ICS 451: Today's plan

- Ethernet hubs and switches
  - comparison to routers
- Spanning Tree Protocol
- Virtual LANs

Ethernet Hubs

- physical-layer (bit-level) forwarding
  - with signal regeneration
  - generates or propagates jamming signal
- topology cannot have any loops
- entire network is one collision domain
  - only one packet live at a time
  - limits size of the network

Ethernet Switches

- data link layer (frame-level) forwarding
  - with error checking
  - behaves like an Ethernet host
- topology cannot have any loops
- forwarding is usually in hardware
  - hardware must store and forward packet
    - expensive switches do wormhole forwarding
  - if not forwarding in hardware, called a *bridge*

Learning Switches/Bridges

- each Ethernet switch keeps a forwarding table
  - mapping MAC addresses to interfaces
  - mappings forgotten after ~30s unless refreshed
- when a packet is received on interface *x*:
  - its source address and *x* are added to the table
  - if the packet's destination MAC address is in the table, the packet is forwarded to that interface
  - otherwise, the packet is broadcast
- this is a *learning switch* or *learning bridge*

Switches vs. Hubs

Switches: data link

- Each interface is a separate collision domain
- only broadcasts if needed
- packet-time latency
  - unless wormhole
- no topology loops

Hubs: physical layer

- all interfaces are same collision domain
- always broadcasts
  - useful for network monitoring
- bit-time latency
- no topology loops

Switches vs Routers

- Switches interconnect networks
- networks have same IP number
- if not in table, broadcast
- works automatically
- data link layer

- Routers interconnect networks
- networks have different IP numbers
- if not in table, drop
- usually needs configuration
- network layer

Spanning Tree Protocol/STP

- Suppose we wanted to have redundant links in a switched ethernet network
  - configure switches to only forward over selected interfaces that don't form any loops
- distributed computation can:
  - connect every *Ethernet segment*
  - without any loops
- this computation gives a spanning tree

STP Overview

- STP switches generate and receive BPDUs
  - Bridge Protocol Data Unit frames
  - addressed to multicast `01:80:C2:00:00:00`
- switches elect as root of the spanning tree the switch with the lowest ID (MAC address)
- each switch finds the shortest path to the root from each of its segments
- corresponding interfaces are kept open
  - the remaining interfaces are blocked

BPDUs

- A BPDU contains:
  - identifier R of the root switch
    - useful in electing the root
    - useful in building the tree
  - identifier T of the sender of this BPDU
  - distance/cost c between T and R
    - inversely proportional to link bandwidth
  - interface/port p over which this BPDU was sent
- <R, c, T, p>

comparing BPDUs

- when we compare BPDUs, better BPDU have
  - smaller identifier R of the root switch
    - if R is the same,
  - smaller cost c to R
    - if c and R are the same,
  - smaller sender identifier T
    - if T, c, and R are the same,
  - smaller port number p
- so we can always find the better of two BPDUs

interface/port states

- a *root port* connects this switch to the root
- a *designated port* is in the spanning tree
- a *blocked port* is not used
  - except for receiving BPDUs
- frames are forwarded over root or designated ports
- BPDUs are only sent over designated ports

STP algorithm

- when receiving <R, c, T, p> on interface q:
  - add the cost of q to the cost c, then
  - save the BPDU as the latest for port q
  - if my ID < R, I am the root
    - set all my interfaces to *designated ports*
  - otherwise, I find the best saved BPDU *bv*
    - the port of *bv* is my *root port*
    - I compute my outgoing BPDU <R', c', T', _>
    - for each port, if my BPDU is better than the latest BPDU received from that port, I set that port to a *designated port*
    - otherwise, I *block* that port

STP details

- no traffic is forwarded during initial STP computation
- on link or switch failure, BPDUs eventually expire, and STP computation is restarted

- STP is almost plug-and-play
  - not enabled by default
    - perhaps not available on low-end switches
  - incurs additional traffic and delays

Virtual LANs

- a switch can be configured to group some of its interfaces into a Virtual LAN (VLAN)
- broadcasting (and STP) is only over the interfaces in the same VLAN
  - this can be combined with routing among the different VLANs
- VLANs over multiple switches require VLAN identifcation of received frames
  - additional header carries this ID (802.1q)
  - header also includes frame priority