Protocol Stack and Layers

Summary

CSC 249
March 29, 2017

Interconnecting with hubs

Switch: traffic isolation
- Switch installation breaks subnet into LAN segments - separate collision regions, or domains
- Switches filter packets:
  - same-LAN-segment frames not usually forwarded onto other LAN segments
  - segments become separate collision domains
Switches: Self learning

- A switch has a **switch table**
- An entry in a switch table contains:
  - (MAC Address, Interface, Time Stamp)
  - stale entries in table dropped (TTL can be 60 min)
- switch *learns* which hosts can be reached through which interfaces
  - when frame received, switch "learns" location of sender: incoming LAN segment
  - records sender/location pair in switch table

---

Switch example

**Suppose C sends frame to B**

- Switch receives frame from from C
  - notes that B and C are in same segment
  - switch does nothing

---

**Suppose C sends frame to D**

- Switch receives frame from from C
  - notes in switch table that C is on interface 1
  - because D is not in table, switch forwards frame into interfaces 2 and 3
- frame received by D; (but nothing is added to the table)

---

**Suppose D replies back with frame to C.**

- Switch receives frame from from D
  - notes in switch table that D is on interface 2
  - because C is in table, switch forwards frame only to interface 1
- frame received by C
**Switch: allows multiple simultaneous transmissions**

- Hosts can have dedicated, direct connection to switch
- Switches buffer packets
- Ethernet protocol used on each incoming link, but no collisions: full duplex
  - each link is its own collision domain
- Switching: A-to-A’ and B-to-B’ simultaneously, without collisions
  - not possible with dumb hub

**Switch features can’t**

- **Transparent**
  - hosts are unaware of presence of switches, so do not address frames to switches
  - Switches are transparent to IP
- **Self-learning** to build address table
  - plug-and-play = self-learning
  - switches do not need to be configured by a person

---

**Question (from Friday’s handout)**

- Replace the router between subnets 1 and 2 with a switch S1, and label the router between 2 and 3 as R 1.
  - Consider sending an IP datagram from Host E to Host F. Will Host E ask router R 1 to help forward the datagram? Why? In the Ethernet frame containing the IP datagram, what are the source and destination IP and MAC addresses?
  - Suppose E would like to send an IP datagram to B, and assume that E’s ARP cache does not contain B’s MAC address. Will B perform an ARP query to find B’s MAC address? Why? In the Ethernet frame (containing the IP datagram destined to B) that is delivered to router R 1, what are the source and destination IP and MAC addresses?
  - Suppose Host A would like to send an IP datagram to Host B, and neither A’s ARP cache contains B’s MAC address nor does B’s ARP cache contain A’s MAC address. Further suppose that the switch S1’s forwarding table contains entries for Host B and router R 1 only. Thus, A will broadcast an ARP request message.

---

**Figure for Question**
Summary comparison

- Hub
- Switch
- Router

Switches vs. Routers

- both store-and-forward devices
  - routers: network layer devices
  - switches are link layer devices
- routers maintain routing tables, implement routing algorithms
- switches maintain switch tables, implement filtering, learning algorithms

Hub-Switch-Router Comparison

- IP vs. MAC address used
- Layer implemented through - affect speed of processing
- Collisions prevented - Hubs allows collisions, switches create different collision domains
- Routing - global algorithm vs. Switch self-learning
  - Requires or not set-up by system administrator → Switch ARP tables are self-learning creations
  - Hierarchical vs. flat addressing
  - Size of table created
  - Geographic range possible with router v. switch given size of tables required
- Able to interconnect heterogeneous links/technologies

Chapter 6: Summary

- Data link layer services:
  - error detection and correction
  - sharing a broadcast channel: multiple access
  - link layer addressing
  - Plug-and-play for ARP and switch table learning
- Link layer technologies
  - Ethernet
  - switched LANS (switches v. hubs)
**Encapsulation**

**Four sources of packet delay**

- **Transmission delay**:
  - $d_{\text{trans}}$: transmission delay
  - $L$: packet length (bits)
  - $R$: link bandwidth (bps)
  - $d_{\text{trans}} = \frac{L}{R}$

- **Propagation delay**:
  - $d_{\text{prop}}$: propagation delay
  - $d$: length of physical link
  - $s$: propagation speed in medium (~2x10^8 m/sec)
  - $d_{\text{prop}} = \frac{d}{s}$

- **Nodal processing**
- **Queueing delay**

**What is a protocol?**

a human protocol and a computer network protocol:

**HTTP overview**

HTTP Server

PC running Explorer

HTTP request

HTTP response

Server running Apache Web server

Mac running Navigator

HTTP request

HTTP response

Get http://www.smith.edu

<file>
**Sockets**

- process sends/receives messages to/from its socket
  - sending process sends message through the socket
  - sending process relies on transport infrastructure, UDP or TCP as programmed into the operating system, to deliver the message to the socket at the receiving host & process

**DNS: a distributed, hierarchical database**

- a host, or client, wants the IP address for www.google.com
  1. Client (local server) queries root server to find the .com DNS server
  2. Client queries .com DNS server to get google.com DNS server
  3. Client queries google.com DNS server to get the IP address for www.google.com

**TCP: slow start & congestion avoidance**

**Implementation:**
- variable CWND
- variable ssthresh
- Loss event
  - Triple duplicate ACK
  - timeout

**DHCP client-server scenario**

yiaddr = 'your internet address'
broadcast address, 255.255.255.255 → sent to every host in the subnet
Router architecture overview

- **Routing processor**: Computes forwarding tables and pushes them to input ports.
- **High-seed switching fabric**: Facilitates data forwarding between input and output ports.
- **Routing, management control plane (software)**: Handles routing and management functionalities.
- **Forwarding data plane (hardware)**: Performs packet switching and forwarding.

Routing: Dijkstra’s algorithm

<table>
<thead>
<tr>
<th>Step</th>
<th>N'</th>
<th>D(v),p(v)</th>
<th>D(w),p(w)</th>
<th>D(x),p(x)</th>
<th>D(y),p(y)</th>
<th>D(z),p(z)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>u</td>
<td>2, u</td>
<td>5, u</td>
<td>1, u</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>1</td>
<td>ux</td>
<td>2, u</td>
<td>4, x</td>
<td>2, x</td>
<td>∞</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>uxy</td>
<td>2, u</td>
<td>3, y</td>
<td>4, y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>uxyv</td>
<td>3, y</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>uxyvw</td>
<td>4, y</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>uxyvwz</td>
<td>4, y</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Network Layer

CSMA/CD (collision detection)

- **Spatial layout of nodes**: Illustrates the physical arrangement of nodes in the network.
- **Collision detect/abort time**: Represents the time delay before a collision is detected and packets are aborted.

Layer | Happens to Packet | Action caused by packet | Action/event happens on own |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>App.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trans.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Netw.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Link</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>