CSCI-1680
Link Layer II
Nick DeMarinis

Based partly on lecture notes by Rodrigo Fonseca, David Mazières, Phil Levis, John Jannotti
Today: Link Layer (cont.)

• Framing
• Reliability
  – Sliding window
  – Error correction
• Medium Access Control
  – (Short) Case study: Ethernet
• Link Layer Switching
Medium Access Control

• Control access to shared physical medium
  – E.g., who can talk when?
  – If everyone talks at once, no one hears anything

• Two conflicting goals
  – Maximize utilization when one node sending
  – Approach 1/N allocation when N nodes sending
Different Approaches

• Partitioned Access
  – Time Division Multiple Access (TDMA)
  – Frequency Division Multiple Access (FDMA)
  – Code Division Multiple Access (CDMA)

• Random Access
  – ALOHA/ Slotted ALOHA
  – Carrier Sense Multiple Access / Collision Detection (CSMA/CD)
  – Carrier Sense Multiple Access / Collision Avoidance (CSMA/CA)
  – RTS/CTS (Request to Send/Clear to Send)
  – Token-based
Ethernet (IEEE 802.3)

- Dominant wired LAN technology
- Original version (1983): 10Mbps
- Now: 1Gbps (1000BASE-T), 10Gbps, ...
- CSMA/CD: Carrier Sense / Multiple Access / Collision Detection
- L1: Manchester encoding
Ethernet Addressing

Globally unique, 48-bit unicast address per adapter
- Example: 00:1c:43:00:3d:09 (Samsung adapter)
- First 24 bits: Registered to manufacturers

Other protocols have adopted this address format (eg. Wifi, Bluetooth, …)

- Nowadays, we call them “mac addresses” or “hardware addresses”
Ethernet’s evolution

Originally, a shared medium with all hosts

- Basic idea: all hosts can see all frames, read a frame if it matches your hardware address
- Implications?
Ethernet MAC: CSMA/CD

- Problem: shared medium, all hosts in the same “collision domain”
- Transmit algorithm
  - If line is idle, transmit immediately
  - Upper bound message size of 1500 bytes
  - If line is busy: wait until idle and transmit immediately
- Generally possible to detect collisions
When to transmit again?

• Delay and try again: exponential backoff
• \( n \)th time: \( k \times 51.2\mu s \), for \( k = U\{0..(2^{\min(n,10)}-1)\} \)
  – 1\textsuperscript{st} time: 0 or 51.2\( \mu \)s
  – 2\textsuperscript{nd} time: 0, 51.2, 102.4, or 153.6\( \mu \)s
• Give up after several times (usually 16)

• Exponential backoff is a useful, general technique
Capture Effect

- Exponential backoff leads to self-adaptive use of channel
- A and B are trying to transmit, and collide
- Both will back off either 0 or 51.2μs
- Say A wins.
- Next time, collide again.
  - A will wait between 0 or 1 slots
  - B will wait between 0, 1, 2, or 3 slots
- ...
Ethernet Recap

• Service provided: send frames among stations with specific addresses
• Addresses are just names, no topology information
  – Special broadcast and multicast addresses
• All nodes in the same “broadcast domain”
  – Is this what we want?
Bridges and Extended LANs

- Single Ethernet collision domain has limitations
  - Limits performance, distance, ...
- Next step: separate collision domains with *bridges*
  - Operates on Ethernet addresses
  - Forwards packets from one collision domain to others
- Modern ethernet uses switches: all hosts directly connected to a bridge
Destinations for packets

- Unicast: forward with filtering
- Broadcast: always forward
- Multicast: always forward or learn groups

- Can try to limit how we direct packets to a destination
Learning Bridges/Switches

• Idea: don’t forward a packet where it isn’t needed
  – If you know recipient is not on that port
• Learn hosts’ locations based on source addresses
  – Build a table as you receive packets
  – Table is a cache: if full, evict old entries. Why is this fine?
• Table says when not to forward a packet
  – Doesn’t need to be complete for correctness
Attack on a Learning Switch

• Eve: wants to sniff all packets sent to Bob
• Same segment: easy (shared medium)
• Different segment on a learning bridge: hard
  – Once bridge learns Bob’s port, stop broadcasting
• How can Eve force the bridge to keep broadcasting?
  – Flood the network with frames with spoofed src addr!
Switching

Switches must be able to, given a packet, determine the outgoing port

• 3 ways to do this:
  – Virtual Circuit Switching
  – Datagram Switching
  – Source Routing
Virtual Circuit Switching

- Explicit set-up and tear down phases
  - Establishes Virtual Circuit Identifier on each link
  - Each switch stores VC table
- Subsequent packets follow same path
  - Switches map [in-port, in-VCI] : [out-port, out-VCI]
- Also called *connection-oriented* model
Virtual Circuit Model

- Requires one RTT before sending first packet
- Connection request contains full destination address, subsequent packets only small VCI
- Setup phase allows reservation of resources, such as bandwidth or buffer-space
  - Any problems here?
- If a link or switch fails, must re-establish whole circuit
- Example: ATM, MPLS
Datagram Switching

- Each packet carries destination address
- Switches maintain address-based tables
  - Maps [destination address]:[out-port]
- Also called connectionless model

<table>
<thead>
<tr>
<th>Addr</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>3</td>
</tr>
<tr>
<td>D</td>
<td>3</td>
</tr>
<tr>
<td>E</td>
<td>2</td>
</tr>
<tr>
<td>F</td>
<td>1</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
</tr>
<tr>
<td>H</td>
<td>0</td>
</tr>
</tbody>
</table>
Datagram Switching

• No delay for connection setup
• Source can’t know if network can deliver a packet
• Possible to route around failures
• Higher overhead per-packet
• Potentially larger tables at switches
Source Routing

• Packets carry entire route: ports
• Switches need no tables!
  – But end hosts must obtain the path information
• Variable packet header
Generic Switch Architecture

• Goal: deliver packets from input to output ports
• Three potential performance concerns:
  – Throughput in bytes/second
  – Throughput in packets/second
  – Latency
Shared Memory Switch

- 1\textsuperscript{st} Generation – like a regular PC
  - NIC DMAs packet to memory over I/O bus
  - CPU examines header, sends to destination NIC
  - I/O bus is serious bottleneck
  - For small packets, CPU may be limited too
  - Typically < 0.5 Gbps
Shared Bus Switch

• 2\textsuperscript{st} Generation
  – NIC has own processor, cache of forwarding table
  – Shared bus, doesn’t have to go to main memory
  – Typically limited to bus bandwidth
    • (Cisco 5600 has a 32Gbps bus)
Point to Point Switch

• 3rd Generation: overcomes single-bus bottleneck
• Example: Cross-bar switch
  – Any input-output permutation
  – Multiple inputs to same output requires trickery
  – Cisco 12000 series: 60Gbps
Cut through vs. Store and Forward

- Two approaches to forwarding a packet
  - Receive a full packet, then send to output port
  - Start retransmitting as soon as you know output port, before full packet
- Cut-through routing can greatly decrease latency
- Disadvantage
  - Can waste transmission (classic optimistic approach)
    - CRC may be bad
    - If Ethernet collision, may have to send runt packet on output link
Buffering

• Buffering of packets can happen at input ports, fabric, and/or output ports

• Queuing discipline is very important

• Consider FIFO + input port buffering
  – Only one packet per output port at any time
  – If multiple packets arrive for port 2, they may block packets to other ports that are free
  – *Head-of-line blocking*: can limit throughput to ~ 58% under some reasonable conditions*

* For independent, uniform traffic, with same-size frames
Head-of-Line Blocking

- Solution: Virtual Output Queueing
  - Each input port has $n$ FIFO queues, one for each output
  - Switch using matching in a bipartite graph
  - Shown to achieve 100% throughput*

*MCKEOWN et al.: ACHIEVING 100% THROUGHPUT IN AN INPUT-QUEUED SWITCH, 1999
Current Developments

- Switches are becoming programmable
  - Match-action paradigm
  - Custom protocols, encapsulation, metering, monitoring

- Current speeds reach 12.8Tbps (32x400Gbps or 256x50Gbps) on a single programmable switching chip
VLANs

Consider: Company network, A and B departments
- Broadcast traffic does not scale
- May not want traffic between the two departments
- Topology has to mirror physical locations
- What if employees move between offices?
VLANs

• Solution: Virtual LANs
  – Assign switch ports to a VLAN ID (color)
  – Isolate traffic: only same color
  – Trunk links may belong to multiple VLANs
  – Encapsulate packets: add 12-bit VLAN ID

• Easy to change, no need to rewire
We did not cover these...
Coming Up

• Connecting multiple networks: IP and the Network Layer