ECE/COMPSCI 356 Computer Network Architecture

Lecture 9: Switching technologies

Neil Gong

neil.gong@duke.edu

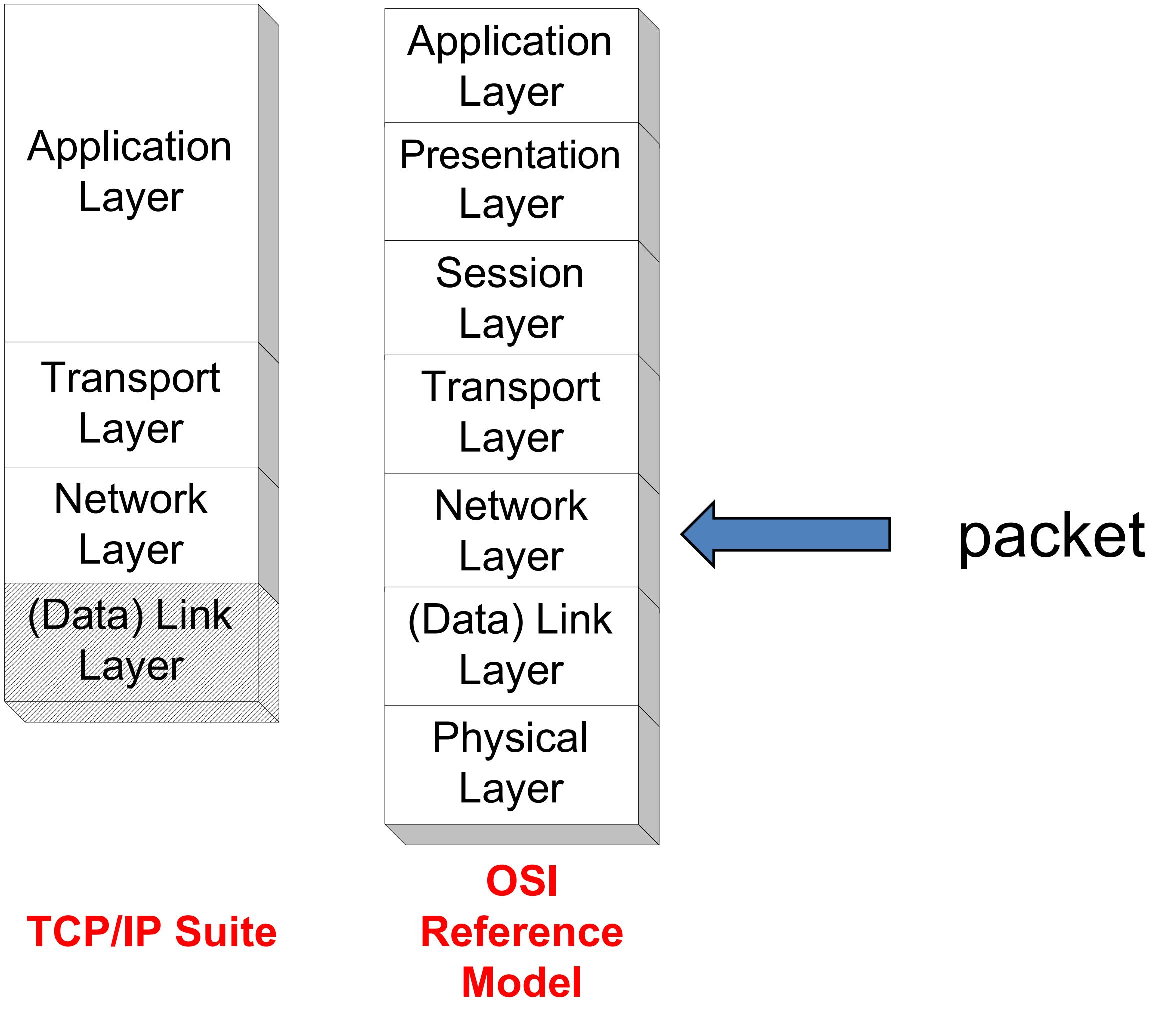
Slides credit: Xiaowei Yang, PD

Overview

- Types of switching
  - Datagram
  - Virtual circuit
  - Source routing
- Bridges (also called LAN switches)
  - Learning bridges
  - Spanning Tree Algorithm

Roadmap

- What we have learnt: single link network
  - Problem: limited scale
  - Point-to-point links connect only two nodes
  - Multiple access links
    - Ethernet < 1024 hosts, 2500 meters
    - Wireless limited by radio ranges
- How to interconnect different types of links and networks to build large, global network?



Application Layer
Transport Layer
Network Layer
(Data) Link Layer
Application Layer
Presentation Layer
Session Layer
Transport Layer
Network Layer
(Data) Link Layer
Physical Layer
packet
TCP/IP Suite
OSI Reference Model

Internetworking

- Switches
  - Devices connecting the same type of links
- Routers/gateways
  - Devices connecting different types of links
- IP and Routing

Internetworking

- **Switches**
  - Devices connecting the same type of links
- Routers/gateways
  - Devices connecting different types of links
- IP and Routing

Switch

- A switch is a device with several inputs and outputs
  - Called ports

A star topology

- A switch has a limited number of input and output ports
- A switch connects with other nodes in a star topology
- Switches can be connected to each other to build larger networks

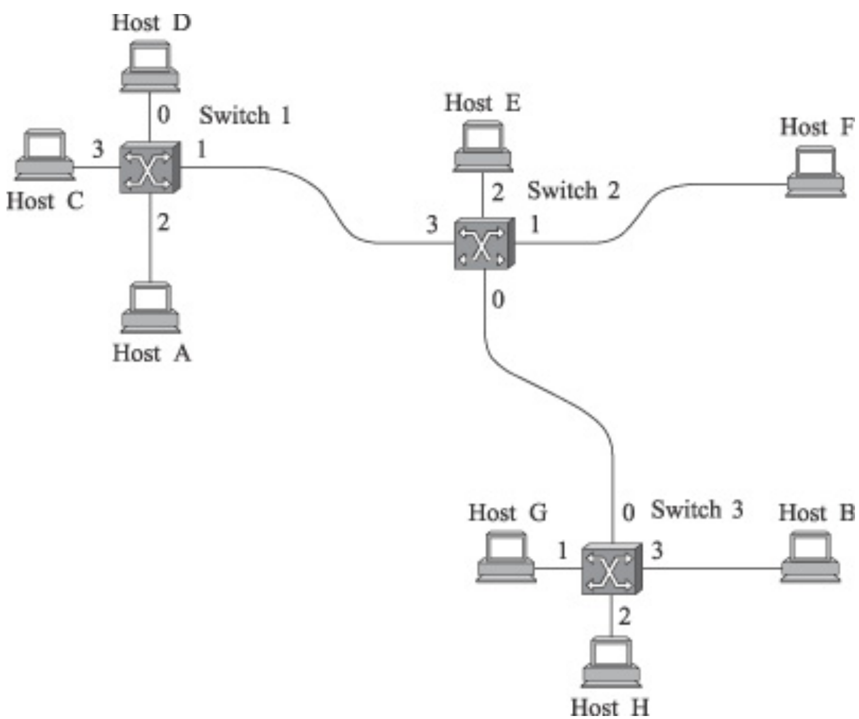
Switching technologies

- Receive a packet on one port
- Which output port?
- Solution
  - Connectionless: datagram
  - Connection oriented: virtual circuit
  - Source routing

Datagram

- Each host has an address
  - A global unique identifier
  - E.g., Ethernet has 48-bit addresses
- Every packet contains the destination address
- A switch maintains a *forwarding table* that maps a packet to an output port

Switch 2’s forwarding table



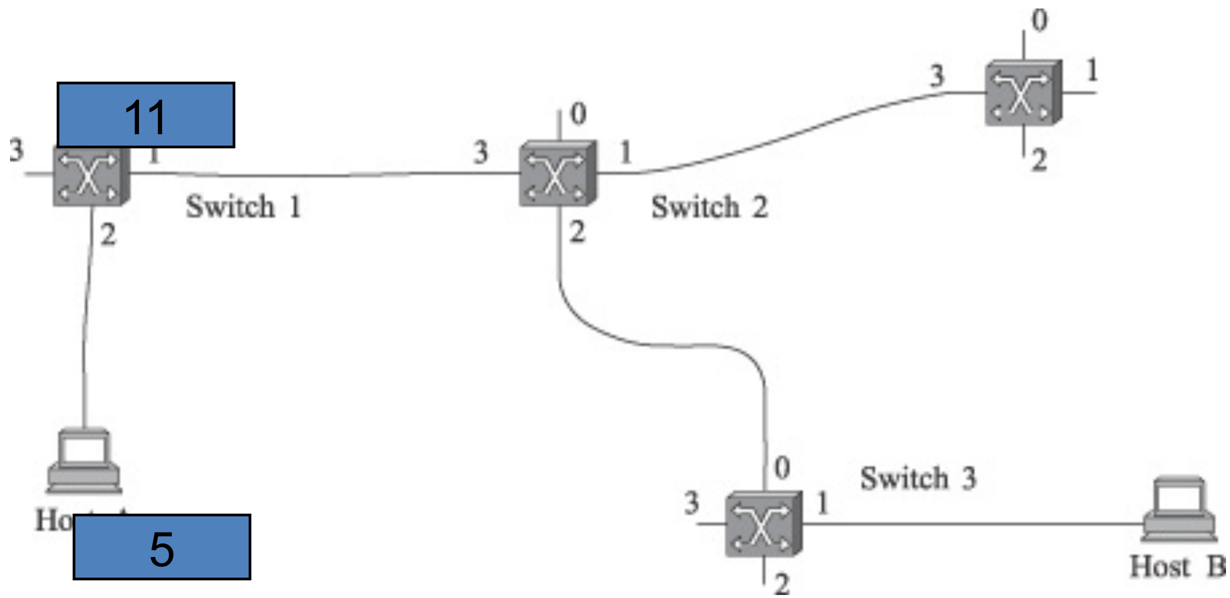
A	3
B	0
C	3
D	
E	
F	
G	
H	

Q: how does a switch compute the table?

Virtual circuit

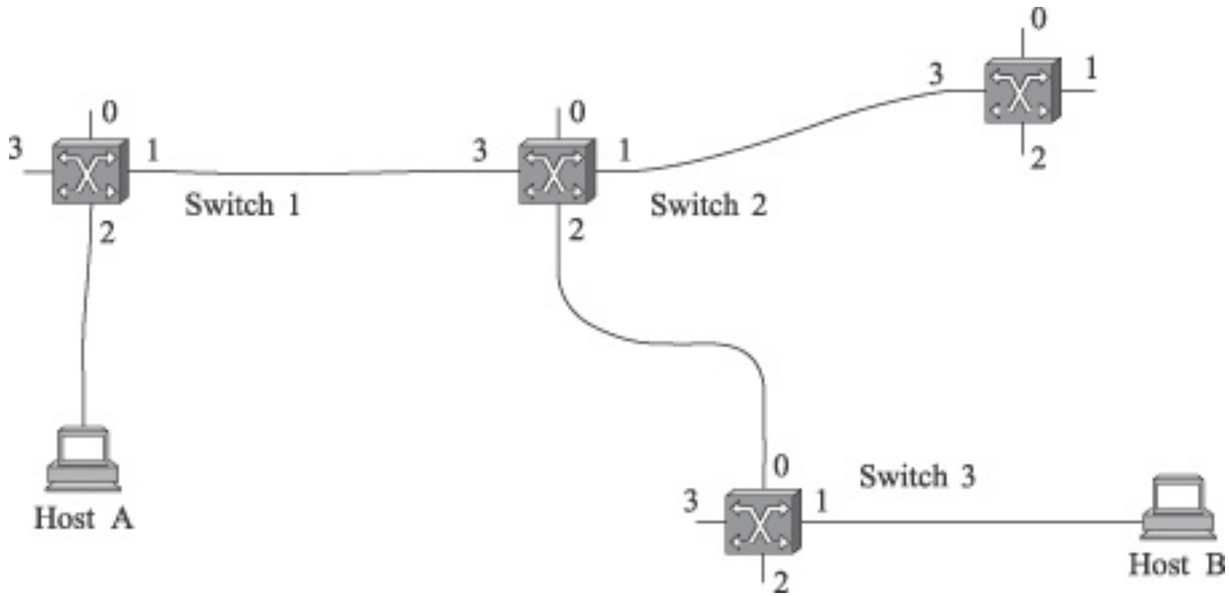
- Connection oriented
  - Set up a virtual circuit
  - Data transfer
- Data transfer
  - Switch maintains a *virtual circuit table*
  - Each entry includes
    - An incoming interface
    - A virtual circuit identifier
    - An outgoing interface
    - An outgoing virtual circuit identifier

Virtual circuit table (switch1)



Incoming interface	Incoming VCI	Outgoing interface	Outgoing VCI
2	5	1	11

Data transfer



- Algorithm:
  - If a packet arrives on the matching incoming port with the matching incoming VCI, it will be sent to the corresponding outgoing port with the corresponding VCI

- VCIs are link-local

How to setup a virtual circuit

- Administrator configured
  - Admin manually sets up VC tables
  - Does not suit large networks
  - Called permanent virtual circuit (PVC)

- Signaling
  - A host sends messages to dynamically setup or tear down a VC
  - Called switched virtual circuit (SVC)

SVC setup protocol

- A host A sends a setup message to first hop switch, including the final destination address
  - Similar to a datagram packet

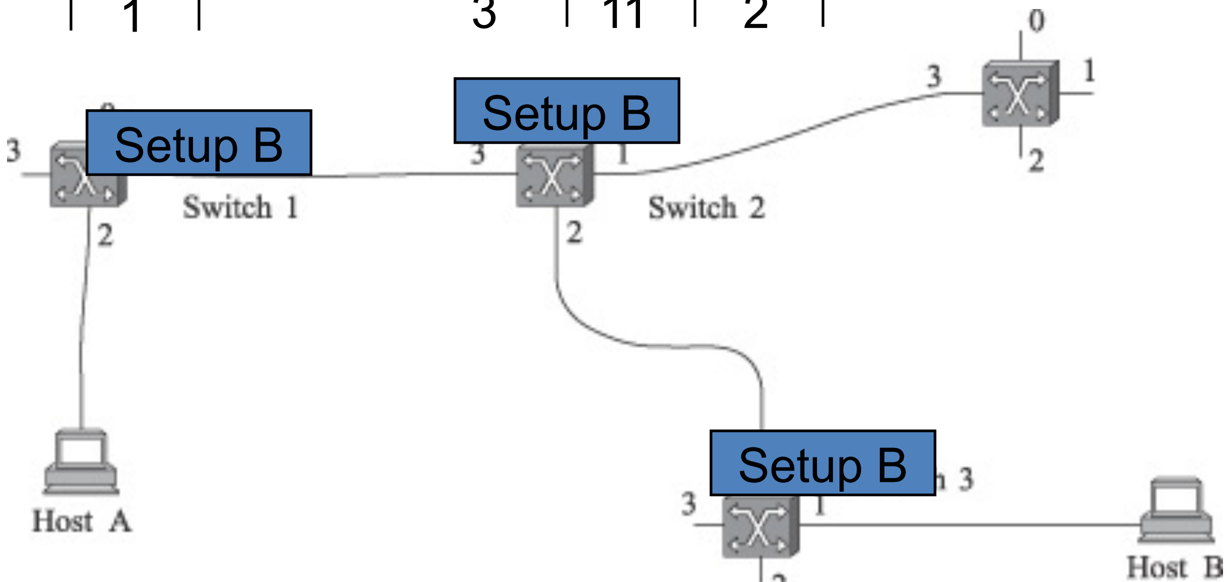
- The switch picks an unused VCI to identify the incoming connection, and fills part of the VC table

- Every switch repeats the process until the packet reaches the destination B

- The destination B sends an ack to inform its upstream switch the VCI for the connection

IF	VCI	OF	VCI
2	5	1	

IF	VCI	OF	VCI
3	11	2	



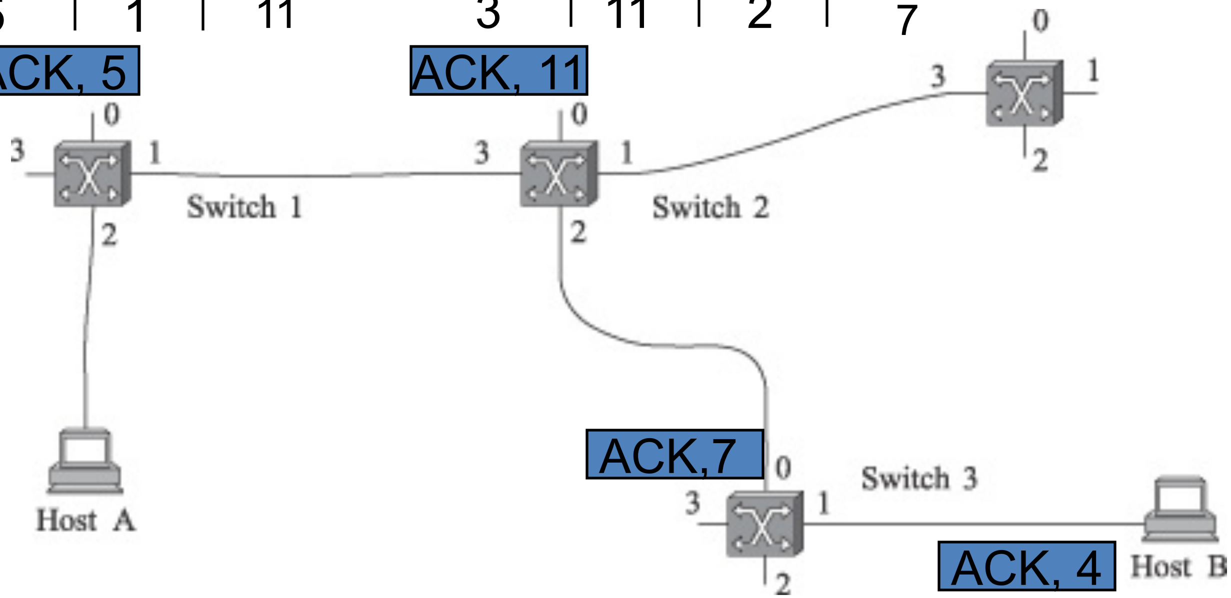
Setup B

IF	VCI	OF	VCI
0	7	1	

VCI
4

IF	VCI	OF	VCI
2	5	1	11

IF	VCI	OF	VCI
3	11	2	7



B: VCI 5

IF	VCI	OF	VCI
0	7	1	4

VCI
4

Characteristics of Virtual Circuit

- Cons
  - Connection setup wait
  - One switch failure tears down the entire connection
- Pros
  - Data packets contain a small VCI, not the full destination addresses
  - Resource reservation
    - Buffers can be allocated during the setup phase for a virtual circuit
    - An example (X.25)
      - Buffers allocated during connection setup
      - Sliding window is run between pairs of nodes (hop-by-hop flow control)
      - Circuit is rejected if no more buffer

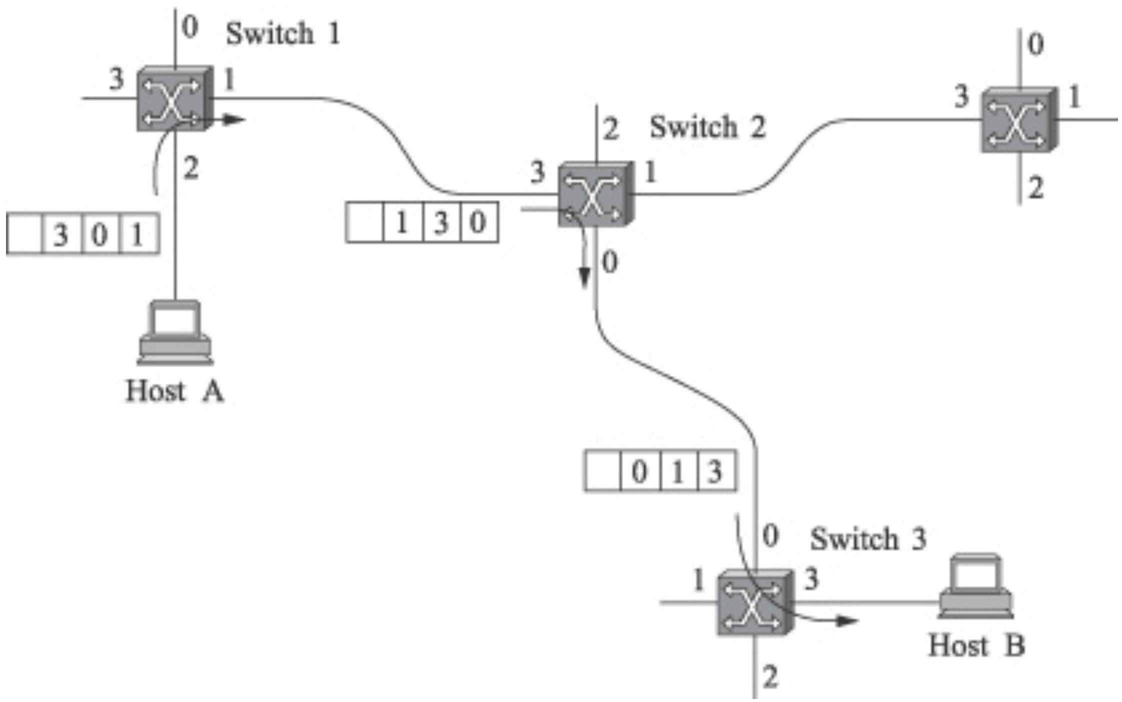
Examples of VC technologies

- X.25
- Frame relay
- Asynchronous Transfer Mode (ATM)
  - Popular in the late 80s and early 90s due to its high speed
    - Major telecoms supported it
  - Popularity faded. IP/Ethernet ruled

Switching technologies

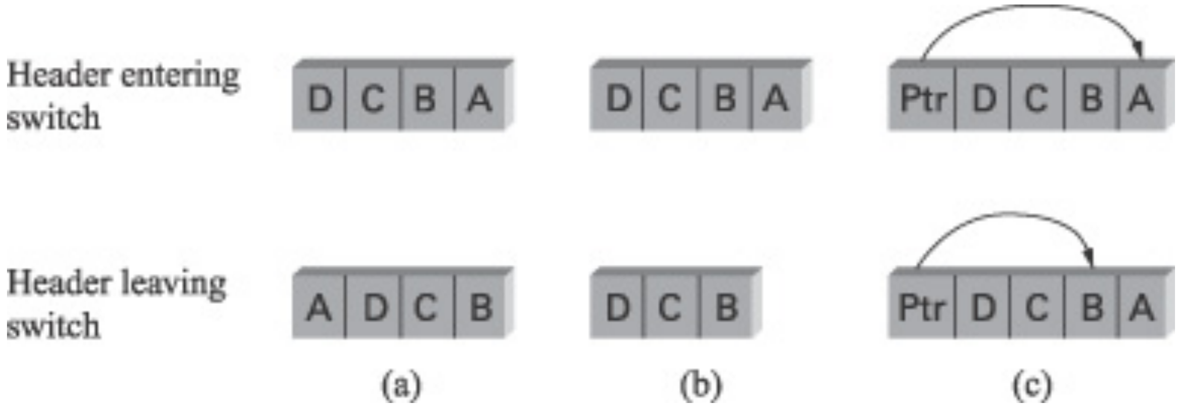
- Connectionless: datagram
- Connection oriented: virtual circuit
  - An example: ATM
- Source routing

Source routing



- Source host provides all the information for packets to travel across the network
  - Packets carry output port numbers
  - Variable header length

Handling source routing headers



a. Rotation
b. Stripping
   – No return path!
c. Pointer

Bridges

- Class of switches that is used to forward packets between LANs such as Ethernets
  - Also known as LAN switches

- Suppose you have a pair of Ethernets that you want to interconnect
  - One approach is put a repeater in between them
    - It might exceed the physical limitation of the Ethernet
      - No more than four repeaters between any pair of hosts
      - No more than a total of 2500 m in length is allowed
  - An alternative would be to put a node between the two Ethernets and have the node forward frames from one Ethernet to the other
    - This node is called a **Bridge**
    - A collection of LANs connected by one or more bridges is usually said to form an **Extended LAN**

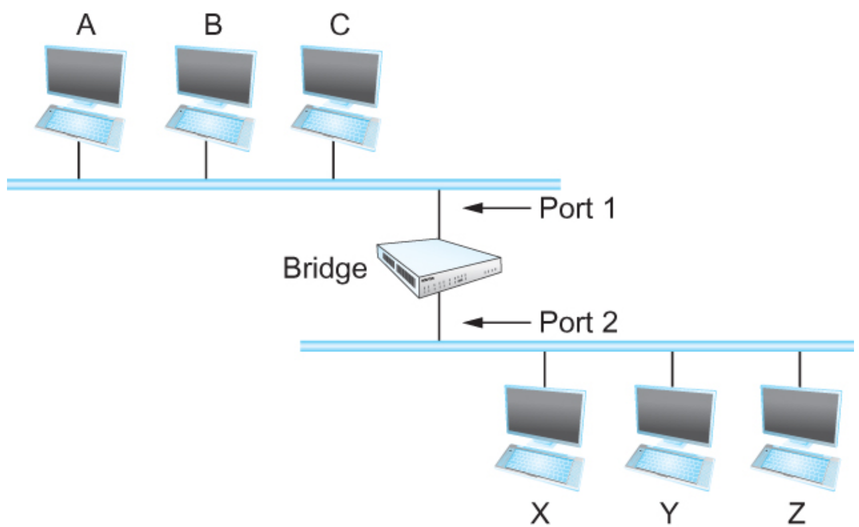
Bridges

- Simplest Strategy for Bridges
  - Accept LAN frames on their inputs and forward them out to all other outputs
  - Used by early bridges

- Learning Bridges
  - Observe that there is no need to forward all the frames that a bridge receives

Bridges

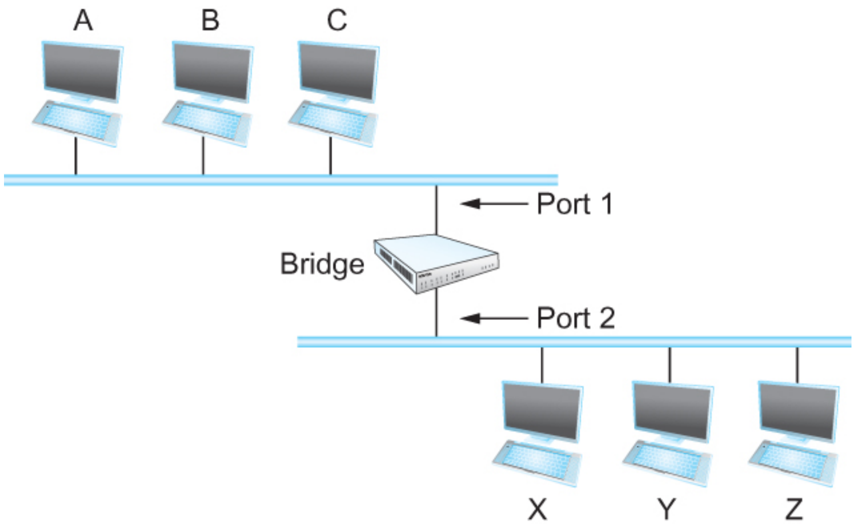
- Consider the following figure
  - When a frame from host A that is addressed to host B arrives on port 1, there is no need for the bridge to forward the frame out over port 2.



  - How does a bridge come to learn on which port the various hosts are?

Bridges

- Solution
  - Download a table into the bridge



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

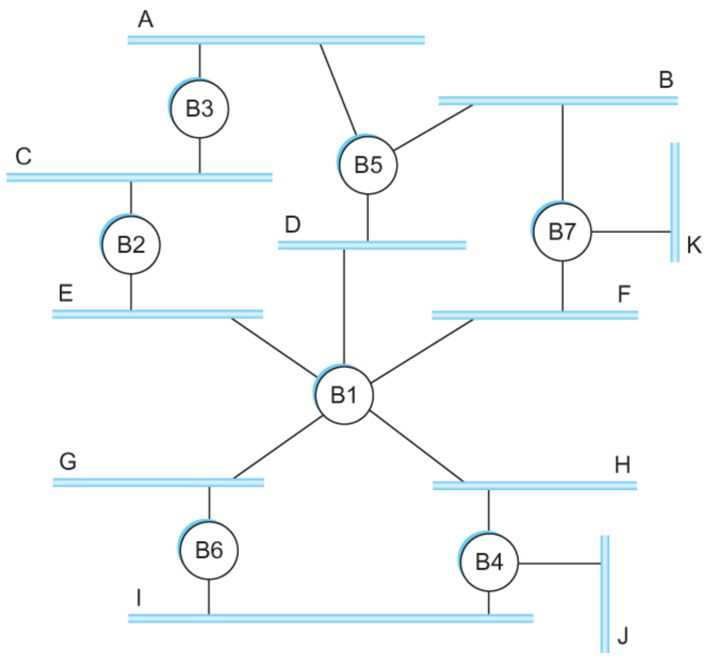
  - Who does the download?
    - Human
      - Too much work for maintenance

Bridges

- Can the bridge learn this information by itself?
  - Yes
- How
  - Each bridge inspects the source address in all the frames it receives
  - Record the information at the bridge and build the table
  - When a bridge first boots, this table is empty
  - Entries are added over time
  - A timeout is associated with each entry
  - The bridge discards the entry after a specified period of time
    - To protect against the situation in which a host is moved from one network to another
- If the bridge receives a frame that is addressed to host not currently in the table
  - Forward the frame out on all other ports

Bridges

- Strategy does not work fine if the extended LAN has a loop in it
- Why?
  - Frames potentially loop through the extended LAN forever



  - Bridges B1, B4, and B6 form a loop

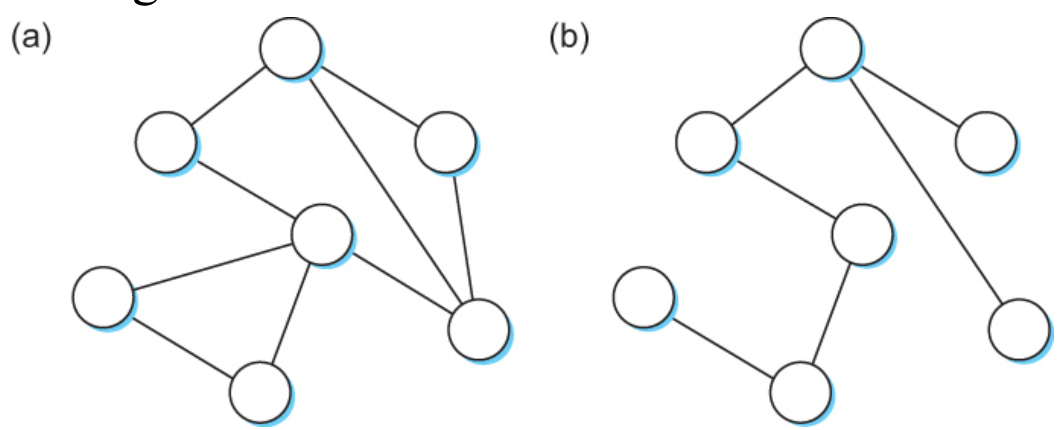
Bridges

- How does an extended LAN come to have a loop in it?
  - Network is managed by more than one administrator
    - For example, it spans multiple departments in an organization
    - It is possible that no single person knows the entire configuration of the network
      - A bridge that closes a loop might be added without anyone knowing
  - Loops are built into the network to provide redundancy in case of failures
- Solution
  - Distributed Spanning Tree Algorithm

Spanning Tree Algorithm

- Think of the extended LAN as being represented by a graph that possibly has loops (cycles)

- A spanning tree is a sub-graph of this graph that covers all the vertices but contains no cycles
  – Spanning tree keeps all the vertices of the original graph but throws out some of the edges



Example of (a) a cyclic graph; (b) a corresponding spanning tree.

Spanning Tree Algorithm

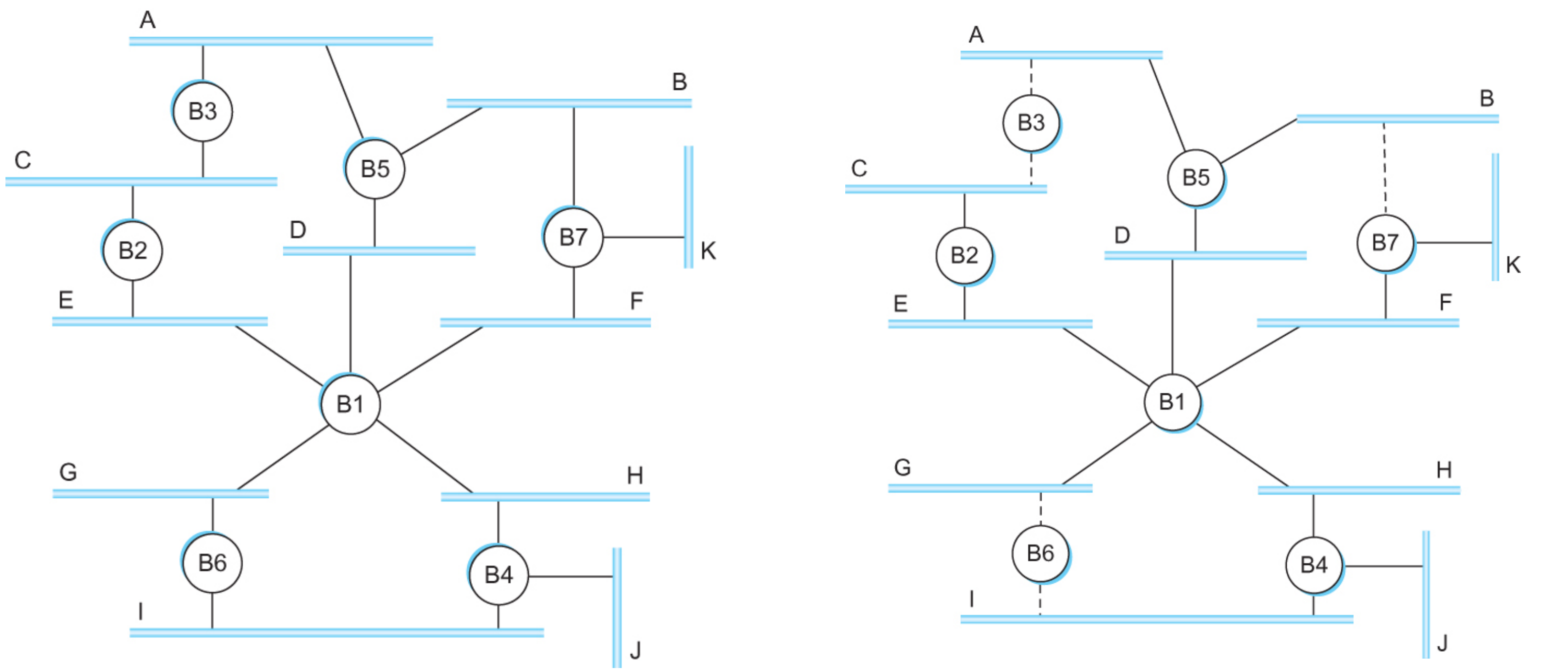
- A protocol used by a set of bridges to agree upon a spanning tree for a particular extended LAN
- IEEE 802.1 specification for LAN bridges is based on this algorithm

- Each bridge decides the ports over which it is and is not willing to forward frames
  - In a sense, it is by removing ports from the topology that the extended LAN is reduced to an acyclic tree
  - It is even possible that an entire bridge will not participate in forwarding frames

Spanning Tree Algorithm

- Main idea
  - Each bridge has a unique identifier
    - B1, B2, B3,…and so on.
  - Select the bridge with the smallest id as the root of the spanning tree
  - Represent the extended LAN as a graph
  - Use *breadth first search* to find a spanning tree

An example



- Each LAN has a single *designated bridge* that is responsible for forwarding frames toward the root bridge
- B5 is the designated bridge for LAN A
- B5 is the designated bridge for LAN B

Distributed Implementation

- Bridges send *configuration messages* to determine the spanning tree
- Denote a configuration message from bridge X in which it claims to be distance d from the root bridge Y as (Y, d, X)
- Initially each bridge thinks it is the root, so it sends a configuration message on each of its ports identifying itself as the root and giving a distance to the root of 0
- Upon receiving a configuration message over a particular port, the bridge checks to see if the new message is *better* than the current best configuration message recorded for that port
- The new configuration is better than the currently recorded information if
  - It identifies a root with a smaller id or
  - It identifies a root with an equal id but with a shorter distance or
  - The root id and distance are equal, but the sending bridge has a smaller id

Distributed Implementation

- If the new message is better than the currently recorded one,
  - The bridge discards the old information and saves the new information
  - It adds 1 to the distance-to-root field

- When a bridge receives a configuration message indicating that it is not the root bridge (that is, a message from a bridge with smaller id)
  - The bridge stops generating configuration messages on its own
  - Only forwards configuration messages from other bridges after adding 1 to the distance field

Distributed Implementation

- When a bridge receives a configuration message that indicates it is not the designated bridge for that port

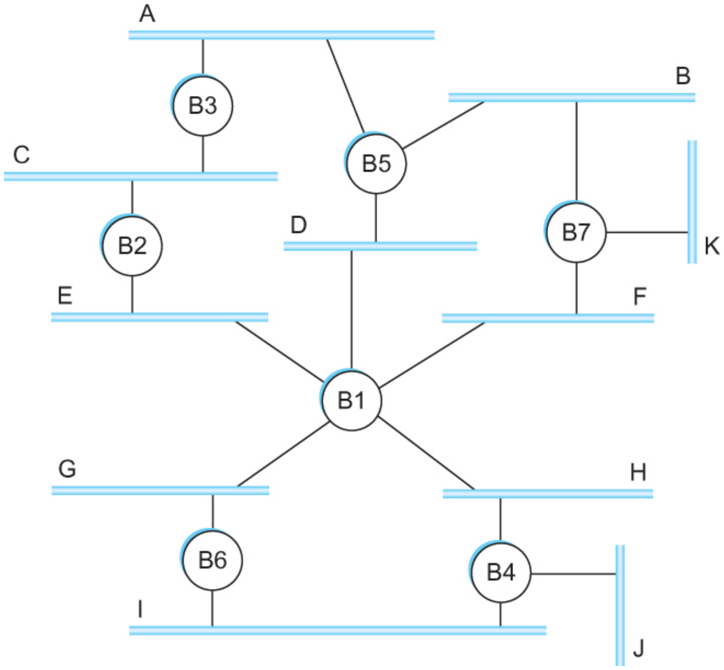
  => a message from a bridge that is closer to the root or equally far from the root but with a smaller id

    - The bridge stops sending configuration messages over that port

- When the system stabilizes,
  - Only the root bridge is still generating configuration messages.
  - Other bridges are forwarding these messages only over ports for which they are the designated bridge

An example

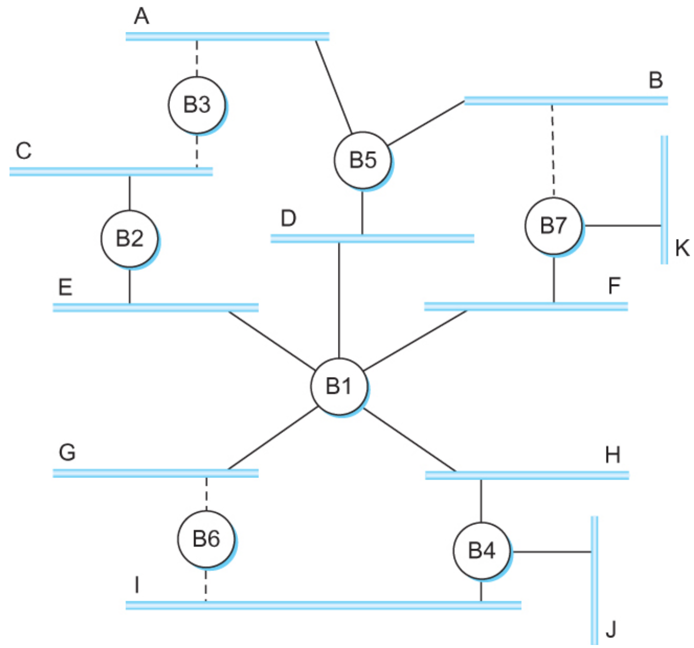
- Consider the situation when the power had just been restored to the building housing the following network



- All bridges would start off by claiming to be the root

An example

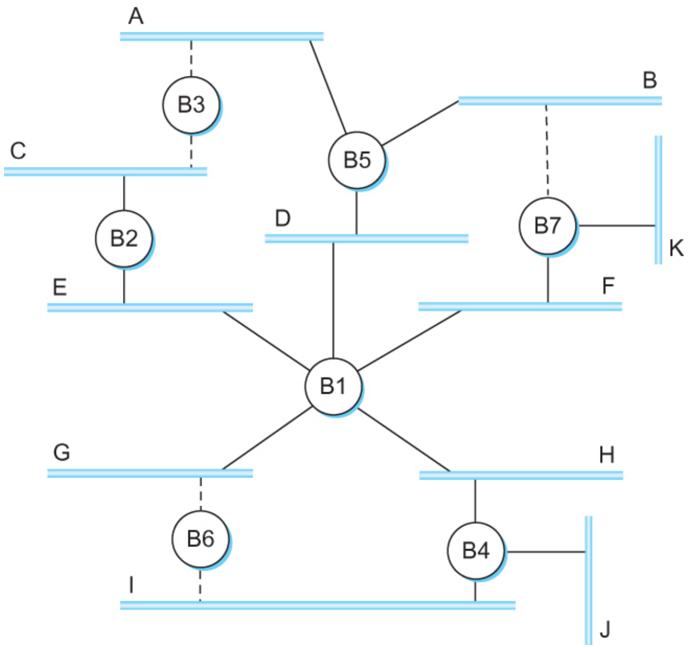
- Recall we denote a configuration message from node X in which it claims to be distance d from the root node Y as (Y, d, X)



- Consider the activity at node B3

An example

- B3 initial configuration message
  - B3-B2: (B3, 0, B3)
  - B3-B5: (B3, 0, B3)
- B3 receives (B2, 0, B2)
- Since 2 < 3, B3 accepts B2 as root
  - B3-B2: (B2, 1, B2)
  - B3-B5: (B3, 0, B3)
- B3 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
  - B3-B2: (B1, 2, B2)
  - B3-B5: (B3, 0, B3)
- B5 accepts B1 as root and sends (B1, 1, B5) to B3
  - B3-B2: (B1, 2, B2)
  - B3-B5: (B1, 2, B5)
- B3 closes port to B5
  - The remaining only port to B2 can also be closed



Spanning Tree Algorithm

- Even after the system has stabilized, the root bridge continues to send configuration messages periodically
  - Other bridges continue to forward these messages

- When a bridge fails, the downstream bridges will not receive the configuration messages

- After waiting a specified period of time, they will once again claim to be the root and the algorithm starts again

Spanning Tree Algorithm

- Broadcast and Multicast
  - Forward all broadcast/multicast frames to all selected ports
  - Host decides whether to accept a frame or not

Spanning Tree Algorithm

- Limitation of Bridges
  - Do not scale
    - Spanning tree algorithm does not scale
    - Broadcast does not scale
  - Security
    - Every host can snoop
  - Do not accommodate heterogeneity
    - Ethernet-bridge-Ethernet: ok
    - Ethernet-bridge-ATM: not ok
      - Bridge examines frame headers

Summary

- Types of switching
  - Datagram
  - Virtual circuit
  - Source routing
- Bridges (also called LAN switches)
  - Learning bridges
  - Spanning Tree Algorithm