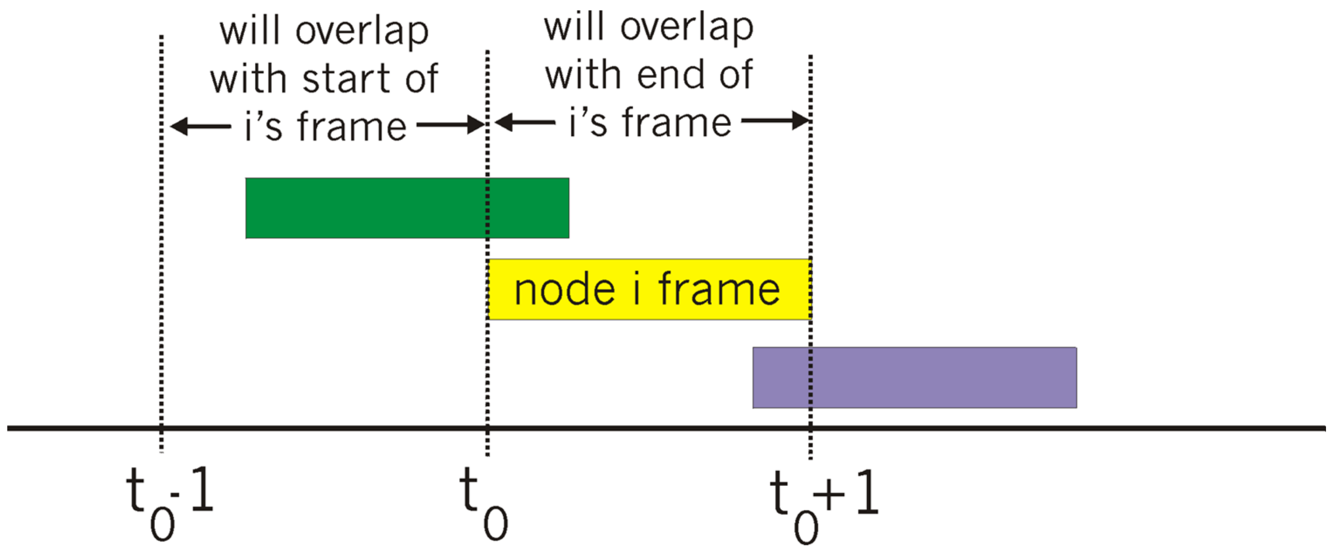
- Potentially higher throughput for transmissions
  - Active nodes get full channel bandwidth

Aloha

- Designed in 1970 to support wireless data connectivity
  - Between Hawaiian Islands—rough!
- Goal: distributed access control (no central arbitrator)
  - Over a shared *broadcast* channel
- Aloha protocol in a nutshell:
  - When you have data **send it**
  - If data doesn't get through (receiver sends acknowledgement) then **retransmit after a random delay**
  - Why not a fixed delay?

Collisions

- Frame sent at t_0 collides with frames sent in [t_0-1, t_0+1]
  - Assuming unit-length frames
  - Ignores propagation delay



Slotted Aloha

- Time is divided into equal size slots (frame size)
- Host wanting to transmit starts at start of next slot
  - Retransmit like w/Aloha, but quantize to nearest next slot
- Requires **time synchronization** between hosts

Success (S), Collision (C), Empty (E) slots

Channel Efficiency

Q: What is max fraction slots successful?

A: Suppose *n* stations have packets to send

- Each transmits in slot with probability *p*
- Prob[successful transmission], *S,* is:

$$S = p\,(1-p)^{(n-1)}$$

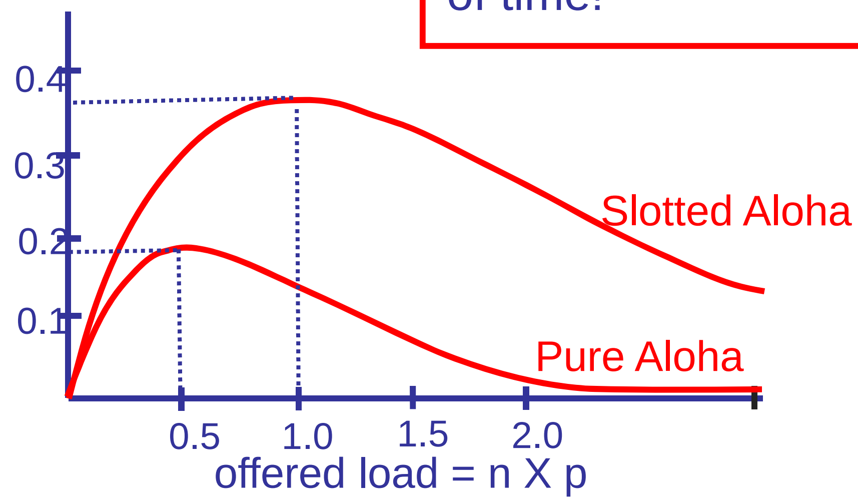
- any of *n* nodes:

$$S = \text{Prob[one transmits]} = np(1-p)^{(n-1)}$$

(optimal *p* as *n*->infinity = *1/n*)

$$= 1/e = .37$$

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



Carrier Sense (CSMA)

- Aloha transmits even if another host is transmitting
  - Thus guaranteeing a *collision*
- Instead, *listen first* to make sure channel is idle
  - Useful only if channel is frequently idle
  - Why?
- Listen how long to be confident channel is idle?
  - Depends on maximum propagation delay
  - Small (<<1 frame length) for LANs
  - Large (>>1 frame length) for satellites

Retransmission Options

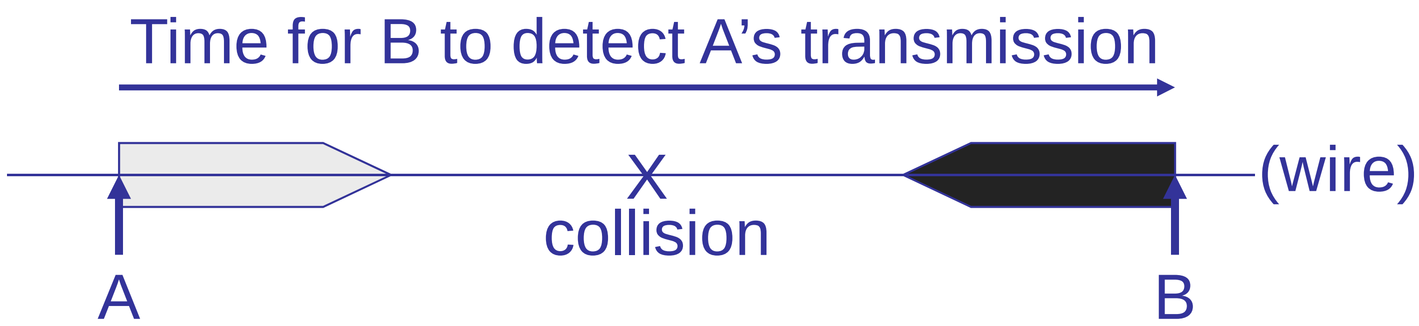
- non-persistent CSMA
  - Give up, or send after some random delay
  - Problem: may incur larger delay when channel is idle

- 1-persistent CSMA
  - Send as soon as channel is idle
  - Problem: blocked senders all try to send at once

- *P*-persistent CSMA
  - If idle, send packet with probability p; repeat
  - Make sure $(p * n) < 1$

Jamming

- Even with CSMA there can still be collisions. Why?



- If nodes can detect collisions, abort! (CSMA/CD)
  - Requires a minimum frame size ("acquiring the medium")
  - *B* must continue sending ("jam") until *A* detects collision

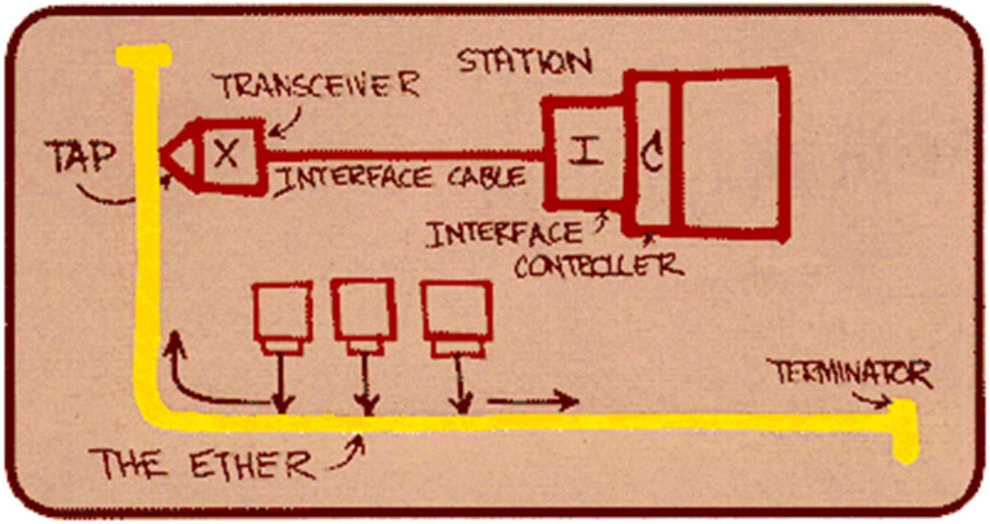
- Requires a full duplex channel
  - Aside: wireless is typically half duplex; need an alternative for wireless channels (we'll return to this)

Collision Detection

- How can A know that a collision has taken place?
  - Worst case:
    - Latency between nodes A & B is d
    - A sends a message at time t and B sends a message at t + d – epsilon (i.e., just before receiving A's message)
  - B knows there is a collision, but not A… B must keep transmitting so A knows that its packet has collided
  - How long? *2 * d* *(keep talking until you know everyone has heard it)*
- IEEE 802.3 Ethernet specifies max value of *2d* to be 51.2us
  - This relates to maximum distance of 2500m between hosts
  - At 10Mbps it takes 0.1us to transmit one bit so 512 bits take 51.2us to send
  - So, Ethernet frames must be at least 64B (512 bits) long
    - Padding is used if data is too small
- Transmit jamming signal to insure all hosts see collision
  - 48 bit times sufficient

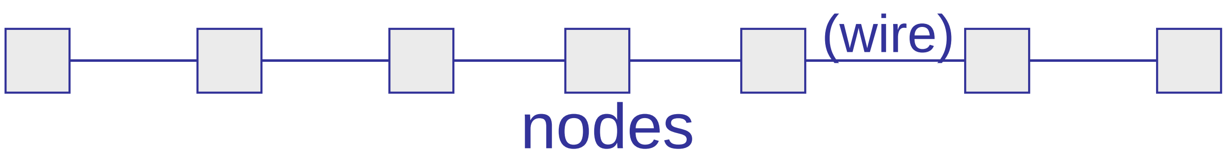
Ethernet

- First local area network (LAN)
  - Developed in early ’70s by Metcalfe and Boggs at PARC
  - Originally 1Mbps, now supports 10Mbps, 100Mbps, 1Gbps and 10Gbps flavors (40/100G in development)
- Currently *the* dominant LAN technology
  - Becoming the dominant WAN technology



Classic Ethernet

- IEEE 802.3 standard wired LAN
  (modified 1-persistent CSMA/CD)
- Classic Ethernet: 10 Mbps over coaxial cable
  - All nodes share same physical wire
  - Max length 2.5km, max between stations 500m



- Framing
  - Preamble, 32-bit CRC, variable length data
  - Unique 48-bit address per host (bcast & multicast addrs too)

Preamble (8)	Source (6)	Dest (6)	Len (2)	Payload (var)	Pad (var)	CRC (4)

Ethernet improvements

- Problems with random delay with fixed mean
  - Few senders = unnecessary delay
  - Many senders = unnecessary collisions

- Binary exponential back-off balances delay w/load
  - First collision: wait 0 or 1 min frame times at random, retry
  - Second time: wait 0, 1, 2, or 3 times
  - *N*th time (n<=10): wait 0, 1, …, 2^n-1 times
  - Max wait 1023 frames; give up after 16 attempts

Capture Effect

- Randomized access scheme is not fair

- Suppose stations *A* and *B* always have data to send
  - They *will* collide at some time
  - Both pick random number of “slots” (0, 1) to wait
  - Suppose *A* wins and sends
  - Next time the collide, *B*’s chance of winning is halved
    - » *B* will select from 0,1,2,3 due to exponential back-off

- *A* keeps winning: said to have captured the channel

Ethernet Performance

- Much better than Aloha or CSMA in practice

- Source of protocol inefficiency: collisions
  - More efficient to send larger frames
    - » Acquire the medium and send lots of data
    - » Less time on arbitration (figuring out who gets to send)

  - Less efficient if
    - » More hosts – more collisions needed to identify single sender
    - » Smaller packet sizes – more frequent arbitration
    - » Longer links – collisions take longer to observe, more wasted bandwidth

Contention-free Protocols

- Problem with channel partitioning
  - Inefficient at low load (idle subchannels)
- Problem with contention-based protocols
  - Inefficient at high load (collisions)

- Contention-free protocols
  - Try to do both by explicitly taking turns
  - Can potentially also offer guaranteed bandwidth, latency, etc.

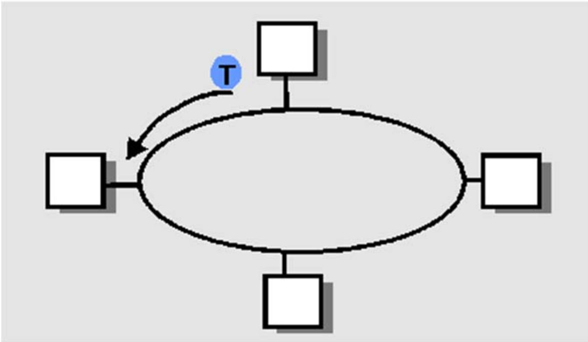
Two contention-free approaches

Polling

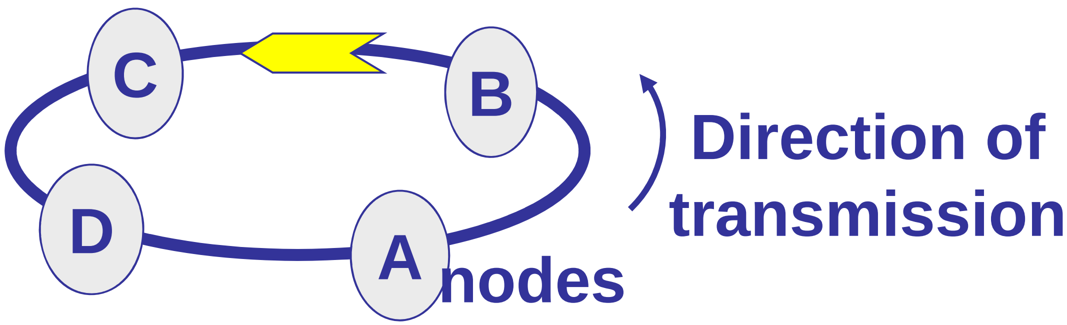
- Master node “invites” slave nodes to transmit in turn
- Request to Send (RTS), Clear to Send (CTS) messages
- Problems:
  - Polling overhead
  - Latency
  - Single point of failure (master)

Token Passing

- Control token passed from one node to next sequentially.
- Point-to-point links can be fast
- Problems:
  - Token overhead
  - Latency
  - Single point of failure (token)



Token Ring (802.5)



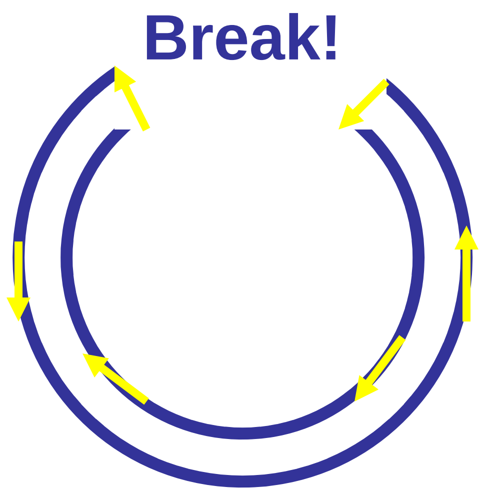
- Token rotates permission to send around ring of nodes
- Sender injects packet into ring and removes later
  - Maximum token holding time (THT) bounds access time
  - Early or delayed token release
  - Round robin service, acknowledgments and priorities
- Monitor nodes ensure health of ring (alerts on failures)

SD (1)	AC (1)	FC (1)	Destination Address (6)	Source Address (6)	Data	Checksum (4)	ED (1)	FS (1)

FDDI
(Fiber Distributed Data Interface)

- Roughly a large, fast token ring
  - 100 Mbps and 200km vs 4/16 Mbps and local
  - Dual counter-rotating rings for redundancy
  - Complex token holding policies for voice etc. traffic

- Token ring advantages
  - No contention, bounded access delay
  - Support fair, reserved, priority access
- Disadvantages
  - Complexity, reliability, scalability



Why Did Ethernet Win?

- Failure modes
  - Token rings – network unusable
  - Ethernet – single node detached
- Good performance in common case
- Completely distributed, easy to maintain/administer
- Easy incremental deployment
- Volume → lower cost → higher volume ….

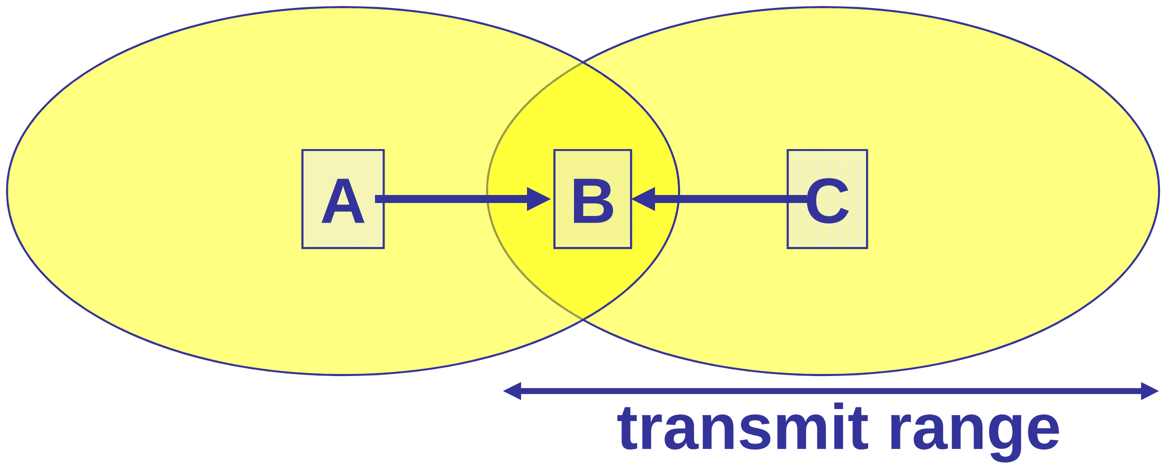
Wireless Media Access

Wireless is more complicated than wired …

- Cannot detect collisions
  - Transmitter swamps co-located receiver
  - Collision avoidance

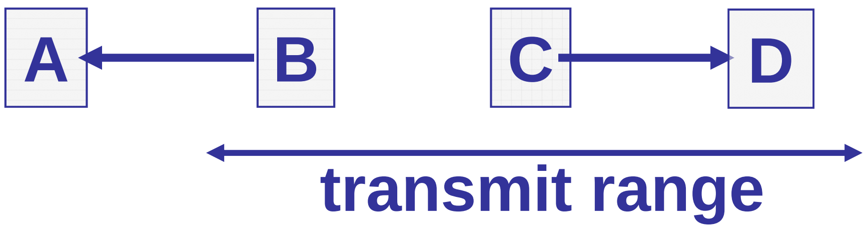
- Different transmitters have different coverage areas
  - Asymmetries lead to hidden/exposed terminal problems
    - » Just because A is talking to B, doesn't mean that C can't talk to D
    - » A is aware of B and B is aware of C, but A and C are unaware of each other (they are hidden nodes). Both will send to B and frames will collide at B without A or C realizing
  - Also can use contention-free protocols (RTS/CTS)

Hidden Terminals



- A and C can both send to B but can't hear each other
  - A is a hidden terminal for C and vice versa
  - Packets will collide at B and create interference
- CSMA will be ineffective – want to sense at receiver

Exposed Terminals

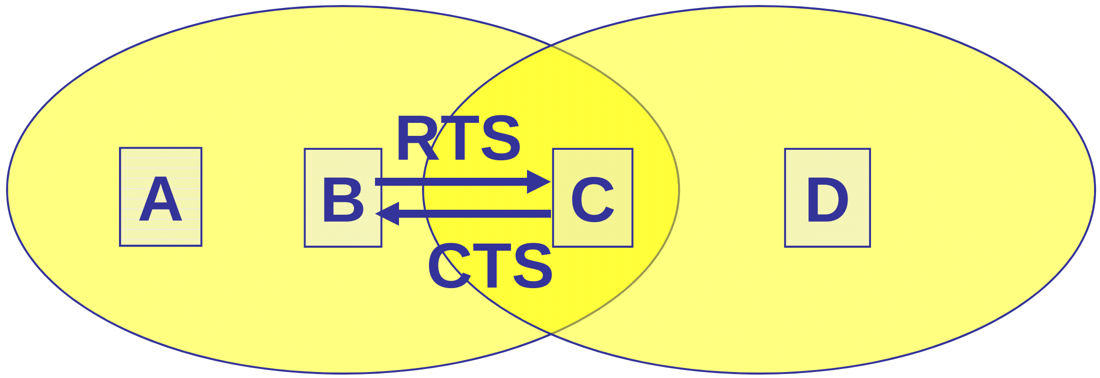


- B, C can hear each other but can safely send to A, D
  - B and C won't transmit, when they should

CSMA with Collision Avoidance (CSMA/CA)

- Since we can't detect collisions, we try to *avoid* them
- When medium busy, choose random interval (*contention window*)
  - Wait for that many **idle** timeslots to pass before sending
  - Remember p-persistence … a refinement
- When a collision is inferred, retransmit with binary exponential backoff (like Ethernet)
  - Use ACK from receiver to infer "no collision"
  - Again, exponential backoff helps us adapt "p" as needed
- CSMA/CA vs. CSMA/CD
  - In CSMA/CA, backoff *before* collision
  - In CSMA/CD, backoff *after* collision

RTS / CTS Protocols (MACA)



Overcome exposed/hidden terminal problems with contention-free protocol

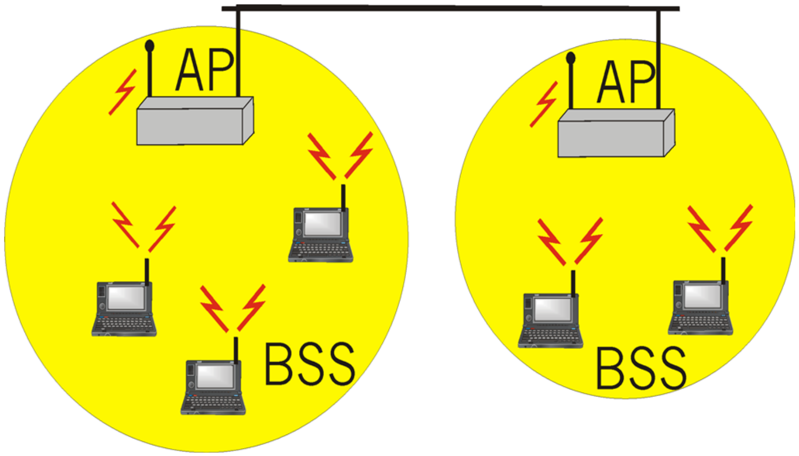
1. B stimulates C with Request To Send (RTS) packet
2. A hears RTS and defers to allow the CTS
3. C replies to B with Clear To Send (CTS)
4. D hears CTS and defers to allow the data
5. B sends to C

IEEE 802.11 Wireless LAN

- 802.11b
  - 2.4-5 GHz unlicensed radio spectrum
  - up to 11 Mbps
  - direct sequence spread spectrum (DSSS) in physical layer
    - All hosts use same code
  - Widely deployed, using base stations
- 802.11a
  - 5-6 GHz range
  - up to 54 Mbps
- 802.11g
  - 2.4-5 GHz range
  - up to 54 Mbps
- All use CSMA/CA for multiple access
- Optional RTS/CTS
- All have base-station and ad-hoc network versions

IEEE 802.11 Wireless LAN

- Wireless host communicates with a base station
  - Base station = access point (AP)
- Basic Service Set (BSS) (a.k.a. "cell") contains:
  - Wireless hosts
  - Access point (AP): base station
- BSS's combined to form distribution system (DS)



802.11 Twists

- How to support different speeds on same channel?
  - Physical layer header encoded at lowest bitrate and indicates bitrate of rest of packet

- Network Allocation Vector (NAV)
  - Each frame contains field that indicates the amount of time that will be used for the communication (channel reservation)
  - All receivers defer transmit for that time
  - Allows for long or multi-frame exchange

Misc issue: Addressing Alternatives

- On a broadcast channel all nodes receive all packets
  - Addressing determines which packets are kept and which packets are thrown away
  - Packets can be sent to:
    - » Unicast – one destination
    - » Multicast – group of nodes (e.g. “everyone playing WoW”)
    - » Broadcast – everybody on wire
- Dynamic addresses (e.g. Appletalk)
  - Pick an address at random
  - Broadcast “is anyone using address XX?”
  - If yes, repeat
- Static address (e.g. Ethernet, 802.11, TokenRing, etc)

IEEE addressing (Ethernet, etc)

- Addresses: 6 bytes (48 bits)
  - Each adapter is given a globally unique address at manufacturing time
    - » Address space is allocated to manufacturers
      - 24 bits identify manufacturer
      - E.g., 0:0:15:* → 3com adapter
    - » Frame is received by all adapters on a LAN and dropped if address does not match
  - Special addresses
    - » Broadcast – FF:FF:FF:FF:FF:FF is “everybody”
    - » Range of addresses allocated to multicast
      - Adapter maintains list of multicast groups node is interested in
  - Practical problems: non-unique addresses

Summary

- Ways to share a channel
  - Subdivide the channel into subchannels
  - Contention-based protocols
    - » Try and retry if it fails
  - Contention-free protocols
    - » Explicit control over who gets to send at each time
- Particular issues for wireless
  - Hidden/exposed terminal problems
- Addressing

- For next time: bridging/switching, read P&D 3-1-3.2, 3.1