Goals of Today’s Lecture

- MAC (Media Access Control) protocols, esp.
  - CSMA/CD
    - Carrier Sense Multiple Access / Collision Detection
- Ethernet: single segment
  - Frame structure
  - Length/timing constraints due to Collision Detection
- Ethernet: spanning multiple segments
  - Repeaters and hubs
  - Bridges and switches
  - Self-learning (plug-and-play)
  - Spanning trees (time permitting)

Three Ways to Share the Media

- Channel partitioning MAC protocols (TDMA, FDMA):
  - Share channel efficiently and fairly at high load
  - Inefficient at low load (where load = # senders):
    - 1/N bandwidth allocated even if only 1 active node!
- “Taking turns” protocols (discussed in Section)
  - Eliminates empty slots without causing collisions
  - Overhead in acquiring the token
  - Vulnerable to failures (e.g., failed node or lost token)
- Random access MAC protocols
  - Efficient at low load: single node can fully utilize channel
  - High load: collision overhead

Key Ideas of Random Access

- Carrier sense
  - Listen before speaking, and don’t interrupt
  - Checking if someone else is already sending data
  - ... and waiting till the other node is done
- Collision detection
  - If someone else starts talking at the same time, stop
  - Realizing when two nodes are transmitting at once
  - ...by detecting that the data on the wire is garbled
- Randomness
  - Don’t start talking again right away
  - Waiting for a random time before trying again
CSMA Collisions

Collisions can still occur: propagation delay means two nodes may not hear each other’s transmission in time.

At time $t_0$, node $D$ still hasn’t heard node $B$’s signal sent at the earlier time $t_0$, so $D$ goes ahead and transmits: failure of carrier sense.

Collision: entire packet transmission time wasted

CSMA/CD (Collision Detection)

- CSMA/CD: carrier sensing, deferral as in CSMA
  - Collisions detected within short time
  - Colliding transmissions aborted, reducing wastage
  
- Collision detection
  - Easy in wired LANs: measure signal strengths, compare transmitted, received signals
  - Difficult in wireless LANs
    - Reception shut off while transmitting
    - Even if on, might not be able to hear the other sender, even though the receiver can
    - Leads to use of collision avoidance instead

CSMA/CD Collision Detection

Both $B$ and $D$ can tell that collision occurred.

This lets them (1) know that they need to resend the frame, and (2) recognize that there’s contention and adopt a strategy for dealing with it.

Note: for this to work, we need restrictions on minimum frame size and maximum distance

Ethernet: CSMA/CD Protocol

- Carrier sense: wait for link to be idle
- Collision detection: listen while transmitting
  - No collision: transmission is complete
  - Collision: abort transmission & send jam signal
- Random access: exponential back-off
  - After collision, wait a random time before trying again
  - After $m$th collision, choose $K$ randomly from $\{0, \ldots, 2^{m-1}\}$
  - … and wait for $K\times512$ bit times before trying again
- The wired LAN technology
  - Hugely successful: 3/10/100/1000/10000 Mbps
Minimum Packet Size

- Why enforce a minimum packet size?
- Give a host enough time to detect collisions
- In Ethernet, minimum packet size = 64 bytes (two 6-byte addresses, 2-byte type, 4-byte CRC, and 46 bytes of data)
- If host has less than 46 bytes to send, the adaptor pads (adds) bytes to make it 46 bytes
- What is the relationship between minimum packet size and the length of the LAN?

Minimum Packet Size (more)

a) Time = t; Host 1 starts to send frame
b) Time = t + d; Host 2 starts to send a frame, just before it hears from host 1’s frame
c) Time = t + 2d; Host 1 hears Host 2’s frame → detects collision

LAN length = (min_frame_size)*(light_speed)/(2*bandwidth) =
= (8*64b)*(2.5*10^8mps)/(2*10^7 bps) = 6400m approx

What about 100 mbps? 1 gbps? 10 gbps?

Limits on CSMA/CD Network Length

- Latency depends on physical length of link
  - Time to propagate a packet from one end to the other
- Suppose A sends a packet at time \( t \)
  - And B sees an idle line at a time just before \( t + d \)
  - ... so B happily starts transmitting a packet
- B detects a collision, and sends jamming signal
  - But A can’t see collision until \( t + 2d \)

Limits on CSMA/CD Network Length

- A needs to wait for time \( 2d \) to detect collision
  - So, A should keep transmitting during this period
  - ... and keep an eye out for a possible collision
- Imposes restrictions. E.g., for 10 Mbps Ethernet:
  - Maximum length of the wire: 2,500 meters
  - Minimum length of a frame: 512 bits (64 bytes)
    - 512 bits = 51.2 µsec (at 10 Mbit/sec)
    - For light in vacuum, 51.2 µsec ≈ 15,000 meters
      vs. 5,000 meters “round trip” to wait for collision
Ethernet Frame Structure

- Sending adapter encapsulates packet in frame
  - **Preamble**: synchronization
    - Seven bytes with pattern 10101010, followed by one byte with pattern 10101011
    - Used to synchronize receiver & sender
  - **Type**: indicates the higher layer protocol
    - Usually IP (but also Novell IPX, AppleTalk, …)
  - **CRC**: cyclic redundancy check
    - Receiver checks & simply drops frames with errors

---

Ethernet Frame Structure (Continued)

- **Addresses**: 48-bit source and destination MAC addresses
  - Receiver’s adaptor passes frame to network-level protocol
    - If destination address matches the adapter’s
      - Or the destination address is the broadcast address (ff:ff:ff:ff:ff:ff)
      - Or the destination address is a multicast group receiver belongs to
        - Or the adapter is in promiscuous mode
    - Addresses are globally unique
      - Assigned by NIC vendors (top three octets specify vendor)
        - During any given week, > 500 vendor codes seen at LBNL
  - **Data**: maximum: 1,500 bytes
    - Minimum: 46 bytes (+14 bytes header + 4 byte trailer = 512 bits)

---

Ethernet, con’t

- Connectionless
  - No handshaking between sending and receiving adapter
- Unreliable
  - Receiving adapter doesn’t send ACKs or NACKs
  - Packets passed to network layer can have gaps
    - Gaps will be filled if application is using TCP
    - Otherwise, application will see the gaps
  - 2,700 page IEEE 802.3 standardization
    - http://standards.ieee.org/getieee802/802.3.html
    - Note, “classical” Ethernet has no length field …
      - … instead, sender pauses 9.2 µsec when done
      - 802.3 shoehorns in a length field

---

Benefits of Ethernet

- Easy to administer and maintain
- Inexpensive
- Increasingly higher speed

- Evolved from shared media to switches
  - And from electrical signaling to also optical
  - Changes everything except the frame format
- A good general lesson for evolving the Internet:
  - The right interface (service model) can often accommodate unanticipated changes
### Shutting Data at Different Layers

- Different devices switch different things
  - Physical layer: electrical signals (*repeaters* and *hubs*)
  - Link layer: frames (*bridges* and *switches*)
  - Network layer: packets (*routers*)

### Physical Layer: Repeaters

- Distance limitation in local-area networks
  - Electrical signal becomes weaker as it travels
  - Imposes a limit on the length of a LAN
    - In addition to limit imposed by collision detection
- Repeaters join LANs together
  - Analog electronic device
  - Continuously monitors electrical signals on each LAN
  - Transmits an amplified copy

### Physical Layer: Hubs

- Joins multiple input lines electrically
  - Do not necessarily amplify the signal
- Very similar to repeaters
  - Also operates at the physical layer

### Limitations of Repeaters and Hubs

- One large collision domain
  - Every bit is sent everywhere
  - So, aggregate throughput is limited
  - E.g., three departments each get 10 Mbps independently
    - … and then if connect via a hub must share 10 Mbps
- Cannot support multiple LAN technologies
  - Repeaters/hubs do not buffer or interpret frames
  - So, can’t interconnect between different rates or formats
  - E.g., no mixing 10 Mbps Ethernet & 100 Mbps Ethernet
5 Minute Break

Questions Before We Proceed?

Switches & Concurrent Communication

- Host A can talk to C, while B talks to D

  A
  switch
  B

  C
  D

- If host has (dedicated) point-to-point link to switch:
  - **Full-duplex**: each connection can send in both directions
    o At the same time (otherwise, "half-duplex")
  - **Completely avoids collisions**
    o No need for carrier sense, collision detection, and so on

Link Layer: Switches / Bridges

- Connect two or more LANs at the **link layer**
  - Extracts destination address from the frame
  - Looks up the destination in a table
  - Forwards the frame to the appropriate LAN segment
    - Or point-to-point link, for higher-speed Ethernet
  - Each segment is its own collision domain

![Link Layer Diagram]

Advantages Over Hubs & Repeaters

- Only forwards frames as needed
  - Filters frames to avoid unnecessary load on segments
  - Sends frames only to segments that need to see them
- Extends the geographic span of the network
  - Separate collision domains allow longer distances
- Improves privacy by limiting scope of frames
  - Hosts can “snoop” the traffic traversing their segment
    … but not all the rest of the traffic
- If needed, applies carrier sense & collision detection
  - Does not transmit when the link is busy
  - Applies exponential back-off after a collision
- Joins segments using different technologies
Disadvantages Over Hubs & Repeaters

- Higher cost
  - More complicated devices that cost more money
- Delay in forwarding frames
  - Bridge/switch must receive and parse the frame
  - ... and perform a look-up to decide where to forward
  - Introduces store-and-forward delay
    - Can ameliorate using cut-through switching
      - Start forwarding after only header received
- Need to learn where to forward frames
  - Bridge/switch needs to construct a forwarding table
  - Ideally, without intervention from network administrators
- Solution: self-learning

Motivation For Self Learning

- Large benefit if switch/bridge forward frames only on segments that need them
  - Allows concurrent use of other links
- Switch table
  - Maps destination MAC address to outgoing interface
  - Goal: construct the switch table automatically

Self Learning: Building the Table

- When a frame arrives
  - Inspect source MAC address
  - Associate address with the incoming interface
  - Store mapping in the switch table
  - Use time-to-live field to eventually forget the mapping
    - Soft state

Self Learning: Handling Misses

- When frame arrives with unfamiliar destination
  - Forward the frame out all of the interfaces (“flooding”)
  - ... except for the one where the frame arrived
  - Hopefully, this case won’t happen very often
  - When destination replies, switch learns that node, too

When in doubt, shout!
Switch Filtering / Forwarding

When switch receives a frame:
- index the switch table using MAC dest address
- if entry found for destination {
  - if dest on segment from which frame arrived
    - then drop frame
  - else forward frame on interface indicated
- } else flood

Problems?
- forward on all but the interface on which the frame arrived

Flooding Can Lead to Loops
- Switches sometimes need to broadcast frames
  - Upon receiving a frame with an unfamiliar destination
  - Upon receiving a frame sent to the broadcast address
  - Implemented by flooding
- Flooding can lead to forwarding loops
  - E.g., if the network contains a cycle of switches
    - Either accidentally, or by design for higher reliability
    - “Broadcast storm”

Solution: Spanning Trees
- Ensure the forwarding topology has no loops
  - Avoid using some of the links when flooding
  - … to prevent loop from forming
- Spanning tree (K&R pp. 406-408)
  - Sub-graph that covers all vertices but contains no cycles
  - Links not in the spanning tree do not forward frames

Constructing a Spanning Tree
- Need a distributed algorithm
  - Switches cooperate to build the spanning tree
  - … and adapt automatically when failures occur
- Key ingredients of the algorithm
  - Switches need to elect a root
    - The switch w/ smallest identifier (MAC addr)
  - Each switch determines if its interface is on the shortest path from the root
    - Excludes it from the tree if not
  - Messages (Y, d, X)
    - From node X
    - Proposing Y as the root
    - And the distance is d
Steps in Spanning Tree Algorithm

- Initially, each switch proposes itself as the root
  - Switch sends a message out every interface
  - … proposing itself as the root with distance 0
  - Example: switch X announces (X, 0, X)
- Switches update their view of the root
  - Upon receiving message (Y, d, Z) from Z, check Y’s id
  - If new id smaller, start viewing that switch as root
- Switches compute their distance from the root
  - Add 1 to the distance received from a neighbor
  - Identify interfaces not on shortest path to the root
  - … and exclude them from the spanning tree
  - If root or shortest distance to it changed, flood updated message (Y, d+1, X)

Example From Switch #4’s Viewpoint

- Switch #4 thinks it is the root
  - Sends (4, 0, 4) message to 2 and 7
- Then, switch #4 hears from #2
  - Receives (2, 0, 2) message from 2
  - … and thinks that #2 is the root
  - And realizes it is just one hop away
- Then, switch #4 hears from #7
  - Receives (2, 1, 7) from 7
  - And realizes this is a longer path
  - So, prefers its own one-hop path
  - And removes 4-7 link from the tree

Example From Switch #4’s Viewpoint

- Switch #2 hears about switch #1
  - Switch 2 hears (1, 1, 3) from 3
  - Switch 2 starts treating 1 as root
  - And sends (1, 2, 2) to neighbors
  - Switch #4 hears from switch #2
  - Switch 4 starts treating 1 as root
  - And sends (1, 3, 4) to neighbors
  - Switch #4 hears from switch #7
  - Switch 4 receives (1, 3, 7) from 7
  - And realizes this is a longer path
  - So, prefers its own three-hop path
  - And removes 4-7 link from the tree

Robust Spanning Tree Algorithm

- Algorithm must react to failures
  - Failure of the root node
    - Need to elect a new root, with the next lowest identifier
  - Failure of other switches and links
    - Need to recompute the spanning tree
- Root switch continues sending messages
  - Periodically reannouncing itself as the root (1, 0, 1)
  - Other switches continue forwarding messages
- Detecting failures through timeout (soft state)
  - Switch waits to hear from others
  - Eventually times out and claims to be the root
Moving From Switches to Routers

- Advantages of switches over routers
  - Plug-and-play
  - Fast filtering and forwarding of frames

- Disadvantages of switches over routers
  - Topology restricted to a spanning tree
  - Large networks require large ARP tables
  - Broadcast storms can cause the network to collapse
  - Can’t accommodate non-Ethernet segments (why not?)

Comparing Hubs, Switches & Routers

<table>
<thead>
<tr>
<th></th>
<th>hubs</th>
<th>switches</th>
<th>routers</th>
</tr>
</thead>
<tbody>
<tr>
<td>traffic isolation</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>plug &amp; play</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>optimized routing</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>cut-through</td>
<td>yes</td>
<td>yes*</td>
<td>no*</td>
</tr>
</tbody>
</table>

Summary

- Ethernet as an exemplar of link-layer technology
- Simplest form, single segment:
  - Carrier sense, collision detection, and random access
- Extended to span multiple segments:
  - Hubs & repeaters: physical-layer interconnects
  - Bridges / switches: link-layer interconnects
- Key ideas in switches
  - Self learning of the switch table
  - Spanning trees