Switching and Forwarding

Outline

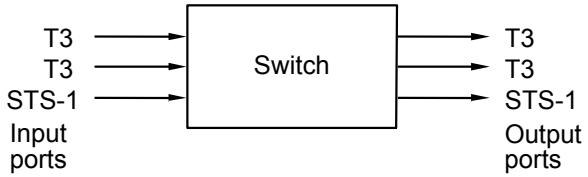
- Store-and-Forward Switches
- Bridges and Extended LANs

Housekeeping

- Homework handback
- Questions about Hwk 2

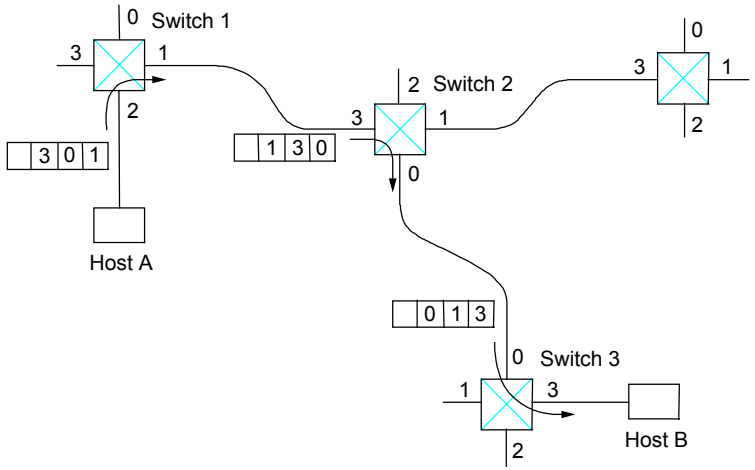
Scalable Networks

- Switch
  - forwards packets from input port to output port
  - port selected based on address in packet header



- Advantages
  - cover large geographic area (tolerate latency)
  - support large numbers of hosts (scalable bandwidth)

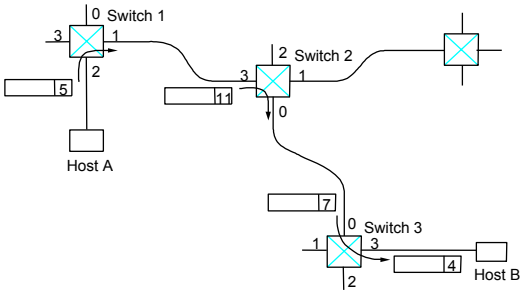
Source Routing



Virtual Circuit Switching

- Explicit connection setup (and tear-down) phase
- Subsequence packets follow same circuit
- Sometimes called *connection-oriented* model

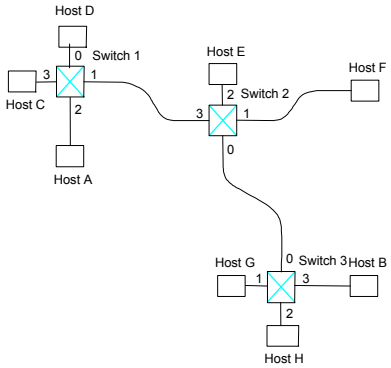
- Analogy: phone call
- Each switch maintains a VC table



Datagram Switching

- No connection setup phase
- Each packet forwarded independently
- Sometimes called *connectionless* model

- Analogy: postal system
- Each switch maintains a forwarding (routing) table



Virtual Circuit Model

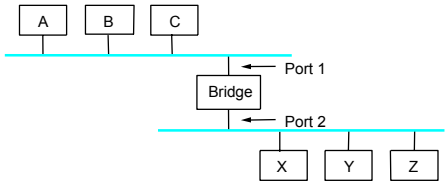
- Typically wait full RTT for connection setup before sending first data packet.
- While the connection request contains the full address for destination, each data packet contains only a small identifier, making the per-packet header overhead small.
- If a switch or a link in a connection fails, the connection is broken and a new one needs to be established.
- Connection setup provides an opportunity to reserve resources.

Datagram Model

- There is no round trip time delay waiting for connection setup; a host can send data as soon as it is ready.
- Source host has no way of knowing if the network is capable of delivering a packet or if the destination host is even up.
- Since packets are treated independently, it is possible to route around link and node failures.
- Since every packet must carry the full address of the destination, the overhead per packet is higher than for the connection-oriented model.

Bridges and Extended LANs

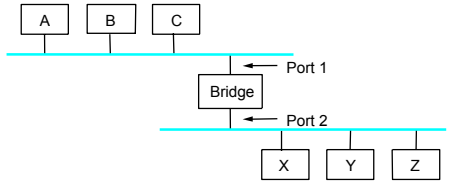
- LANs have physical limitations (e.g., 2500m)
- Connect two or more LANs with a *bridge*
  - accept and forward strategy
  - level 2 connection (does not add packet header)



- Ethernet Switch = Bridge on Steroids

Learning Bridges

- Do not forward when unnecessary
- Maintain forwarding table

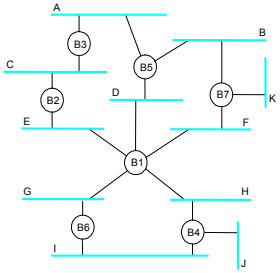


Host	Port
A	1
B	1
C	1
X	2
Y	2
Z	2

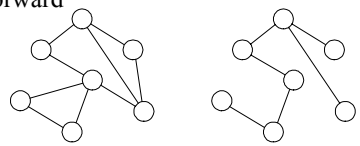
- Learn table entries based on source address
- Table is an optimization; need not be complete
- Always forward broadcast frames

Spanning Tree Algorithm

- Problem: loops

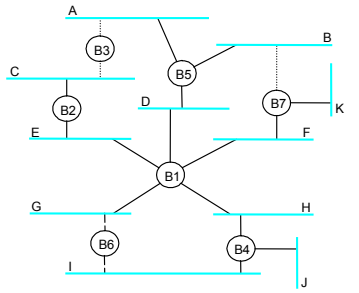


- Bridges run a distributed spanning tree algorithm
  - select which bridges actively forward
  - developed by Radia Perlman
  - now IEEE 802.1 specification



Algorithm Overview

- Each bridge has unique id (e.g., B1, B2, B3)
- Select bridge with smallest id as root
- Select bridge on each LAN closest to root as designated bridge (use id to break ties)
- Each bridge forwards frames over each LAN for which it is the designated bridge



Algorithm Details

- Bridges exchange configuration messages
  - id for bridge sending the message
  - id for what the sending bridge believes to be root bridge
  - distance (hops) from sending bridge to root bridge
- Each bridge records current best configuration message for each port
- Initially, each bridge believes it is the root

Algorithm Detail (cont)

- When learn not root, stop generating config messages
  - in steady state, only root generates configuration messages
- When learn not designated bridge, stop forwarding config messages
  - in steady state, only designated bridges forward config messages
- Root continues to periodically send config messages
- If any bridge does not receive config message after a period of time, it starts generating config messages claiming to be the root

Properties of the algorithm

- Resources consumed
  - Time
  - Total number of messages
- Correctness
  - End with a spanning tree: no cycles, every network node reachable

Broadcast and Multicast

- Forward all broadcast/multicast frames
  - current practice
- Learn when no group members downstream
- Accomplished by having each member of group G send a frame to bridge multicast address with G in source field

Limitations of Bridges

- Do not scale
  - spanning tree algorithm does not scale
  - broadcast does not scale
- Do not accommodate heterogeneity
- Caution: beware of transparency