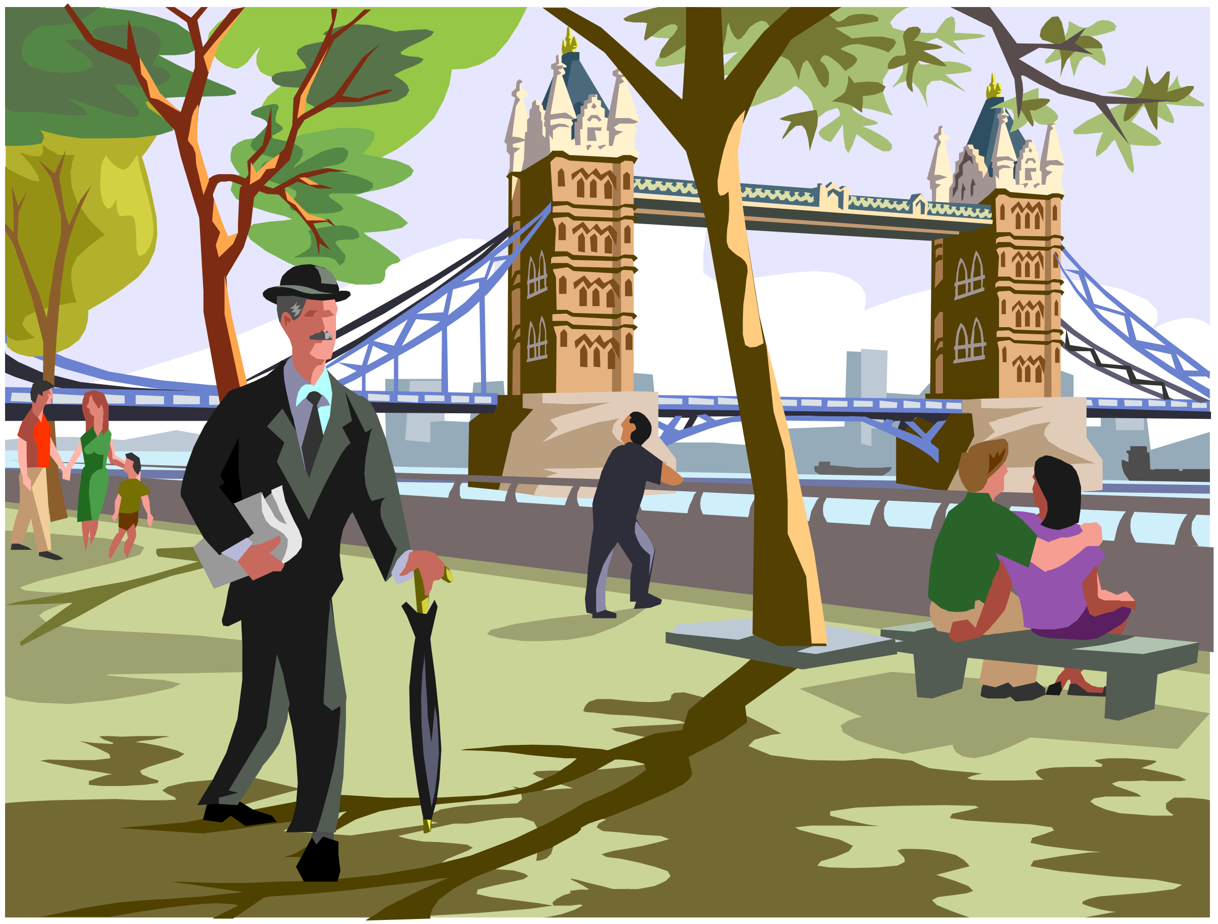
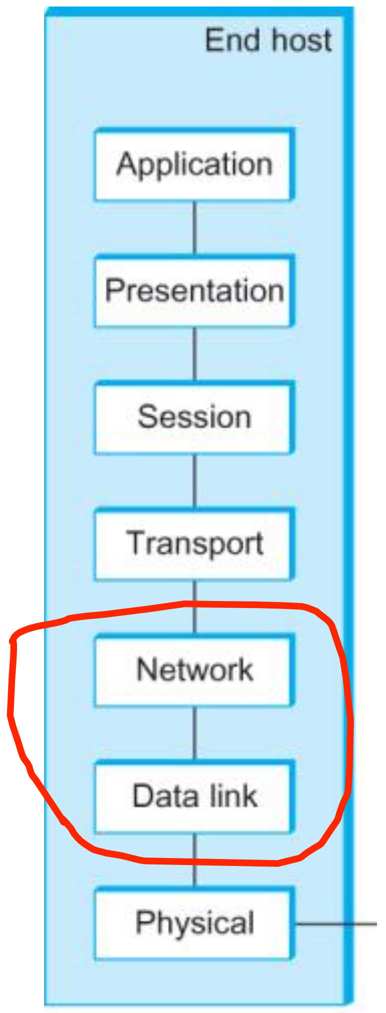
Switching and bridging

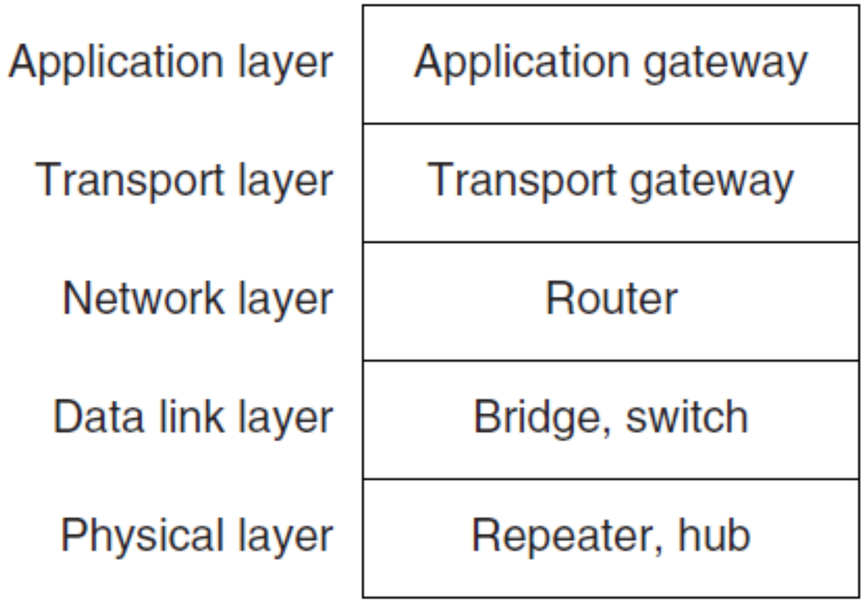


Overview

- Last chapter:
  - Creating networks from:
    - Point-to-point links
    - Shared medium (wireless)
- This chapter:
  - Software and hardware connecting networks together
  - How do packets find their way?



Hardware terminology



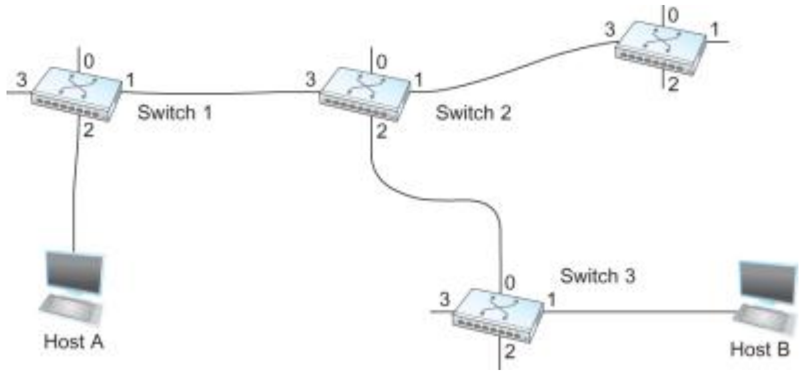
Application layer	Application gateway	Understands format and contents of data, e.g. translate Internet message to SMS message
Transport layer	Transport gateway	Connect different connection-oriented protocols, e.g. TCP/IP to SCTP
Network layer	Router	Operates on packets, uses IP addresses
Data link layer	Bridge, switch	Operates on frames, looks at MAC addresses
Physical layer	Repeater, hub	Analog devices, clean up signal, amplify, put out on another cable

Bridges and switches

- Bridge ≈ switch
  - Switch modern term, typically many ported device
- Connect multiple LANs together
  - Why?
    - Multiple departments built independent LANs
    - Geographic separation
      - Distances require small number of long fiber links
    - Handle load of many computers

Switching

- Switch
  - Multi-input, multi-output device
  - Receive data on a port
  - Decide which port(s) to send data out on a port
    - Based on MAC address in frame and knowledge of some sort (in the frame or in the switch)
  - Main function of the network layer



Approaches to switching

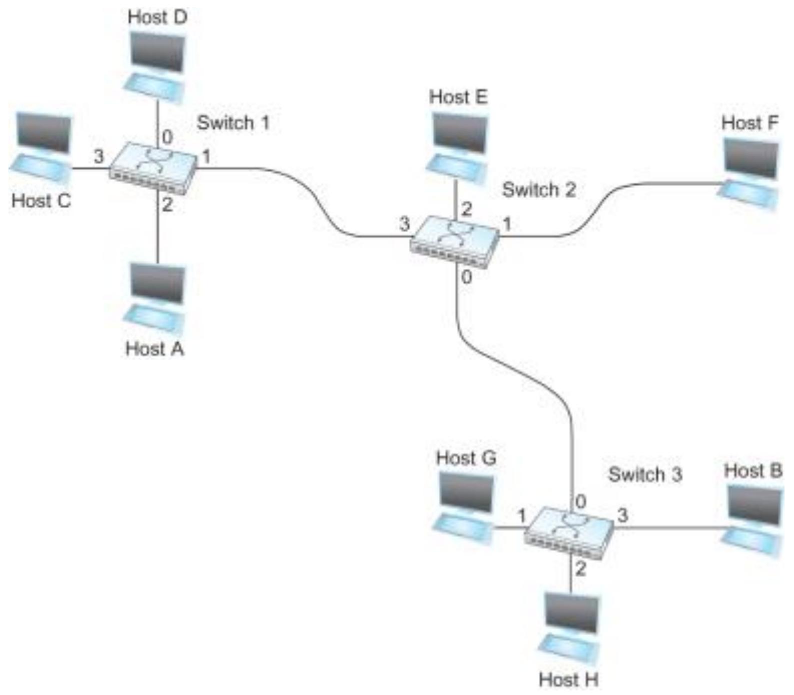
- How to switch based on frame's address info?
  1. Datagram / connectionless approach
  2. Virtual circuit / connection-oriented approach
  3. Source routing (less common)

Connectionless approach

- Datagram model
  - Each frame has enough info to get it to destination (its MAC address)
  - To forward, switch consults a forwarding table

Destination	Port
A	3
B	0
C	3
D	3
E	2
F	1
G	0
H	0

Forwarding Table for Switch 2



Connectionless approach

- Datagram model
  - Advantages:
    - Host can send frame anytime
      - No initial connection setup
    - Each frame forwarded independently of others
      - May go a different route each time
    - Robust to switch/link failure
  - Disadvantages:
    - Each frame competes with all other frame for buffer
      - If no buffer, frame dropped
    - Destination could be down (host has no idea)

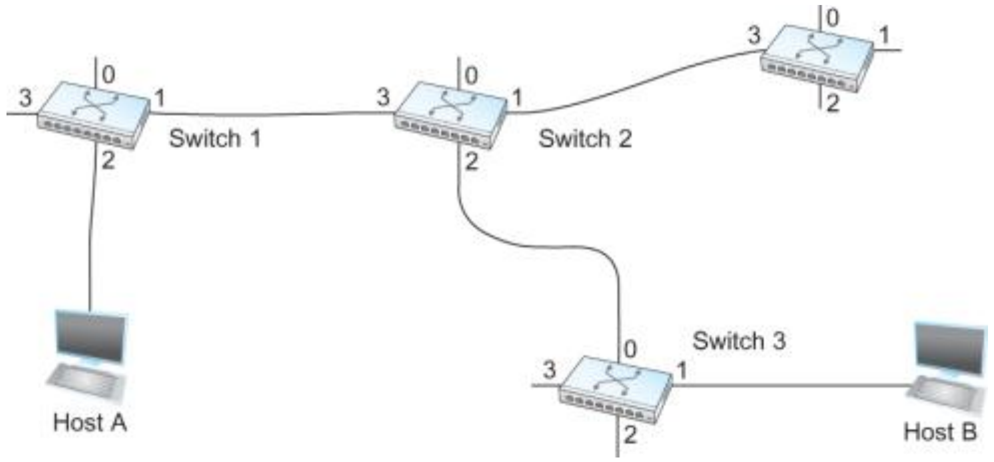
Connection-oriented approach

- Virtual circuit switching
  - Establish a virtual circuit (VC)
  - Requires initial setup from host to destination
  - e.g. ATM, Frame relay X.25

A wants to send data to B.

Establish a connection state (VC table) in each switch between A and B.

VC table entry has a virtual circuit identifier (VCI) that will be in frames belonging to this connection.

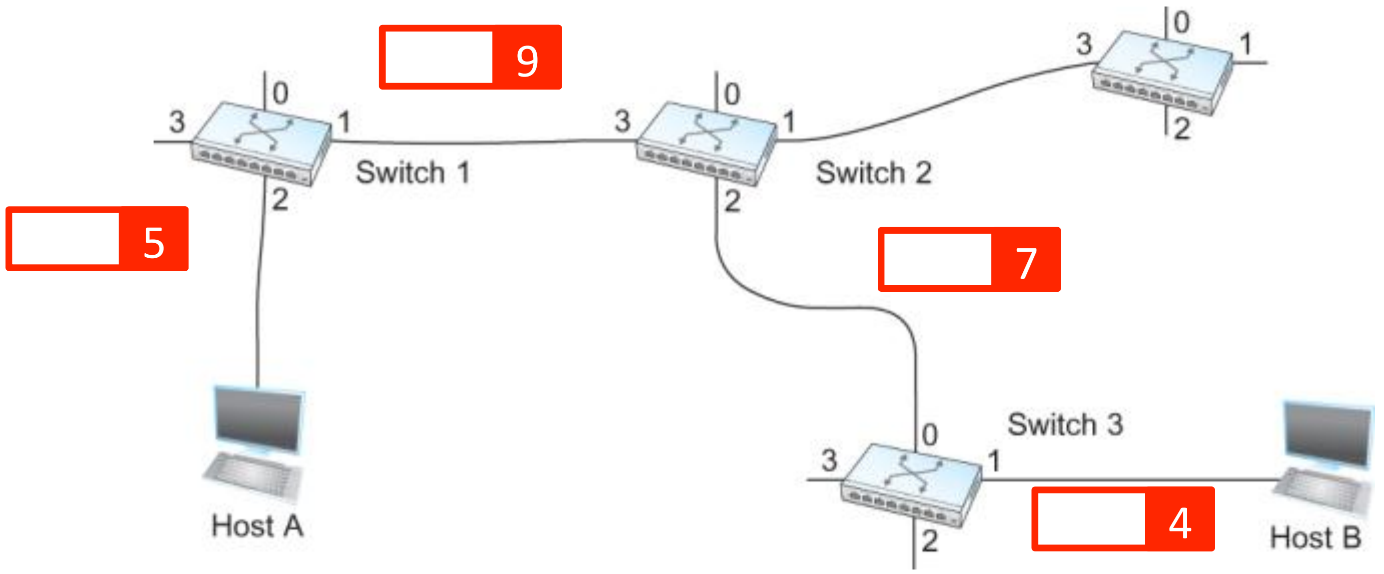


Establishing a connection

- Network admin based
  - Virtual circuit is permanent
  - Setup by admin and long-lived
- Host setup
  - Host sends messages into network (signalling)
  - Avoids the need for admin involvement

Admin setup example

- Admin finds path A → B, sets up tables in switch 1-3



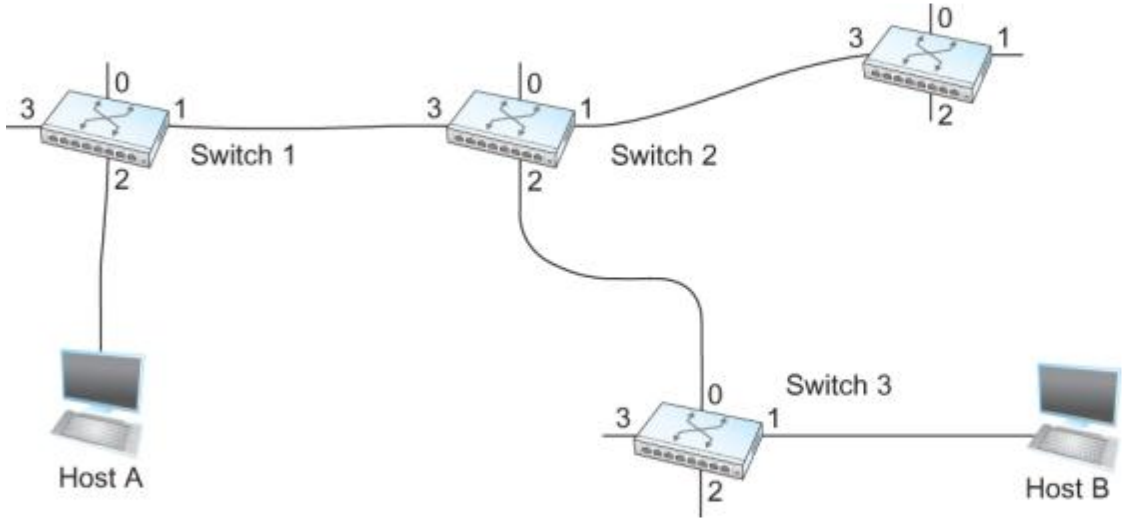
Switch	Incoming Interface	Incoming VC	Outgoing Interface	Outgoing VC
1	2	5	1	9
2	3	9	2	7
3	0	7	1	4

Signaling

- Signaling
  - Real networks too large for manual setup
  - "Permanent" VCs established by admin using signaling
  - Temporary VCs established by one of the hosts

Signaling

- Signaling process
  - Assume switches know network topology
  - A sends message to switch 1 with address B
  - Each switch on path to B adds VC table entry
  - Signaling back from B to A sets up reverse path
  - Connection terminated via teardown message

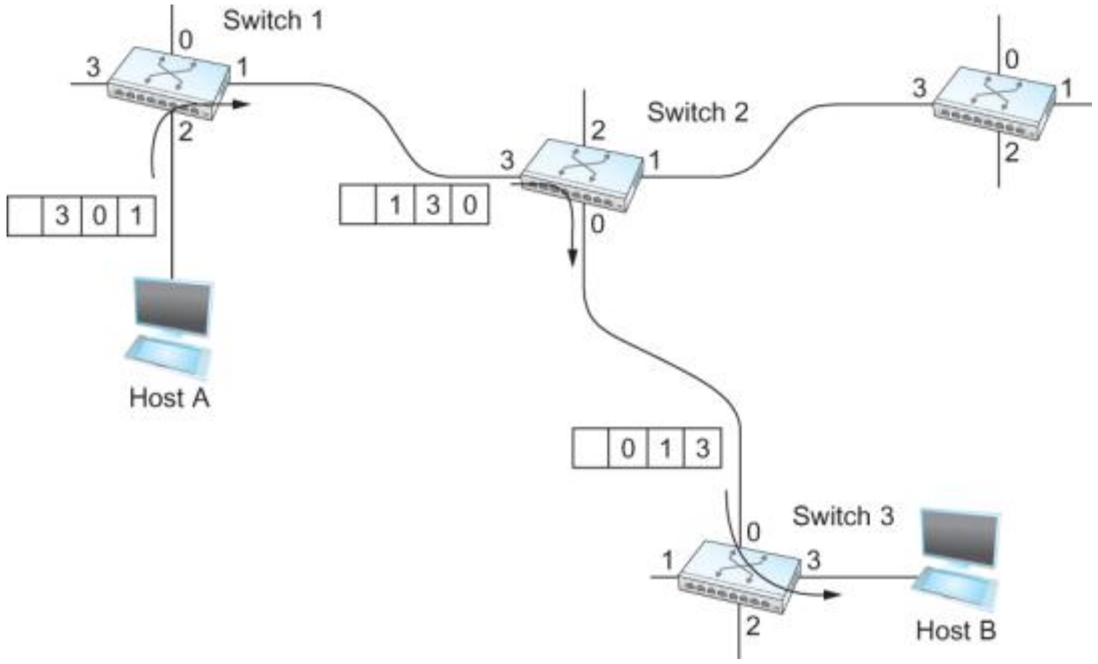


Connection-oriented

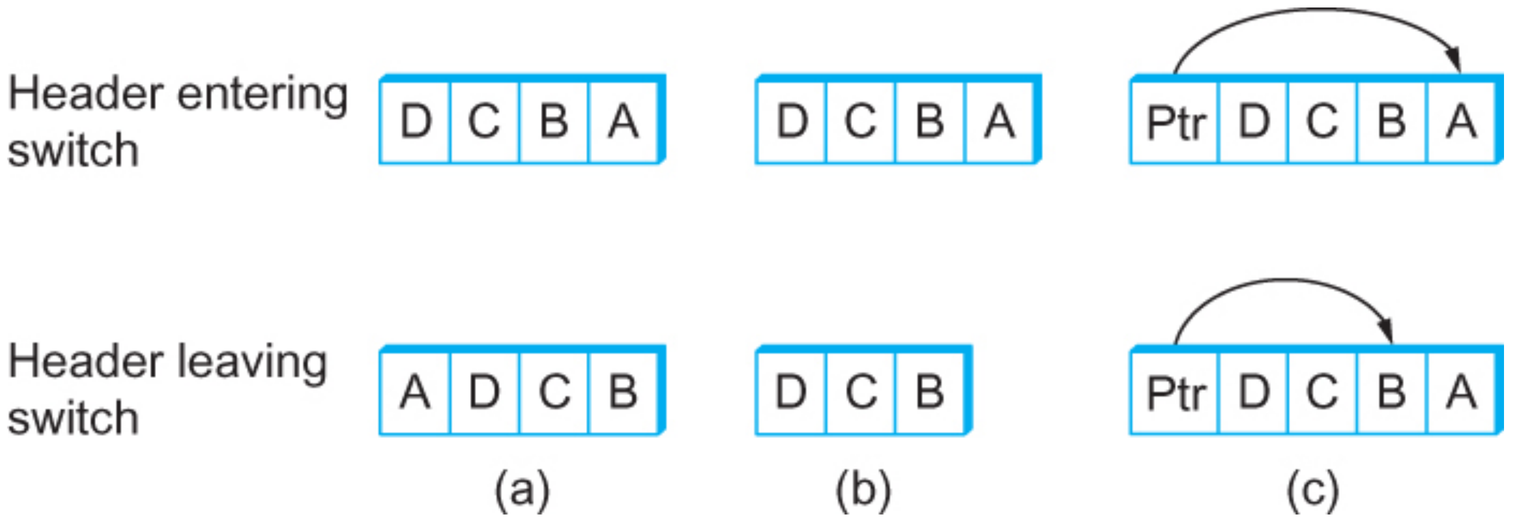
- Advantages:
  - Before data flows, A knows B is alive and well
  - Resources can be preallocated for the circuit
  - VC identifiers small compared to 48-bit MAC
  - Could provide different quality of service (QoS)
- Disadvantages:
  - One RTT to establish connection
  - Link or switch failure breaks connection

Source routing

- Source routing
  - All information required to route data to destination provided by source host



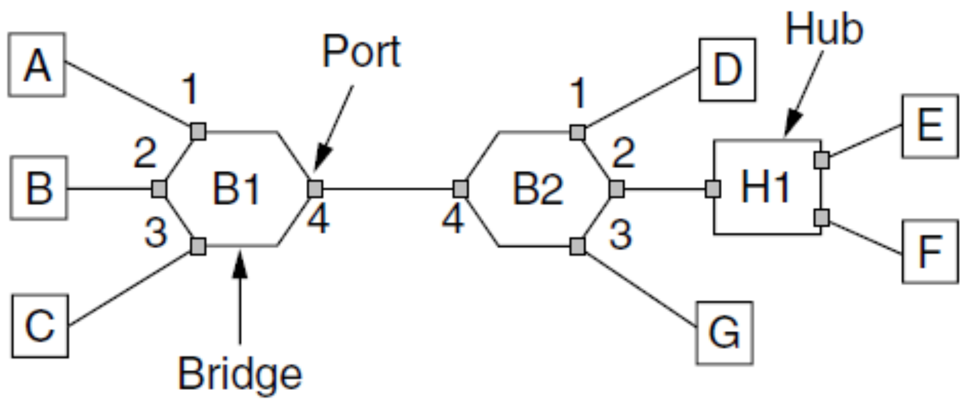
Source routing



a) Rotate header after every switch
b) Delete entry after every switch
c) Maintain a pointer into the list of routing instructions

Learning bridges

- How do bridges/switches learn what to do?
  - Backward learning
    - Who is on what port?
  - Spanning tree
    - What to do about cycles?



Backward learning

- Switch starts knowing nothing
  - Promiscuous mode, listens to all traffic on all ports
  - Hash table, destination → output port
    - Frame arrives on port, add entry based on who sent it
  - Topology can change as machine/bridges powered on and off
    - Table entries purged after a few minutes

Routing procedure

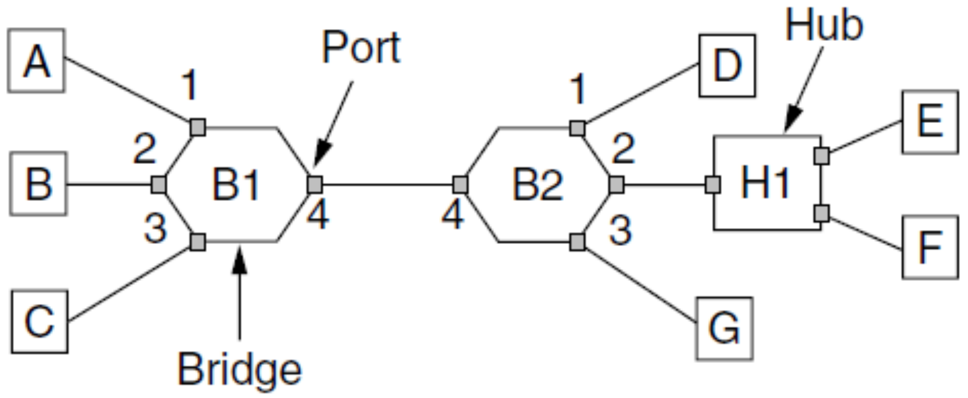
- Port for destination same as source port

  → do nothing
- Port for destination different from source port

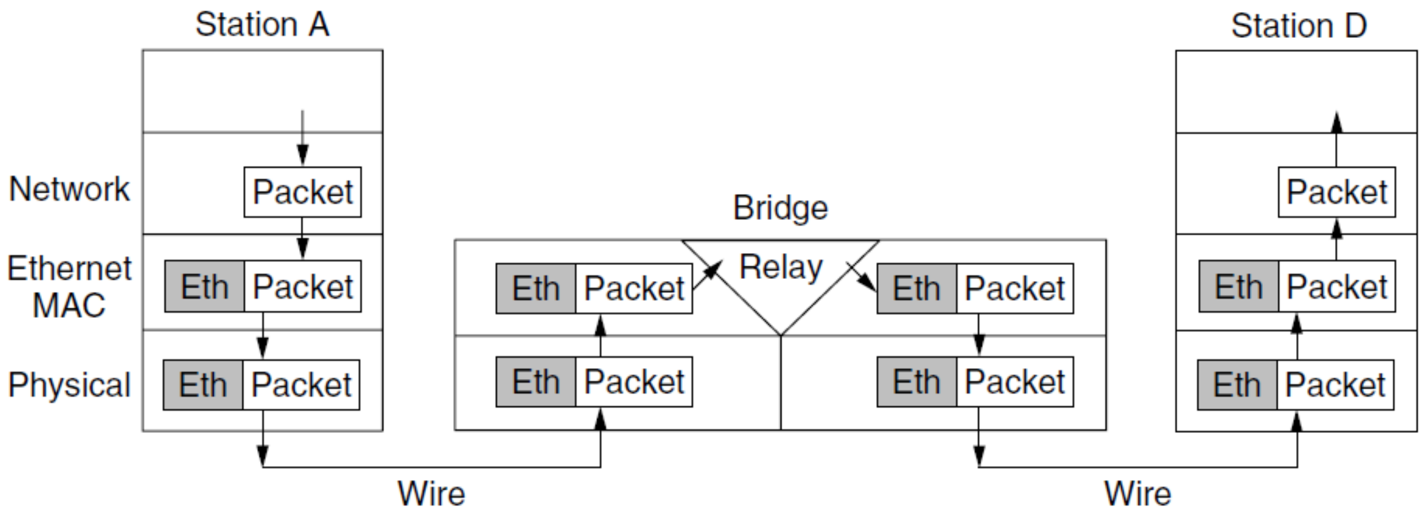
  → forward on destination port
- If destination port unknown

  → flood on all ports except source port

http://www.cisco.com/image/gif/paws/10607/lan-switch-transparent.swf



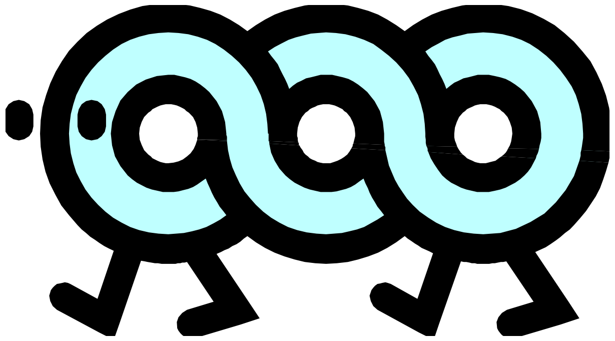
Processing at a bridge



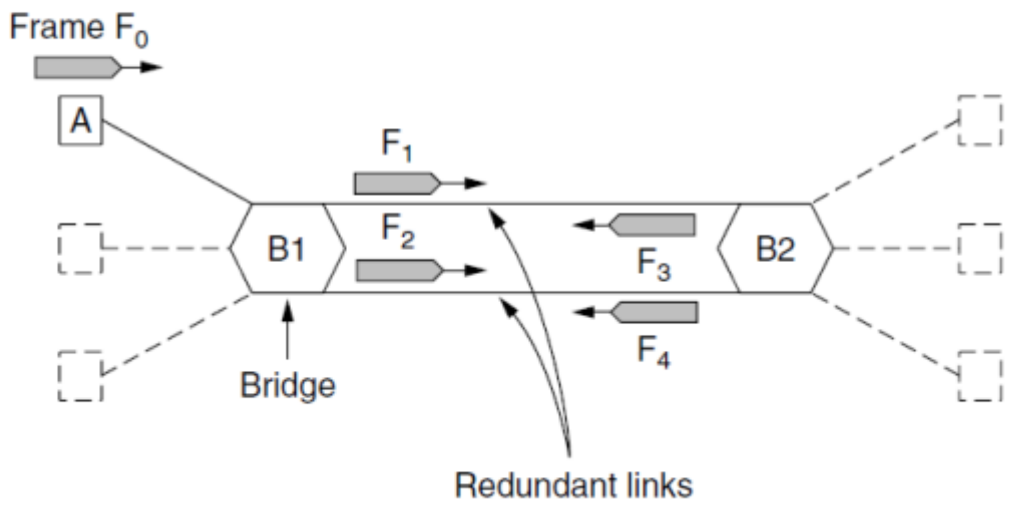
- Bridge only looks at MAC address in frame
  - May start forwarding as soon as destination port known
    - Cut-through switching, wormhole routing
- Bridge may rewrite headers
  - e.g. in VLANs

Loops

- Problem: loops in the network topology
  - May be accidental
  - May be added to provide redundancy
  - Backward learning flooding causes trouble



Loop example



A sends frame F_0, destination some unknown host D

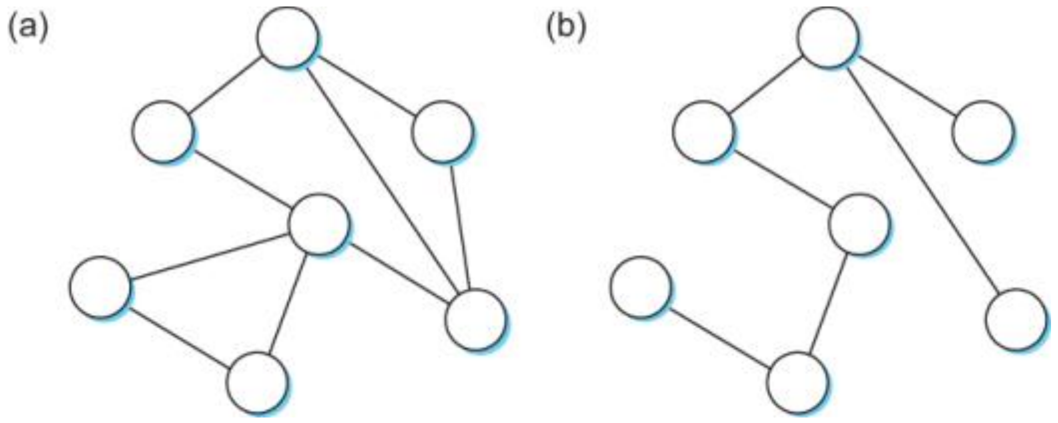
B1 doesn't know destination port, sends out as F_1 on port 1, F_2 on port 2 towards B2 (and all other ports except source port)

B2 gets F_1 on port 1, sends out as F_3 on port 2 back at B1
B2 gets F_2 on port 2, sends out as F_4 on port 1 back at B1

...

Spanning tree

- Spanning tree algorithm
  - Distributed algorithm run on switches
  - Switches converge on a single spanning tree
    - Keep all vertices (switches, network segments)
    - Drop some edges (ports)



Spanning tree algorithm

- Problem: loops in the network topology
  - Radia Perlamn at DEC
  - One week to figure out how to join LANs without loops
  - Took one day, then wrote a poem:

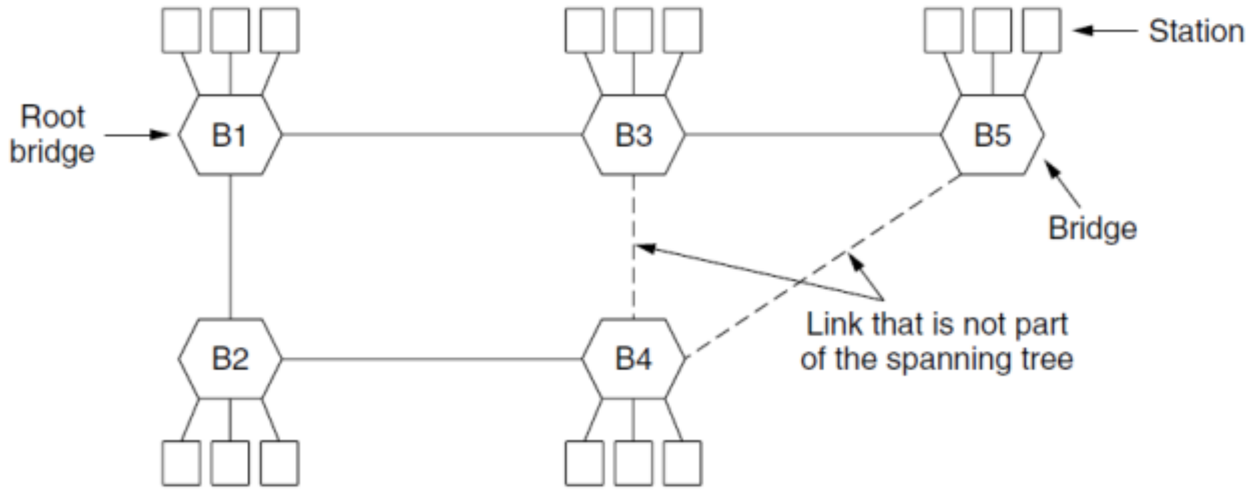
I think that I shall never see
A graph more lovely than a tree.
A tree whose crucial property
Is loop-free connectivity.
A tree which must be sure to span.
So packets can reach even LAN.
First the Root must be selected
by ID it is elected.
Least cost paths from Root are traced
In the tree these paths are placed.
A mesh is made by folks like me
Then bridges find a spanning tree.

http://www.youtube.com/watch?v=iE_AbM8ZykI

Spanning tree algorithm

- Algorithm:
  - Each bridge has unique identifier
    - Based on MAC address of switch
  - Root is bridge with smallest ID
  - Root forwards all frames over all ports
  - Each bridge computes shortest path to root
    - This port is the bridge's root port
  - Each network segment (multi-drop/hub)
    - Bridge closest to root is that segment's designated port

Spanning tree, no shared segments



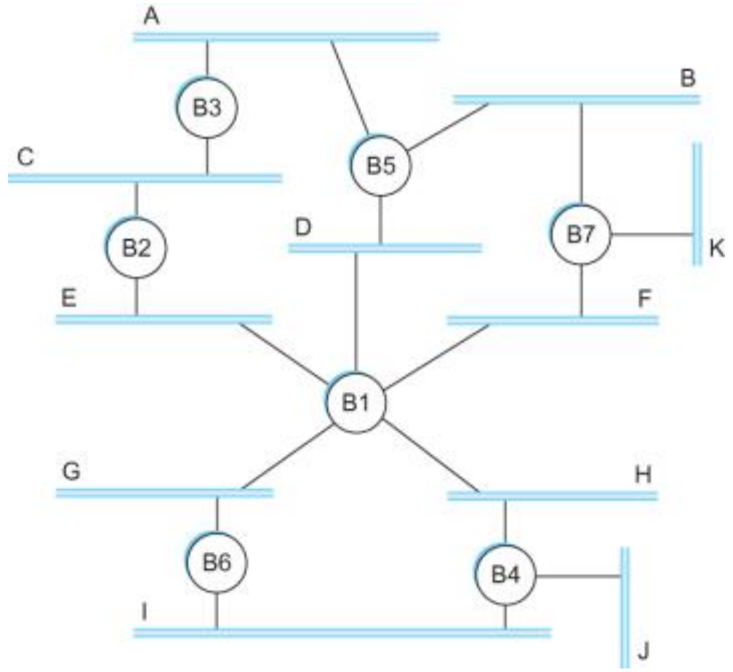
1) B1 has the lowest ID, news spreads, all bridges agree B1 is root.
2) B2 and B3 are directly connect to root, added to tree
3) B4 can reach B1 in two hops via B3 or B2, B2 wins (lower ID)
4) B5 can reach B1 in two hops via B3 (other paths are three hops)
5) Links from B3 to B4 and from B4 to B5 turned off

Spanning tree in detail

- Message (Y, d, X)
  - X claiming to be d away from root node Y
- Initially, all bridges think they are root
  - Send message on all ports (X, 0, X)
  - Stop claiming to be root if you see message with < root ID
- On receiving message on port, is it better than best recorded for port?
  - Root with < ID
  - Root with = ID but shorter distance
  - Root ID and distance equal, but sending bridge has < ID
  - Record better message and forward on with (d + 1)

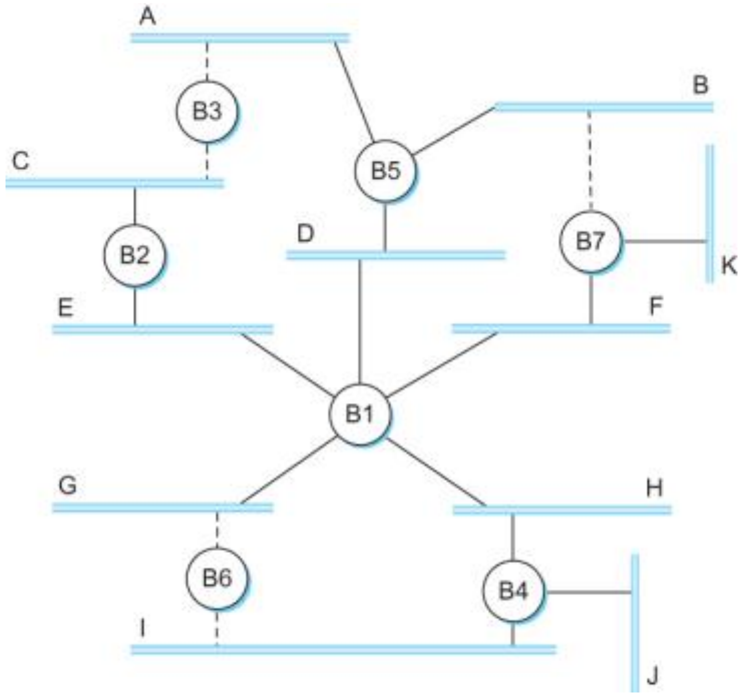
Spanning tree example 2

- Power just restored to network
  - Consider activity at node B3



Spanning tree example 2

- B3 receives (B2, 0, B2)
- Since 2 < 3, B3 accepts B2 as root
- B3 adds 1 to the distance advertised by B2 and sends (B2, 1, B3) to B5
- Meanwhile B2 accepts B1 as root because it has the lower id and it sends (B1, 1, B2) toward B3
- B5 accepts B1 as root and sends (B1, 1, B5) to B3
- B3 accepts B1 as root and it notes that both B2 and B5 are closer to the root than it is on network segments A & C.
  - Thus B3 stops forwarding messages on both its ports
  - This leaves B3 with both ports not selected



Recovery from failure

- After system stabilizes:
  - Root continues to send configuration messages
  - Other bridges forward on using tree
  - If bridge fails, downstream bridges won't receive configuration messages
    - After timeout period, downstream bridges claim to be root, algorithm starts again
- Spanning tree doesn't avoid congested bridges
  - May be another route, but can't use if not in tree

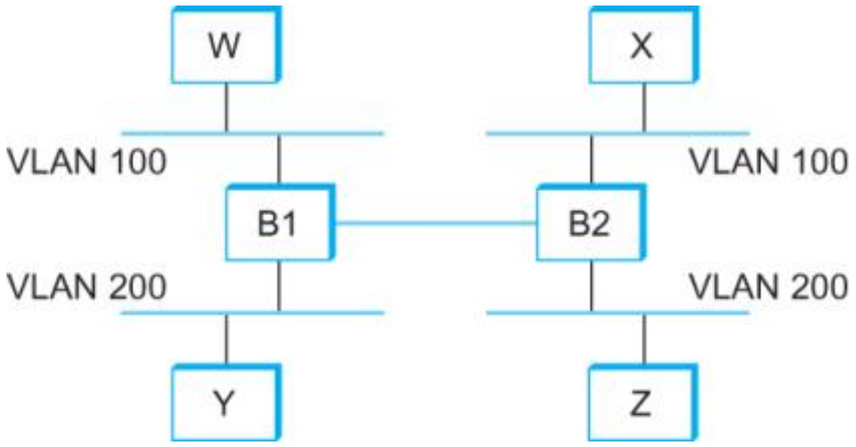
http://www.cisco.com/warp/public/473/spanning_tree1.swf

Limitations

- Bridge limitations
  - Limited to kinds of networks they can interconnect
    - Must support similar address format
    - Ethernet to 802.11 okay, both use 48-bit MAC
    - Ethernet to ATM not okay
  - Doesn't scale to large networks
    - Spanning tree algorithm scales linearly
    - Bridges forward all broadcast traffic

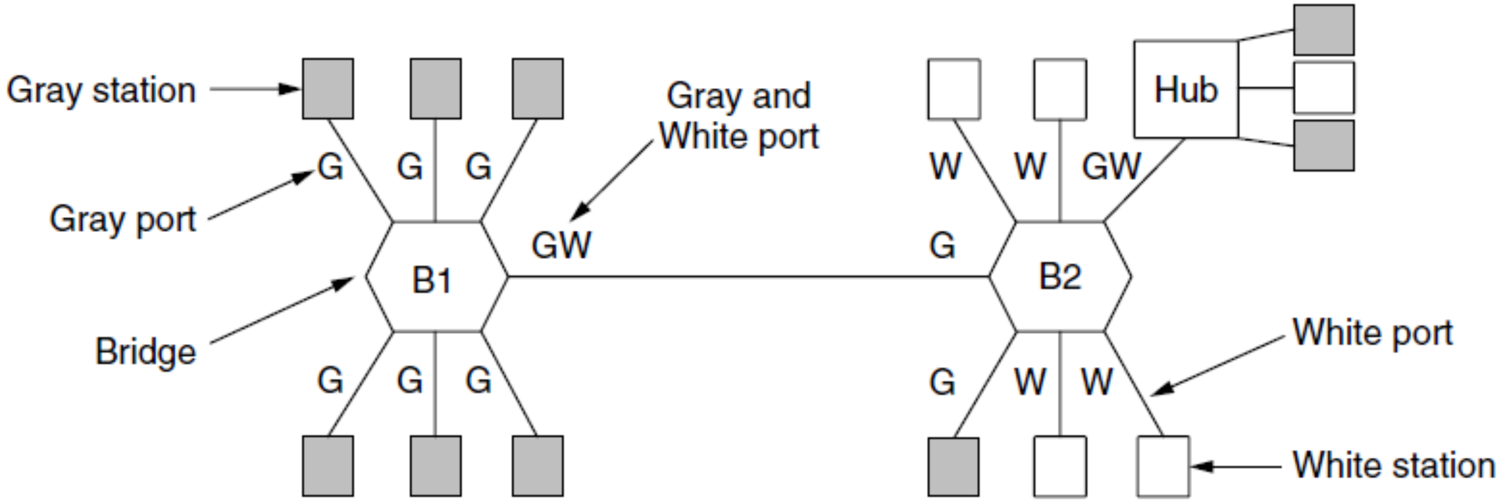
Scalability and security

- Virtual LAN (VLAN)
  - Partition single LAN into multiple virtual LANs
  - Traffic only forwarded to hosts on same VLAN (including broadcast traffic)
  - Segregation by department or security need
    - e.g. web servers versus HR records



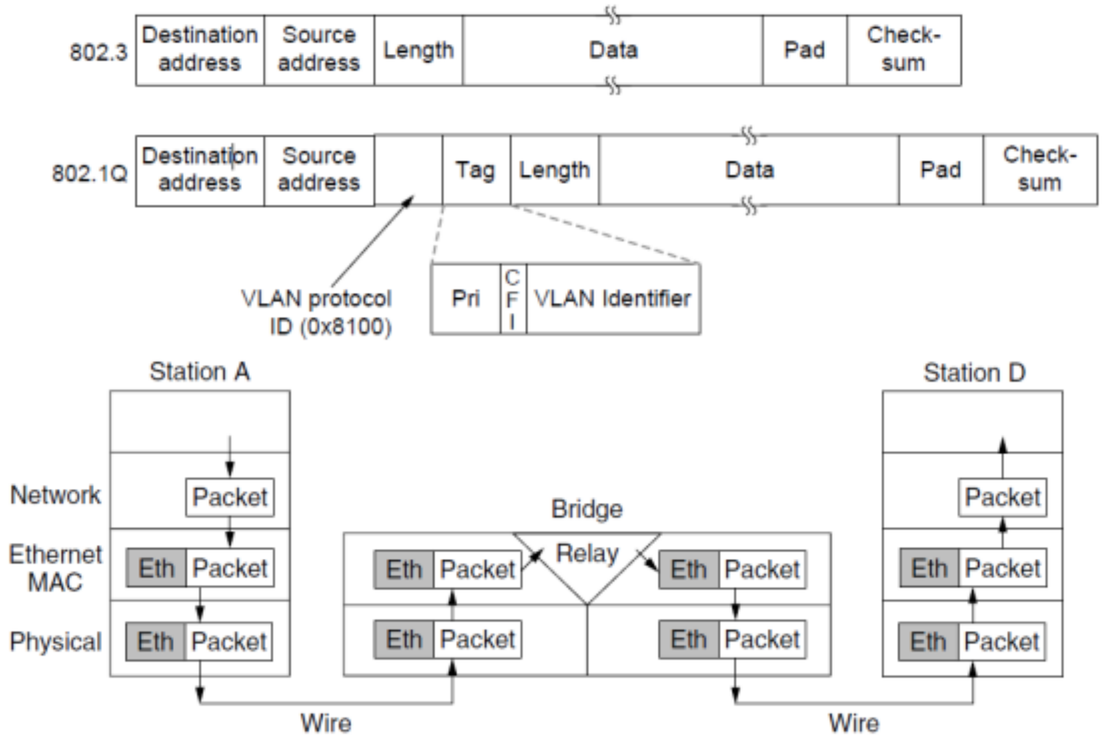
Example VLAN

- Setting up a VLAN
  - Switches must be VLAN-aware
  - Each host given a "color"
  - Configuration tables in the bridges
    - What colors attached to which ports



VLAN details

- Problem: How does bridge know frame color?
  - IEEE 802.1Q
  - Changed Ethernet header to add VLAN identifier



Summary

- Switching and bridging
  - Connect multiple LANs together
    - Operates on network layer
  - Backward learning
    - Which port is connected to which host
  - Spanning tree
    - Avoid cycles when sending messages around
  - Limited scale (10s of switches)
  - Limited heterogeneity (same type of network)