Reminder: Datalink Functions

- **Framing**: encapsulating a network layer datagram into a bit stream.
  - Add header, mark and detect frame boundaries, ...
- **Error control**: error detection and correction to deal with bit errors.
  - May also include other reliability support, e.g. retransmission
- **Flow control**: avoid sender overrunning receiver.
- **Media access control (MAC)**: which frame should be sent over the link next.
  - Easy for point-to-point links
  - Harder for multi-access links: who gets to send?

Datalink Architectures

- **Switches connected by point-to-point links -- store-and-forward**.
  - Used in WAN, LAN, and for home connections
  - Conceptually similar to "routing"
  - But at the datalink layer instead of the network layer
  - MAC = (local) scheduling
- **Multiple access networks** - contention based.
  - Multiple hosts are sharing the same transmission medium
  - Used in LANs and wireless
  - Access control is distributed and much more complex
Datalink Classification

- Datalink
  - Switch-based
    - Virtual Circuits
    - Packet Switching
  - Multiple Access
    - Scheduled Access
    - Random Access

Switching

- Forward units of data based on address in header.
- Many data-link technologies use switching.
  - Virtual circuits: Frame Relay, ATM, X.25, ...
  - Packets: Ethernet, ...
- “Switching” also happens at the network layer.
  - Layer 3: Internet protocol
  - In this case, address is an IP address
  - IP over SONET, IP over ATM, ...
  - Otherwise, operation is very similar
- Switching is different from traditional (hard) circuits
  - E.g., telephone switches (not covered in this course)
  - Switching is based on timing – no addresses

A Switch-based Network

- Switches are connected by point-point links.
- Packets are forwarded hop-by-hop by the switches towards the destination.
  - Forwarding is based on the address
  - How does a switch work?
  - How do nodes exchange packets over a link?
  - How is the destination addressed?

Switch Architecture

- Packets come in one interface, forwarded to output interface based on address.
  - Same idea for bridges, switches, routers: address look up differs
  - Control processor manages the switch and executes higher level protocols.
    - E.g. routing, management, ...
  - The switch fabric directs the traffic to the right output port.
  - The input and output ports deal with transmission and reception of packets.
Connections or Not?

- Two basic approaches to packet forwarding
  - Connectionless
  - (virtual) Circuit switched

- When would you use?

Connectionless

- Host can send anytime anywhere
- No idea if resources are available to get to destination
- Forwarding is independent for each packet
- No setup time
- Fault tolerant
  - More on this later

<table>
<thead>
<tr>
<th>Destination</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>3</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>7</td>
</tr>
<tr>
<td>D</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td></td>
</tr>
</tbody>
</table>

Virtual Circuit Switching

- Two stage process similar to traditional circuits
  - Setup connection + create VC ID
  - Send packets
  - RTT introduced before any data is sent
  - Per packet overhead can be smaller (VCI << adr)
  - Switch failures are hard to deal with
  - Reserves resources for connection possible
  - Widely used in core networks (e.g. MPLS)
  - More on this later

Setup, assign VCIs
Packet Forwarding: Address Lookup

- Address from header.
  - Absolute address (e.g., Ethernet)
  - (VC identifier, e.g., ATM)
  - (IP address for routers)
- Next hop: output port for packet.
- Info: priority, timeout, ...
- Table is filled in by a protocol

<table>
<thead>
<tr>
<th>Address</th>
<th>Next Hop</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>112.123.4</td>
<td>3</td>
<td>13</td>
</tr>
<tr>
<td>201.123.4</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td>25</td>
<td>2</td>
<td>31</td>
</tr>
<tr>
<td>128.2.15.3</td>
<td>1</td>
<td>(2,34)</td>
</tr>
</tbody>
</table>

Datalink Classification

- Datalink
  - Switch-based
  - Multiple Access
    - Virtual Circuits
    - Packet Switching
    - Scheduled Access
    - Random Access

Problem: Sharing a Wire (or the wireless ether)

- Problem: how do you prevent nodes from “talking” at the same time – causes “collision”
- Two classes of solutions:
  - Explicit coordination: schedule transmissions sequentially
  - Randomly access medium: and hope you get lucky

Outline: Multiple Access Protocols

- Scheduled access (short)
- Random access (long)
  - Aloha
  - Ethernet MAC
  - Collisions
  - Ethernet Frames
Scheduled Access MACs

- Reservation systems
  - Central controller
  - Distributed algorithm, e.g. using reservation bits in frame
- Polling: controller polls each nodes
- Token ring: token travels around ring and allows nodes to send one packet
  - Distributer version of polling
  - FDDI, ...

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 → “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 and ALOHA
  - CSMA and CSMA/CD

Aloha – Basic Technique

- First random MAC developed
  - For radio-based communication in Hawaii (1970)
- Basic idea:
  - When ready, transmit
  - Receivers send ACK for data
  - Detect collisions by timing out for ACK
  - Recover from collision by trying after random delay

Aloha Throughput Comparison

- It is possible to calculate throughput for Aloha
  - Many assumptions: exponential arrival, transmitters independent, ...
- Bad news: maximum throughput is low
- Slotted Aloha (a variant) can achieve higher throughput
  - But has higher latency, especially under low load

\[
G = \text{offered load} = N \times p
\]

\[
S = \text{throughput} = \text{[plot of throughput vs. offered load]}
\]

Protocol constrains effective channel throughput!
How Can We Do Better?

- Natural scheme – listen before you talk...
  - Works well in practice
  - A cheap form of coordination
- But sometimes this breaks down
  - Why? How do we fix/prevent this?

Ethernet

- First practical local area network, built at Xerox PARC in 70’s
- Dominant LAN technology:
  - Cheap
  - Kept up with the speed race: 10, 100, 1000, ... Mbps

Ethernet MAC Features

- Carrier Sense: listen before you talk
  - Avoid collision with active transmission
- Collision Detection during transmission
  - Listen while transmitting
  - If you notice interference → assume collision
  - Abort transmission immediately – saves time
- Why didn’t ALOHA have this?
  - Signal strength is reduced by distance for radio
    - May not hear remote transmitter – hidden terminal
  - Very difficult for radios to listen and transmit
  - More on this later in the course

Ethernet MAC – CSMA/CD

- Carrier Sense Multiple Access/Collision Detection
Ethernet CSMA/CD: Making it work

Jam Signal: make sure all other transmitters are aware of collision; 48 bits;
Exponential Backoff:
- If deterministic delay after collision, collision will occur again in lockstep
- Why not random delay with fixed mean?
  - Few senders → needless waiting
  - Too many senders → too many collisions
- Goal: adapt retransmission attempts to estimated current load
  - heavy load: random wait will be longer

Ethernet Backoff Calculation

- Delay is set as K slots – control K
- Exponentially increasing random delay
  - Infer senders from # of collisions
  - More senders → increase wait time
- First collision: choose K from {0,1}; delay is K x 512 bit transmission times
- After second collision: choose K from {0,1,2,3}...
- After ten or more collisions, choose K from {0,1,2,3,4...,1023}

Outline: Multiple Access Protocols

- Scheduled access (short)
- Random access (long)
  - Aloha
  - Ethernet MAC
  - Collisions
  - Ethernet Frames

Collisions
**Minimum Packet Size**

- Packets must be long enough to guarantee all nodes observe collision
- Depends on packet size and length of wire
  - Propagation delay
  - Min packet length > 2x max prop delay

**Delay & Collision Detection**

- Speed in cable \(\sim 60\% \times c \sim 1.8 \times 10^{8} \text{ m/s}\)
- 10Mb Ethernet, 2.5km cable
  - \~12.5\text{us} delay
  - +Introduced repeaters (max 5 segments)
  - Worst case – 51.2\text{us} round trip time!
  - Corresponds to 512 bits
- Also used as slot time = 51.2\text{us} for backoff
  - After this time, sender is guaranteed sole access to link
  - Specifically, will have heard any signal sent in the previous slot

**Scaling Ethernet**

- What about scaling? 10Mbps, 100Mbps, 1Gbps, ...
  - Use a combination of reducing network diameter and increasing minimum minimum packet size
- Reality check: 40 Gbps is 4000 times 10 Mbps
  - 10 Mbps: 2.5 km and 64 bytes -> silly
  - Solution: switched Ethernet – next lecture
- What about a maximum packet size?
  - Needed to prevent node from hogging the network
  - 1500 bytes in Ethernet = 1.2 msec on original Ethernet
  - For 40 Gps -> 0.3 microsec -> silly and inefficient

**Ethernet Frame Structure**

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

- **Preamble**: 8 bytes
  - 101010...1011
  - Used to synchronize receiver, sender clock rates
- **CRC**: 4 bytes
  - Checked at receiver, if error is detected, the frame is simply dropped
- **Type**: 2 bytes
  - Demultiplexing: indicates the higher layer protocol, mostly IP today but historically more protocols (such as Novell IPX and AppleTalk)

Addressing Alternatives

- **Broadcast** → all nodes receive all packets
  - Addressing determines which packets are kept and which are packets are thrown away
  - Packets can be sent to:
    - Unicast – one destination
    - Multicast – group of nodes (e.g. “everyone playing Quake”)  
    - Broadcast – everybody on wire
- **Dynamic addresses** (e.g. Appletalk)
  - Pick an address at random
  - Broadcast “is anyone using address XX?”
  - If yes, repeat
- **Static address** (e.g. Ethernet)

Ethernet Address Assignment

- Each adapter is given a globally unique 6-byte IEEE address at manufacturing time
  - Address space is allocated to manufacturers
    - 24 bits identify manufacturer
      - E.g., 0:0:15:* → 3com adapter
    - Frame is received by all adapters on a LAN and dropped if address does not match
- **Special addresses**
  - Broadcast – FF:FF:FF:FF:FF:FF is “everybody”
  - Range of addresses allocated to multicast
    - Adapter maintains list of multicast groups node is interested in

Why Did Ethernet Win?

- **Failure modes**
  - Token rings – network unusable (or expensive)
- **Good performance in common case**
  - Deals well with bursty traffic
  - Usually used at low load
- **Volume** → lower cost → higher volume ....
- **Adaptable**
  - To higher bandwidths (vs. FDDI)
  - To switching (vs. ATM)
- **Easy incremental deployment** (backwards compatible)
- **Cheap cabling, etc**
And .. It is Easy to Manage

- You plug in the host and it basically works
  - Zero configuration at the datalink layer
  - Today: may need to deal with security
- Protocol is fully distributed
- Broadcast-based.
  - In part explains the easy management
  - Some LAN protocols (e.g. ARP) rely on broadcast
    - Networking would be harder without ARP
  - Not having natural broadcast capabilities adds complexity to a LAN (e.g., ATM)
- Network managers love it!

Summary

- CSMA/CD $\rightarrow$ carrier sense multiple access with collision detection
  - Why do we need exponential backoff?
  - Why does collision happen?
  - Why do we need a minimum packet size?
    - How does this scale with speed?
- Ethernet
  - What is the purpose of different header fields?
  - What do Ethernet addresses look like?
- What are some alternatives to Ethernet design?