Multiple Access Links and Protocols

Two types of "links":
- point-to-point
  - PPP for dial-up access
  - point-to-point link between Ethernet switch and host
- broadcast (shared wire or medium)
  - traditional Ethernet
  - upstream HFC
  - 802.11 wireless LAN

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
- 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

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

MAC Protocols: a taxonomy

Three broad classes:
- Channel Partitioning
  - divide channel into smaller "pieces" (time slots, frequency, code)
  - allocate piece to node for exclusive use
- Random Access
  - channel not divided, allow collisions
  - "recover" from collisions
- "Taking turns"
  - Nodes take turns, but nodes with more to send can take longer turns

Channel Partitioning MAC protocols: TDMA

TDMA: time division multiple access
- access to channel in "rounds"
- each station gets fixed length slot (length = pkt trans time) in each round
- unused slots go idle
- example: 6-station LAN, 1,3,4 have pkt, slots 2,5,6 idle

Channel Partitioning MAC protocols: FDMA

FDMA: frequency division multiple access
- channel spectrum divided into frequency bands
- each station assigned fixed frequency band
- unused transmission time in frequency bands go idle
- example: 6-station LAN, 1,3,4 have pkt, frequency bands 2,5
Random Access Protocols

- When node has packet to send
  - transmit at full channel data rate $R$
  - no 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

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

<table>
<thead>
<tr>
<th>Node 1</th>
<th>Node 2</th>
<th>Node 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

**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

### Slotted ALOHA efficiency

**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 = $Np(1-p)^{N-1}$

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 $1/e \approx .37$

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

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

$P(\text{success by given node}) = P(\text{node transmits})$

- $P(\text{no other node transmits in } [p_1, p_2])$
- $P(\text{no other node transmits in } [p_0-1, p_2])$
- $P(\text{no other node transmits in } [p_0-1, p_0])$

Choosing optimum $p$ and then letting $n \to \infty$...

$= 1/(2e) = .18$

Even worse!
**CSMA (Carrier Sense Multiple Access)**

- **CSMA**: listen before transmit:
  - If channel sensed idle: transmit entire frame
  - If channel sensed busy, defer transmission
- 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

**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**

**“Taking Turns” MAC protocols**

- **Channel partitioning MAC protocols**:
  - Share channel efficiently and fairly at high load
  - Inefficient at low load: delay in channel access, 1/N bandwidth allocated even if only 1 active node!
- **Random access MAC protocols**
  - Efficient at low load: single node can fully utilize channel
  - High load: collision overhead
- “Taking turns” protocols look for best of both worlds!
Summary of MAC protocols

❒ What do you do with a shared media?
  ❒ Channel partitioning, by time, frequency or code
    ❒ Time division, frequency division
  ❒ Random partitioning (dynamic),
    ❒ ALOHA, S-ALOHA, CSMA, CSMA/CD
    ❒ Carrier sensing: easy in some technologies (wire), hard in others (wireless)
    ❒ CSMA/CD used in Ethernet
    ❒ CSMA/CA used in 802.11
  ❒ Taking Turns
    ❒ Polling from a central site, token passing

LAN technologies

Data link layer so far:
  ❒ Services, error detection/correction, multiple access

Next: LAN technologies
  ❒ Addressing
  ❒ Ethernet
  ❒ Hubs, switches
  ❒ PPP

MAC Addresses and ARP

❒ 32-bit IP address:
  ❒ Network-layer address
  ❒ Used to get datagram to destination IP subnet

❒ MAC (or LAN or physical or Ethernet) address:
  ❒ Used to get frame from one interface to another physically-connected interface (same network)
  ❒ 48-bit MAC address (for most LANs) burned in the adapter ROM

LAN Addresses and ARP

Each adapter on LAN has unique LAN address

Broadcast address = FF-FF-FF-FF-FF-FF

<table>
<thead>
<tr>
<th>IP address</th>
<th>MAC address</th>
<th>TTL</th>
</tr>
</thead>
<tbody>
<tr>
<td>237.196.7.23</td>
<td>1A-2F-BB-76-09-AD</td>
<td>20</td>
</tr>
<tr>
<td>237.196.7.14</td>
<td>1A-2F-BB-76-09-AD</td>
<td>20</td>
</tr>
<tr>
<td>237.196.7.14</td>
<td>1A-2F-BB-76-09-AD</td>
<td>20</td>
</tr>
</tbody>
</table>

LAN Address (more)

❒ MAC address allocation administered by IEEE
❒ Manufacturer buys portion of MAC address space
  (to assure uniqueness)
❒ Analogy:
  (a) MAC address: like Social Security Number
  (b) IP address: like postal address
❒ MAC flat address ➔ portability
  ❒ Can move LAN card from one LAN to another
❒ IP hierarchical address NOT portable
  ❒ Depends on IP subnet to which node is attached

ARP: Address Resolution Protocol

Question: how to determine MAC address of B knowing B’s IP address?

❒ Each IP node (Host, Router) on LAN has ARP table
❒ ARP Table: IP/MAC address mappings for some LAN nodes
  (IP address: MAC address: TTL)
  ❒ TTL (Time To Live): time after which address mapping will be forgotten (typically 20 min)
**ARP protocol: Same LAN (network)**

- A wants to send datagram to B, and B’s MAC address not in A’s ARP table.
- A broadcasts ARP query packet, containing B’s IP address
  - Dest MAC address = FF-FF-FF-FF-FF-FF
  - all machines on LAN receive ARP query
- B receives ARP packet, replies to A with its (B’s) MAC address (unicast)
- A caches (saves) IP-to-MAC address pair in its ARP table until information becomes old (times out)
- ARP is “plug-and-play”:
  - nodes create their ARP tables without intervention from net administrator

**Routing to another LAN**

walkthrough: send datagram from A to B via R

- A knows B's IP address

- Two ARP tables in router R, one for each IP network (LAN)