CSE 461: Multiple Access

Homework:
Chapter 2, problems 1, 8, 12, 18, 23, 24, 35, 43, 46, and 58
Key Focus: How do multiple parties share a wire?

This is the Medium Access Control (MAC) portion of the Link Layer

Examples of access protocols:
- Aloha
- CSMA variants
- Classic Ethernet
- Wireless
What is it all about?

- Consider an audio conference where
  - if one person speaks, all can hear
  - if more than one person speaks at the same time, both voices are garbled
- How should participants coordinate actions so that
  - the number of messages exchanged per second is maximized
  - time spent waiting for a chance to speak is minimized
- This is the *multiple access problem*
Some simple solutions

- Use a moderator
  - a speaker must wait for moderator to call on him or her, even if no one else wants to speak
  - what if the moderator’s connection breaks?
- Distributed solution
  - speak if no one else is speaking
  - but if two speakers are waiting for a third to finish, guarantee collision
- Designing good schemes is surprisingly hard!
Multiple Access Protocols

- Single shared broadcast channel
- Two or more simultaneous transmissions by nodes: interference
  - Collision if node receives two or more signals at the same time

**Multiple Access Protocol**

- Distributed algorithm that determines how nodes share channel, i.e., determine when node can transmit
- Communication about channel sharing must use channel itself!
  - No out-of-band channel for coordination
Computer Network Characteristics

- Transmission needs vary
  - Between different nodes
  - Over time
- Network is not fully utilized
Ideal Multiple Access Protocol

Broadcast channel of rate $R$ bps

1. When one node wants to transmit, it can send at rate $R$.

2. When $M$ nodes want to transmit, each can send at average rate $R/M$.

3. Fully decentralized:
   - no special node to coordinate transmissions
   - no synchronization of clocks, slots

4. Simple
Base technologies

- Isolates data from different sources
- Three basic choices
  - Frequency division multiple access (FDMA)
  - Time division multiple access (TDMA)
  - Code division multiple access (CDMA)
FDMA

- Simplest
- Best suited for analog links
- Each station has its own frequency band, separated by guard bands
- Receivers tune to the right frequency
- Number of frequencies is limited
  - reduce transmitter power; reuse frequencies in non-adjacent cells
  - example: voice channel = 30 KHz
  - 833 channels in 25 MHz band
  - with hexagonal cells, partition into 118 channels each
  - but with $N$ cells in a city, can get $118N$ calls => win if $N > 7$
TDMA

- All stations transmit data on same frequency, but at different times
- Needs time synchronization
- Pros
  - users can be given different amounts of bandwidth
  - mobiles can use idle times to determine best base station
  - can switch off power when not transmitting
- Cons
  - synchronization overhead
  - greater problems with multipath interference on wireless links
CDMA

- Users separated both by time and frequency
- Send at a different frequency at each time slot (*frequency hopping*)
- Or, convert a single bit to a code (*direct sequence*)
  - receiver can decipher bit by inverse process
- Pros
  - hard to spy
  - immune from narrowband noise
  - no need for all stations to synchronize
CDMA

- **Cons**
  - implementation complexity
  - need for power control
    - to avoid capture
  - need for a large contiguous frequency band (for direct sequence)
FDD and TDD

- Two ways of converting a wireless medium to a duplex channel
- In Frequency Division Duplex, uplink and downlink use different frequencies
- In Time Division Duplex, uplink and downlink use different time slots
- Can combine with FDMA/TDMA
- Examples
  - TDD/FDMA in second-generation cordless phones
  - FDD/TDMA/FDMA in digital cellular phones
Centralized access schemes

- One station is master, and the other are slaves
  - slave can transmit only when master allows
- Natural fit in some situations
  - wireless LAN, where base station is the only station that can see everyone
  - cellular telephony, where base station is the only one capable of high transmit power
Centralized access schemes

- **Pros**
  - simple
  - master provides single point of coordination

- **Cons**
  - master is a single point of failure
    - need a re-election protocol
    - master is involved in every single transfer => added delay
Polling and reservations

- **Polling**
  - master asks each station in turn if it wants to send (roll-call polling)
  - inefficient if only a few stations are active, overhead for polling messages is high, or system has many terminals

- **Reservation**
  - Some time slots devoted to reservation messages
    - can be smaller than data slots => *minislots*
  - Stations contend for a minislot (or own one)
  - Master decides winners and grants them access to link
Distributed schemes

- Compared to a centralized scheme
  - more reliable
  - have lower message delays
  - often allow higher network utilization
  - but are more complicated
Random Access Protocols

- When node has packet to send
  - transmit at full channel data rate R.
  - no \textit{a priori} coordination among nodes
- two or more transmitting nodes $\Rightarrow$ “collision”
- random access MAC protocol specifies:
  - how to detect collisions
  - how to recover from collisions (e.g., via delayed retransmissions)
- Examples of random access MAC protocols:
  - slotted ALOHA
  - ALOHA
  - CSMA, CSMA/CD, CSMA/CA
ALOHA

- Wireless links between the Hawaiian islands in the 70s
- Want distributed allocation
  - no special channels, or single point of failure

- Aloha protocol:
  - Just send when you have data!
  - There will be some collisions of course ...
  - Detect error frames and retransmit a random time later
Slotted ALOHA

**Assumptions**
- all frames same size
- time is divided into equal size slots, time to transmit 1 frame
- nodes start to transmit frames only at beginning of slots
- nodes are synchronized
- if 2 or more nodes transmit in slot, all nodes detect collision

**Operation**
- when node obtains fresh frame, it transmits in next slot
- no collision, node can send new frame in next slot
- if collision, node retransmits frame in each subsequent slot with prob. $p$ until success
**Slotted ALOHA**

**Pros**
- Single active node can continuously transmit at full rate of channel
- Highly decentralized: only slots in nodes need to be in sync
- Simple

**Cons**
- Collisions, wasting slots
- Idle slots
- Nodes may be able to detect collision in less than time to transmit packet
- Clock synchronization
Efficiency is the long-run fraction of successful slots when there are many nodes, each with many frames to send.

Suppose N nodes with many frames to send, each transmits in slot with probability $p$.

- prob that node 1 has success in a slot = $p(1-p)^{N-1}$
- prob that any node has a success = $Np(1-p)^{N-1}$
Optimal choice of $p$

- For max efficiency with $N$ nodes, find $p^*$ that maximizes $Np(1-p)^{N-1}$
- For many nodes, take limit of $Np^*(1-p^*)^{N-1}$ as $N$ goes to infinity, gives $1/e = 0.37$
- Efficiency is 37%, even with optimal $p$
Pure (unslotted) ALOHA

- unslotted Aloha: simpler, no synchronization
- when frame first arrives
  - transmit immediately
- collision probability increases:
  - frame sent at $t_0$ collides with other frames sent in $[t_0-1,t_0+1]$
Pure Aloha efficiency

\[ P(\text{success by given node}) = P(\text{node transmits}) \cdot P(\text{no other node transmits in } [t_0-1, t_0]) \cdot P(\text{no other node transmits in } [t_0, t_0+1]) \]
\[ = p \cdot (1-p)^{N-1} \cdot (1-p)^{N-1} \]
\[ = p \cdot (1-p)^{2(N-1)} \]

... choosing optimum \( p \) and then letting \( n \to \infty \) ...

Efficiency \( = 1/(2e) = .18 \)

Even worse!
Carrier Sense Multiple Access (CSMA)

- A fundamental advance: listen before you transmit
  - check whether the medium is active before sending a packet (i.e. *carrier sensing*)
  - If channel sensed is idle, transmit entire frame
  - If channel is busy, defer transmission
  - A node with something to send doesn’t have to wait for a master, or for its turn in a schedule

- Human analogy: don’t interrupt others!
CSMA collisions

collisions can still occur:
propagation delay means
two nodes may not hear
each other’s transmission

collision:
entire packet transmission
time wasted

note:
role of distance & propagation delay in
determining collision probability
2. Carrier Sense Multiple Access

- Good defense against collisions only if “a” is small (LANs)
  - “a” parameter: number of packets that fit on the wire
    - a = bandwidth * delay / packet size
    - Small (<<1) for LANs, large (>>1) for satellites
Simplest CSMA scheme

- Send a packet as soon as medium becomes idle
- 1-persistent CSMA
  - Wait until idle then go for it
  - Problem: Blocked senders can queue up and collide
Avoiding Collisions: p-persistent CSMA

- p-persistent CSMA
  - If idle send with prob p until done; assumed slotted time
  - Choose p so $p \times \# \text{senders} < 1$; avoids collisions at cost of delay
Avoiding Collisions: Exponential Backoff

- **exponential backoff**
  - on collision, choose timeout randomly from doubled range
  - backoff range adapts to number of contending stations
  - no need to choose $p$
  - need to detect collisions: *collision detect circuit* $\Rightarrow$ CSMA/CD
CSMA/CD (Collision Detection)

**CSMA/CD**: carrier sensing, deferral as in CSMA

- collisions *detected* within short time
- colliding transmissions aborted, reducing channel wastage

- collision detection:
  - easy in wired LANs: measure signal strengths, compare transmitted, received signals
  - difficult in wireless LANs: receiver shut off while transmitting

- human analogy: the polite conversationalist
CSMA/CD collision detection
Ethernet (IEEE 802.3)

dominant wired LAN technology:
- cheap <$20 for Gigabit!
- first widely used LAN technology
- Simpler, cheaper than token LANs and ATM
- Kept up with speed race: 10 Mbps – 10 Gbps

Metcalfe’s Ethernet sketch
Ethernet Topologies

Bus Topology: Shared
All nodes connected to a wire

Star Topology:
All nodes connected to a central repeater (hub or switch)
Ethernet Connectivity

10Base5 – ThickNet
< 500m

Controller

Vampire Tap

Bus Topology

Transceiver
Ethernet Connectivity

10Base2 – ThinNet
< 200m

Controller

Transceiver

BNC T-Junction

Bus Topology
Ethernet Connectivity

10BaseT
< 100m

Controller

Star Topology
**Ethernet II Frame Structure**

Sending adapter encapsulates IP datagram (or other network layer protocol packet) in Ethernet frame.

<table>
<thead>
<tr>
<th>Preamble (8)</th>
<th>Dest (6)</th>
<th>Source (6)</th>
<th>Type (2)</th>
<th>Payload (var)</th>
<th>Pad (var)</th>
<th>CRC (4)</th>
</tr>
</thead>
</table>

**Preamble:**

- 7 bytes with pattern 10101010 followed by one byte with pattern 10101011
- Used to synchronize receiver, sender clock rates (Manchester encoding)
Ethernet Frame Structure (more)

- **Addresses:** 6 bytes
  - if adapter receives frame with matching destination address, or with broadcast address (e.g. ARP packet), it passes data in frame to net-layer protocol
  - otherwise, adapter discards frame
- **Type:** higher layer protocol (usually IP, but Novell IPX, Apple Talk, and others supported)
- **Data:** min 64 bytes (why?), max 1500 bytes
- **CRC:** checked at receiver, if error is detected, the frame is simply dropped
Ethernet Specifications

- **Coaxial Cable**
  - Max between stations 500m
  - Max length 2.5km with repeaters

- **Taps**
  - > 2.5m apart

- **Transceiver**
  - Idle detection
  - Sends/Receives signal

- **Repeater**
  - Joins multiple Ethernet segments
  - < 5 repeaters between any two hosts

- < 1024 hosts
Ethernet MAC Algorithm

- **Sender/Transmitter**
  - If line is idle (carrier sensed)
    - Send immediately
    - Send maximum of 1500B data (1527B total)
    - Wait 9.6 µs before sending again
  - If line is busy (no carrier sensed)
    - Wait until line becomes idle
    - Send immediately
  - If collision detected
    - Stop sending and jam signal
    - Try again later
Ethernet MAC Algorithm

Node A starts transmission at time 0

At time almost T, node A’s message has almost arrived

Node B starts transmission at time T

How can we ensure that A knows about the collision?
Collision Detection

- Example
  - Node A’s message reaches node B at time T
  - Node B’s message reaches node A at time 2T
  - For node A to detect a collision, node A must still be transmitting at time 2T

- 802.3
  - 2T is bounded to 51.2µs
  - At 10Mbps $51.2\,\mu s = 512b$ or $64B$
  - Packet length $\geq 64B$

- Jam after collision
  - Ensures that all hosts notice the collision
Ethernet MAC Algorithm

Node A starts transmission at time 0.

At time almost T, node A’s message has almost arrived.

Node B starts transmission at time T.

At time 2T, A is still transmitting and notices a collision.
Binary Exponential Backoff

- How long should a host wait to retry after a collision?
- Build on 1-persistent CSMA/CD
- On collision: jam and exponential backoff
- Binary Exponential Backoff:
  - Colliding hosts pick a random number from 0 to $2^{(N-1)}$
  - First collision: wait 0 or 1 slot 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
Binary Exponential Backoff

Choices after 1 collision

0
Ts
2Ts
3Ts

Time of collision

Why use fixed time slots?

How long should the slots be?
CSMA/CD efficiency

- Much better than ALOHA, but still decentralized, simple, and cheap

- $t_{\text{trans}} = \text{time to transmit max-size frame}$

- $t_{\text{prop}} = \text{max prop between 2 nodes in LAN}$

- More efficient to send larger frames (Efficiency $\rightarrow 1$ as $t_{\text{trans}} \rightarrow \infty$)
  - Acquire the medium and send lots of data
  - Worse for Fast, Gigabit Ethernet where $t_{\text{trans}}$ is short

- Smaller networks more efficient (Efficiency $\rightarrow 1$ as $t_{\text{prop}} \rightarrow 0$)
  - Worse as path gets longer (e.g., satellite)
Ethernet Capture

- Randomized access scheme is not fair

- Stations A and B always have data to send
  - They will collide at some time
  - Suppose A wins and sends, while B backs off
  - Next time they collide and B’s chances of winning are halved!
Frame Reception

- Sender handles all access control
- Receiver simply pulls the frame from the network
- Ethernet controller/card
  - Sees all frames
  - Selectively passes frames to host processor
- Acceptable frames
  - Addressed to host
  - Addressed to broadcast
  - Addressed to multicast address to which host belongs
  - Anything (if in promiscuous mode)
    - Need this for packet sniffers/TCPDump
Contestation-free Protocols

- Collisions are the main difficulty with random schemes
  - Inefficiency, limit to scalability

- Q: Can we avoid collisions?
- A: Yes. By taking turns or with reservations
  - Token Ring / FDDI, DQDB

- More generally, what else might we want?
  - Deterministic service, priorities/QOS, reliability
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
  - Token release after sending data
  - Round robin service, acknowledgments and priorities
- Monitor nodes ensure health of ring
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
    - Supports both single attached and dual attached stations
  - Complex token holding policies for voice etc. traffic
    - Guaranteed rotation every Target Token Rotation Time (TTRT)

- Token ring advantages
  - No contention, bounded access delay
  - Supports fair, reserved, priority access

- Disadvantages
  - Complexity, reliability, scalability

Break!
Token passing

- In distributed polling, every station has to wait for its turn
- Time wasted because idle stations are still given a slot
- What if we can quickly skip past idle stations?
- This is the key idea of token ring
- Special packet called ‘token’ gives station the right to transmit data
- When done, it passes token to ‘next’ station
  - => stations form a logical ring
- No station will starve
Logical rings

- Can be on a non-ring physical topology

(a) Single ring
(b) Dual ring
(c) Token bus
(d) Hub or star-ring
Ring operation

- During normal operation, copy packets from input buffer to output
- If packet is a token, check if packets ready to send
  - If not, forward token
  - If so, delete token, and send packets
- Receiver copies packet and sets ‘ack’ flag
- Sender removes packet and deletes it
- When done, reinserts token
- If ring idle and no token for a long time, regenerate token
Hub or star-ring

- Simplifies wiring
- Active hub is predecessor and successor to every station
  - can monitor ring for station and link failures
- Passive hub only serves as wiring concentrator
  - but provides a single test point
- Because of these benefits, hubs are practically the only form of wiring used in real networks
  - even for Ethernet
Evaluating token ring

**Pros**
- medium access protocol is simple and explicit
- no need for carrier sensing, time synchronization or complex protocols to resolve contention
- guarantees zero collisions
- can give some stations priority over others

**Cons**
- token is a single point of failure
  - lost or corrupted token trashes network
  - need to carefully protect and, if necessary, regenerate token
- all stations must cooperate
  - network must detect and cut off unresponsive stations
- stations must actively monitor network
  - usually elect one station as monitor
Key Concepts

- Multiple access networks
  - Share medium by dividing up time, frequency, code
  - Are either controlled or fully distributed
- Key concerns: fairness and efficiency
  - Overhead: collisions and uselessly waiting
- Popular standards:
  - Ethernet (random access, CSMA/CD)
  - Token ring (contention-free)