Review of Multiple Access Protocols

- Channel partitioning
  - divide channel into smaller “pieces” (time slots, frequency, code), allocate piece to node for exclusive use
  - Time Division, Frequency Division, Code Division

- Random access
  - channel not divided, allow collisions, recover from collisions
  - ALOHA, S-ALOHA, CSMA, CSMA/CD

- Centrally controlled/coordinate
  - coordinate shared access to avoid collisions

Ethernet

Developed by Metcalfe at Xerox PARC in 1970s
First widely used technology for local-area networks (now dominant)
Standardized as IEEE 802.3
For shared access link

Ethernet Frame Structure

Sender encapsulates data payload in Ethernet frame

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

- Preamble: 7 bytes with pattern 10101010 followed by one byte with pattern 10101011 indicating the beginning of a frame, synchronizing receiver/sender clocks
- Addresses: link layer addresses, different from IP addresses
- Length: length of the body field
  - max 1500bytes standard, higher in jumbo frames
- CRC: error detection
Not Supported

- **Connectionless**:
  - no handshaking between sender and receiver
  - no connection-based resource management, flow control

- **Unreliable**:
  - corrupted frames are dropped, no attempt to retransmit
  - lost frames are ignored (in fact, not even bother to detect)
  - no guarantee of in-order delivery

**Rationale**:
- chance of corruption/loss is small
- corrupted or lost packets may be recovered by higher layer protocols, e.g., TCP

---

Ethernet uses CSMA/CD

- Random access, no synchronized clocks

- One doesn’t transmit if it senses that another is transmitting, that is, **carrier sense**

- Transmitter aborts when it senses that another is transmitting, that is, **collision detection**

- Before attempting a retransmission, one waits a non-deterministic delay

---

Ethernet’s CSMA/CD Backoff

**Load-based backoff**:
- Longer backoff when more are competing for the link
  - Previous conflicts indicate high load, repeated conflicts indicate higher load

**Specifically**:
- First collision: delay $K$ units of time, $K$ is randomly chosen from $\{0,1\}$
  - a unit of time separation should be sufficient for carrier sense to avoid conflict (e.g., 5 microsec)
- After second collision: choose $K$ from $\{0,1,2,3\}$
  - ... ...
- After tenth collisions, choose $K$ from $\{0,1,2,3,4,\ldots,1023\}$
- Also called **exponential backoff**

---

Ethernet Efficiency

- $t_{\text{prop}} = \text{max prop between 2 nodes}$
- $t_{\text{trans}} = \text{time to transmit a frame}$

\[
\text{efficiency} \approx \frac{1}{1 + 5t_{\text{prop}} / t_{\text{trans}}}
\]

- Higher efficiency at lower $t_{\text{prop}}$ or higher $t_{\text{trans}}$

More efficient than ALOHA, but still decentralized and simple.
Ethernet Technologies: 10Base2

- **10**: 10Mbps; **2**: under 200 meters max cable length
- thin coaxial cable in a bus topology

10BaseT and 100BaseT

- 10/100 Mbps rate; latter called "fast ethernet"
- **T** stands for Twisted Pair
- Nodes connect to a hub/switch: "star topology"; 100 m max distance between nodes and hub/switch

Encoding

- 10BaseT, 10Base2 – Manchester encoding
- 100BaseT – 4B5B encoding
- Allows clocks in sending and receiving nodes to synchronize to each other
  - no need for a centralized, global clock among nodes
- Still remember their efficiencies??

Gbit Ethernet

- 1Gbps and even 10Gbps
  - if latency doesn’t change, can all apps benefit?
- efficiency of CSMA/CD
  \[
  \text{efficiency} = \frac{1}{1 + 5 \frac{t_{\text{prop}}}{t_{\text{trans}}}}
  \]
- short node distances, large frames to be efficient!
- fortunately today’s Ethernet links are not shared
Where is link layer implemented?

- Link layer implemented in “adaptor” (aka NIC)
  - Ethernet card, wireless card
- Sending side:
  - Encapsulates data in a frame
  - Adds error checking bits, rdt, flow control, etc.
- Receiving side:
  - Looks for errors, rdt, flow control, etc.
  - Extracts data, passes to OS at receiving node
  - Adapter is semi-autonomous
  - Has its own processor/buffer

Interaction with the Host

- Control from the host:
  - Read/write control status registers
- Notifying the host:
  - Interrupts
- Moving data:
  - Direct memory access (DMA)

Link-layer Addresses

Link-layer (or MAC, LAN) Address:

- Uniquely identifying each node (or adapter) connected to a shared link
  - Integral to multiple access protocols
- 48 bits
- Broadcast address – “FF-FF-FF-FF-FF-FF”

Properties of Link-layer Addresses

- Per-device static address:
  - Factory pre-set address for each adapter
  - Change possible (OS may instruct the adapter to use a different MAC address), but not usual
- Universal uniqueness:
  - No two adapters anywhere have the same pre-set address
  - How is this achieved? MAC address allocation administered by IEEE and a manufacturer gets portion of address space.
  - Portability: can move adapter from one network to another (no configuration required!)
A Problem

Question: how to determine B’s link-layer address knowing B’s IP address?

ARP: Address Resolution Protocol

- Each node (host, router) on LAN has ARP table
  - IP/Link-layer address mappings for LAN nodes <IP addr; Link-layer addr>
  - Use ARP table to find link-layer address

How does ARP work?

First principle: no manual configuration - “plug-and-play”; must deal with dynamic network changes

- A wants to send datagram to B with an IP address.
- Suppose B’s link-layer address is not in A’s ARP table.
- A broadcasts ARP query packet, containing B’s IP address.
- B receives ARP packet, replies to A with its (B’s) link-layer address.

Link-layer protocol or network-layer protocol?

Two addresses

- Co-existence of link-layer and network-layer addresses
- Why not just one?
Disclaimer

- Parts of the lecture slides contain original work of James Kurose, Larry Peterson, and Keith Ross. The slides are intended for the sole purpose of instruction of computer networks at the University of Rochester. All copyrighted materials belong to their original owner(s).