Housekeeping

- Look at `select` system call
- See homework 2 solutions on the Lecture Notes web page for answer to the probability difficulties we (I) had last time
- No satisfactory answer to the double-bit error problem that was pointed out last time

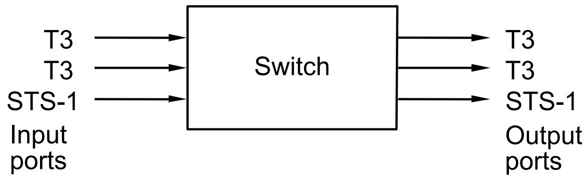
Switching and Forwarding - continued

Outline

Spanning Tree – completed
Datagram Switching Characteristics
Virtual Circuit Switching Characteristics
Introduction to ATM

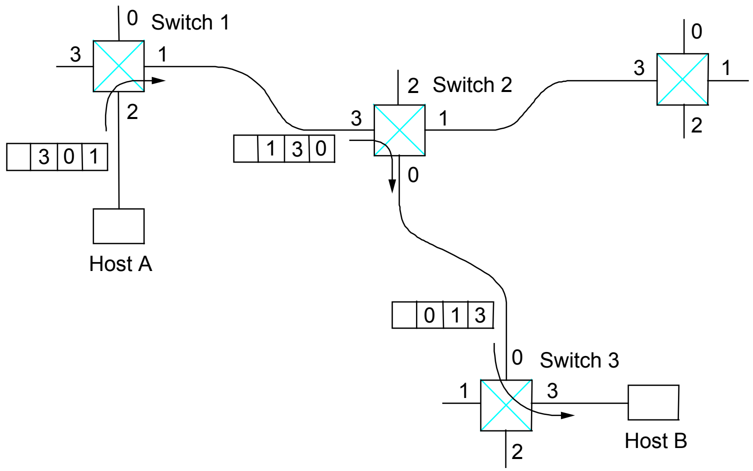
Scalable Networks

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



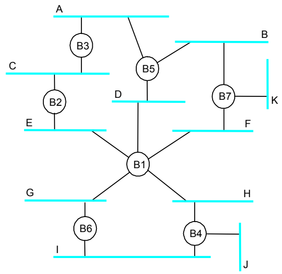
- Datagram Switching vs. Virtual Circuit Switching vs Source Routing

Source Routing

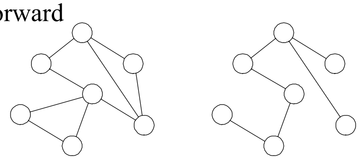


Spanning Tree Algorithm

- Problem: loops

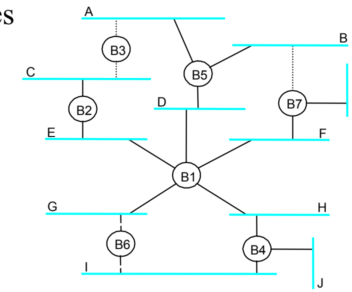


- Bridges run a distributed spanning tree algorithm
  - select which bridges actively forward
  - developed by Radia Perlman
  - now IEEE 802.1 specification



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

Spanning Tree Algorithm Properties

- How long to converge?
- How many total messages?

Broadcast and Multicast

- Current Practice
  - Forward all broadcast/multicast frames
- Better - learn when group members are downstream
  - Each member of group G sends a frame to multicast address with G in source field
  - Like normal “smart” bridging

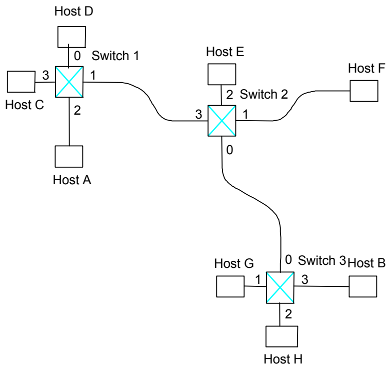
Limitations of Bridges

- Do not scale
  - spanning tree algorithm does not scale
  - broadcast does not scale
- Do not accommodate heterogeneity
- Caution: beware of non-transparency
  - Dropped and re-ordered packets
  - Increased delays

Datagram Switching

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

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



Datagram Model

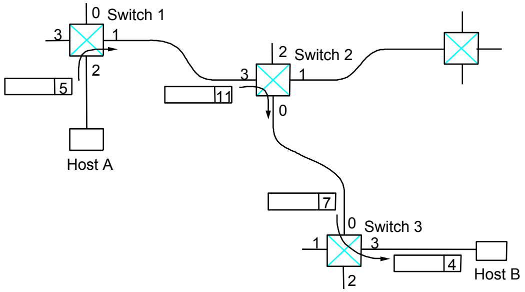
- There is no round trip time delay waiting for connection setup; a host can send data as soon as it is ready.
- Source host has no way of knowing if the network is capable of delivering a packet or if the destination host is even up.
- Since packets are treated independently
  - every packet must carry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

Virtual Circuit Model

- wait full RTT for connection setup before first data packet sent
- data packets contain only a small identifier, making the per-packet header overhead smaller than a datagram header
- switch or link failure, breaks the connection: a new one must be established.
- Connection setup provides an opportunity to reserve resources for a flow

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

Cell Switching (ATM)

- Connection-oriented packet-switched network
- Used in both WAN and LAN settings
- Signaling (connection setup) Protocol: Q.2931
- Specified by ATM forum
- Packets are called *cells*
  - 5-byte header + 48-byte payload: why?
- Commonly transmitted over SONET
  - other physical layers possible

Variable vs Fixed-Length Packets

- No Optimal Length
  - if small: high header-to-data overhead
  - if large: low utilization for small messages
- Fixed-Length Easier to Switch in Hardware
  - simpler
  - enables parallelism

Big vs Small Packets

- Small Improves Queue behavior
  - finer-grained pre-emption point for scheduling link
    - maximum packet = 4KB
    - link speed = 100Mbps
    - transmission time = 4096 x 8/100 = 327.68us
    - high priority packet may sit in the queue 327.68us
    - in contrast, 53 x 8/100 = 4.24us for ATM
  - near cut-through behavior
    - two 4KB packets arrive at same time
    - link idle for 327.68us while both arrive
    - at end of 327.68us, still have 8KB to transmit
    - in contrast, can transmit first cell after 4.24us
    - at end of 327.68us, just over 4KB left in queue

Big vs Small (cont)

- Small Improves Latency (for voice)
  - voice digitally encoded at 64KBps (8-bit samples at 8KHz)
  - need full cell's worth of samples before sending cell
  - example: 1000-byte cells implies 125ms per cell (too long)
  - smaller latency implies no need for echo cancellors
- ATM Compromise: 48 bytes = (32+64)/2