C:\cygwin\home\zahorjan\cse461\12au\06-multiaccess.cp3

Multiaccess Protocols

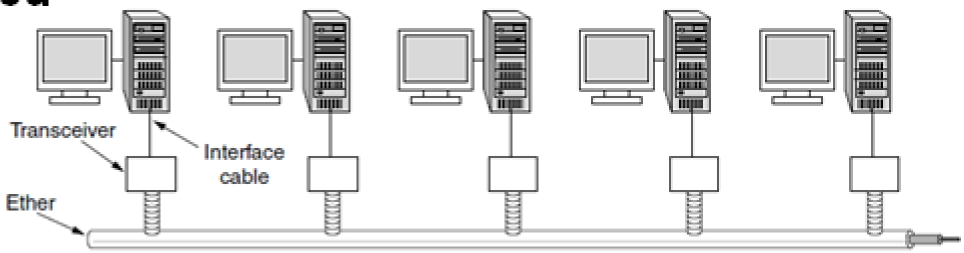
Recap from last class

- Two ways to handle possible errors in the link
  - FEC
  - ARQ (e.g., sliding window)

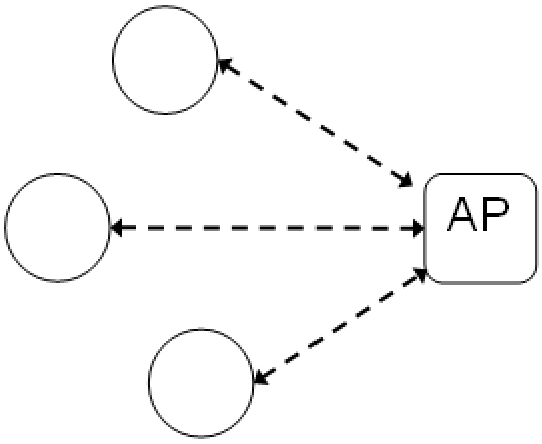
- Why are there errors in the link, again?

Multi-access topologies

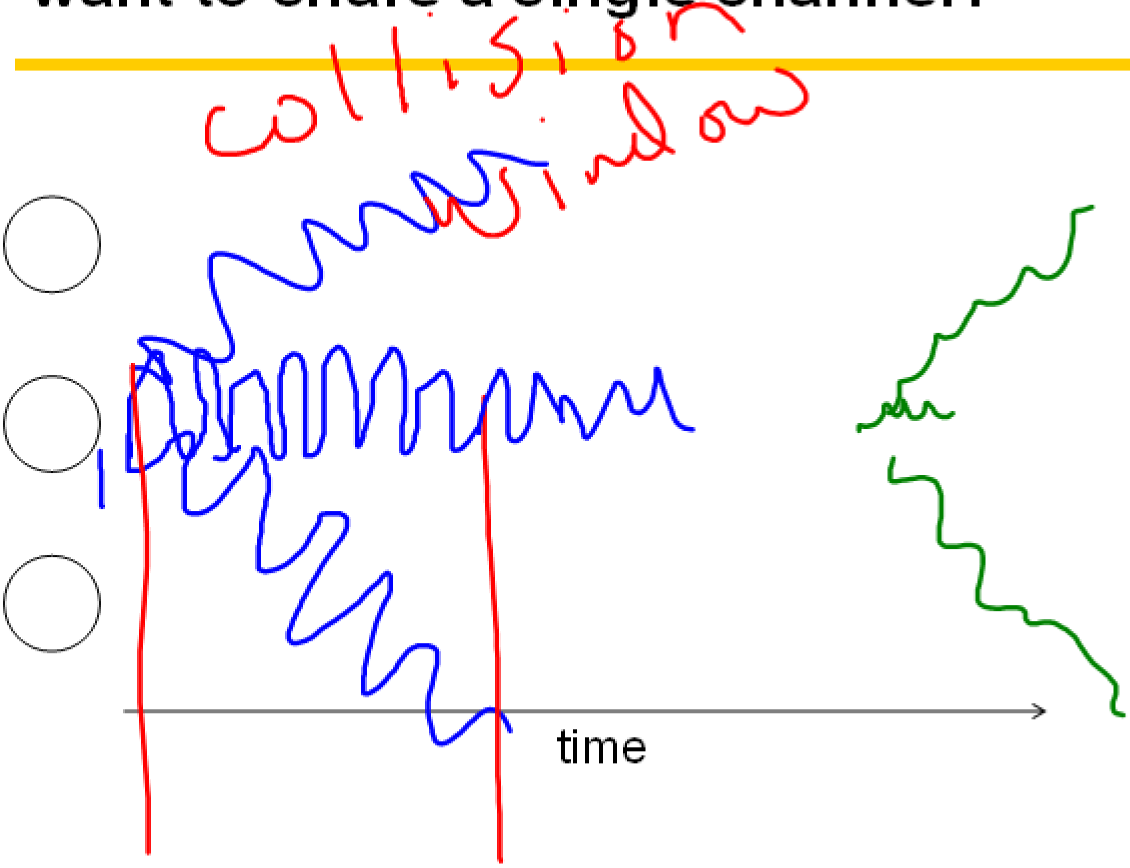
Wired



Wireless



What happens when multiple nodes want to share a single channel?

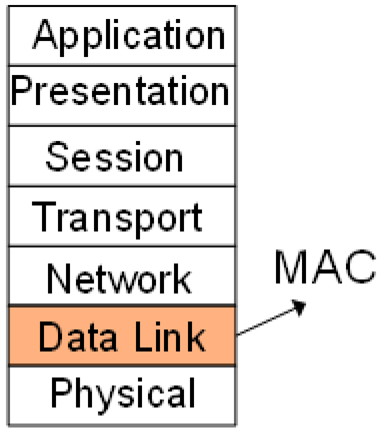


What can you do about it?

partition token ring Frequencies Time code	Statistical muxing

MAC (Medium Access Control)

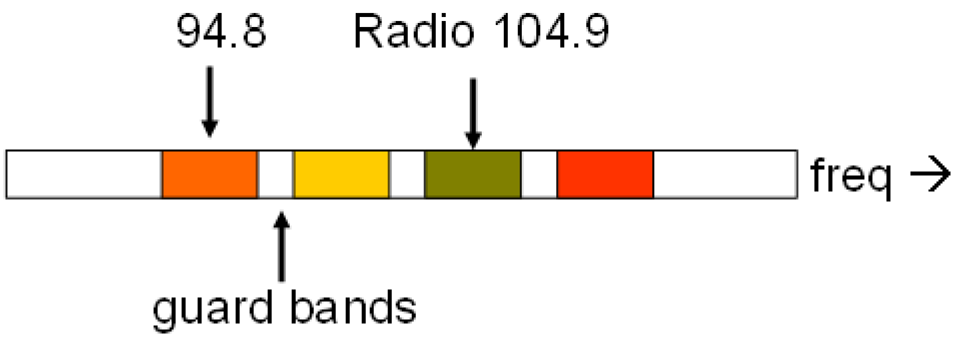
- Ideas:
  - Static partitioning: Division via time, frequency and codes
  - Randomized access protocols
    - Aloha
    - CSMA variants
    - Classic Ethernet
    - Wireless
  - Contention-free protocols
    - Token ring
    - DQDB



Approach #1: Static Partitioning

Frequency Division Multiple Access

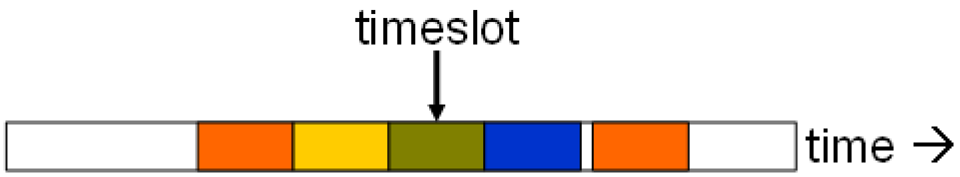
- Simultaneous transmission in different frequency bands
  - Analog: Radio/TV, AMPS cell phones (800MHz)



"Speaking at different pitches"

Time Division Multiple Access

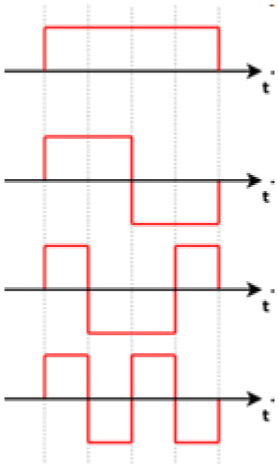
- Timeslice given frequency band among users
  - Digital: used extensively inside the telephone network (GSM)
  - T1 (1.5Mbps) line was originally developed to carry 24 voice signals, each 125us; also E1 (2Mbps, 32 slots)



"Speaking at different times"

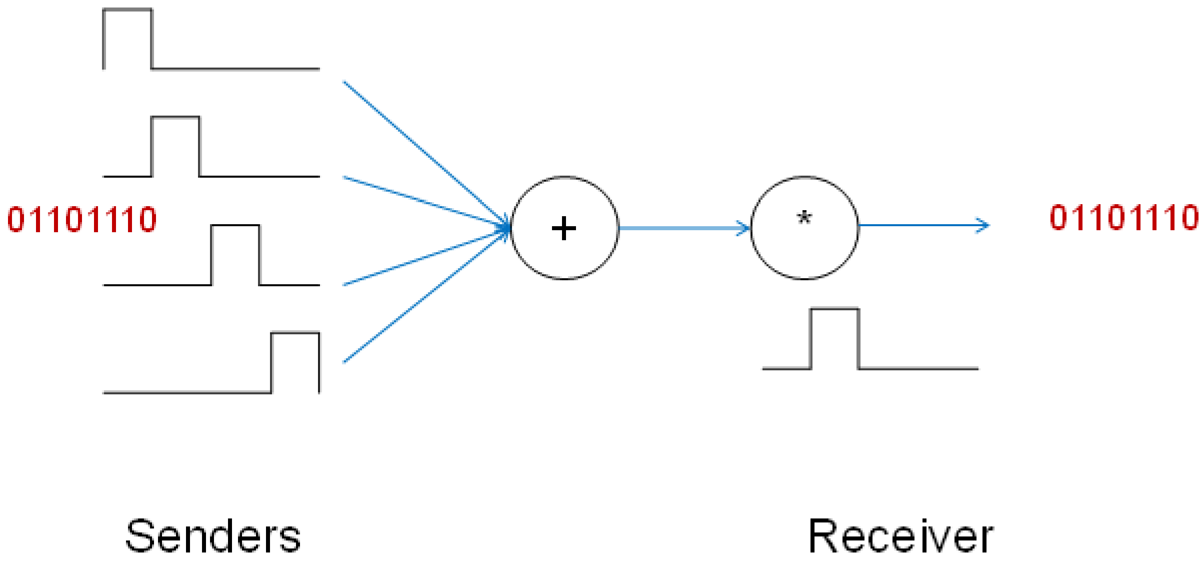
Code Division Multiple Access

- Give each user a different code
  - All users send at once
  - Each user is given a chip code
  - User data is modulated using their chip code
- Codes are orthogonal to each other
  - Can correlate for one code
    - This will ignore the rest
- Widely used for 3G mobile phones

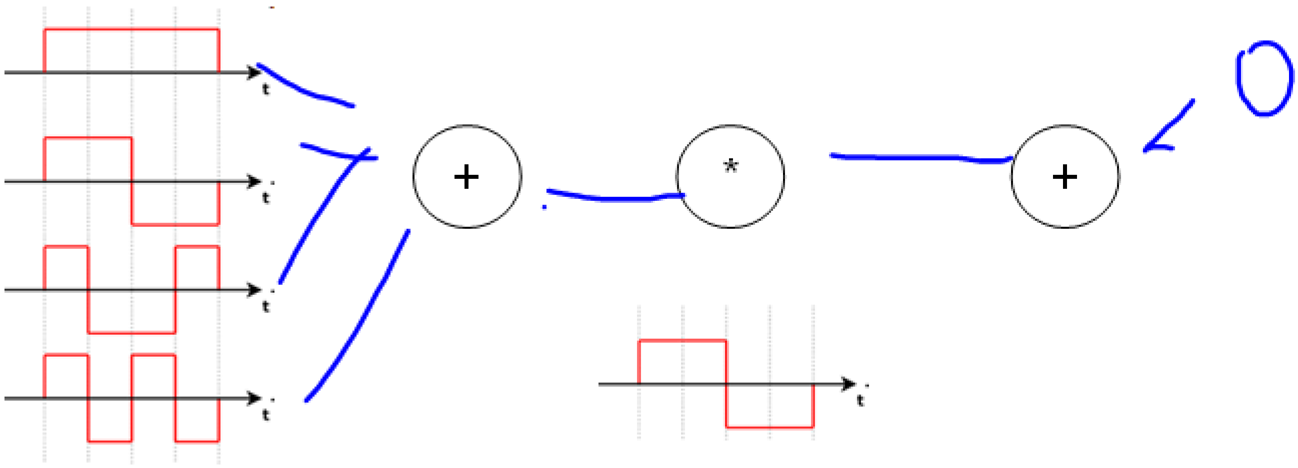


Four mutually orthogonal chip codes

Relationship to TDMA



CDMA Operation



Static partitioning schemes aren't always attractive

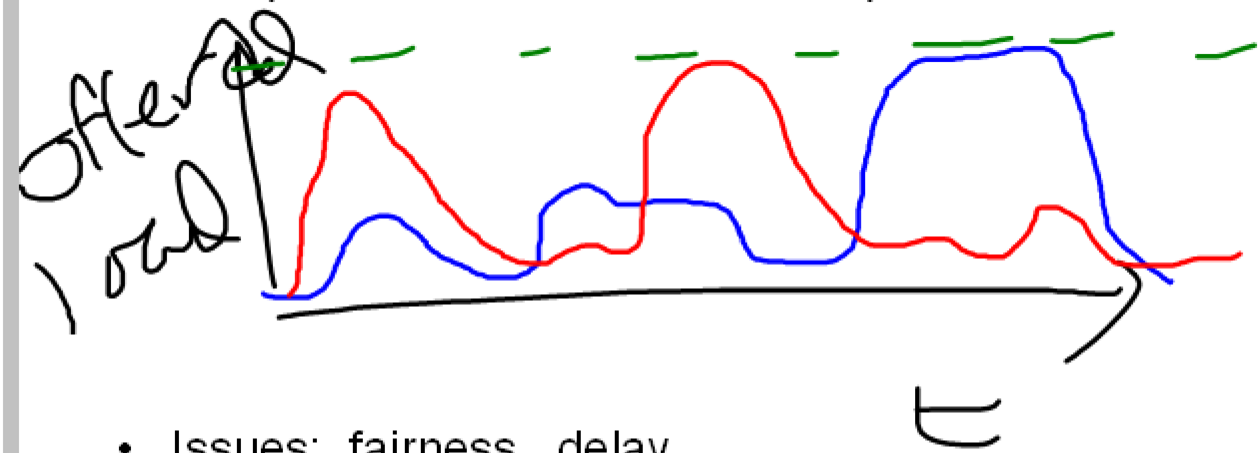
- When are they really bad?

- When are they really good?

- An alternate is statistical multiplexing
  - "On demand" use

Statistical Multiplexing

- Many workloads tend to be bursty

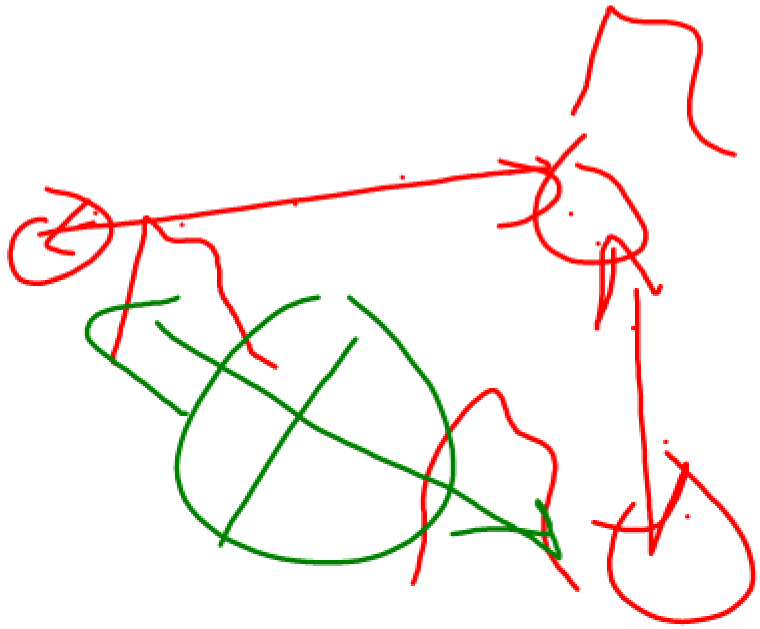


- Issues: fairness, delay

Randomized access: ALOHA

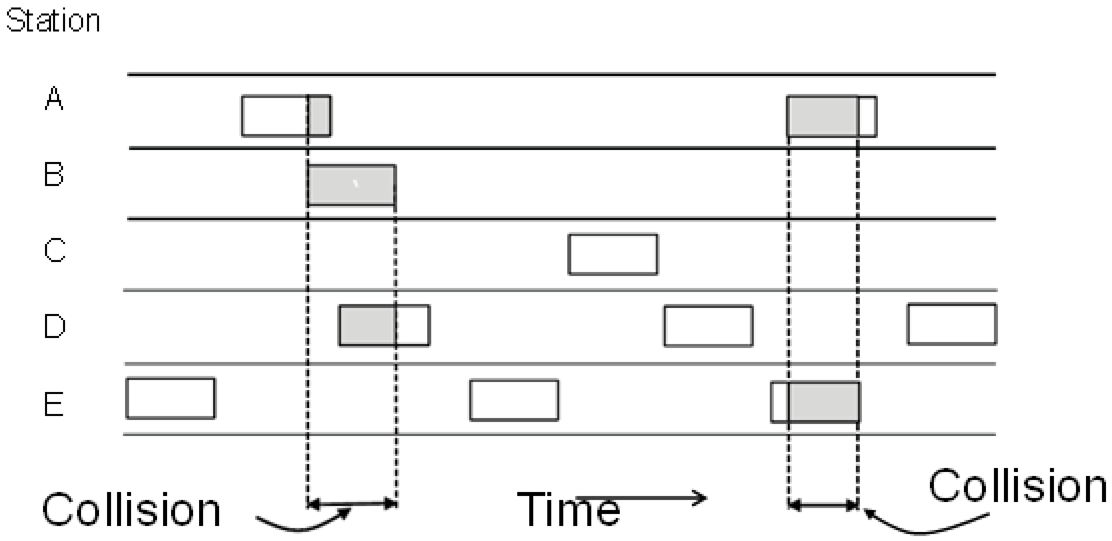
- Wireless links between the Hawaiian islands in the 70s
- Aloha protocol:
  - Just send when you have data!
  - There will be some collisions of course …
  - Detect errored frames and retransmit a random time later
    - How?
    - Why a random time?
- Simple, decentralized and works well for low load
  - For many users, analytic traffic model, max efficiency is 18%

Hawaii



ALOHA (1)

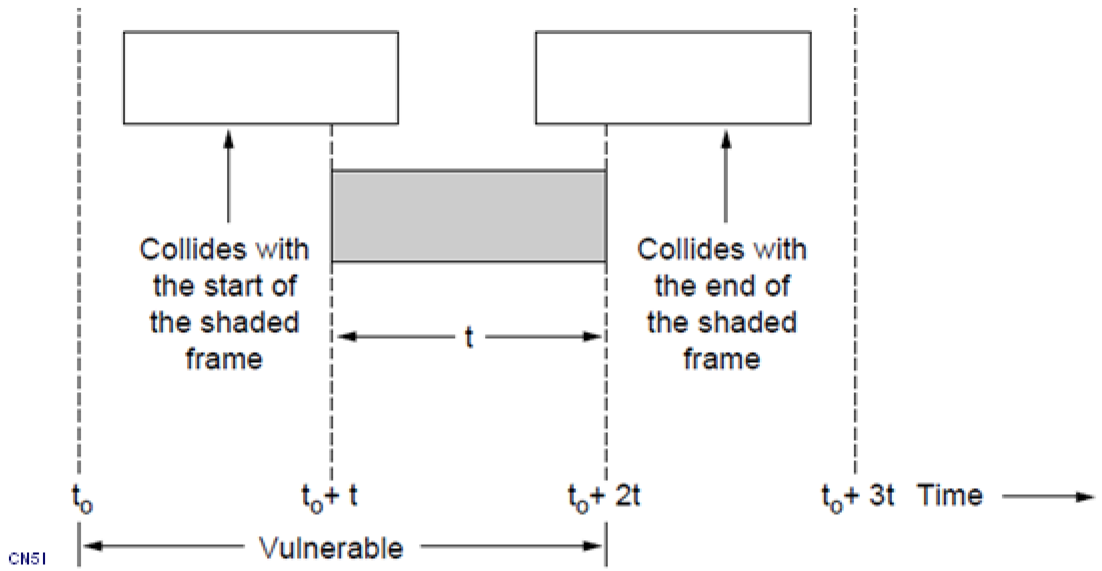
In pure ALOHA, stations transmit frames whenever they have data; stations retry after a random time for collisions



ALOHA (2)

Collisions happen when other users transmit during a vulnerable period that is twice the frame time

- Synchronizing senders to slots can reduce collisions



Slotted Aloha

- Divide time into slots
- Each transmission can start only at the beginning of the slot
- How much better is slotted Aloha compared to Aloha?

Slotted Aloha

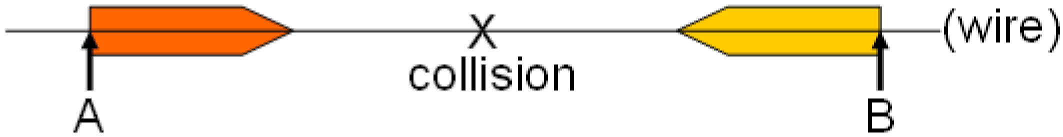
Why does this help?

Recap

- Which layer takes care of access control?
- What are examples of static medium access protocol?
  - Why are static medium access protocols not efficient?
- Aloha/Slotted Aloha as examples of randomized access
- Can we improve efficiency of the Alohas?

Most commonly used randomized MAC: Carrier Sense Multiple Access

- We can do better by listening before we send (CSMA)

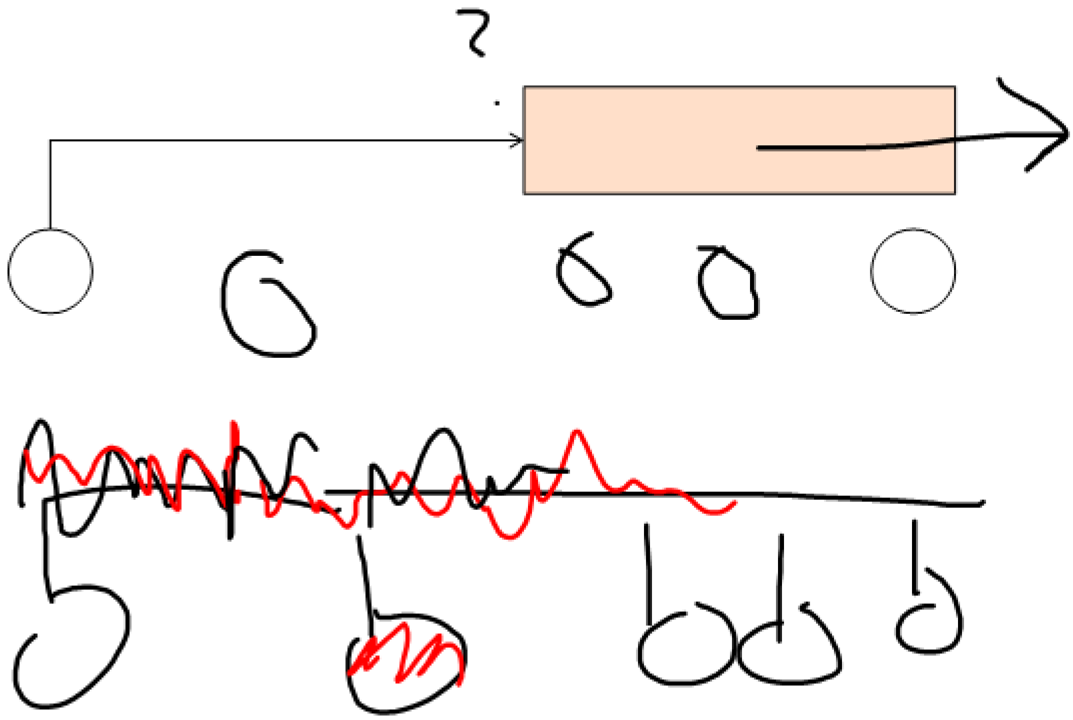


- Since we sense the channel before sending, there should be no collisions, right?

- CSMA works best if bandwidth delay product is low
  - number of packets that fit on the wire = bandwidth * delay / packet size
  - Small (<<1) for LANs, large (>>1) for satellites

Carrier sense and BD product

What if the Channel is sensed busy?



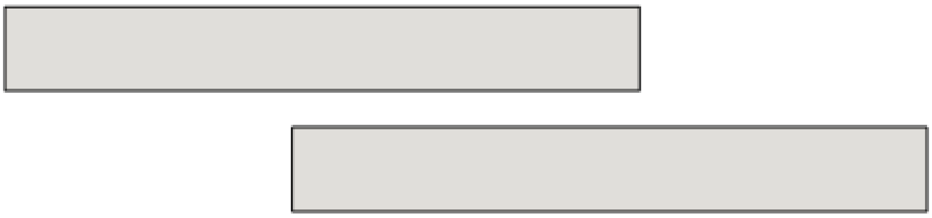
What if the Channel is sensed busy?

- 1-persistent CSMA
  - Wait until idle then go for it
  - Blocked senders can queue up and collide---too greedy
- non-persistent CSMA
  - Wait a random time and try again
- p-persistent CSMA
  - If idle send with probability p until done; assumed slotted time

- Are the number of collisions with p-persistent CSMA likely to be less or more than 1-persistant?
  - How about delay?

CSMA with Collision Detection (CSMA/CD)

- Goal: Make collision losses short:



- For wired media we can detect all collisions and abort (CSMA/CD):
  - Requires a minimum frame size ("acquiring the medium")
  - B must continue sending ("jam") until A detects collision

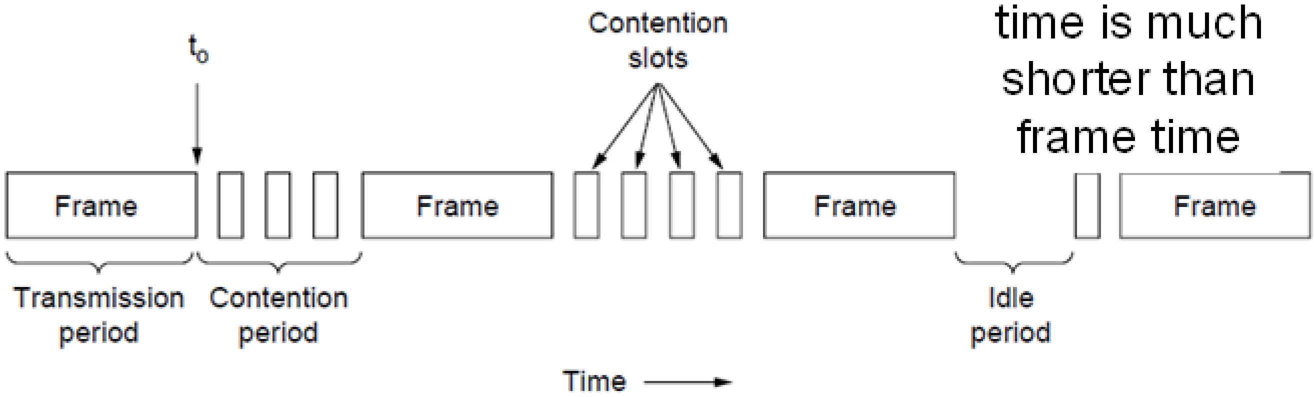
Detecting Collions / Jamming

Minimum frame length is required to guarantee that collision detection works

CSMA/CD

CSMA/CD improvement is to detect/abort collisions

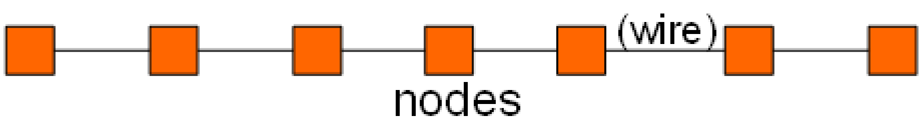
- Reduced contention times improve performance



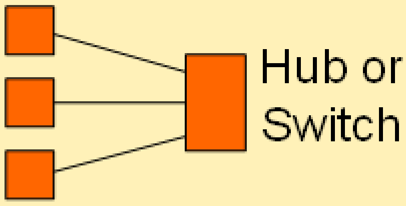
Contention time is much shorter than frame time

Classic Ethernet uses CSMA/CD

- IEEE 802.3 standard wired LAN (1-persistent CSMA/CD)
- Classic Ethernet: 10 Mbps over coaxial cable
  - baseband signals, Manchester encoding, preamble, 32 bit CRC



- BUT: Newer versions are much faster
  - Gigabit Ethernet (1 Gbps)
- Modern equipment isn't one long wire
  - We cover switches later



Ethernet Frames

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

- Min frame 64 bytes, max 1500 bytes
- Max length 2.5km, max between stations 500m (repeaters)
- Addresses unique per adaptor; globally assigned
- Broadcast media

CSMA/CD with Exponential Backoff

- Build on 1-persistent CSMA/CD
- On collision: jam and exponential backoff
  - Jamming: send 48 bit sequence to ensure collision detection
- Backoff:
  - First collision: wait 0 or 1 frame times at random and retry
  - Second time: wait 0, 1, 2, or 3 frame times
  - Nth time (N<=10): wait 0, 1, …, 2^N-1 times
  - Max wait 1023 frames, give up after 16 attempts
  - Scheme balances average wait with load

Classic Ethernet Performance

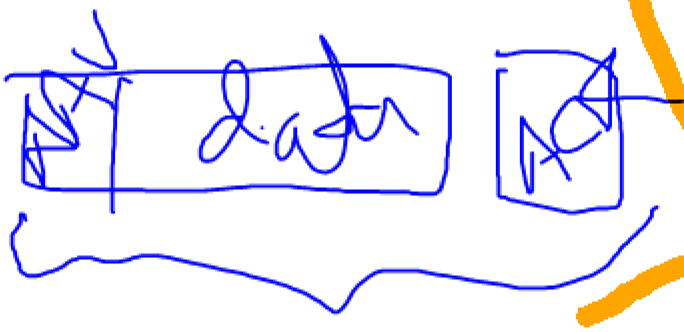
- Much better than Aloha or CSMA!
  - Works very well in practice
  - "Measured Capacity of an Ethernet: Myths and Reality," Boggs, Mogul and Kent, SIGCOMM 88.
- Source of protocol inefficiency: collisions
  - More efficient to send larger frames
    - Acquire the medium and send lots of data
  - Less efficient as the network grows in terms of frames
    - recall "a" = delay / (frame size * transmission rate)
    - "a" grows as the path gets longer (satellite)
    - "a" grows as the bit rates increase (Fast, Gigabit Ethernet)

After collision, what p is optimal for p-persistence?

Recap

hidden

- CSMA senses the channel before sending data. Why do collisions occur in this scenario?
  - What is the important metric that determines if CSMA will work well?
- Why does adding collision detection (CD) improve performance of CSMA?
  - What technology uses CSMA/CD for medium access control?
- What is binary exponential backoff?



Wireless Multiple Access

Wireless is more complicated than wired.

1. Cannot detect collisions
   Why?

2. Every node has a different transmission and reception range
   There is no single "state of the medium"

3. Sender does not know what's happening at the receiver

CSMA with Collision Avoidance

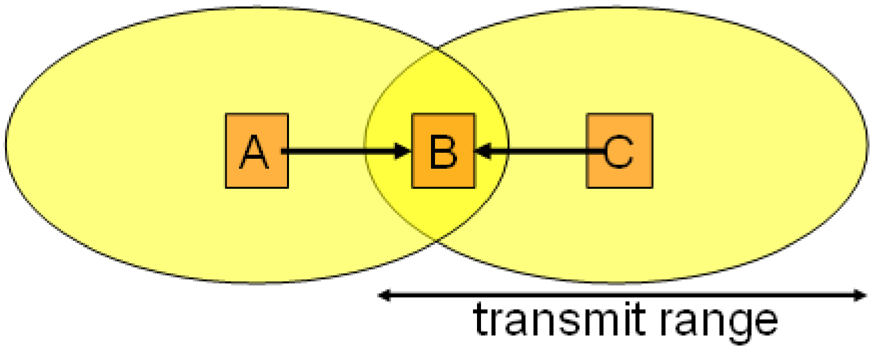
- Since we can't detect collisions, we avoid them
  - CSMA/CA as opposed to CSMA/CD
  - Not greedy like Ethernet
- When medium busy, choose random backoff interval
  - Wait for that many idle timeslots to pass before sending
  - Remember p-persistence? … a refinement
- Unless you're sure frame was received, retransmit
- How do you know frame was received?
  - Use ACKs from receiver

802.11 (WiFi): (Re)transmit scheme

- Wait until sense air is idle
- Send
- If you don't hear an ACK in very short window after you finishing sending, assume frame was lost
  - Binary exponential backoff and try again

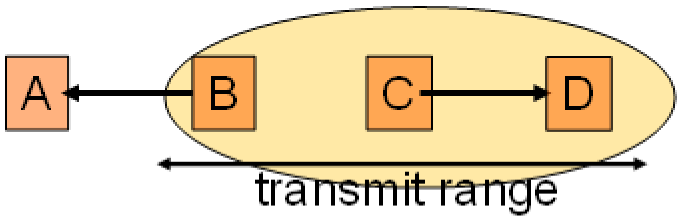
- Why does the 802.11 MAC use ACKs and Ethernet doesn't?
  - What does Ethernet use that 802.11 doesn't?

CSMA not enough: Hidden Terminals

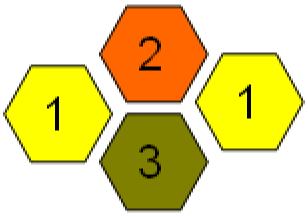


- A and C can both send to B but can't hear each other
  - A is a hidden terminal for C and vice versa
- CSMA will be ineffective – want to sense at receiver

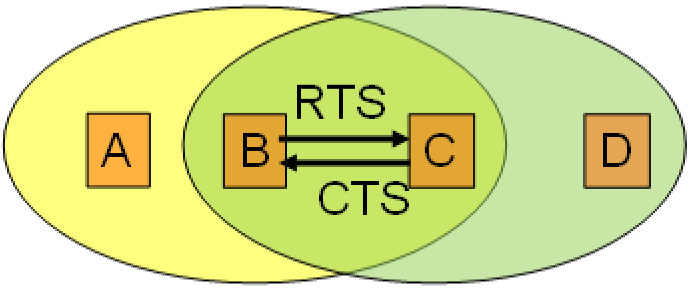
CSMA not enough: Exposed Terminals



- B, C can hear each other but can safely send to A, D
- Compare to spatial reuse in cell phones:



RTS / CTS for hidden terminals



1. B stimulates C with Request To Send (RTS)
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

802.11 Wireless LANs

- Dominant standard with many PHY/MAC options/features

- Wireless plus wired infrastructure
- Avoids collisions with CSMA/CA; RTS/CTS largely unused
- Much PHY processing for high-rate physical links

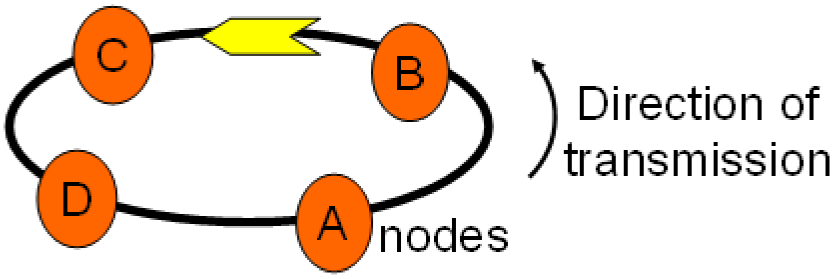
Recap

- CSMA/CD while works well for Ethernet, cannot be used for wireless. Why?
  - Instead, what is used in wireless for medium access control?
- CSMA is still not enough because of _____ terminals.
  - CSMA also causes inefficiences (performance problems) because of ______ terminals
- RTS/CTS can address the ____ terminal problem

Contention-free Protocols

- Collisions are the main difficulty with random schemes
  - To improve efficiency/scalability, many schemes grant ongoing bandwidth and use random schemes for request traffic
- Q: Can we avoid collisions altogether?
- A: Yes. By taking turns or with reservations
  - Token Ring / FDDI, DQDB
- More generally, what else might we want?
  - Deterministic service, priorities/QOS, reliability

Example: Token Ring (802.5)



- Token rotates permission to send around node
- 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
  - Operational issues…