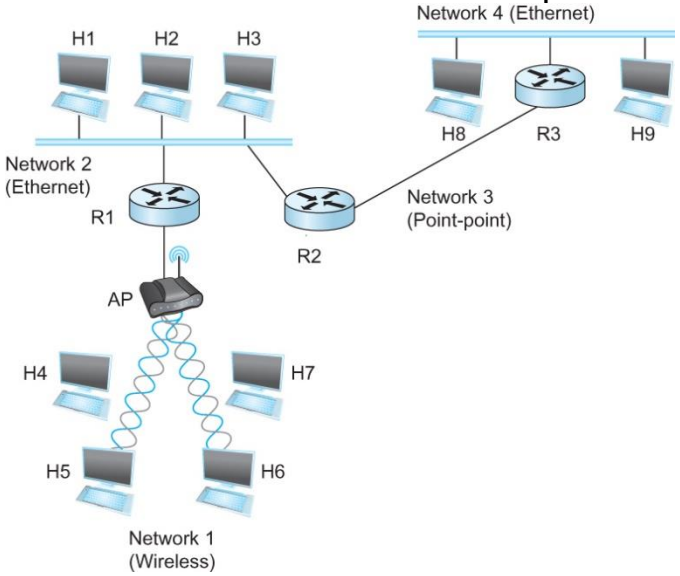
IF2230 Jaringan Komputer
Internetworking
Bridging and Switching

Robithoh Annur
Andreas Bara Timur
Monterico Andrian

Introduction

- What is internetwork
  - An arbitrary collection of networks interconnected to provide some sort of host-host to packet delivery service



A simple internetwork where H represents hosts and R represents routers

Introduction

- It is necessary to connect a LAN to another LAN or to a WAN.
  - Computers within a LAN are often connected using a hub
  - LAN to LAN connections are often performed with a bridge.
  - Segments of a LAN are usually connected using a switch.
  - LAN to WAN connections are usually performed with a router (next lecture).

Topics

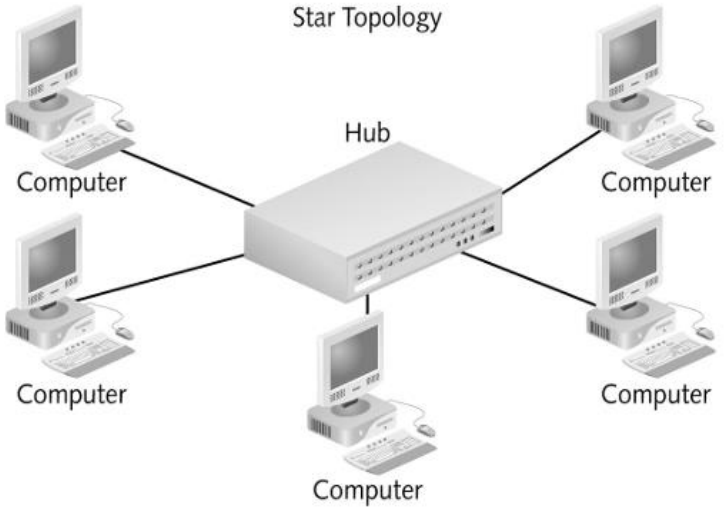
- Interconnecting LAN segments
  - HUB (Physical Layer)
  - Bridge (Link layer)
  - Layer 2 Switch (multi-port bridge, link layer)
- Interconnecting networks
  - Layer 3 Switch (network layer)
  - Router (network layer)

Interconnecting LAN Segments
Three Major Devices

- Hubs (layer 1 devices)
- Bridges (layer 2 devices)
  - Basic Functions
  - Self learning and bridge forwarding table
  - Forwarding/filtering algorithm
  - Bridge looping problem and spanning tree algorithm
- Ethernet Switches
  - Remark: switches are essentially multi-port bridges.
  - What we say about bridges also holds for switches!

Interconnecting with Hubs

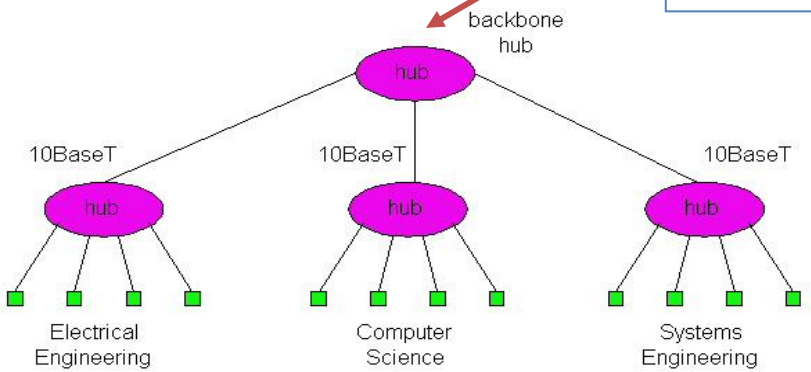
- A hub interconnects two or more nodes into a local area network.
- A hub connects multiple wires coming from different branches, for example, the connector in star topology which connects different stations.
- Hubs cannot filter data, so data packets are sent to all connected devices.
- Hubs do not have the intelligence to find out the best path for data packets which leads to inefficiencies and wastage.



Interconnecting with Hubs

- Backbone hub interconnects LAN segments
- Hubs expand one Ethernet connection into many and extends max distance between nodes
- But individual segment collision domains become one large collision domain
  - if a node in CS and a node EE transmit at same time: collision
- Can't interconnect 10BaseT & 100BaseT
  - Encoding is different: Manchester vs. 4B/5B

Recreates each bit, boosts its energy strength, and transmits the bit to all other interfaces

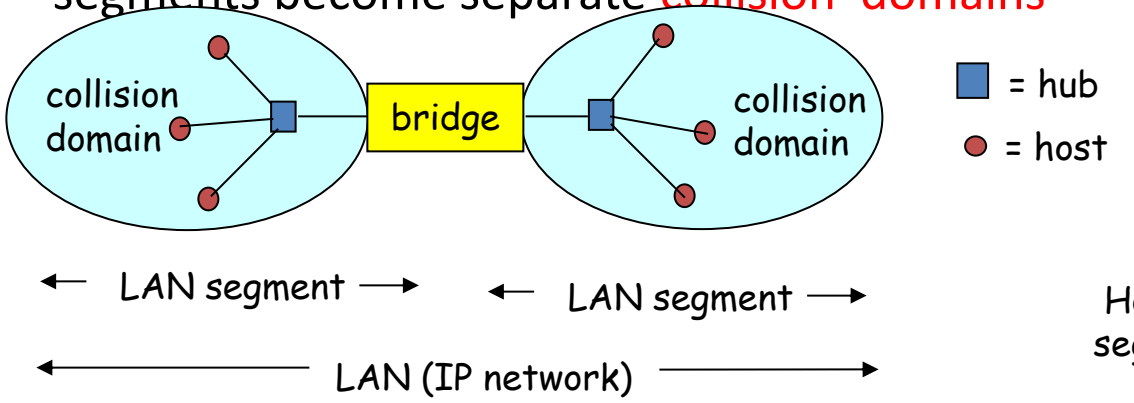


Bridges

- Link layer device
  - stores and forwards Ethernet frames
  - examines frame header and selectively forwards frame based on MAC destination address -- filtering
  - when frame is to be forwarded on a LAN segment, uses CSMA/CD to access the LAN segment
- transparent
  - hosts are unaware of the presence of bridges, it appears to them as a single whole network
- plug-and-play, self-learning
  - bridges do not need to be configured

Bridges: Traffic Isolation

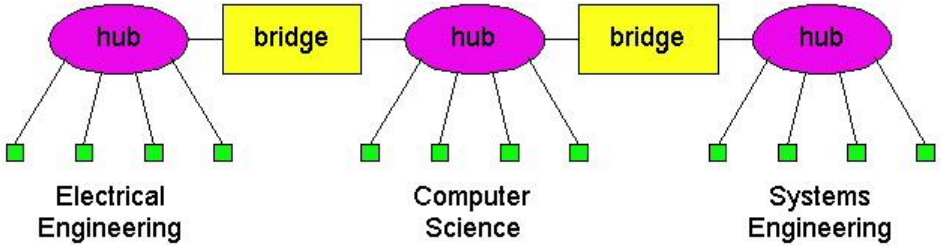
- Bridge installation breaks LAN into LAN segments
- Bridges filter packets:
  - same-LAN-segment frames not usually forwarded onto other LAN segments
  - segments become separate collision domains

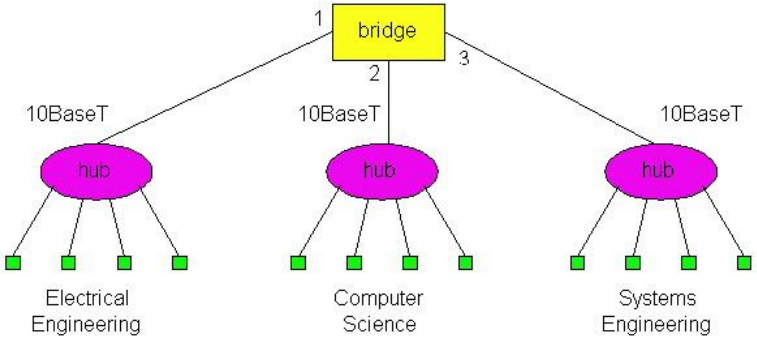


How to determine to which LAN segment to forward frame?

Interconnection without Backbone?

- Not recommended for two reasons:
  - single point of failure at Computer Science hub
  - all traffic between EE and SE must path over CS segment

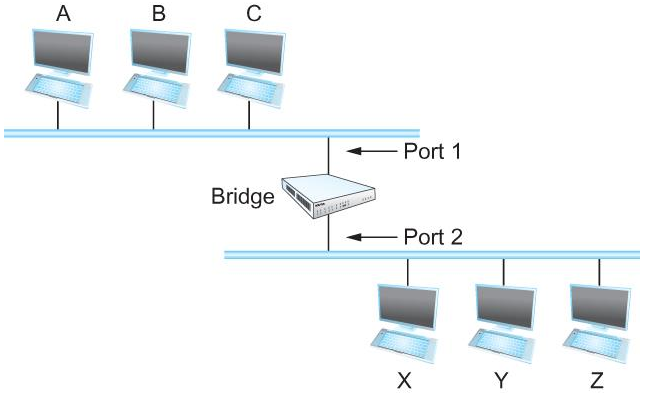




Recommended !

Bridges

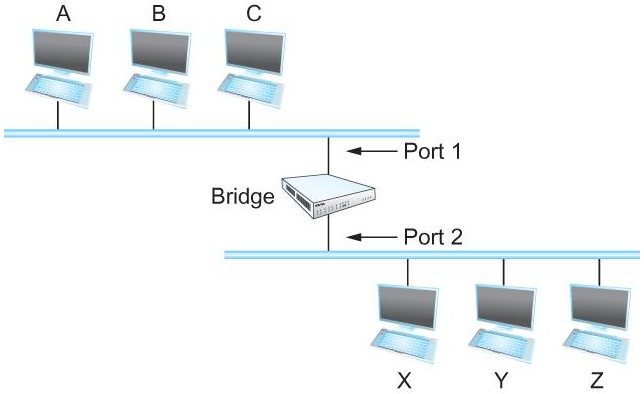
- Consider the following figure
  - When a frame from host A that is addressed to host B arrives on port 1, there is no need for the bridge to forward the frame out over port 2.



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

Bridges

- Solution
  - Download a table into the bridge



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

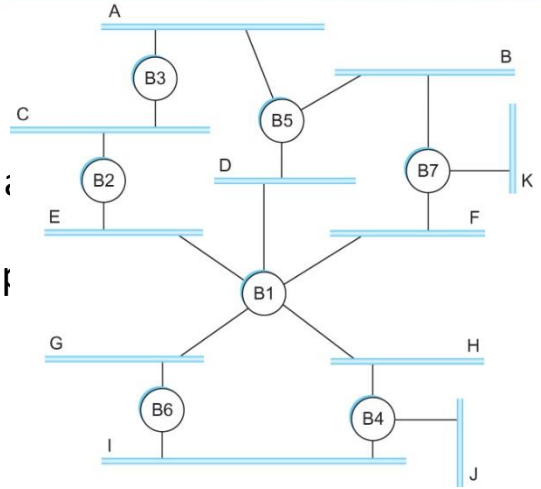
  - Who does the download?
    - Human
      - Too much work for maintenance

Self Learning

- A bridge has a bridge (forwarding) table
- How to perform self learning?
  - 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 works fine if the extended LAN does not have a loop in it
- Looping
  - Pros= for increased reliability, desirable to have redundant, a source to destination
  - Cons= with multiple paths, cycles result - bridges may multip forever
- 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 (by designed)
- Solution
  - Distributed Spanning Tree Algorithm → disabling subset of interfaces



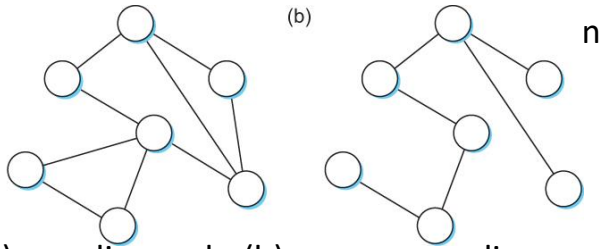
Bridges B1, B4, and B6 form a loop

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
- Developed by Radia Perlman at Digital
  - 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 n forwarding frames



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

Spanning Tree Algorithm

- Algorithm is dynamic
  - The bridges are always prepared to reconfigure themselves into a new spanning tree if some bridges fail
- Main idea
  - Each bridge selects the ports over which they will forward the frames

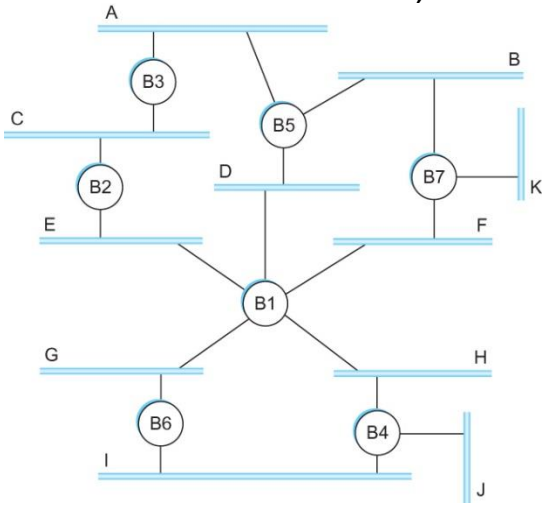
- Algorithm selects ports as follows:
  - Each bridge has a unique identifier
    - B1, B2, B3,…and so on.
  - Elect the bridge with the smallest id as the root of the spanning tree
  - The root bridge always forwards frames out over all of its ports
  - Each bridge computes the shortest path to the root and notes which of its ports is on this path
    - This port is selected as the bridge's preferred path to the root
  - Finally, all the bridges connected to a given LAN elect a single *designated bridge* that will be responsible for forwarding frames toward the root bridge

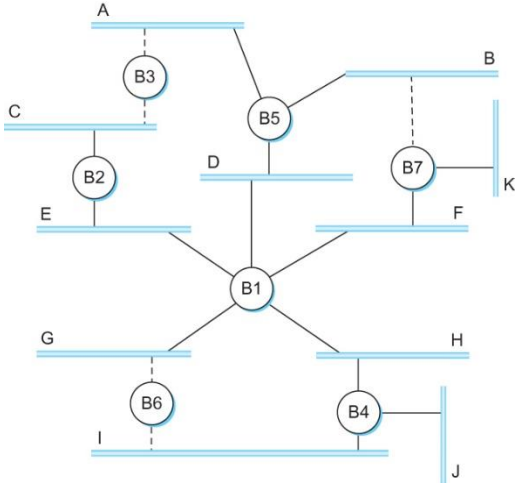
Spanning Tree Algorithm

- Each LAN's designated bridge is the one that is closest to the root
- If two or more bridges are equally close to the root,
  - Then select bridge with the smallest id
- Each bridge is connected to more than one LAN
  - So it participates in the election of a designated bridge for each LAN it is connected to.
  - Each bridge decides if it is the designated bridge relative to each of its ports
  - The bridge forwards frames over those ports for which it is the designated bridge

Spanning Tree Algorithm

- B1 is the root bridge
- B3 and B5 are connected to LAN A, but B5 is the designated bridge
- B5 and B7 are connected to LAN B, but B5 is the designated bridge





Spanning Tree Algorithm

- 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

Spanning Tree Algorithm

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

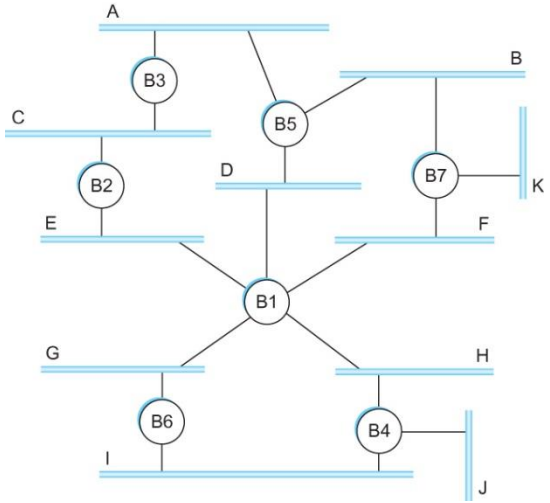
- When a bridge receives a configuration message that indicates it is not the designated bridge for that port

  => 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

Spanning Tree Algorithm

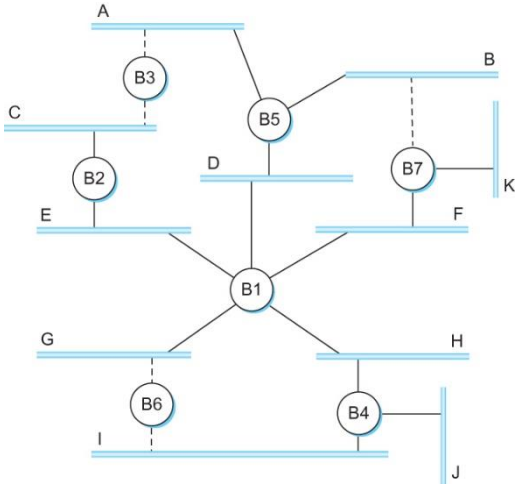
- Consider the situation when the power had just been restored to the building housing the following network



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

Spanning Tree Algorithm

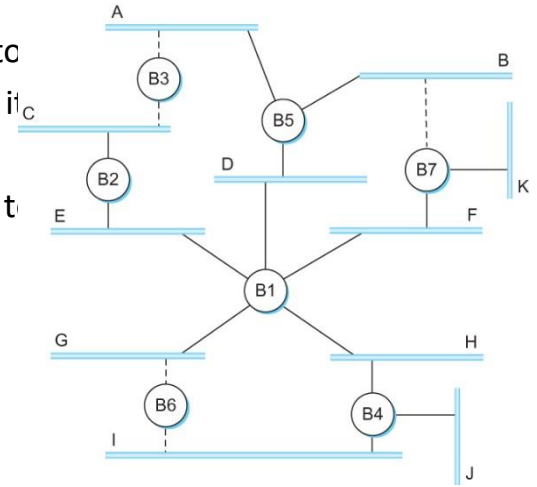
- Denote a configuration message from node X in which it claims to be distance d from the root node Y as (Y, d, X)



- Consider the activity at node B3

Spanning Tree Algorithm

- 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
- Meanwhile B2 accepts B1 as root because it has the lower id and it
- 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 t
  - Thus B3 stops forwarding messages on both its interfaces
  - This leaves B3 with both ports not selected



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
- Note
  - Although the algorithm is able to reconfigure the spanning tree whenever a bridge fails, it is not able to forward frames over alternative paths for the sake of routing around a congested bridge

Spanning Tree Algorithm

- 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

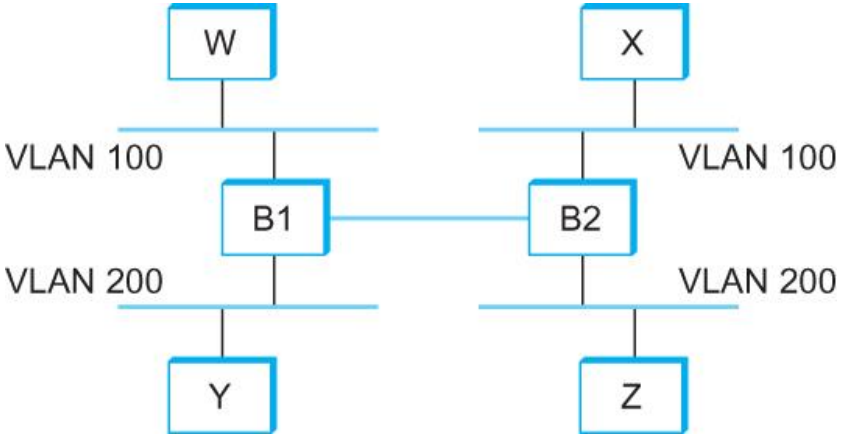
Spanning Tree Algorithm

- Limitation of Bridges
- Do not scale→ it is not realistic to connect more than a few LANs by means of bridges
  - Spanning tree algorithm does not scale
  - Broadcast does not scale

- Do not accommodate heterogeneity

Virtual LAN

- One approach to increasing the scalability of extended LANs is the *virtual LAN* (VLAN).
- VLANs allow a single extended LAN to be partitioned into several seemingly separate LANs

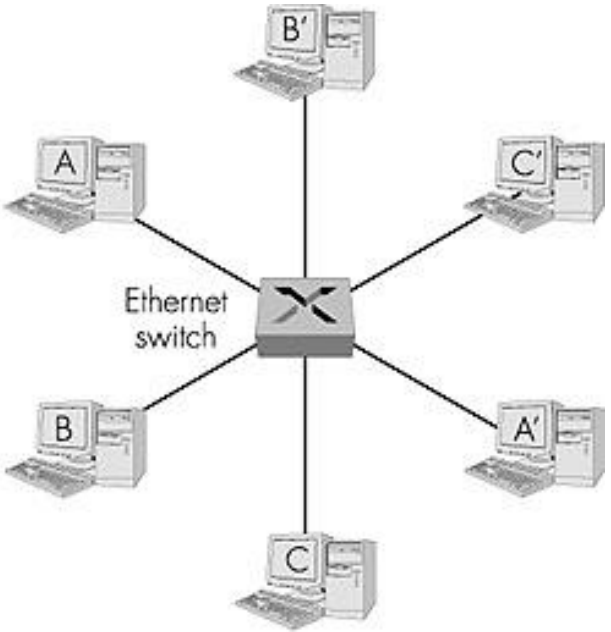


More Words about VLAN

- Virtual LAN (VLAN) – defined in IEEE 802.1q
  - Partition a physical LAN into several “logically separate” LANs
    - reduce broadcast traffic on physical LAN!
    - provide administrative isolation
  - Extend over a WAN (wide area network), e.g., via layer 2 tunnels (e.g., L2TP, MPLS) over IP-based WANs!
- Two types: port-based or MAC address-based
  - each port optionally configured with a VLAN id
  - inbound packets tagged with this “VLAN” id
    - require change of data frames, carry “VLAN id” tags
    - tagged and untagged frames can co-exist
  - “VLAN-aware” switches forward on ports part of same VLAN
- More complex ! - require administrative configuration
  - static (“manual”) configuration
  - Find more in the practical

Ethernet Switches

- Switches A switch is a combination of a hub and a bridge.
- It can interconnect two or more workstations, but like a bridge, it observes traffic flow and learns.
- When a frame arrives at a switch, the switch examines tl forwards the frame out the one necessary connection.
- Essentially a multi-interface bridge
- layer 2 (frame) forwarding, filtering using LAN addresses
- Switching: A-to-A' and B-to-B' simultaneously, no collisio
- large number of interfaces
- often: individual hosts, star-connected into switch
  - Ethernet, but no collisions!



- Major role: isolating traffic patterns and providing multiple access.
- This design is usually done by the network manager. Switches are easy to install and have components that are hotswappable.
- The backplane of a switch is fast enough to support multiple data transfers at one time.
- Multiple workstations connected to a switch use dedicated segments.
- This is a very efficient way to isolate heavy users from the network.

Ethernet Switches

- cut-through switching: frame forwarded from input to output port without awaiting for assembly of entire frame
  - slight reduction in latency
  - Cut-through vs. store and forward
- combinations of shared/dedicated, 10/100/1000 Mbps interfaces

- Switching and Forwarding
  - Generic Switch Architecture
  - Forwarding Tables:
    - Bridges/Layer 2 Switches; VLAN
    - Routers and Layer 3 Switches
- Forwarding in Layer 3 (Network Layer)
  - Network Layer Functions
  - Network Service Models: VC vs. Datagram
    - ATM and IP Datagram Forwarding

Hubs vs. Bridges vs. Routers

- Hubs (aka Repeaters): Layer 1 devices
  - repeat (i.e., regenerate) physical signals
    - don't understand MAC protocols!
    - LANs connected by hubs belong to same collision domain
- Bridges (and Layer-2 Switches): Layer 2 devices
  - store and forward layer-2 frames based on MAC addresses
    - speak and obey MAC protocols
    - bridges segregate LANs into different collision domains
- Routers (and Layer 3 Switches): Layer 3 devices
  - store and forward layer-3 packets based on network layer addresses (e.g., IP addresses)
    - rely on data link layer to deliver packets to (directly connected) next hop
    - network layer addresses are logical (i.e. virtual), need to map to MAC addresses for packet delivery

Forwarding in Layer 3

Putting in context

- What does layer-3 (network layer) do?
  - deliver packets “hop-by-hop” across a network
  - rely on layer-2 to deliver between neighboring hops
- Key Network Layer Functions
  - Addressing: need a global (logical) addressing scheme
  - Routing: build “map” of network, find routes, …
  - Forwarding: actual delivery of packets!
- Two basic network layer service models
  - datagram: “connectionless”
  - virtual circuit (VC): connection-oriented

Network Layer Functions

- Addressing
  - Globally unique address for each routable device
    - Logical address, unlike MAC address (as you've seen earlier)
  - Assigned by network operator
    - Need to map to MAC address (as you'll see later)
- Routing: building a "map" of network
  - Which path to use to forward packets from src to dest
- Forwarding: delivery of packets hop by hop
  - From input port to appropriate output port in a router

Routing and forwarding depend on network service models: *datagram* vs. *virtual circuit*

Virtual Circuit vs. Datagram

- Objective of both: move packets through routers from source to destination
- Datagram Model:
  - *Routing:* determine next hop to each destination a priori
  - *Forwarding:* destination address in packet header, used at each hop to look up for next hop
    - routes may change during "session"
  - analogy: driving, asking directions at every corner gas station, or based on the road signs at every turn
- Virtual Circuit Model:
  - *Routing:* determine a path from source to each destination
  - *"Call" Set-up:* fixed path ("virtual circuit") set up at "*call" setup time*, remains fixed thru "call"
  - *Data Forwarding:* each packet carries "tag" or "label" (virtual circuit id, VCI), which determines next hop
  - *routers maintain "per-call" state*

Datagram

- No connection setup phase
- Each packet forwarded independently
- Sometimes called *connectionless* model
- The idea behind datagrams is incredibly simple
  - just include in every packet enough information i.e. a complete destination address
- Each switch maintains a forwarding (routing) table, to decide how to forward the packet to its destination.

Analogy: postal system

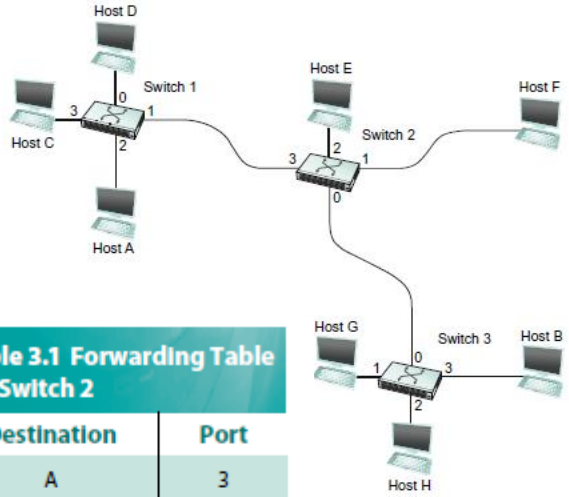


Table 3.1 Forwarding Table for Switch 2

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

Characteristics of Connectionless (Datagram) Network

- A host can send a packet anywhere at any time, since any packet that turns up at the switch can be immediately forwarded (assuming a correctly populated forwarding table)
- When a host sends a packet, it has no way of knowing if the network is capable of delivering it or if the destination host is even up and running
- Each packet is forwarded independently of previous packets that might have been sent to the same destination.
  - Thus two successive packets from host A to host B may follow completely different paths
- A switch or link failure might not have any serious effect on communication if it is possible to find an alternate route around the failure and update the forwarding table accordingly

Datagram (Summary)

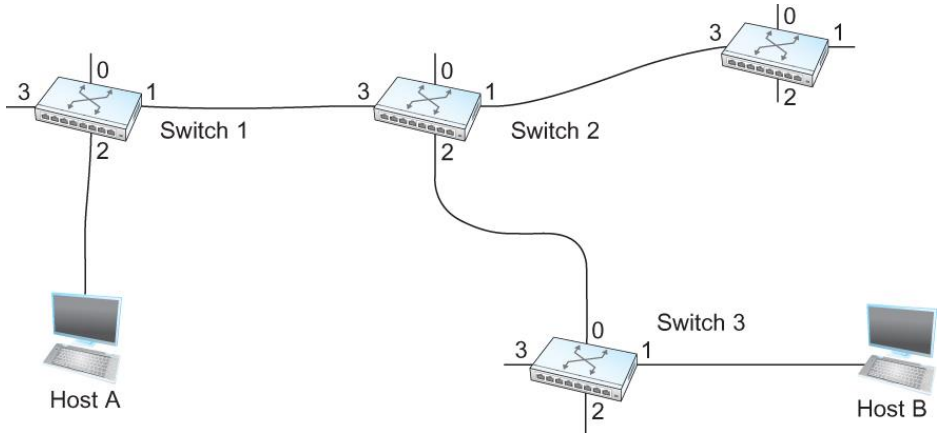
- There is no round trip delay waiting for connection setup; a host can send data as soon as it is ready.
- Source host has no way of knowing if the network is capable of delivering a packet or if the destination host is even up.
- Since packets are treated independently, it is possible to route around 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.

Virtual Circuit Switching

- Widely used technique for packet switching
- Uses the concept of *virtual circuit* (VC)→ Subsequence packets follow same circuit
- Explicit connection setup (and tear-down) phase
- Sometimes called *connection-oriented* model

  still packet switching, not circuit switching!

- Each switch maintains a VC table



Host A wants to send packets to host B. How?

Virtual Circuit Switching

Two-stage process

- Connection setup
- Data Transfer

- Connection setup
  - Establish “connection state” in each of the switches between the source and destination hosts
  - The connection state for a single connection consists of an entry in the “VC table” in each switch through which the connection passes

Virtual Circuit Switching

Two broad classes of approach to establishing connection state

- Network Administrator will configure the state
  - The virtual circuit is permanent (PVC)
  - The network administrator can delete this
  - Can be thought of as a long-lived or administratively configured VC
- A host can send messages into the network to cause the state to be established
  - This is referred as signalling and the resulting virtual circuit is said to be switched (SVC)
  - A host may set up and delete such a VC dynamically without the involvement of a network administrator

Virtual Circuit Switching

One entry in the VC table on a single switch contains

- A virtual circuit identifier (VCI) that uniquely identifies the connection at this switch and that will be carried inside the header of the packets that belong to this connection
- An incoming interface on which packets for this VC arrive at the switch
- An outgoing interface in which packets for this VC leave the switch
- A potentially different VCI that will be used for outgoing packets

- The semantics for one such entry is
  - If a packet arrives on the designated incoming interface and that packet contains the designated VCI value in its header, then the packet should be sent out the specified outgoing interface with the specified outgoing VCI value first having been placed in its header

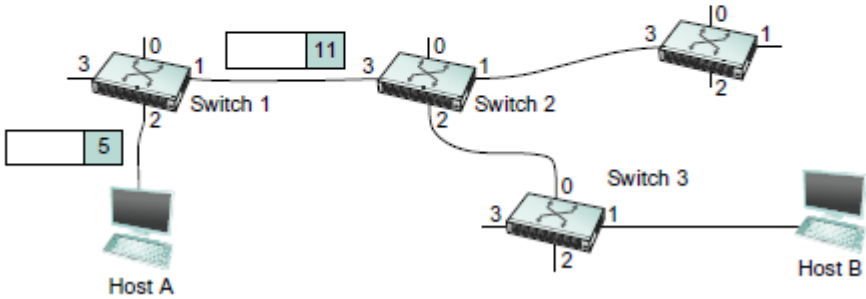
Virtual Circuit Switching

Let's assume that a network administrator wants to manually create a new virtual connection from host A to host B

- First the administrator identifies a path through the network from A to B
- The administrator then picks a VCI value that is currently unused on each link for the connection
- For our example,
  - Suppose the VCI value 5 is chosen for the link from host A to switch 1
  - 11 is chosen for the link from switch 1 to switch 2
  - So the switch 1 will have an entry in the VC table

Table 3.2 Virtual Circuit Table Entry for Switch 1

Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
2	5	1	11



Virtual Circuit Switching

Similarly, suppose

- VCI of 7 is chosen to identify this connection on the link from switch 2 to switch 3
- VCI of 4 is chosen for the link from switch 3 to host B
- Switches 2 and 3 are configured with the following VC table

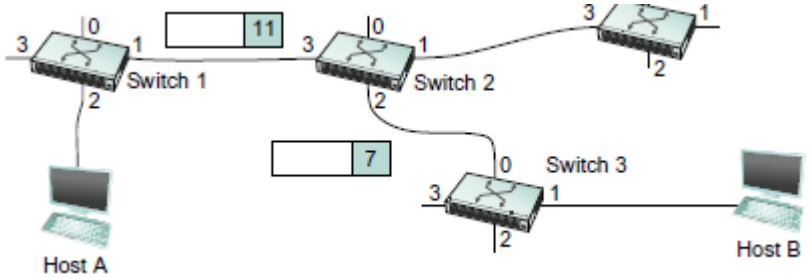


Table 3.3 Virtual Circuit Table Entries for Switches 2 and 3

VC Table Entry at Switch 2

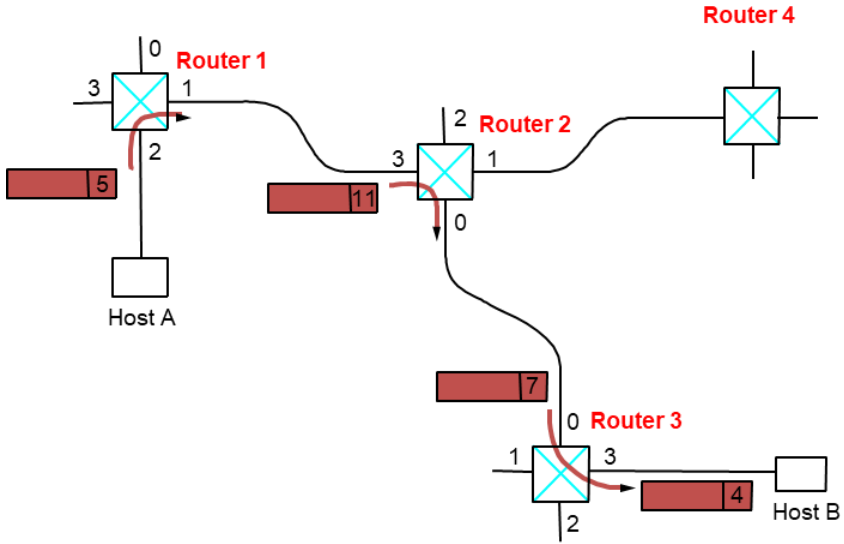
Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
3	11	2	7

VC Table Entry at Switch 3

Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
0	7	1	4

Virtual Circuit Switching

- For any packet that A wants to send to B, A puts the VCI value 5 in the header of the packet and sends it to switch 1
- Switch 1 receives any such packet on interface 2, and it uses the combination of the interface and the VCI in the packet header to find the appropriate VC table entry.
- The table entry on switch 1 tells the switch to forward the packet out of interface 1 and to put the VCI value 11 in the header
- Packet will arrive at switch 2 on interface 3 bearing VCI 11
- Switch 2 looks up interface 3 and VCI 11 in its VC table and sends the packet on to switch 3 after updating the VCI value appropriately
- This process continues until it arrives at host B with the VCI value of 4 in the packet
- To host B, this identifies the packet as having come from host A



"call" from host A to host B along path:
host A→ router 1→ router 2 → router 3 → host B

Virtual Circuit Switching

- In real networks of reasonable size, the burden of configuring VC tables correctly in a large number of switches would quickly become excessive
  - Thus, some sort of signalling is almost always used, even when setting up “permanent” VCs
  - In case of PVCs, signalling is initiated by the network administrator
  - SVCs are usually set up using signalling by one of the hosts
- How does the signalling work
  - To start the signalling process, host A sends a setup message into the network (i.e. to switch 1)
    - The setup message contains (among other things) the complete destination address of B.
    - The setup message needs to get all the way to B to create the necessary connection state in every switch along the way
    - It is like sending a datagram to B where every switch knows which output to send the setup message so that it eventually reaches B
    - Assume that every switch knows the topology to figure out how to do that
  - When switch 1 receives the connection request, in addition to sending it on to switch 2, it creates a new entry in its VC table for this new connection
    - The entry is exactly the same shown in the previous table
    - Switch 1 picks the value 5 for this connection

Virtual Circuit Switching

- Now to complete the connection, everyone needs to be told what their downstream neighbor is using as the VCI for this connection
  - Host B sends an acknowledgement of the connection setup to switch 3 and includes in that message the VCI value that it chose (4)
  - Switch 3 completes the VC table entry for this connection and sends the acknowledgement on to switch 2 specifying the VCI of 7
  - Switch 2 completes the VC table entry for this connection and sends acknowledgement on to switch 1 specifying the VCI of 11
  - Finally switch 1 passes the acknowledgement on to host A telling it to use the VCI value of 5 for this connection
- When host A no longer wants to send data to host B, it tears down the connection by sending a teardown message to switch 1
- The switch 1 removes the relevant entry from its table and forwards the message on to the other switches in the path which similarly delete the appropriate table entries
- At this point, if host A were to send a packet with a VCI of 5 to switch 1, it would be dropped as if the connection had never existed

Virtual Circuit Switching

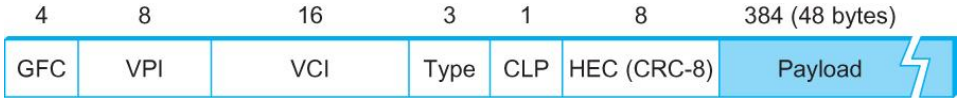
- Characteristics of VC
  - Since host A has to wait for the connection request to reach the far side of the network and return before it can send its first data packet, there is at least one RTT of delay before data is sent
  - While the connection request contains the full address for host B (which might be quite large, being a global identifier on the network), each data packet contains only a small identifier, which is only unique on one link.
    - Thus the per-packet overhead caused by the header is reduced relative to the datagram model
  - If a switch or a link in a connection fails, the connection is broken and a new one will need to be established.
    - Also the old one needs to be torn down to free up table storage space in the switches
  - The issue of how a switch decides which link to forward the connection request on has similarities with the function of a routing algorithm
- Good Properties of VC
  - By the time the host gets the go-ahead to send data, it knows quite a lot about the network-
    - For example, that there is really a route to the receiver and that the receiver is willing to receive data
  - It is also possible to allocate resources to the virtual circuit at the time it is established

Virtual Circuit Switching

- In VC, we could imagine providing each circuit with a different quality of service (QoS)
  - The network gives the user some kind of performance related guarantee
    - Switches set aside the resources they need to meet this guarantee
      - For example, a percentage of each outgoing link's bandwidth
      - Delay tolerance on each switch
- Most popular examples of VC technologies are X.25, Frame Relay and ATM
  - One of the applications of Frame Relay is the construction of VPN
- X.25 is an old standard protocol for connection-oriented packet-switched network (used in old telecommunication companies and Automated Teller Machines (ATM's). This employs the following three-part strategy:
  - Buffers are allocated to each virtual circuit when the circuit is initialized
  - The sliding window protocol is run between each pair of nodes along the virtual circuit, and this protocol is augmented with the flow control to keep the sending node from overrunning the buffers allocated at the receiving node
  - The circuit is rejected by a given node if not enough buffers are available at that node when the connection request message is processed

Virtual Circuit Switching

- ATM (Asynchronous Transfer Mode)
  - Connection-oriented packet-switched network
  - Packets are called cells
    - 5 byte header + 48 byte payload
  - Fixed length packets are easier to switch in hardware
    - Simpler to design
    - Enables parallelism

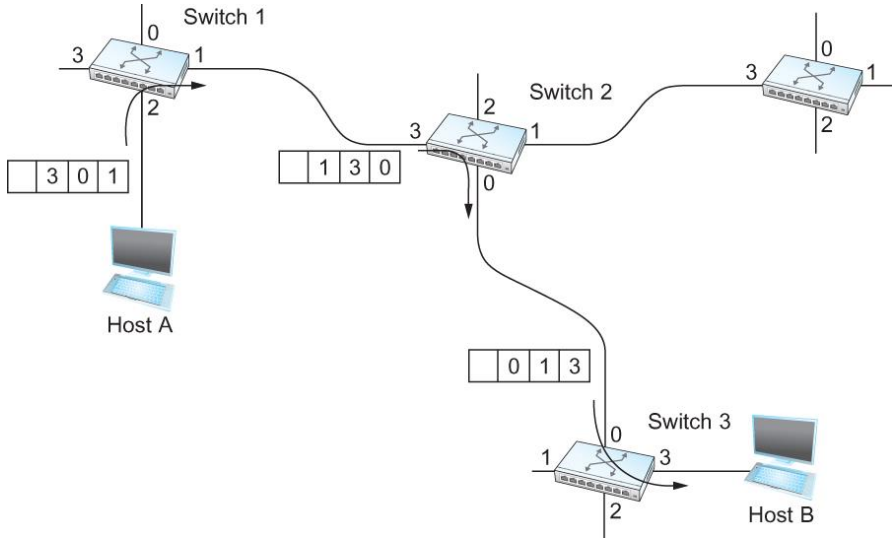


- Host-to-switch format
- GFC: Generic Flow Control
- VCI: Virtual Circuit Identifier
- Type: management, congestion control
- CLP: Cell Loss Priority
- HEC: Header Error Check (CRC-8)

Switching and Forwarding

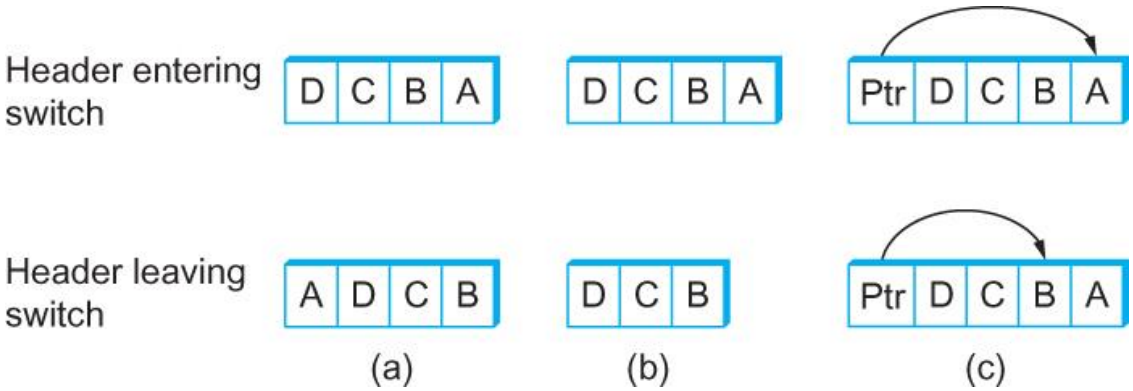
Source Routing

- Source Routing
  - Is the third approach to switching that uses neither virtual circuits nor conventional datagrams
  - All the information about network topology that is required to switch a packet across the network is provided by the source host



Switching and Forwarding

- Other approaches in Source Routing



(a) rotation; (b) stripping; (c) pointer.
The labels are read right to left.