Hubs, Switches, and Routers, Oh My!

Mike Freedman
COS 461: Computer Networks

http://www.cs.princeton.edu/courses/archive/spr20/cos461/

Today: Hubs, Switches, and Routers, Oh My!

Terminology

- **Hubs and Repeaters**
  - Connect machines on same “layer 2” LAN
  - Broadcast: All frames are sent out all physical ports

- **Switches and Bridges**
  - Connect machines on same “layer 2” LAN
  - Only send frames to selected physical port based on destination MAC address

- **Routers**
  - Connect between LANs at “layer 3”, e.g., wide area
  - Only send packet to selected physical port based on destination IP address
“Layer 2”
Hubs and Switches

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
- Repeaters join LANs together
  - Analog electronic device
  - Continuously monitors electrical signals
  - Transmits an amplified copy

Physical Layer: Hubs
- Joins multiple input lines electrically
  - Designed to hold multiple line cards
  - Do not necessarily amplify the signal
- Very similar to repeaters
  - Also operates at the physical layer

Limitations of Repeaters and Hubs
- One large shared link
  - Each bit is sent everywhere
  - So, aggregate throughput is limited
- Cannot support multiple LAN technologies
  - Does not buffer or interpret frames
  - Can’t interconnect between different rates/formats
- Limitations on maximum nodes and distances
  - Shared medium imposes length limits
  - E.g., 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 segment
• Each segment can carry its own traffic

Link Layer: Switches

• Typically connects individual computers
  – A switch is essentially the same as a bridge
  – ... though typically used to connect hosts
• Supports concurrent communication
  – Host A can talk to C, while B talks to D

Bridges/Switches: Traffic Isolation

• Switch filters packets
  – Frame only forwarded to the necessary segments
  – Segments can support separate transmissions

Self Learning: Building the Table

• When a frame arrives
  – Inspect the source MAC address
  – Associate the address with the incoming interface
  – Store the mapping in the switch table
  – Use a timer to eventually forget the mapping

Switch learns how to reach A
**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!

**Switches vs. Hubs**

- Compared to hubs, Ethernet switches support
  - (Y) Larger geographic span
  - (M) Similar span
  - (C) Smaller span

- Compared to hubs, switches provides
  - (Y) Higher load on links
  - (M) Less privacy
  - (C) Heterogenous communication technologies

**Basic Router Architecture**

- Each switch/router has a forwarding table
  - Maps destination address to outgoing interface

- Basic operation
  1. Receive packet
  2. Look at header to determine destination address
  3. Look in forwarding table to determine output interface
  4. Modify packet header (e.g., decr TTL, update chksum)
  5. Send packet to output interface
Basic operation:
1. Receive packet
2. Look at header to determine destination address
3. Look in forwarding table to determine output interface
4. Modify packet header (e.g., decrement TTL, update checksum)
5. Send packet to output interface

Lookup algorithm depends on protocol:

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Mechanism</th>
<th>Techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet (48 bits)</td>
<td>Exact Match</td>
<td>- Direct lookup</td>
</tr>
<tr>
<td>MPLS</td>
<td></td>
<td>- Associative lookup</td>
</tr>
<tr>
<td>ATM</td>
<td></td>
<td>- Hashing</td>
</tr>
<tr>
<td>IPv4 (32 bits)</td>
<td>Longest-Prefix Match</td>
<td>- Radix trie</td>
</tr>
<tr>
<td>IPv6 (128 bits)</td>
<td></td>
<td>- Compressed trie</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- TCAM</td>
</tr>
</tbody>
</table>
Longest Prefix Match

- Each packet has destination IP address
- Router looks up table entry that matches address

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>68.208.0.0/12</td>
<td>1</td>
</tr>
<tr>
<td><strong>68.211.0.0/17</strong></td>
<td>1</td>
</tr>
<tr>
<td>68.211.128.0/19</td>
<td>2</td>
</tr>
<tr>
<td>68.211.160.0/19</td>
<td>2</td>
</tr>
<tr>
<td>68.211.192.0/18</td>
<td>1</td>
</tr>
</tbody>
</table>

68.211.6.120

Longest Prefix Match

- Each packet has destination IP address
- Router looks up table entry that matches address

- Benefit of CIDR allocation and LPR
  - Efficiency: Prefixes can be allocated at much finer granularity
  - Hierarchical aggregation: Upstream ISP can aggregate 2 contiguous prefixes from downstream ISPs to shorter prefix

How? Address lookup using trie

- Prefixes “spelled out” by following path from root
- To find the best prefix spell out address in trie

<table>
<thead>
<tr>
<th>Prefixes</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
</tr>
<tr>
<td>B</td>
</tr>
<tr>
<td>C</td>
</tr>
<tr>
<td>D</td>
</tr>
</tbody>
</table>

How? Address lookup using trie

- Prefixes “spelled out” by following path from root
- To find the best prefix spell out address in trie

- 1 per /lvl = max 32 lookups!
- Too slow:
  - E.g., "Optical Carrier 48" line (2.5 Gbps) requires 160ns lookup … or 4 memory accesses
K-ary trie (k=2)

<table>
<thead>
<tr>
<th>Prefixes</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>111*</td>
<td>10*</td>
<td>1010*</td>
<td>10101</td>
</tr>
</tbody>
</table>

Alternatives to trie for LPM

- Content-Address Memory (CAM)
  - Input: tag (address)
  - Output: value (port)
  - Exact match, but $O(1)$ in hardware

- Ternary CAM
  - 0, 1, *
  - “value” memory cell and “mask” (care / don’t care) cell

- LPM via TCAM
  - In parallel, search all prefixes for all matches
  - Then choose longest match
    - Trick: choose first match, but already sorted by prefix length

Decision: Forwarding table per line card

1. Each line card has own forwarding table copy
2. Prevents central table bottleneck (vs. early routers had table across shared bus)
Decision: Crossbar switch

- Shared bus
  - Only one input can speak to one output at a time

- Crossbar switch / switched backplane
  - Input/output pairs that don’t compete can send in same timeslot

Crossbar switching

- Every input port has connection to every output port
- In each timeslot, each input connected to zero or more outputs

- Good parallelism
- Needs scheduling
Everything gets complicated...

- **Problem: Head-of-line blocking**
  - The packet in front of queue blocks packets behind it from being processed
  - Say first packet at input 1 wants to go to output 5; second packet at 1 that wants 4 is still blocked

Everything gets complicated...

- **Solution: Virtual output queues**
  - One queue at input per output port
  - So avoids head-of-line blocking during crossbar scheduling

---

**Data, Control, & Management Planes**

<table>
<thead>
<tr>
<th>Time-scale</th>
<th>Data</th>
<th>Control</th>
<th>Management</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Packet (ns)</strong></td>
<td><strong>Event (10 ms to sec)</strong></td>
<td><strong>Human (min to hours)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Tasks</strong></td>
<td>Forwarding, buffering, filtering, scheduling</td>
<td>Routing, signaling</td>
<td>Analysis, configuration</td>
</tr>
<tr>
<td><strong>Location</strong></td>
<td>Line-card hardware</td>
<td>Router software</td>
<td>Humans or scripts</td>
</tr>
</tbody>
</table>

---

**Cisco 8000 Series Routers**

- Up to 648 400 GbE
- 260 Tbps backplane
Conclusions

• Physical devices sharing L2 & L3 networks have many common features
  – Forward table lookups
  – Queueing and backplane switching
  – Fast vs. slow paths
    • Switches and routers separate routing decisions (control plane) from forwarding actions (data plane)

• High speed necessitates innovation
  – Specialized hardware
  – Software algorithms