Ethernet: Links, Hubs, Switches

EE 122: Intro to Communication Networks
Fall 2006 (MW 4-5:30 in Donner 155)

Vern Paxson
TAs: Dilip Antony Joseph and Sukun Kim

http://inst.eecs.berkeley.edu/~ee122/

Materials with thanks to Jennifer Rexford, Ion Stoica, and colleagues at Princeton and UC Berkeley

Announcements

• Office hours (329 Soda)
  – Regular slot moving to Weds 3-4PM (half hour later)
  – Extra office hours: Monday Oct 16 1:30-3:30PM
  – Also by appointment, but not this Thursday/Friday
Goals of Today’s Lecture

• Ethernet: single segment
  – Carrier sense, collision detection, and random access
  – Frame structure

• Ethernet: spanning multiple segments
  – Repeaters and hubs
  – Bridges and switches
  – Cut-through switching
  – Self-learning (plug-and-play)
  – Spanning trees
  – Virtual LANs (VLANs)

• The spectrum of interconnections
  – Hubs vs. switches vs. routers

---

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\textsuperscript{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
CSMA/CD Collision Detection

Limitations on Ethernet 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 $t+2d$
Limitations on Ethernet 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 on Ethernet. For 10 Mbps:
  – Maximum length of the wire: 2,500 meters
  – Minimum length of the packet: 512 bits (64 bytes)
    • 512 bits = 51.2 \( \mu \)sec (at 10 Mbit/sec)
    • For light in vacuum, 51.2 \( \mu \)sec \( \approx \) 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 clock rates

• 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 adaptor’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 adaptor 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)

---

Unreliable, Connectionless Service

• **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
Benefits of Ethernet

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

• Evolved from shared media to switches
  – Changes everything except the frame format
  – A good general lesson for evolving the Internet:
    • The right interface (service model) can often accommodate unanticipated changes
  – In fact, Ethernet framing used for wildly different technologies, e.g., 802.11 wireless

Shuttling 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

- Limitations on maximum nodes and distances
  - Does not circumvent limitations of shared media
  - E.g., still cannot go beyond 2500 meters on Ethernet

Link Layer: Bridges

- Connects 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

- Each segment is its own collision domain
Link Layer: Switches

- Typically connects individual computers
  - Essentially the same as a bridge
  - ... though connecting hosts, not LANs
    - In a point-to-point fashion
- Like bridges, support concurrent communication
  - Host A can talk to C, while B talks to D

Dedicated Access and Full Duplex

- **Dedicated** access
  - Host has direct connection to the switch
  - ... rather than a shared LAN connection
- **Full duplex**
  - Each connection can send in both directions
    - At the same time (otherwise, "half duplex")
  - Host sending to switch, and host receiving from switch
- **Completely avoids collisions**
  - Each connection is a bidirectional point-to-point link
  - No need for carrier sense, collision detection, and so on
Bridges/Switches: Traffic Isolation

- Breaks subnet into LAN segments
- Filters packets
  - Frame only forwarded to the necessary segments
  - Segments become separate collision domains

5 Minute Break

Questions Before We Proceed?
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

- 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
  - Solution: cut-through switching
- Need to learn where to forward frames
  - Bridge/switch needs to construct a forwarding table
  - Ideally, without intervention from network administrators
  - Solution: self-learning
- Higher cost
  - More complicated devices that cost more money
Cut-Through Switching

• Buffering a frame takes time
  – If \( L \) is length of the frame, \( R \) is the transmission rate …
  – … then receiving the frame takes \( L/R \) time units
  – When will this be significant?

• Cut-Through: Begin sending as soon as possible
  – Inspect frame header & look-up destination
  – If outgoing link idle, start forwarding
  – Can transmit head of packet while still receiving tail

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

Switch just learned how to reach A.

A B C D

Self Learning: Handling Misses

• When frame arrives with unfamiliar destination
  – Forward the frame out all of the interfaces
  – … except for the one where the frame arrived
  – Hopefully, this case won’t happen very often

When in doubt, shout!

A B C D
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

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
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
  – 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

See Section 3.2.2 in the textbook for details and another example.
Virtual LANs

• Once we have switches, we can enforce policies regarding isolation
  – Group users based on organizational structure rather than physical layout of building

• Implemented as “virtual LANs” or VLANs
  – Associate a “color” (tag) with either each switch interface
    • Assuming entire segment it serves on same VLAN
  – … or with each MAC address
    • Also allows hosts to move from one physical location to another

• Security:
  – Prevents nodes from seeing traffic not meant for them
  – Can force traffic leaving the VLAN to transit control point
    • E.g., firewall or Intrusion Detection System (IDS)

Example: Two Virtual LANs

Red VLAN and Orange VLAN
Switches forward traffic as needed
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: physical-layer interconnects
  – Bridges & switches: link-layer interconnects

• Key ideas in switches
  – Cut-through switching
  – Self learning of the switch table
  – Spanning trees
  – Virtual LANs (VLANs)

• Next time: midterm review