Shared Access Networks

Outline
- Bus (Ethernet)
- Token ring (FDDI)
- Wireless (802.11)

Ethernet Overview

• History
  - developed by Xerox PARC in mid-1970s
  - roots in Aloha packet-radio network
  - standardized by Xerox, DEC, and Intel in 1978
  - similar to IEEE 802.3 standard

• CSMA/CD
  - carrier sense
  - multiple access
  - collision detection

• Frame Format

<table>
<thead>
<tr>
<th></th>
<th>64</th>
<th>48</th>
<th>48</th>
<th>16</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preamble</td>
<td>Dest addr</td>
<td>Src addr</td>
<td>Type</td>
<td>Body</td>
<td>CRC</td>
</tr>
</tbody>
</table>

Ethernet (cont)

• Addresses
  - unique, 48-bit unicast address assigned to each adapter
  - example: 8:0:e4:b1:2
  - broadcast: all 1s
  - multicast: first bit is 1

• Bandwidth: 10Mbps, 100Mbps, 1Gbps
• Length: 2500m (500m segments with 4 repeaters)
• Problem: Distributed algorithm that provides fair access

Transmit Algorithm

• If line is idle...
  - send immediately
  - upper bound message size of 1500 bytes
  - must wait 9.6us between back-to-back frames

• If line is busy...
  - wait until idle and transmit immediately
  - called 1-persistent (special case of p-persistent)
Algorithm (cont)

- If collision...
  - jam for 32 bits, then stop transmitting frame
  - minimum frame is 64 bytes (header + 46 bytes of data)
  - delay and try again
    - 1st time: 0 or 51.2us
    - 2nd time: 0, 51.2, 102.4, or 153.6us
    - nth time: $k \times 51.2$us, for randomly selected $k=0..2^{n-1}$
  - give up after several tries (usually 16)
  - exponential backoff

Collisions

Token Ring Overview

- Examples
  - 16Mbps IEEE 802.5 (based on earlier IBM ring)
  - 100Mbps Fiber Distributed Data Interface (FDDI)

Token Ring (cont)

- Idea
  - Frames flow in one direction: upstream to downstream
  - special bit pattern (token) rotates around ring
  - must capture token before transmitting
  - release token after done transmitting
    - immediate release
    - delayed release
  - remove your frame when it comes back around
  - stations get round-robin service

Frame Format

<table>
<thead>
<tr>
<th>8</th>
<th>8</th>
<th>8</th>
<th>48</th>
<th>48</th>
<th>Variable</th>
<th>32</th>
<th>8</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start delimiter</td>
<td>Access control</td>
<td>Frame control</td>
<td>Dest addr</td>
<td>Src addr</td>
<td>Body</td>
<td>Checksum</td>
<td>End delimiter</td>
<td>Frame status</td>
</tr>
</tbody>
</table>
**Timed Token Algorithm**

- **Token Holding Time (THT)**
  - upper limit on how long a station can hold the token

- **Token Rotation Time (TRT)**
  - how long it takes the token to traverse the ring
  - $TRT \leq ActiveNodes \times THT + RingLatency$

- **Target Token Rotation Time (TTRT)**
  - agreed-upon upper bound on TRT

**Algorithm (cont)**

- Each node measures TRT between successive tokens
  - if measured-TRT > TTRT: token is late so don’t send
  - if measured-TRT < TTRT: token is early so OK to send

- Two classes of traffic
  - synchronous: can always send
  - asynchronous: can send only if token is early

- Worse case: $2 \times TTRT$ between seeing token
- Back-to-back $2 \times TTRT$ rotations not possible

**Token Maintenance**

- **Lost Token**
  - no token when initializing ring
  - bit error corrupts token pattern
  - node holding token crashes

- **Generating a Token (and agreeing on TTRT)**
  - execute when join ring or suspect a failure
  - send a *claim frame* that includes the node’s TTRT bid
  - when receive claim frame, update the bid and forward
  - if your claim frame makes it all the way around the ring:
    - your bid was the lowest
    - everyone knows TTRT
    - you insert new token

**Maintenance (cont)**

- Monitoring for a Valid Token
  - should periodically see valid transmission (frame or token)
  - maximum gap = ring latency + max frame $\leq 2.5ms$
  - set timer at 2.5ms and send claim frame if it fires
Wireless LANs

- IEEE 802.11
- Bandwidth: 1 - 11 Mbps
- Physical Media
  - spread spectrum radio (2.4GHz)
  - diffused infrared (10m)

Spread Spectrum

- Idea
  - spread signal over wider frequency band than required
  - originally designed to thwart jamming
- Frequency Hopping
  - transmit over random sequence of frequencies
  - sender and receiver share
    - pseudorandom number generator
    - seed
  - 802.11 uses 79 x 1MHz-wide frequency bands

Spread Spectrum (cont)

- Direct Sequence
  - for each bit, send XOR of that bit and \( n \) random bits
  - random sequence known to both sender and receiver
  - called \( n \)-bit chipping code
  - 802.11 defines an 11-bit chipping code

Collisions Avoidance

- Similar to Ethernet
- Problem: hidden and exposed nodes
MACAW

- Sender transmits \texttt{RequestToSend} (RTS) frame
- Receiver replies with \texttt{ClearToSend} (CTS) frame
- Neighbors...
  - see CTS: keep quiet
  - see RTS but not CTS: ok to transmit
- Receive sends ACK when has frame
  - neighbors silent until see ACK
- Collisions
  - no collisions detection
  - known when don’t receive CTS
  - exponential backoff

Supporting Mobility

- Case 1: \textit{ad hoc} networking
- Case 2: \textit{access points} (AP)
  - tethered
  - each mobile node associates with an AP

Mobility (cont)

- Scanning (selecting an AP)
  - node sends \texttt{Probe} frame
  - all AP’s w/in reach reply with \texttt{ProbeResponse} frame
  - node selects one AP; sends it \texttt{AssociateRequest} frame
  - AP replies with \texttt{AssociationResponse} frame
  - new AP informs old AP via tethered network
- When
  - active: when join or move
  - passive: AP periodically sends \texttt{Beacon} frame