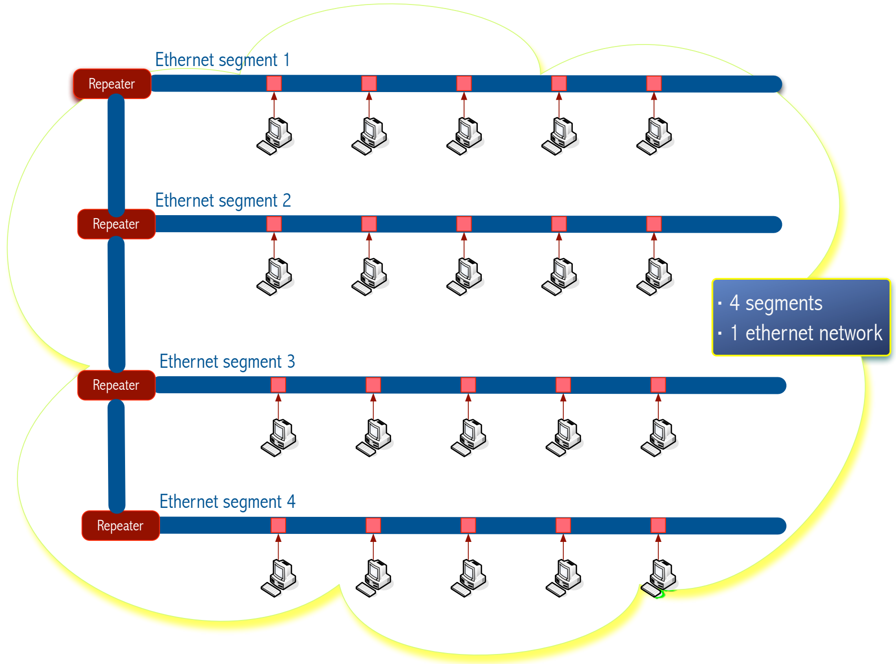
CH. 3
ETHERNET SWITCHING

V 1.0.2 8th/May/2019

Ethernet LAN summary

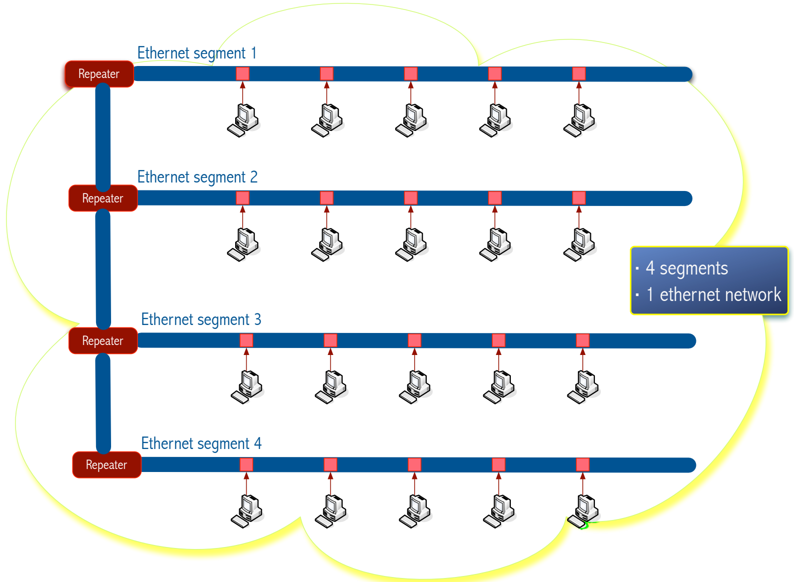
Limitations of a maximally configured bus Ethernet:

- No more than four repeaters between any pair of hosts.
- An Ethernet has a total reach of only **2500 m**
- Bandwidth (10 Mbps) is shared among all the stations



Conceptual Ethernet

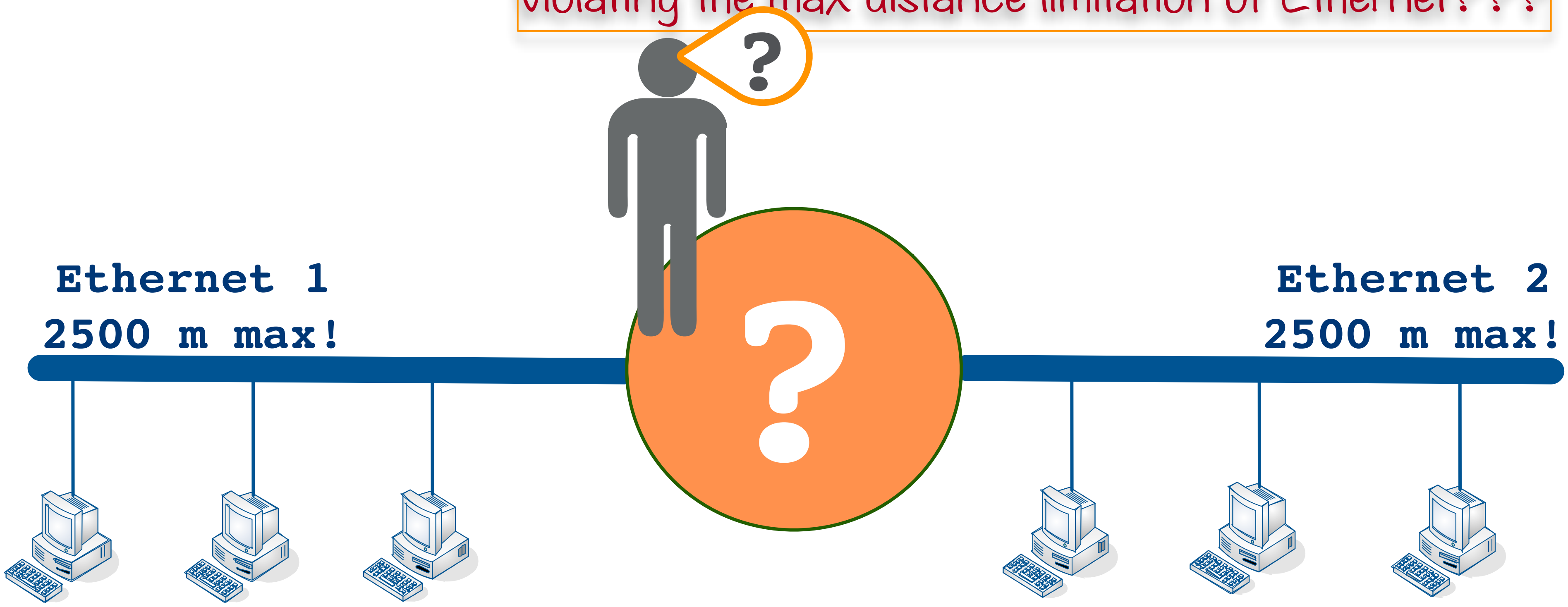
- Shared media:
  - Inherently BROADCAST
  - Every frame is delivered to all hosts, inevitably
    - Coax: Bus topology
    - Hub: Star topology



- Half-duplex
  - Only one flow active at a time

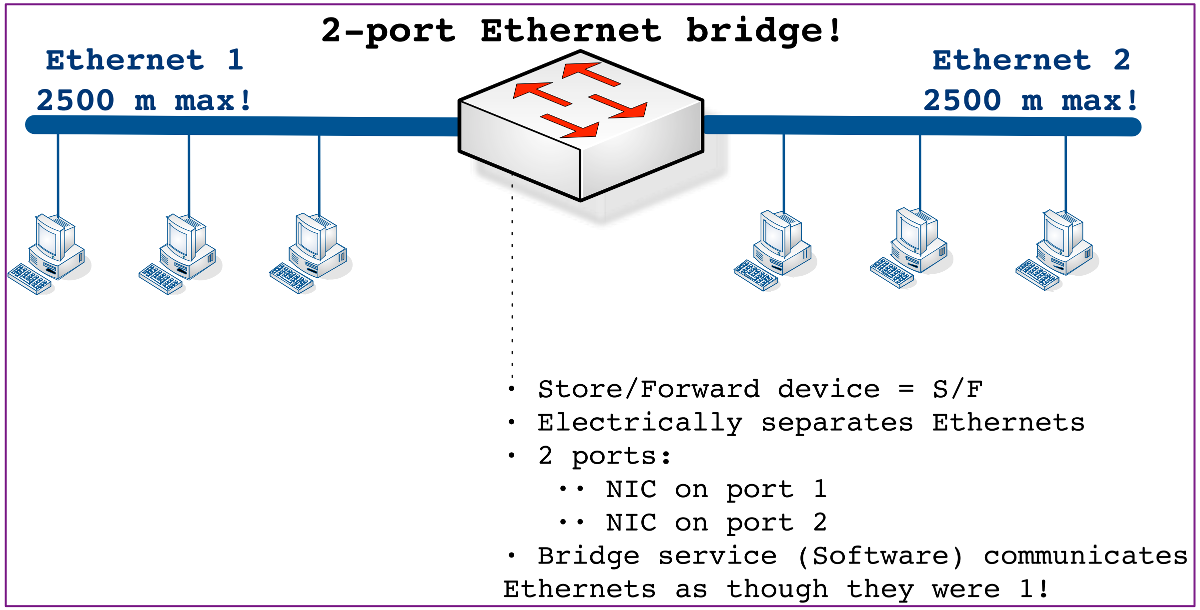
Communicating two max Ethernets

What network device can communicate E1 with E2
without
violating the max distance limitation of Ethernet???



Connecting two Ethernets: Bridge

- A) Repeater in between them?
  - It might exceed the physical limitation of the Ethernet
- B) Hubs regenerate **electrical** signals
  - Hubs are layer-1 devices (OSI)
- C) Bridge? New network equipment that forwards **frames** between two LANs
  - Bridges/switches are layer-2 devices (OSI)

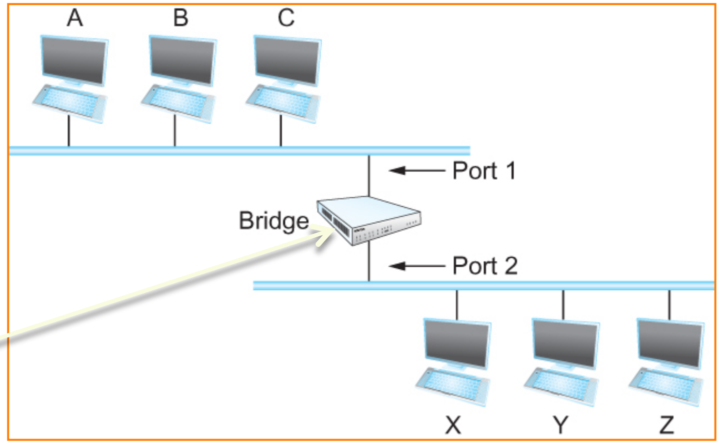


Extended LAN, bridge or switch

- A bridge is a store-and-forward device

bridge?

- The *no-frills* bridge (simplest, oldest, not used today)
  - Each frame received on a port is forwarded to all its other ports
  - Old
  - CNPro!



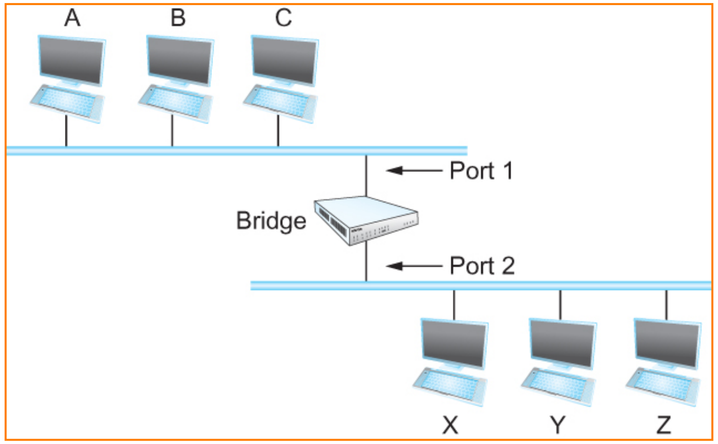
- Learning Bridge
  - Learn MAC addresses as nodes send traffic
  - Have a Station cache or Forwarding Table
    - It contains a MAC – Port table
    - Station sends a frame onto the network for the first time
    - Switch records its source MAC and the port number it was received onto

Extended LAN, bridge or switch

- Basic functioning of a bridge

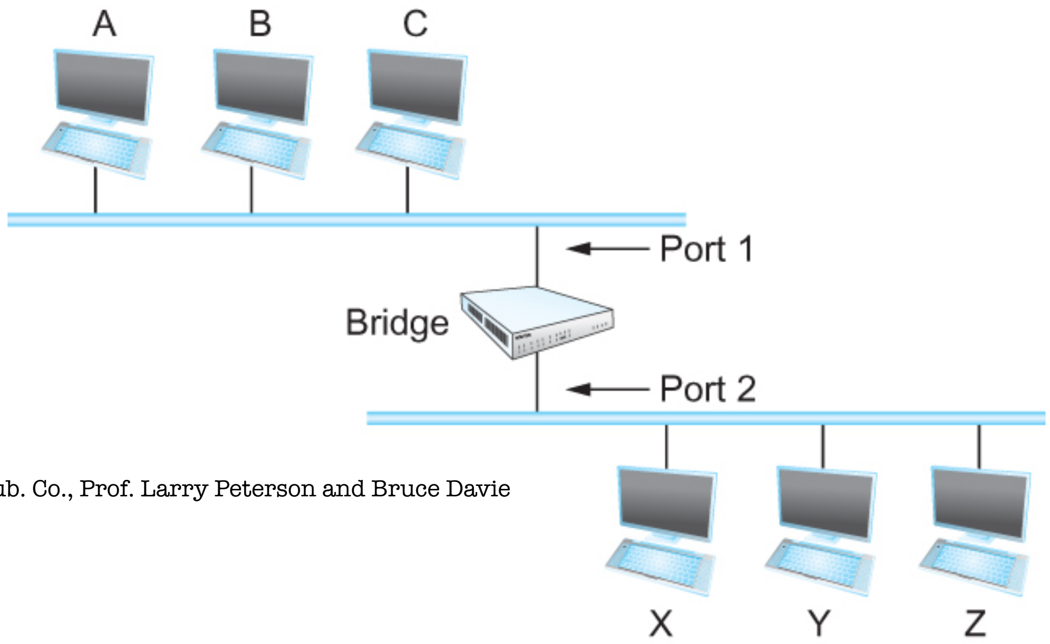
1. Receive a frame on a port and store it into the incoming frame buffer
2. Consult forwarding table
   1. Record the source MAC address into de forwarding table
   2. If destination MAC belongs to the another port, send it onto that port when possible
   3. If destination MAC belongs to receiving port, do nothing
   4. If destination MAC has not been recorded into the forwarding table yet, flood the frame (Send it onto all ports except the one it was received onto)

bridge?



Learning and forwarding examples

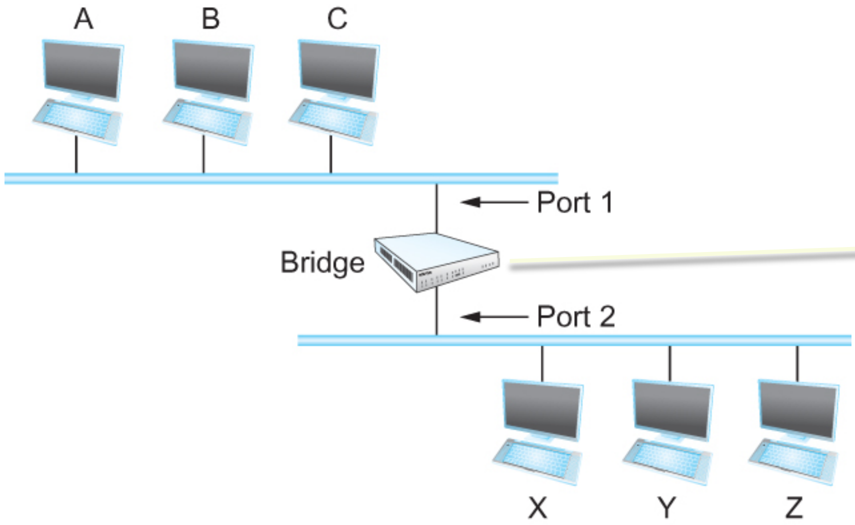
- Frame from host A to host B arrives on port 1
  - No need for the bridge to forward the frame out onto port 2



  - How does a bridge come to learn on which port each host resides?

Learning and forwarding examples

- Learning on which **port** each host resides?
  - *Download a table into the bridge ☺ NO!* (Too much maintenance)
  - Record new source MAC A into the Forwarding Table when host A sends its first frame

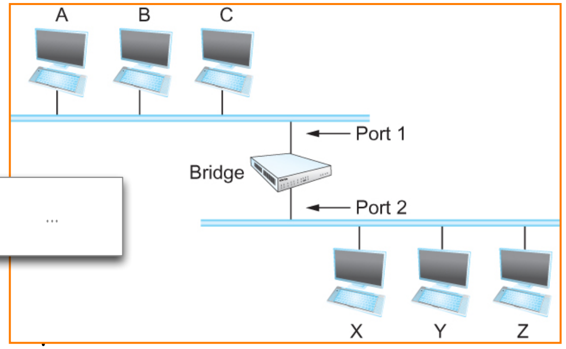


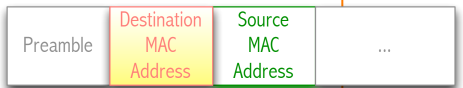
Forwarding table

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

Learning and forwarding examples

- Can the bridge *learn* this information by itself?
  - Yes: this is the *learning bridge*





- Here's how:
  - A bridge inspects the source MAC address in every Ethernet frame it receives
  - Record that information into the forwarding table (FT)
  - When a bridge first boots, this table is **empty**
  - Entries are added over time as hosts inject frames into their ports
    - A timeout is associated with each entry (aging)
    - The bridge discards the entry after a specified period of time
    - It server 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 a host not currently in the table
    - Send the frame onto all *other* ports (Not on the one it was received on): flooding

Forwarding Table

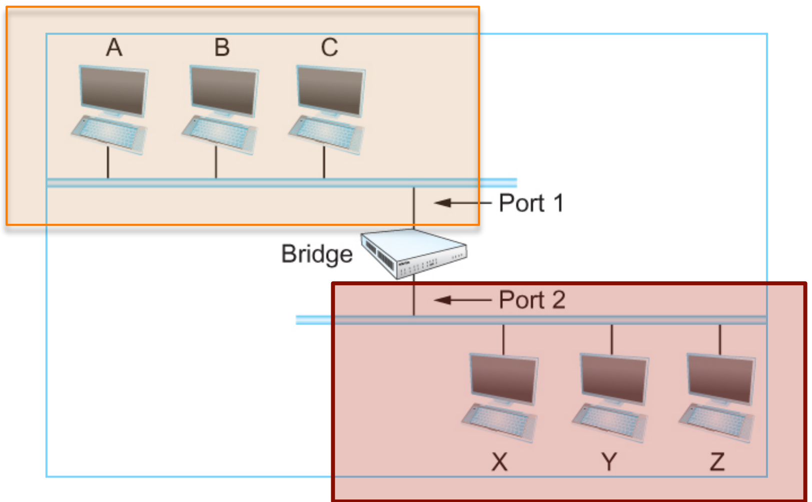
```
Host
MAC Port
--------
A     1
B     1
C     1
X     2
Y     2
Z     2
```

Extended LAN domains

- Can A collide with B? Yes, it can since A and B are connected to the *same* Ethernet segment
- Can A collide with X? No, since A and X belong to different Ethernet segments

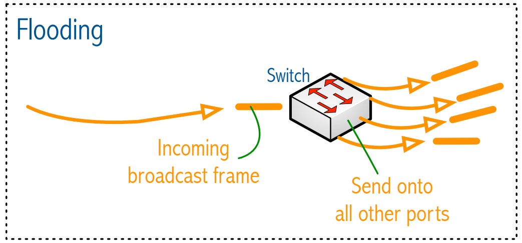
- There exist TWO segments or collision domains
  - A, B, C and bridge port 1
  - X, Y, Z and bridge port 2

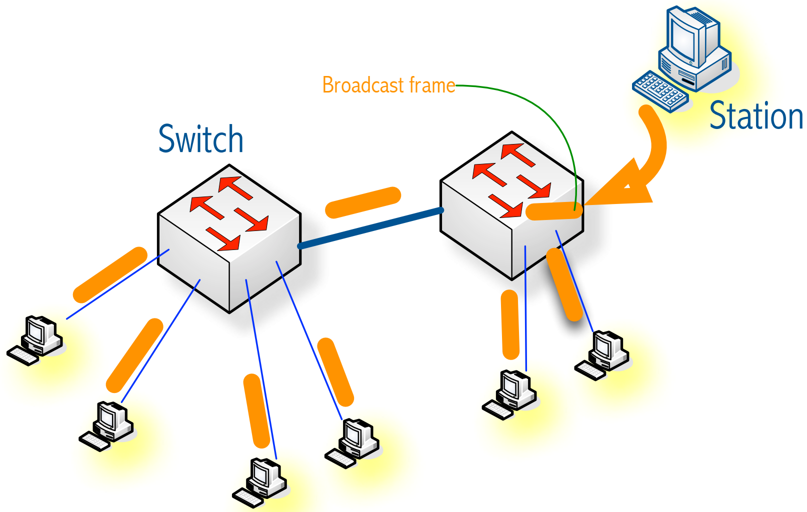
- HOWEVER, there is only one Extended LAN (Network)
  - When a broadcast frame is sent, it is received by all network hosts, we say that it contains a single BROADCAST DOMAIN



Ethernet vs. Switched Ethernet

- Shared Ethernet:
  - Inherently BROADCAST
  - Every frame is delivered to all hosts, inevitably
  - Half-duplex
  - Only one flow active at a time
    - Bus topology and Star topology (hub)

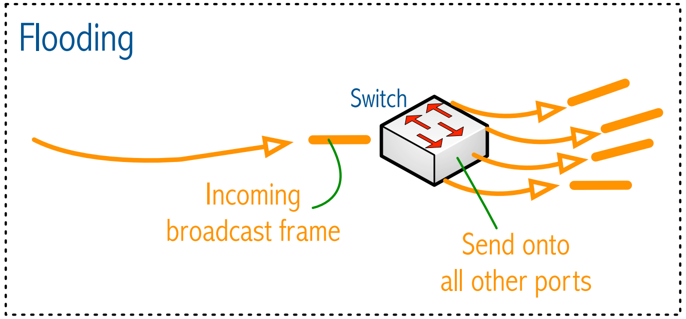


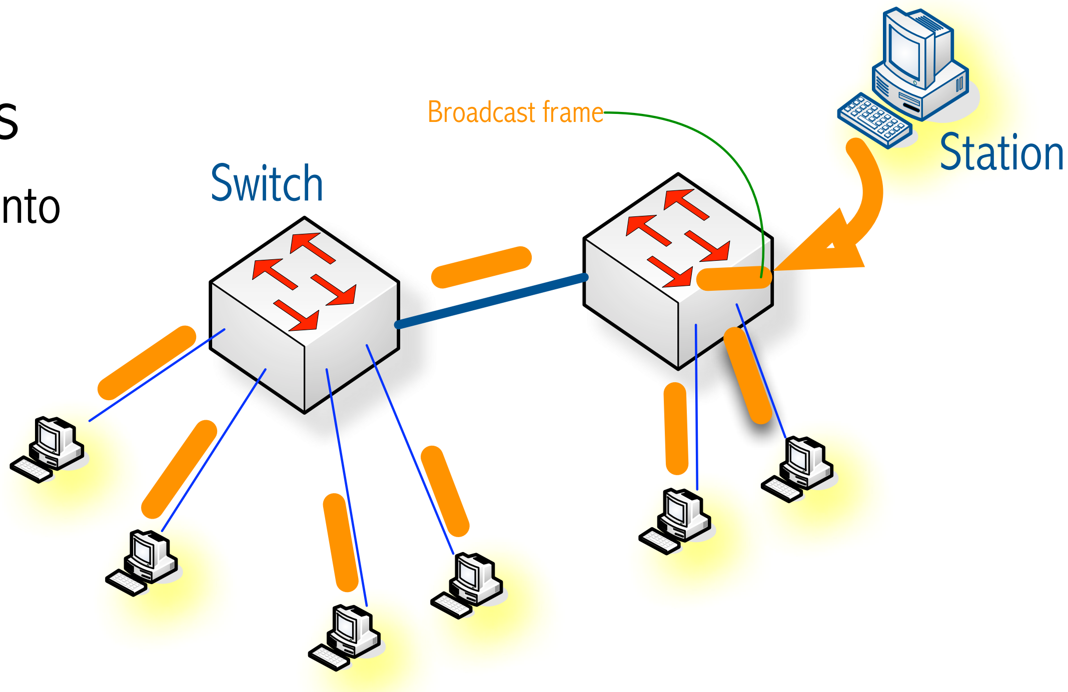


- Switched Ethernet:
  - An Extended LAN based on the interconnection of LAN segments by using bridges and switches
  - BROADCAST is possible but not inherent to the technology, how?
  - Full-duplex
  - Several simultaneous communication paths (Flows) active
    - Star topology, only

Switches do support broadcast

- As usual, a frame can be addressed to the **broadcast** address
- The switch will forward a broadcast frame to all ports
  - Except the port it was received onto
  - Known as **Flooding**





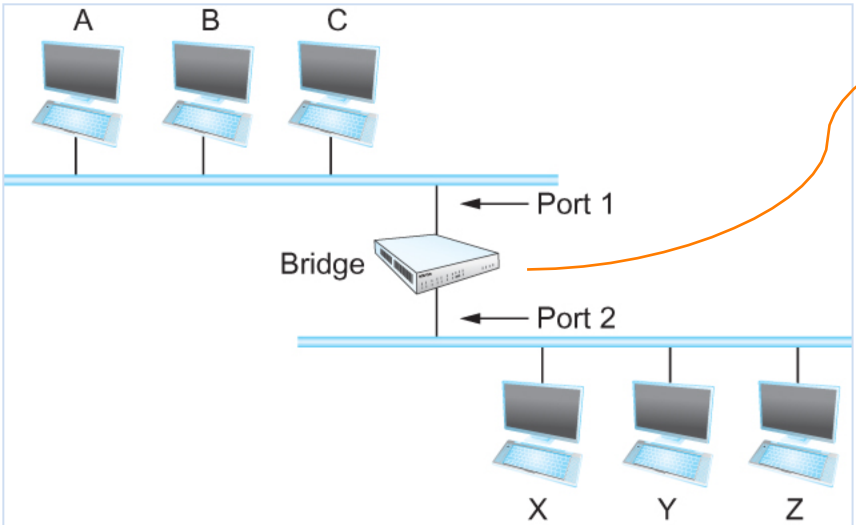
14

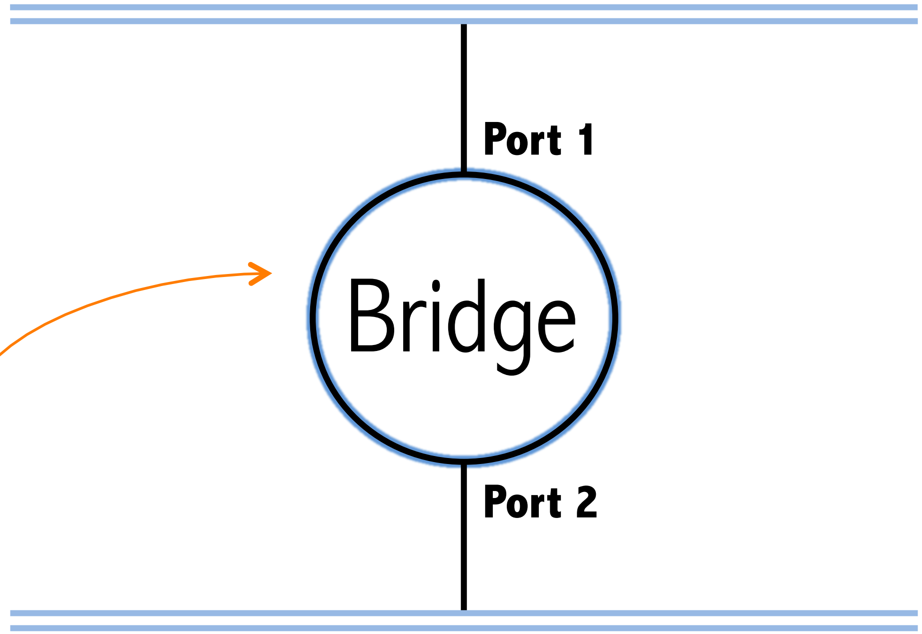
Switched, Extended LANs

More scalable Ethernets

Switched Extended LANs

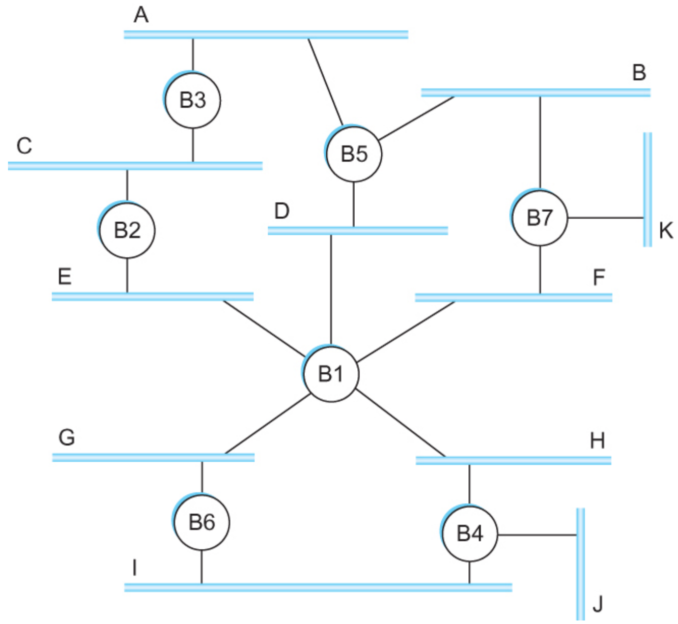
- Our abstract representation of a 2-port bridge





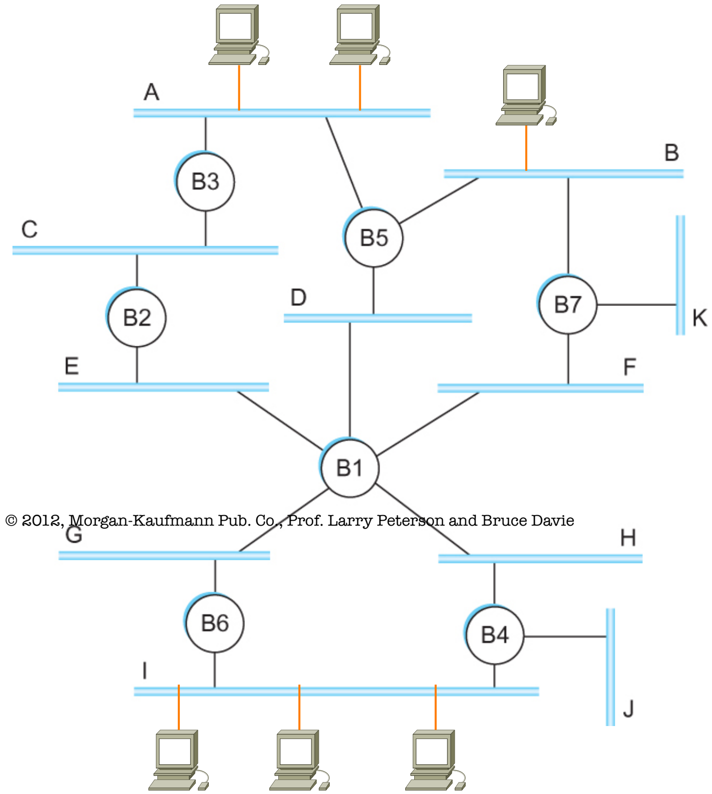
Switched Extended LANs

- Example:
  - What is this? A network: <u>A single network</u>
  - B1, B2 … = Bridges (Switches)
  - A, B, C, D, E … = Several LAN segments (Several Collision domains). Recall: shared media
  - Where are the stations (hosts)?
    - Each station is connected to a segment
  - Only one broadcast domain



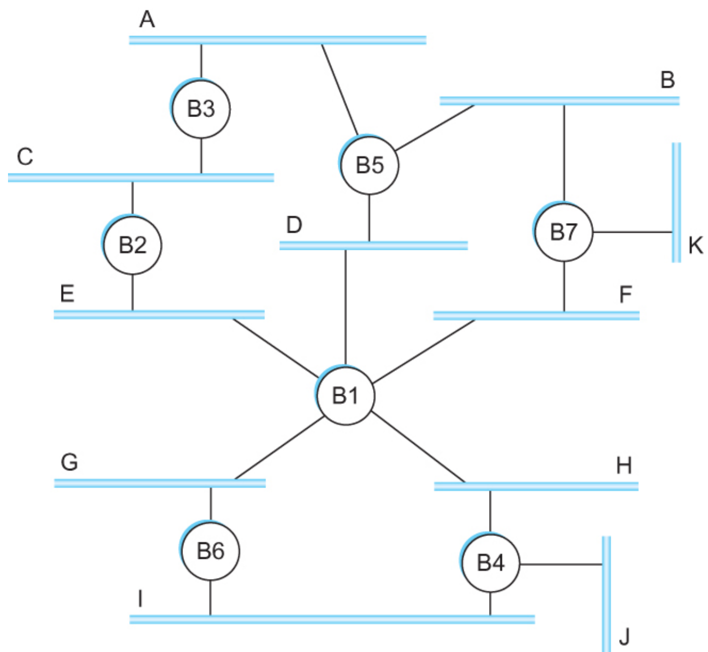
Switched Extended LANs

- Example:
  - What is this? A network: ONE network
  - B1, B2 … = Bridges
    - Remember: basically a switch
  - A, B, C: LAN segments (Collision domains)
    - Several collision domains
    - One broadcast domain: *The extended LAN*
  - Where are the end-nodes (hosts)?
    - Each end-host is connected to a segment



Switched Extended LANs

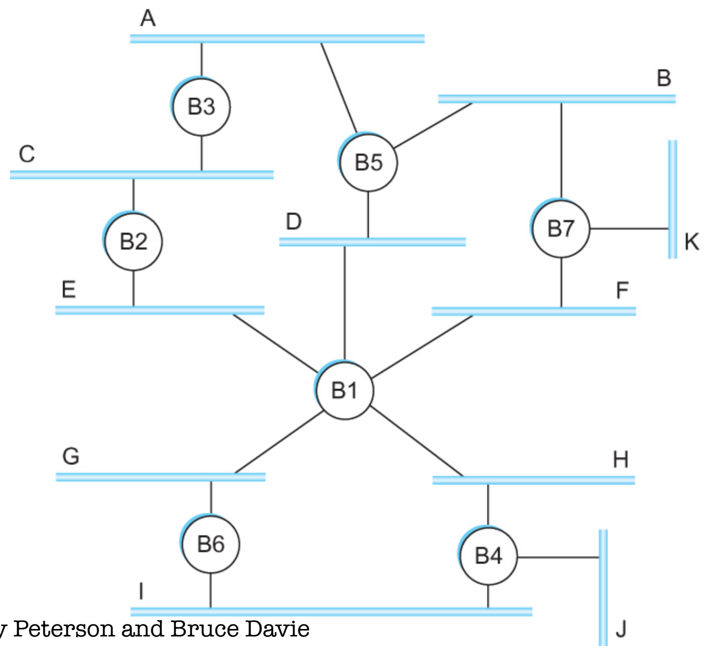
- Example:
  - What is this? A network: ONE network
  - B1, B2 … = Bridges
    - Remember: basically a switch
  - A, B, C: LAN segments (Collision domains)
    - Several collision domains
    - One broadcast domain: *The extended LAN*
  - Where are the end-nodes (hosts)?
    - Each end-node is connected to a segment



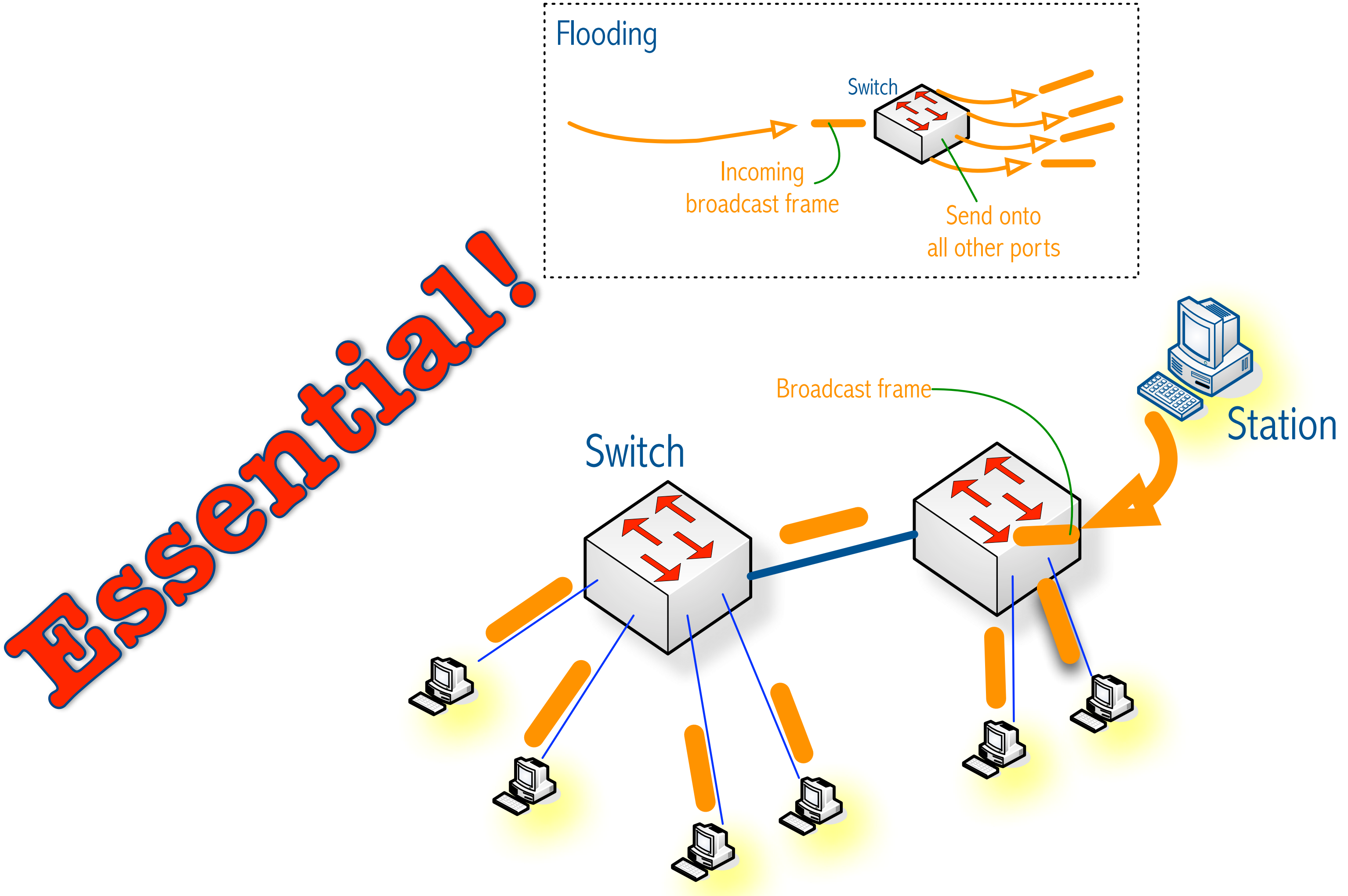
The switched LAN and loops

- How does an extended LAN come to have a **loop** in it?
  - Managed by more than one administrator
  - **Loops** provide **redundancy** in case of failures: good
  - Loops cause trouble with broadcast traffic

- Solution:
  - Detect loops
  - *Logically* open loops by disabling some bridge ports
  - Turn the network graph into a tree
  - **Spanning Tree Algorithm (ST)**

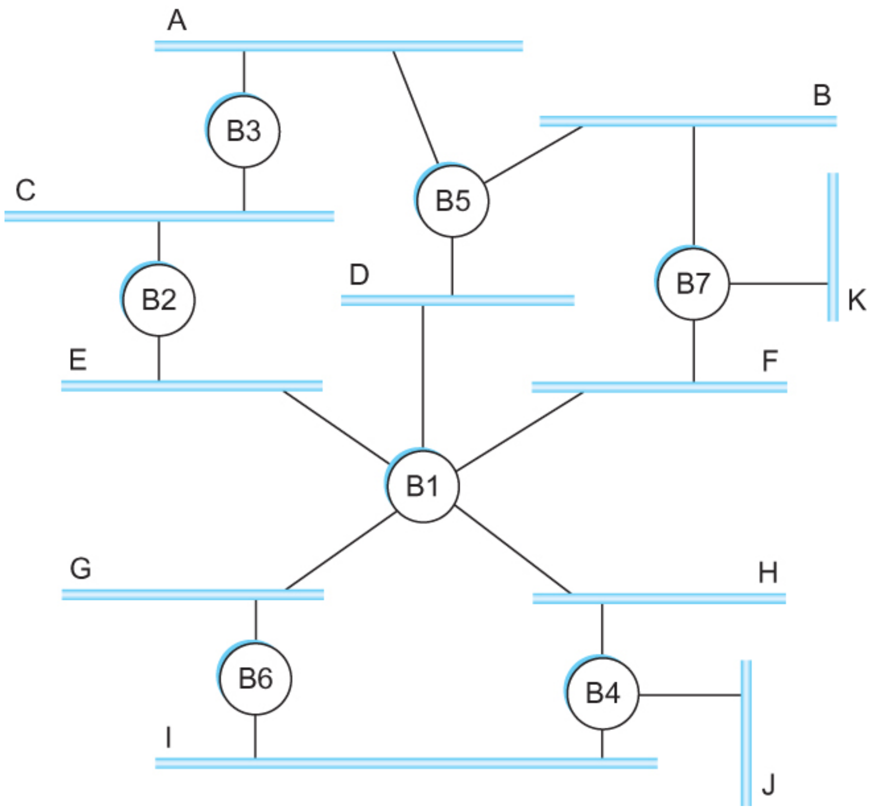


Recall: Switches flood broadcast traffic



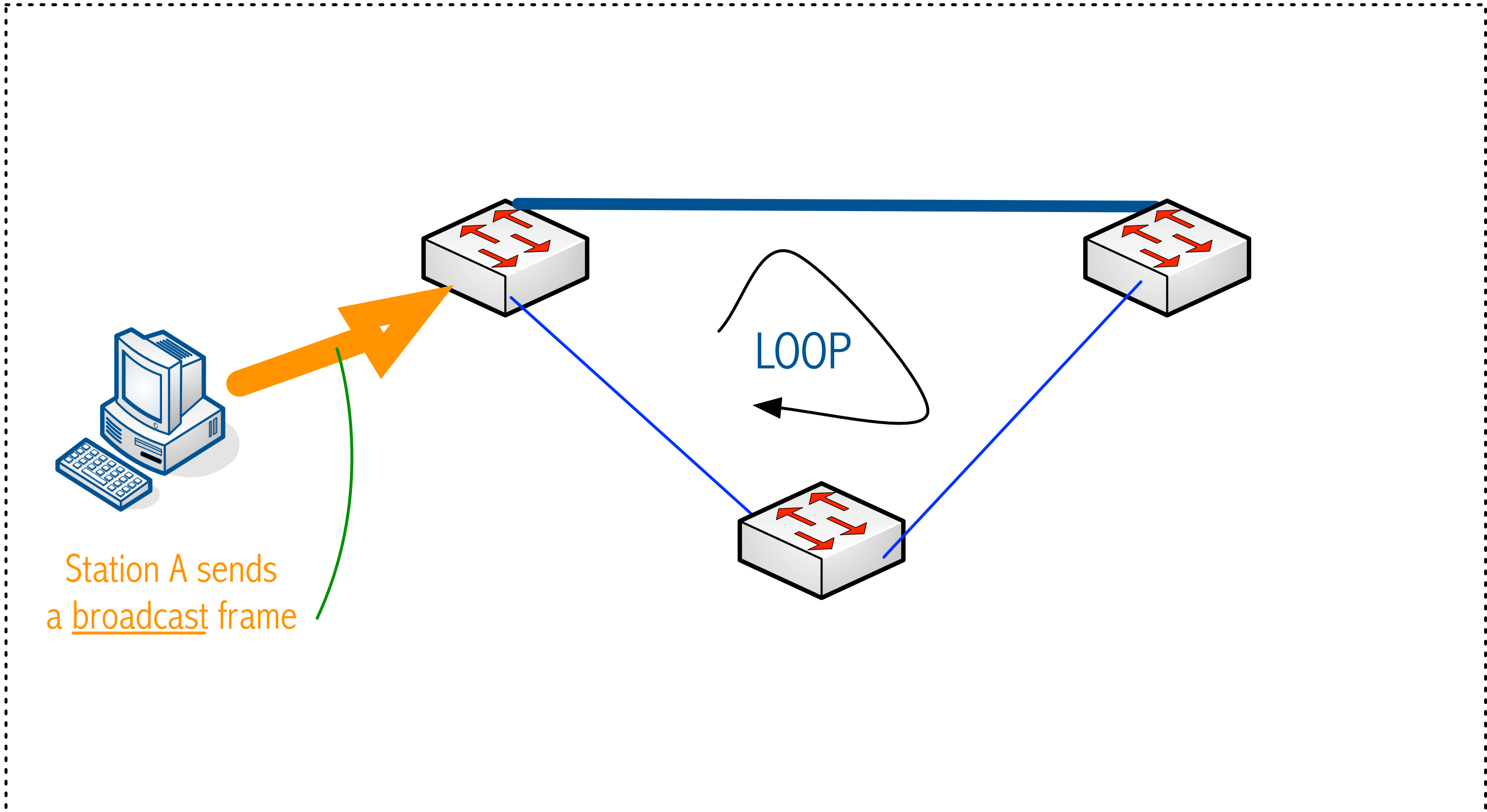
Switched LANs and loops

- Broadcast frames loop through the extended LAN forever
- B1, B4 and B6 form a loop



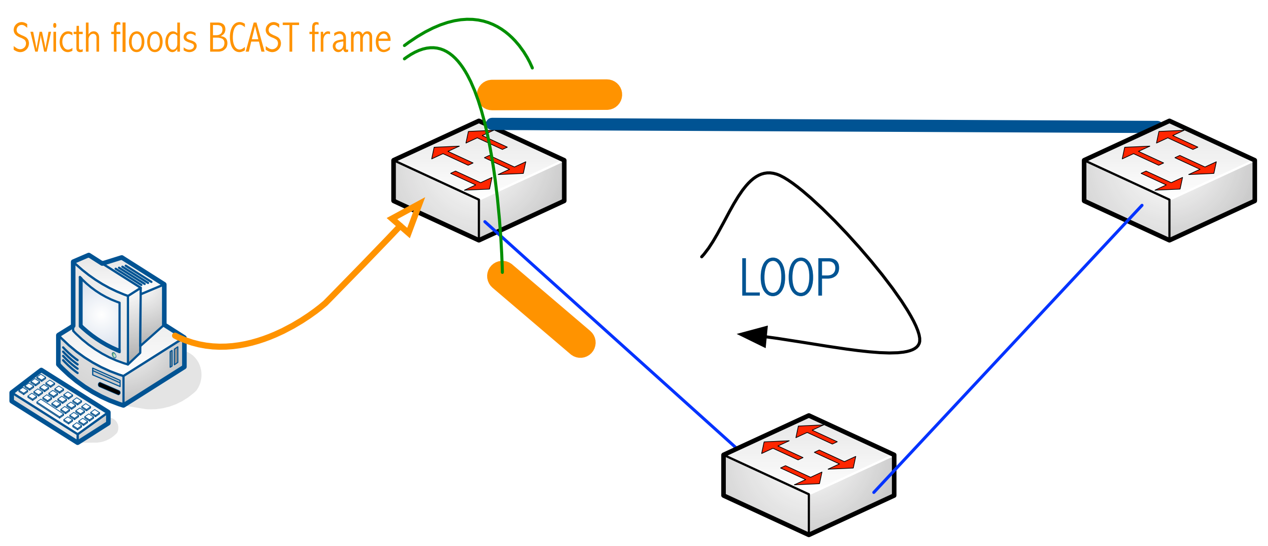
Switched LANs, broadcast and loops

- Station A sends a BROADCAST frame



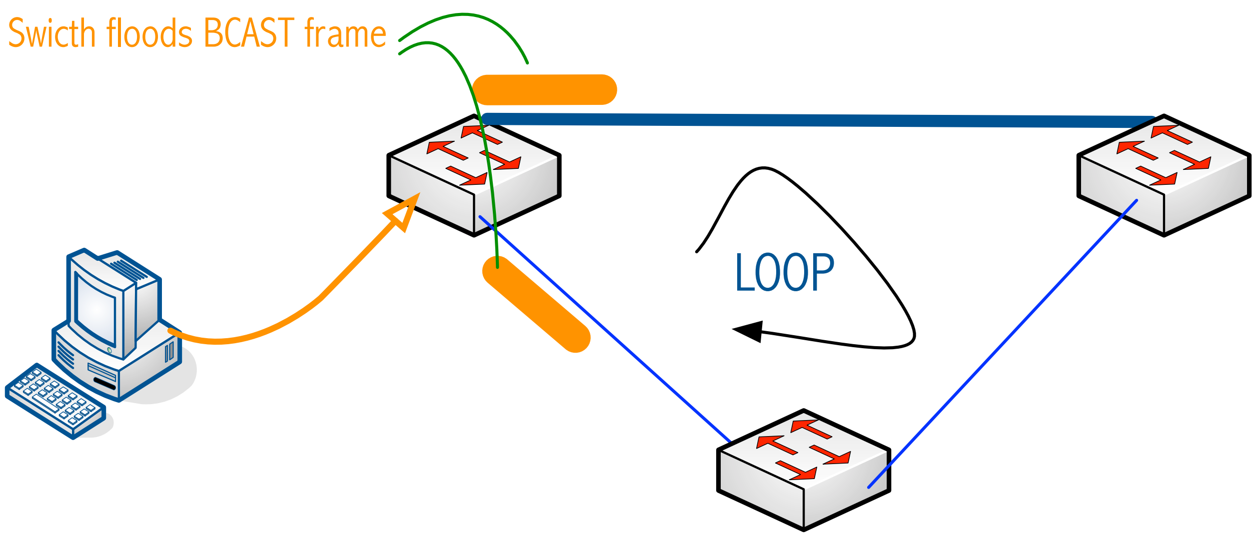
Switched LANs, broadcast and loops

- BCAST frame ingresses in switch
  - Switch will <u>flood</u> it: send it over all ports except the port over which it was received



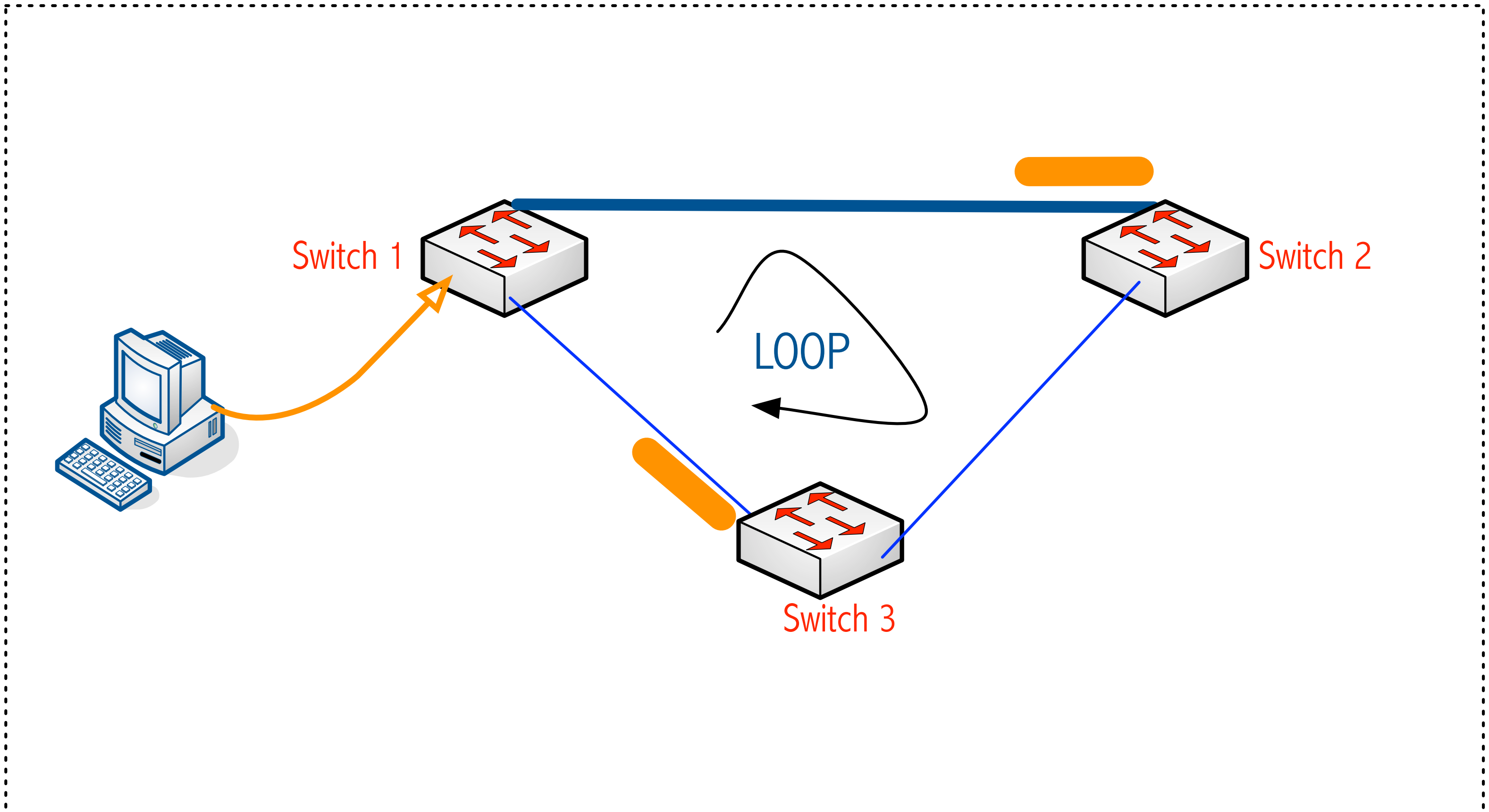
Switched LANs, broadcast and loops

- Switch 1 floods BCAST frame



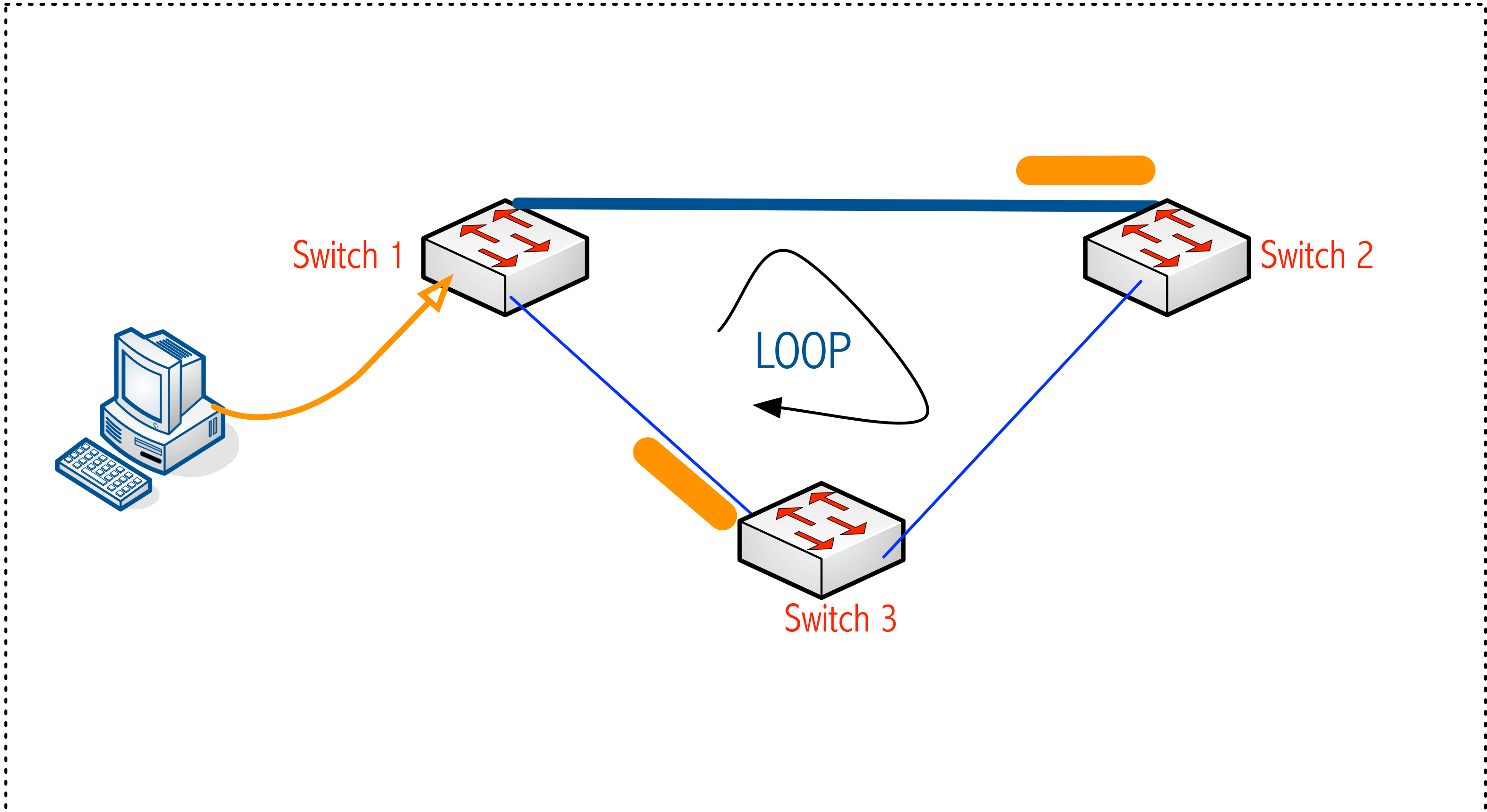
Switched LANs, broadcast and loops

- BCAST frame is delivered to Switch 2 and Switch 3



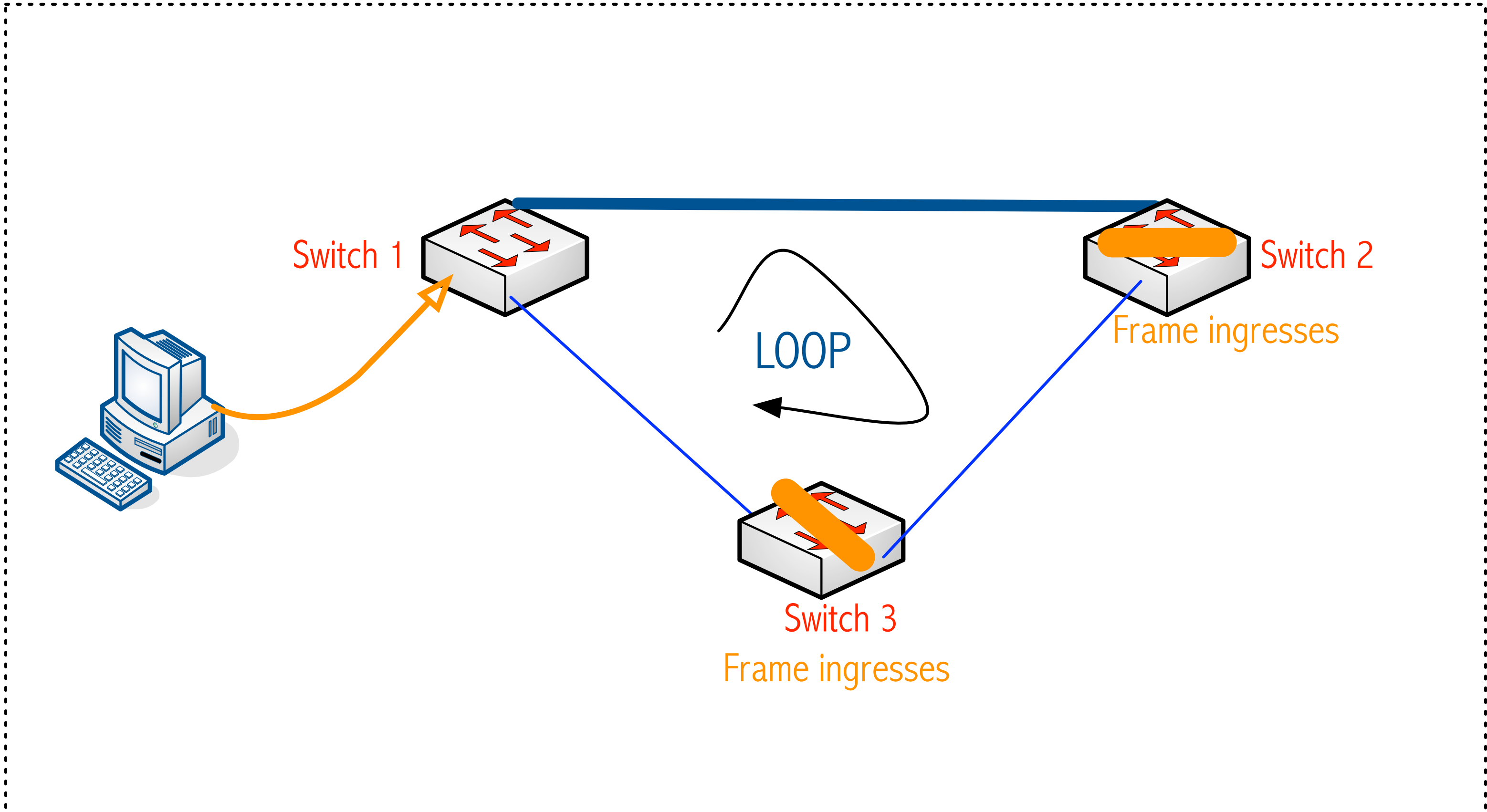
Switched LANs, broadcast and loops

- BCAST frame ingresses into Switch 2 and Switch 3



Switched LANs, broadcast and loops

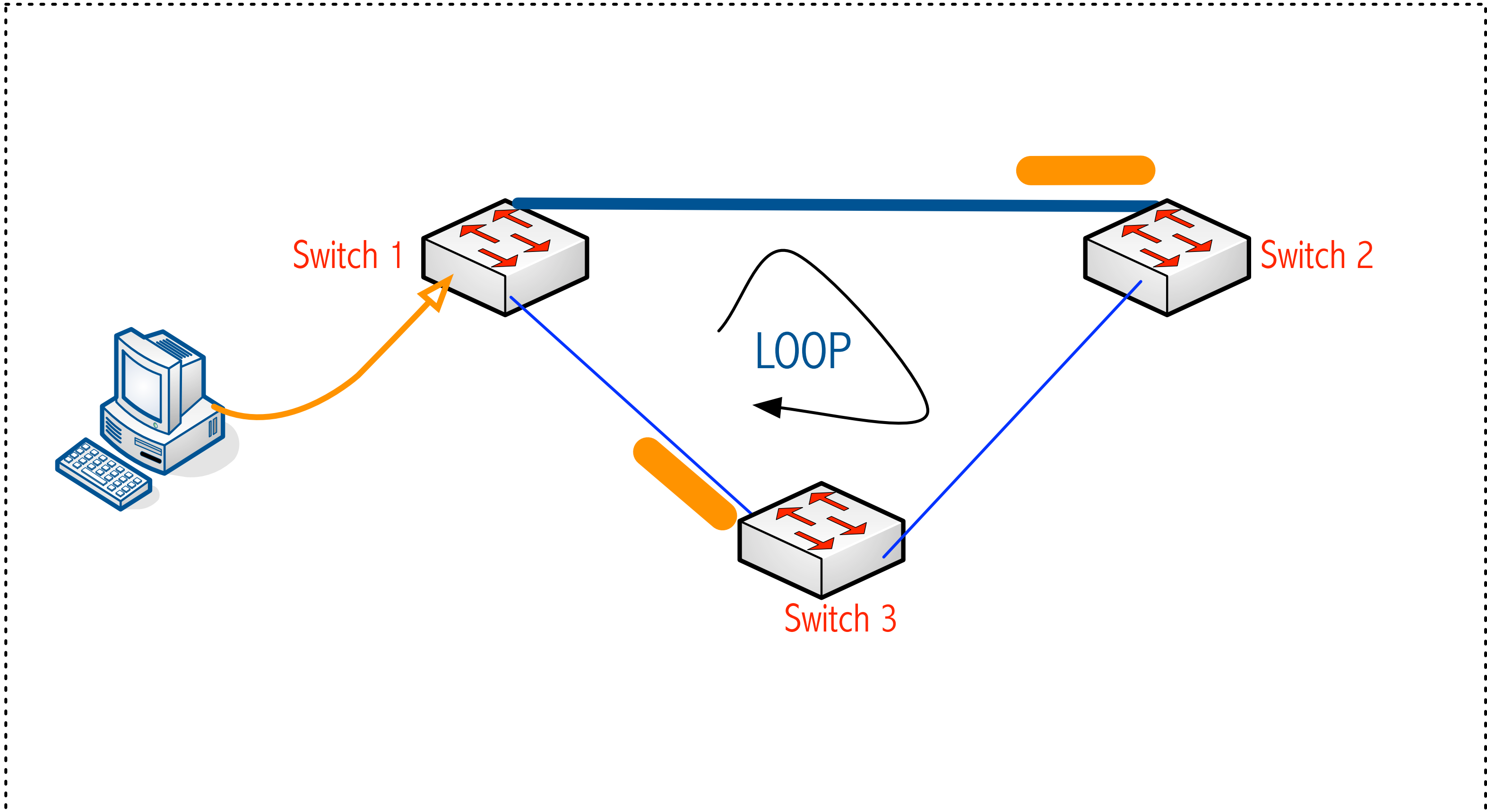
- BCAST frame ingresses into Switch 2 and Switch 3



Switched LANs, broadcast and loops

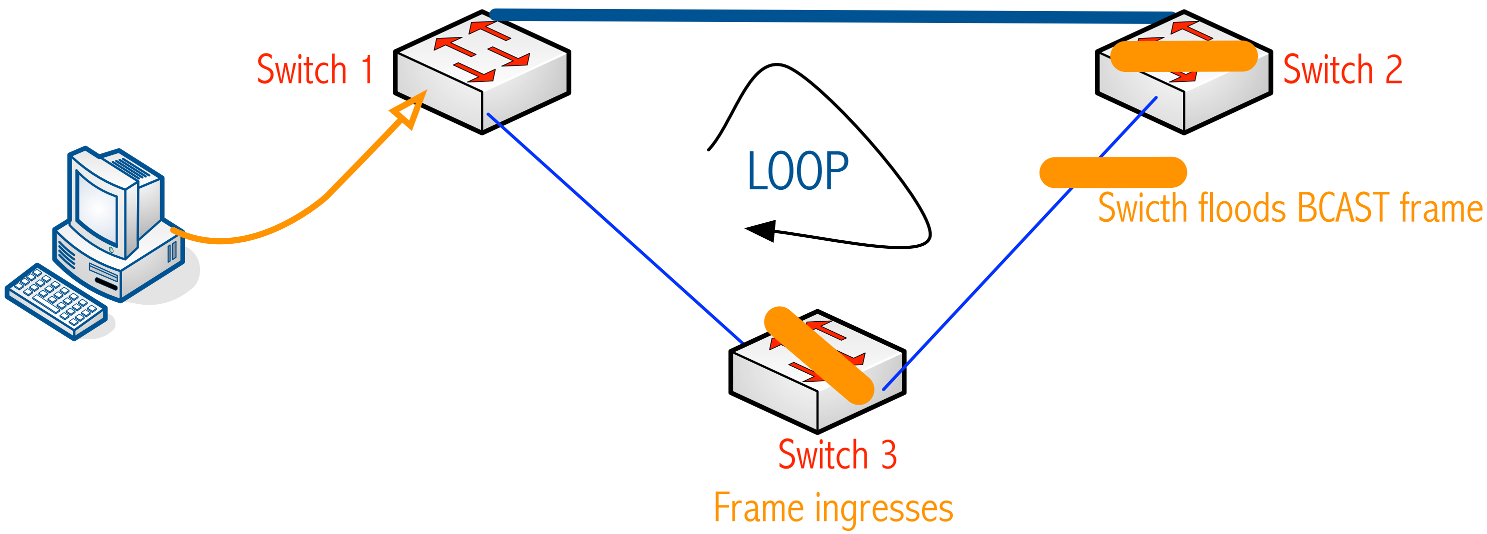
28

- BCAST frame ingresses into Switch 2 and Switch 3



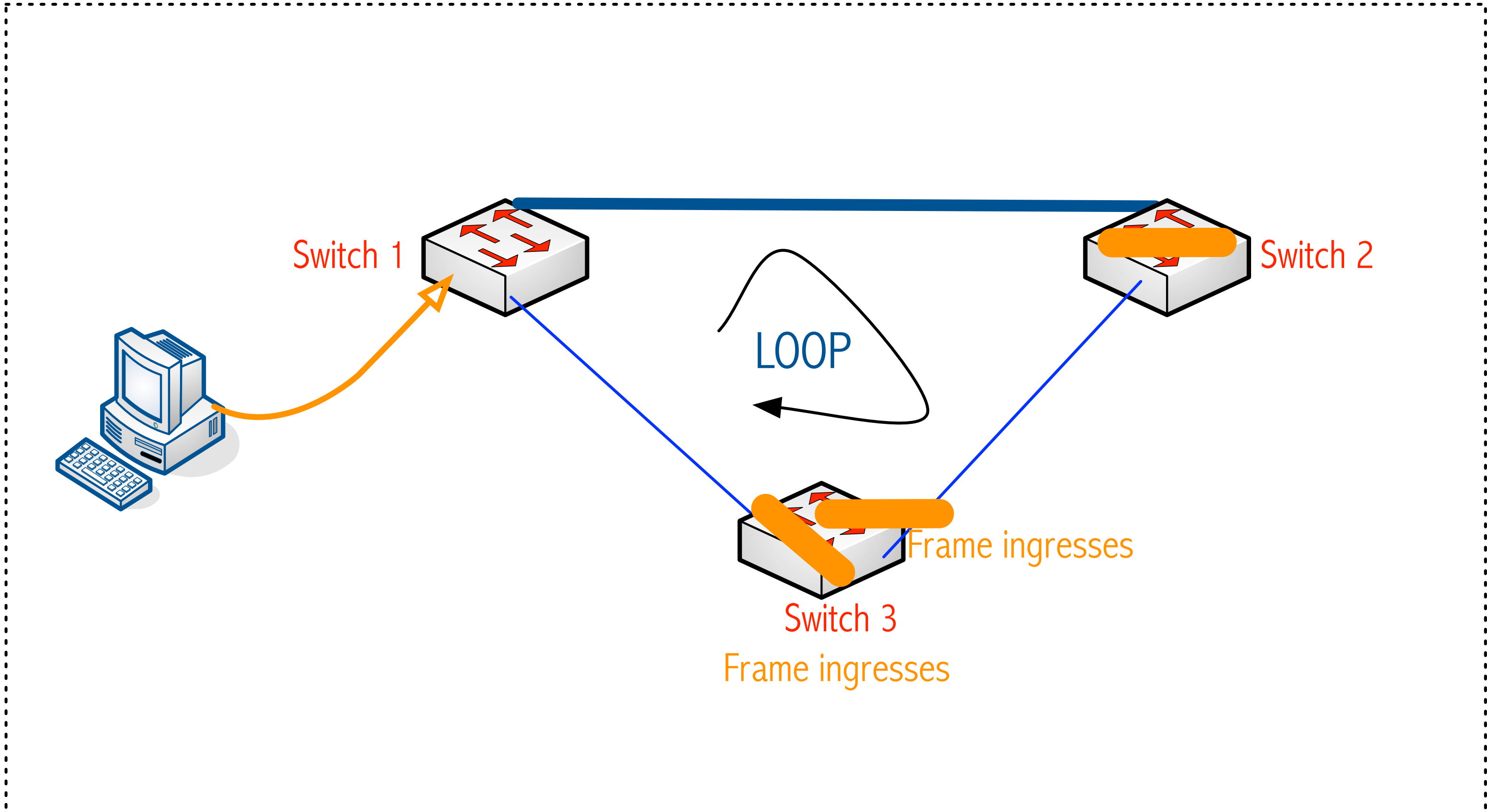
Switched LANs, broadcast and loops

- Switch 2 floods BCAST frame



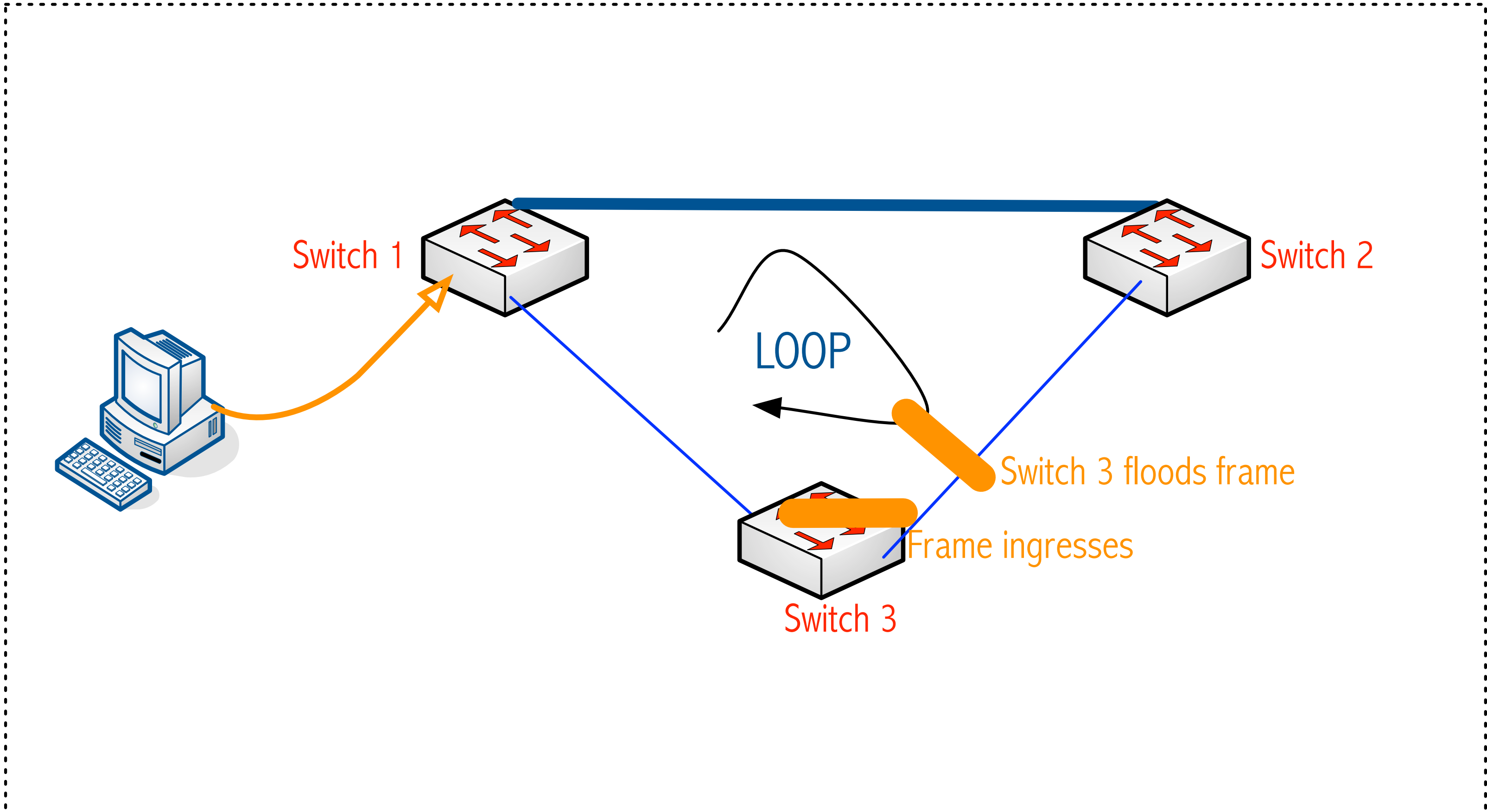
Switched LANs, broadcast and loops

- Switch 2 floods BCAST frame

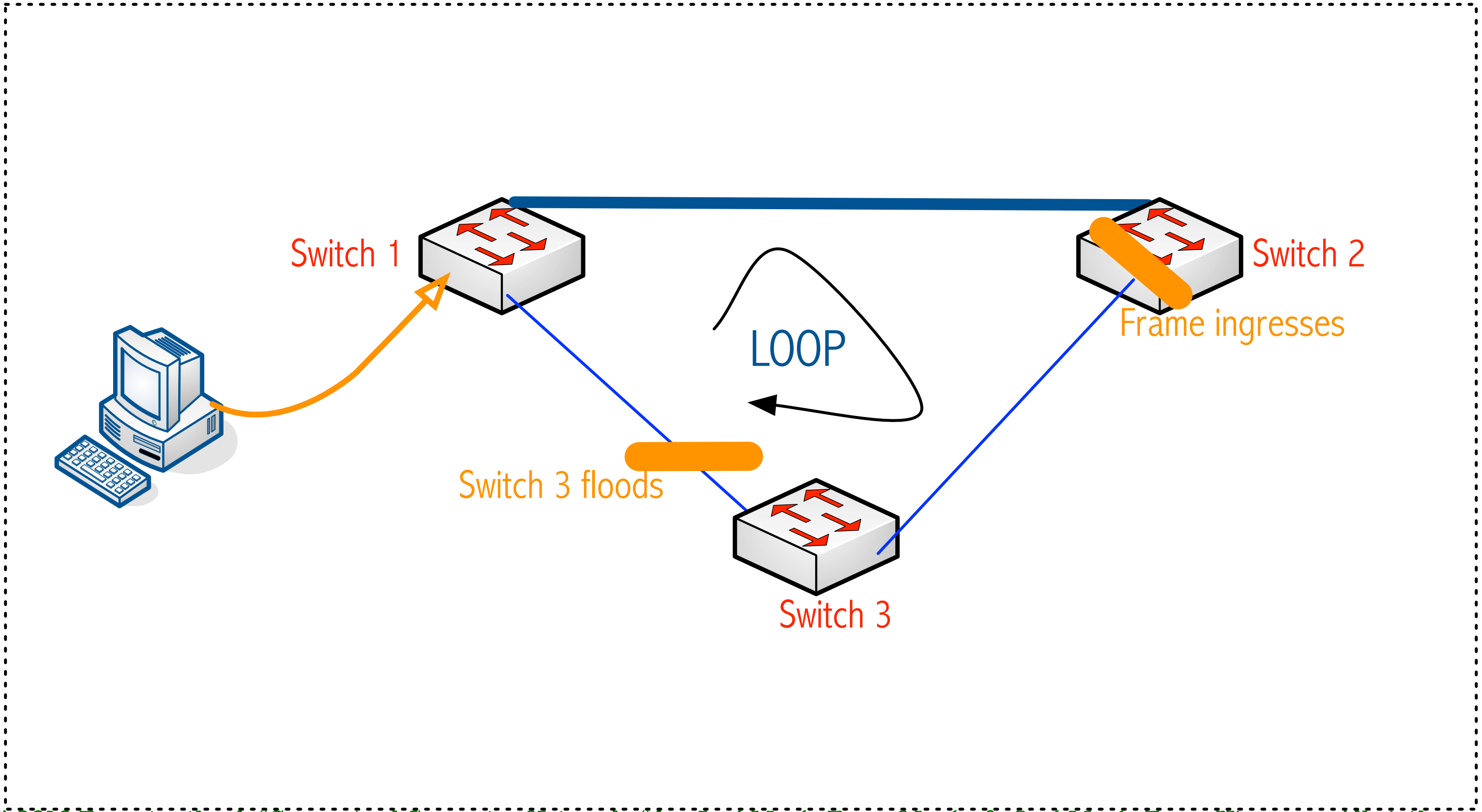


Switched LANs, broadcast and loops

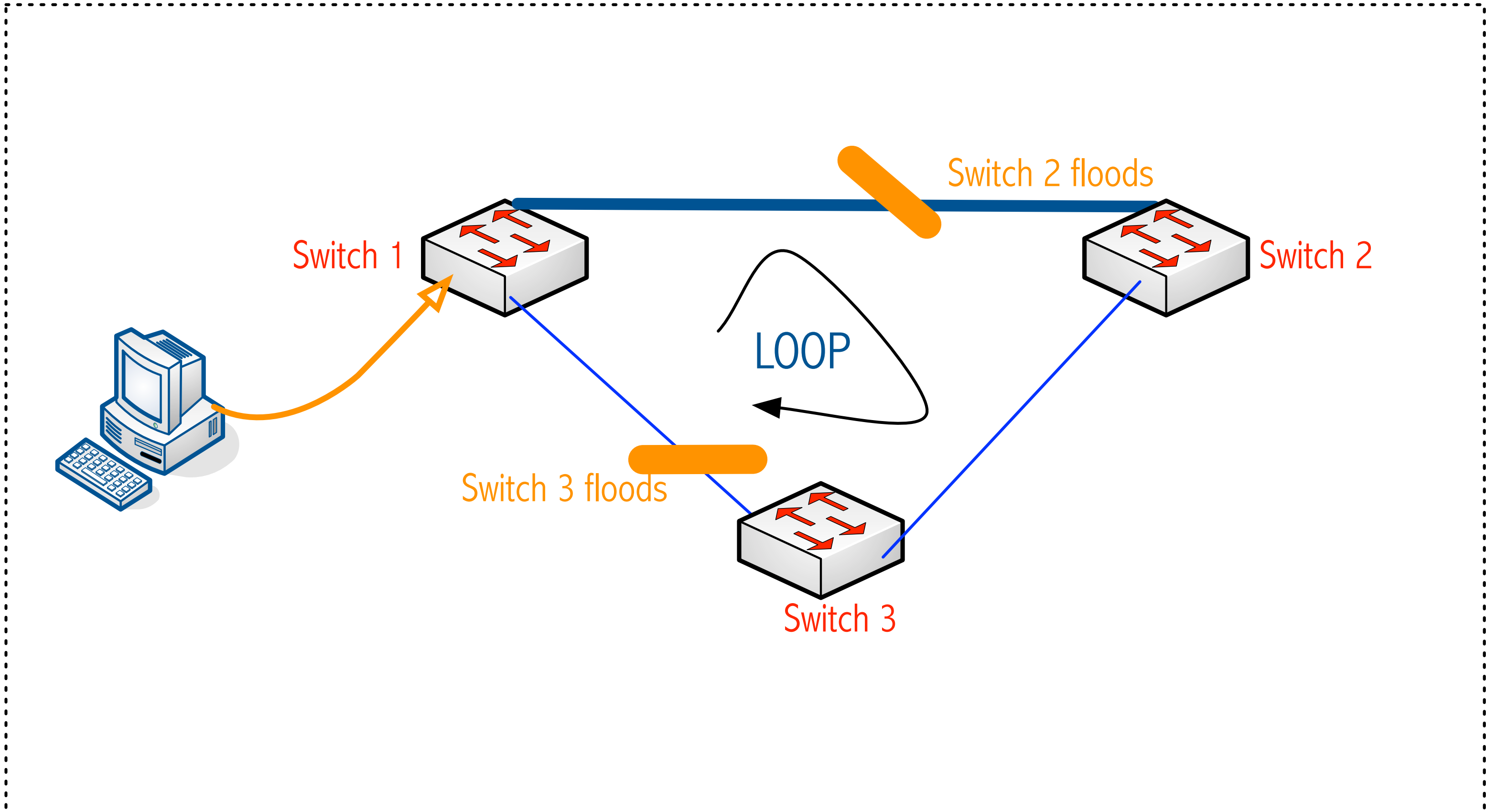
- Switch 2 floods BCAST frame



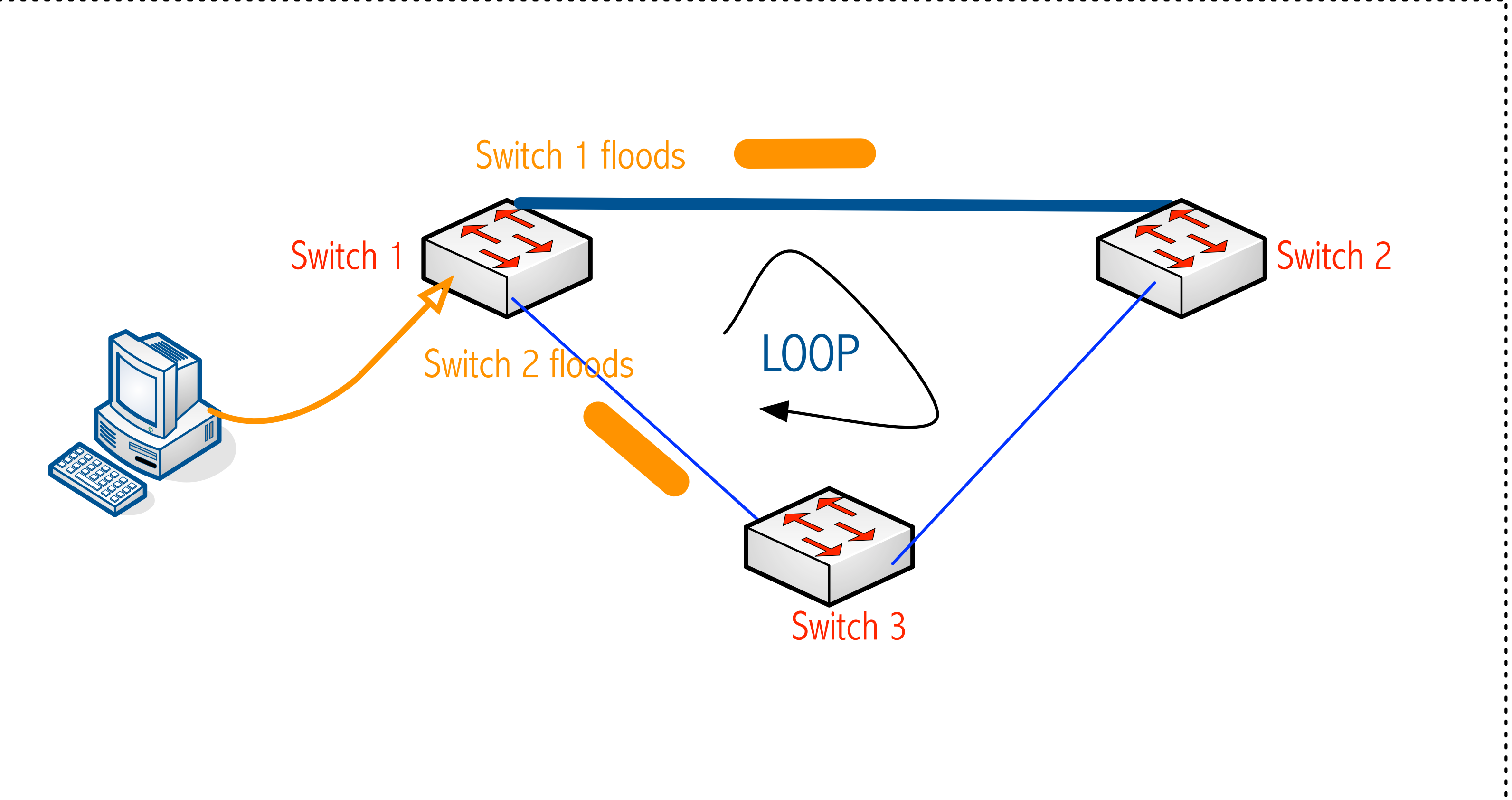
Switched LANs, broadcast and loops



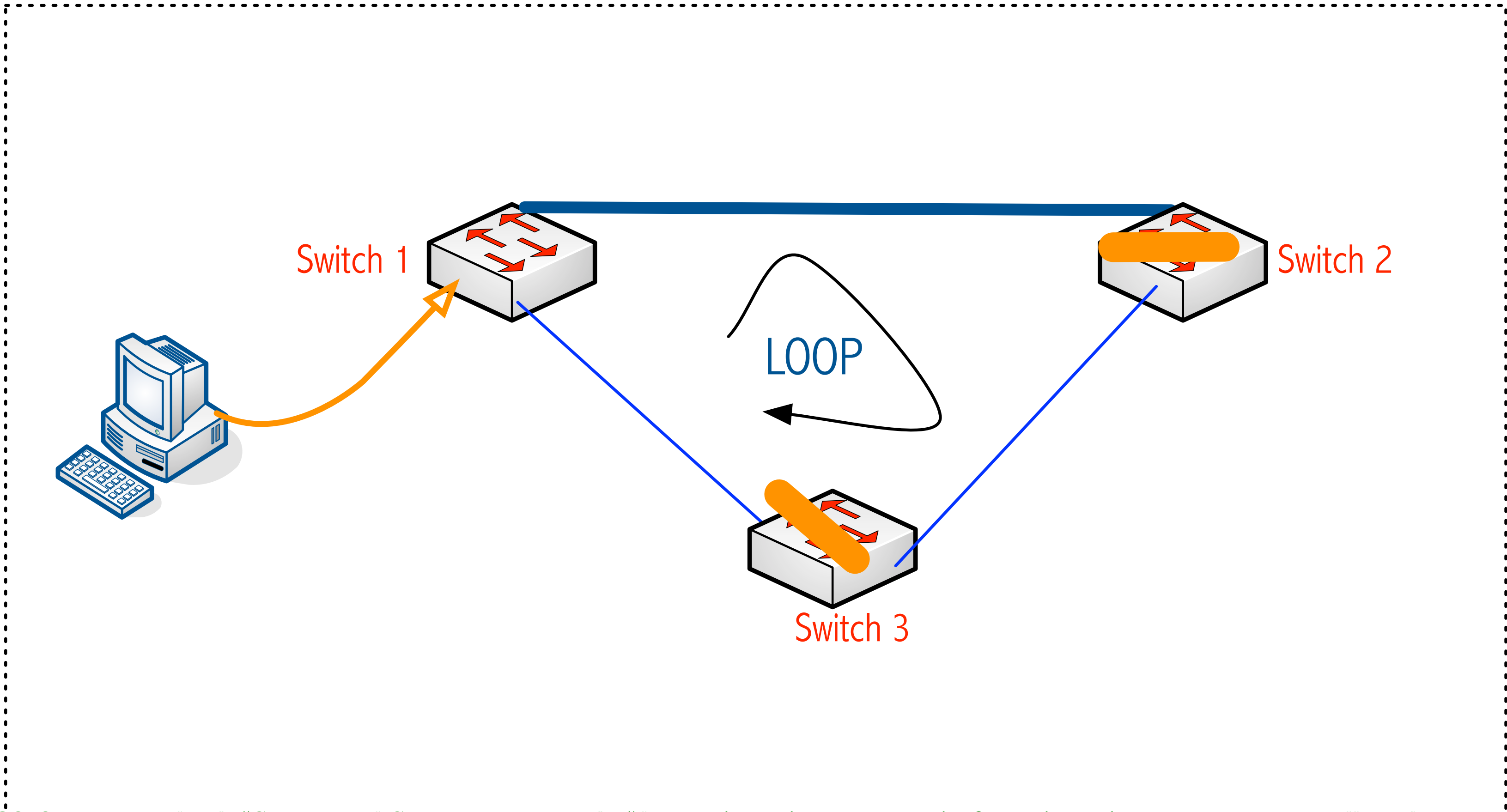
Switched LANs, broadcast and loops



Switched LANs, broadcast and loops



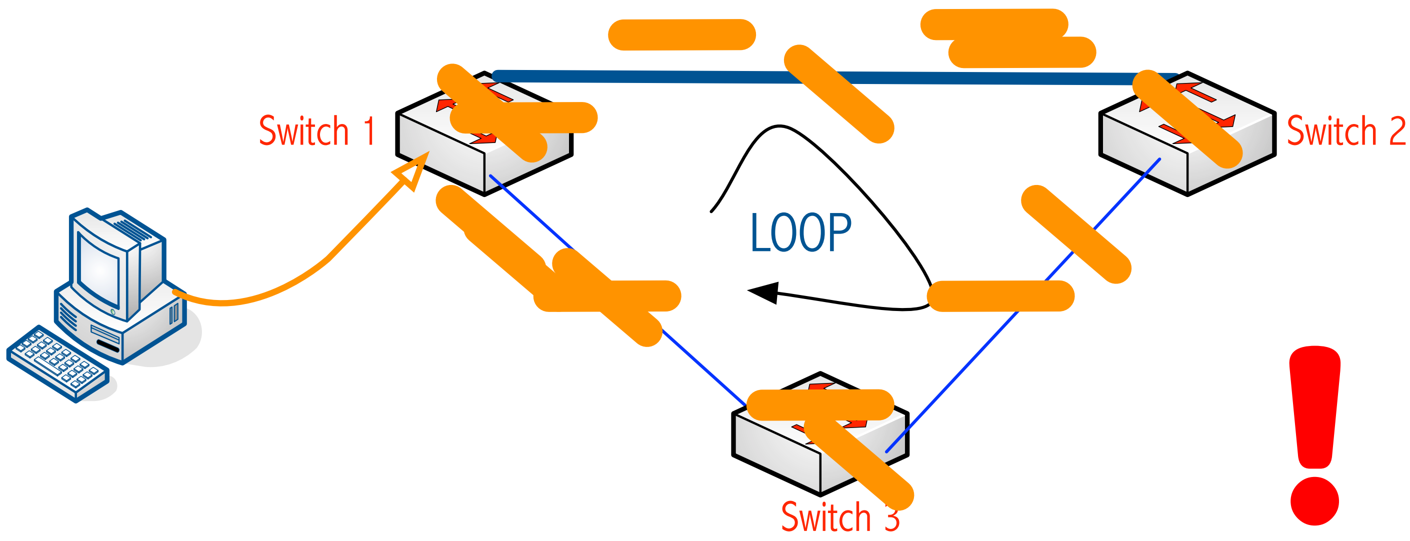
Switched LANs, broadcast and loops



Switched LANs, broadcast and loops

36

- This process never ends and takes up all the network bandwidth!
- Loops provide redundant ways in case of failures BUT
- LOOPS CAUSE BROADCAST STORMS!!!

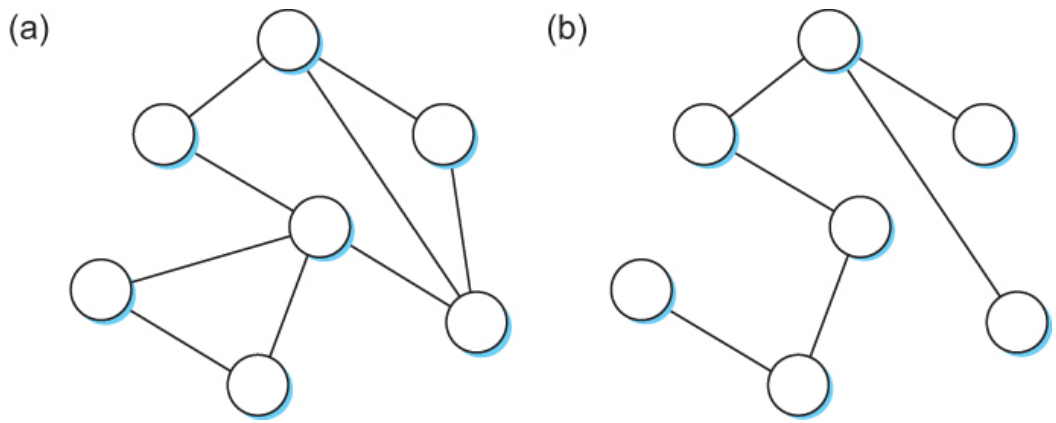


Spanning Tree Protocol (STP)

- Spanning Tree Algorithm is a distributed algorithm
- STP is based on it

Spanning Tree Algorithm

- The extended LAN may contain loops
- A **spanning tree** is a sub-graph of a graph that covers all its vertices but contains no cycles
  - It offers the same –*abstract*- connectivity but with no cycles



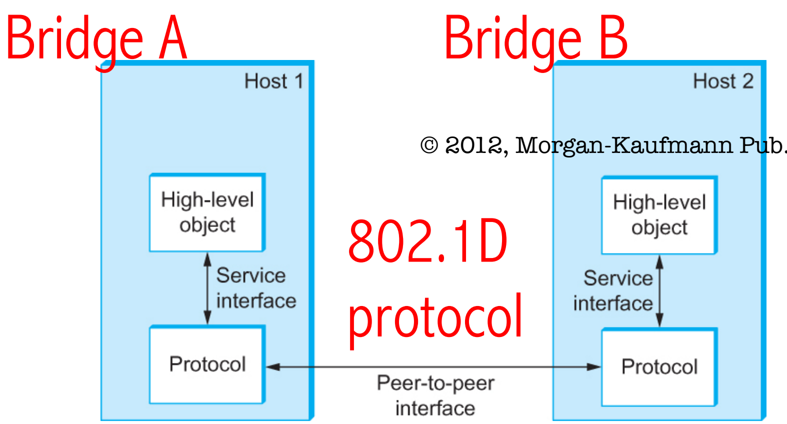
Example of (a) a cyclic graph; (b) a corresponding spanning tree.

Spanning Tree Protocol (STP)

- A protocol used by a set of bridges to agree upon a spanning tree for a particular extended LAN
  - STP is based on the Spanning Tree Algorithm

- The IEEE 802.1D 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
  - By removing ports from the topology the extended LAN is reduced to an acyclic tree
  - It is possible that an entire bridge will not participate in forwarding frames



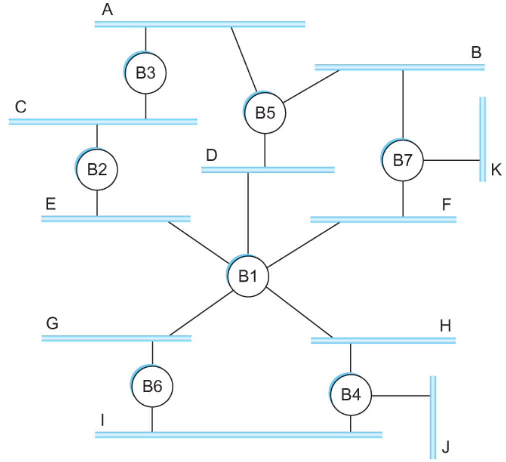
Spanning Tree Protocol (STP)

- Spanning Tree is executed in a distributed way (It's a distributed algorithm)
  - It is executed among a set of switches
  - The switches interchange STP messages (Look previous slide)

- The bridges are always ready to reconfigure themselves into a new spanning tree if some bridge or link fails

- Main idea
  - Each bridge selects the ports over which they will forward the frames

Spanning Tree *Algorithm*



- The distributed algorithm selects ports as follows:

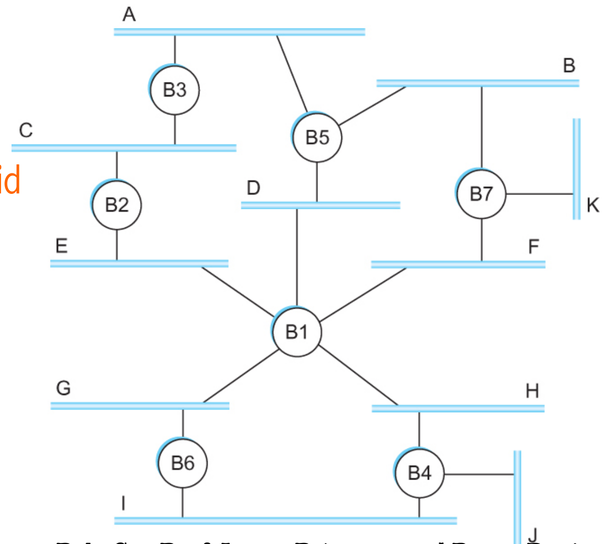
1. Each bridge has a unique identifier
   - B1, B2, B3…
2. Bridge with the smallest id becomes root of the spanning tree
   - The root bridge always forwards frames out over all of its ports
3. RP = Root Port:
   - Each bridge computes the shortest path to the root and notes which of its ports is on this path
   - This port is selected as the bridge's preferred path to the root

4. DBP = Designated Bridge Port:
   - All bridges connected to a given LAN elect a single DBP
   - Responsible for forwarding frames toward the root bridge

Spanning Tree Algorithm

- Each bridge has a **root port (RP)**
  - The closest port to the root
  - Used for communication with the root

- If two or more ports are equally close to the root
  - *Break ties* by selecting the port with the smallest next-bridge id
  - If still equal cost, then *break ties* by choosing the port with lowest port id



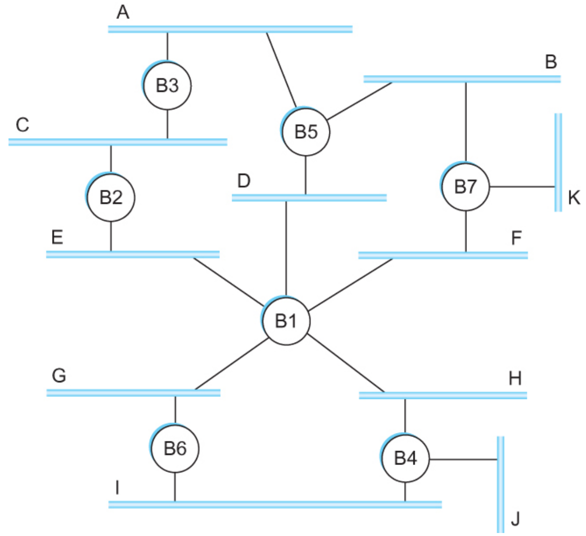
- **Example: Which is B3's root port?**
  - B1 is root
  - Shortest distance from B3 to B1 (The root bridge)
    - Through A = 2
    - Through C = 2
    - Equal, then break ties:
      - A: Next bridge on least-cost path is B5
      - C: Next is B2 which has a lower ID than B5, THEREFORE ROOT of B3 is its port C

Spanning Tree Algorithm

- Each LAN has a **Designated Bridge Port (DBP)**
  - It's the one that is closest to the root

- If two or more bridges are equally close to the root,
  - *Break ties by* selecting the bridge with the smallest bridge id
  - If the bridge selected so far has two or more ports connected to a LAN, choose the port with lowest port id

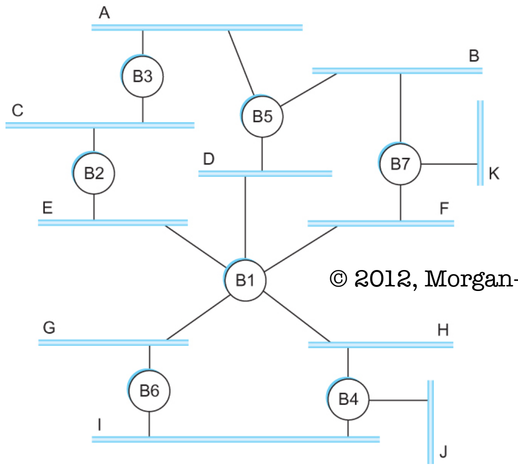
- **Example: Which is the DBP of LAN B?**
  - Shortest distance from B → B1 (root) is 2 via B5 and via B7
  - Since B5 < B7, we select B5 as the *Designated Bridge on B,* specifically the port on the upper right of B5 is the Designated Bridge Port of LAN B



Spanning Tree Algorithm

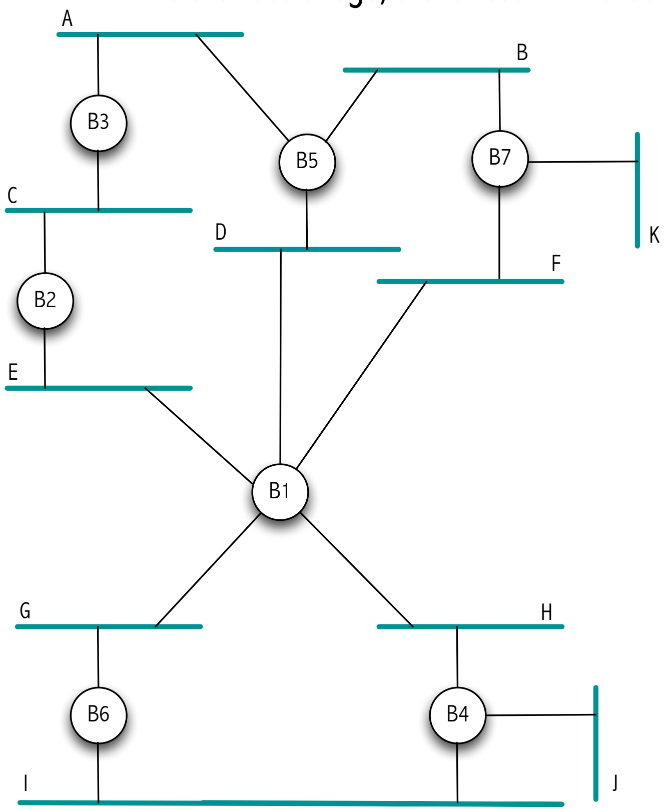
- Each LAN has a **Designated Bridge Port (DBP)**
  - It's the one that is closest to the root

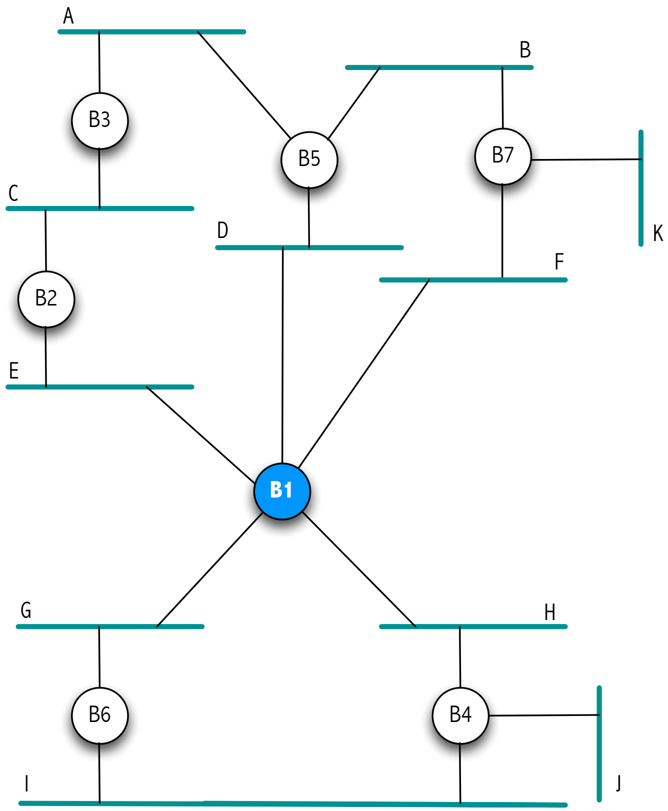
  - *Each bridge is connected to more than one LAN*
  - *So it participates in the election of a designated bridge for each LAN it is connected to.*
  - *Each bridge decides if it is the designated bridge relative to each of its ports*
  - *The **bridge forwards frames over those ports for which it is the designated bridge***



Spanning Tree Algorithm

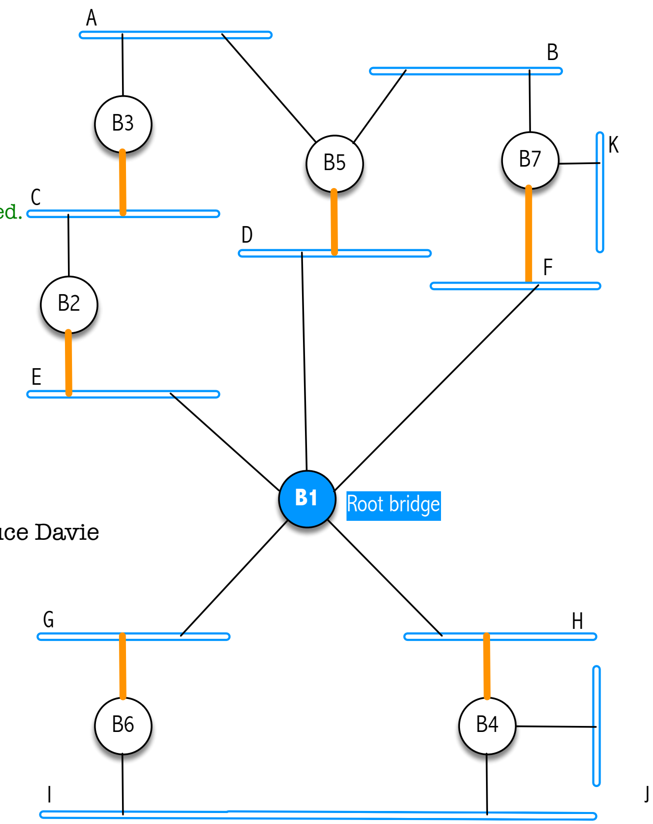
- Example from textbook pg. 194 (Fig. 3.10): Extended LAN with loops
- **Step 1: Root bridge**
  - B1 is the root bridge, the lowest numbered bridge





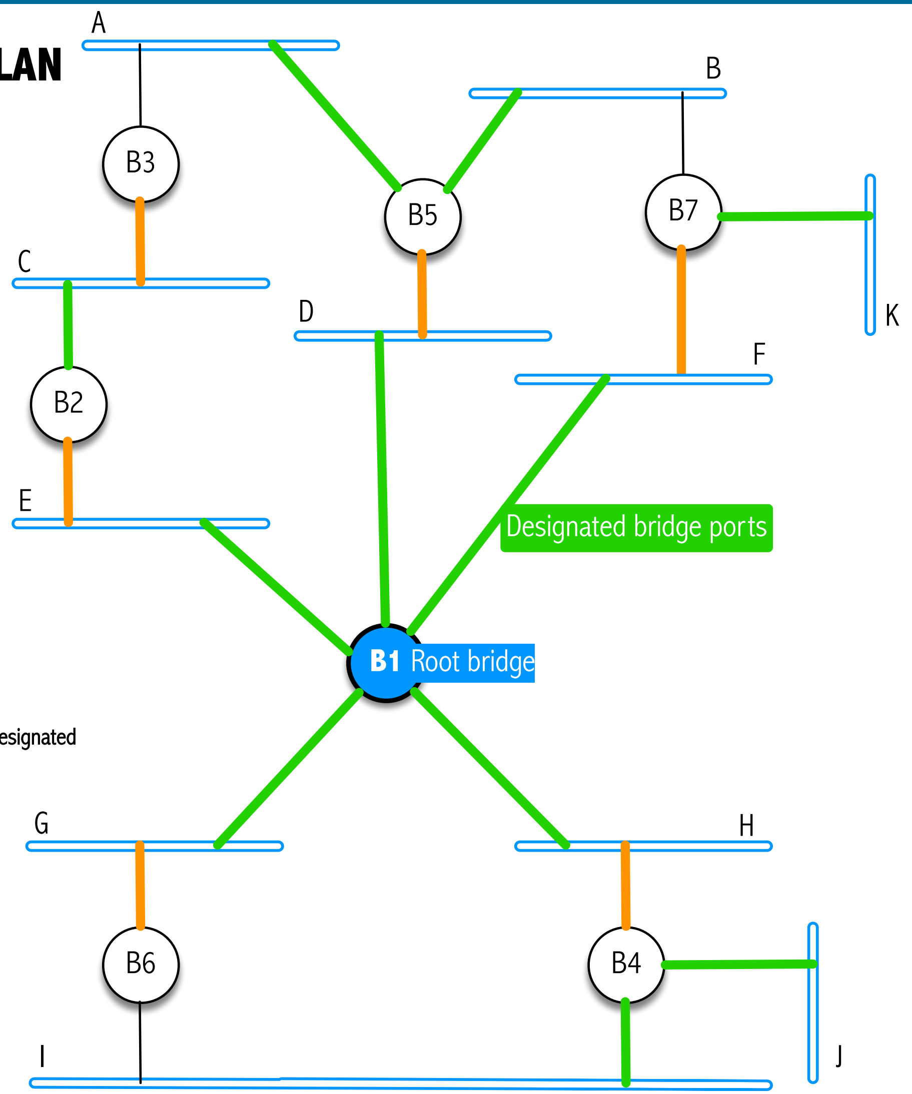
Spanning Tree Algorithm

- Textbook pg. 194 (Fig. 3.10): Extended LAN with loops
- **Step 2: Root port (RP) of each bridge**
  - B3 least cost to root is 2 (Via A and via C)
    - Break ties by lower label of next bridge: Choose B2 since label is lower numbered, B2 < B5
  - B4 least cost to root is 1 (Via H)
  - Calculate the root port of each bridge



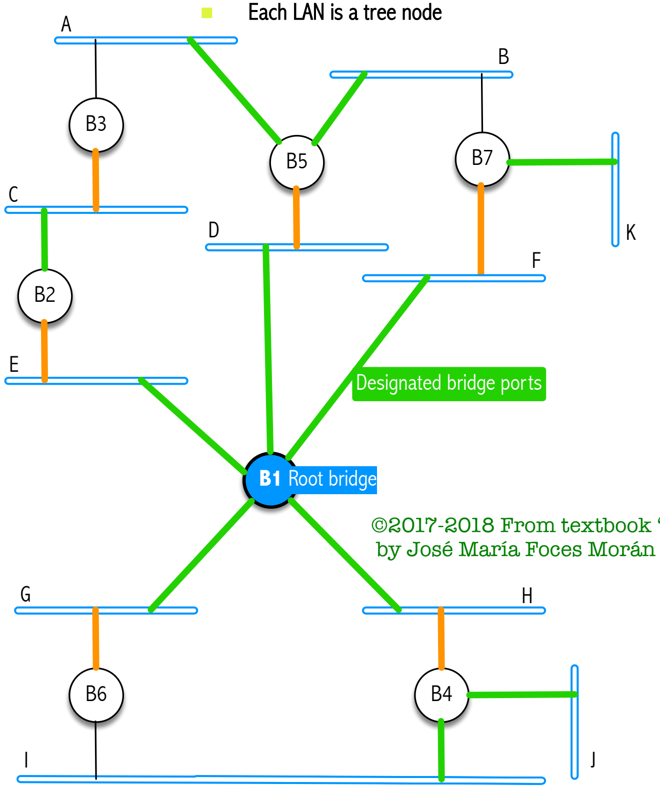
Spanning Tree Algorithm

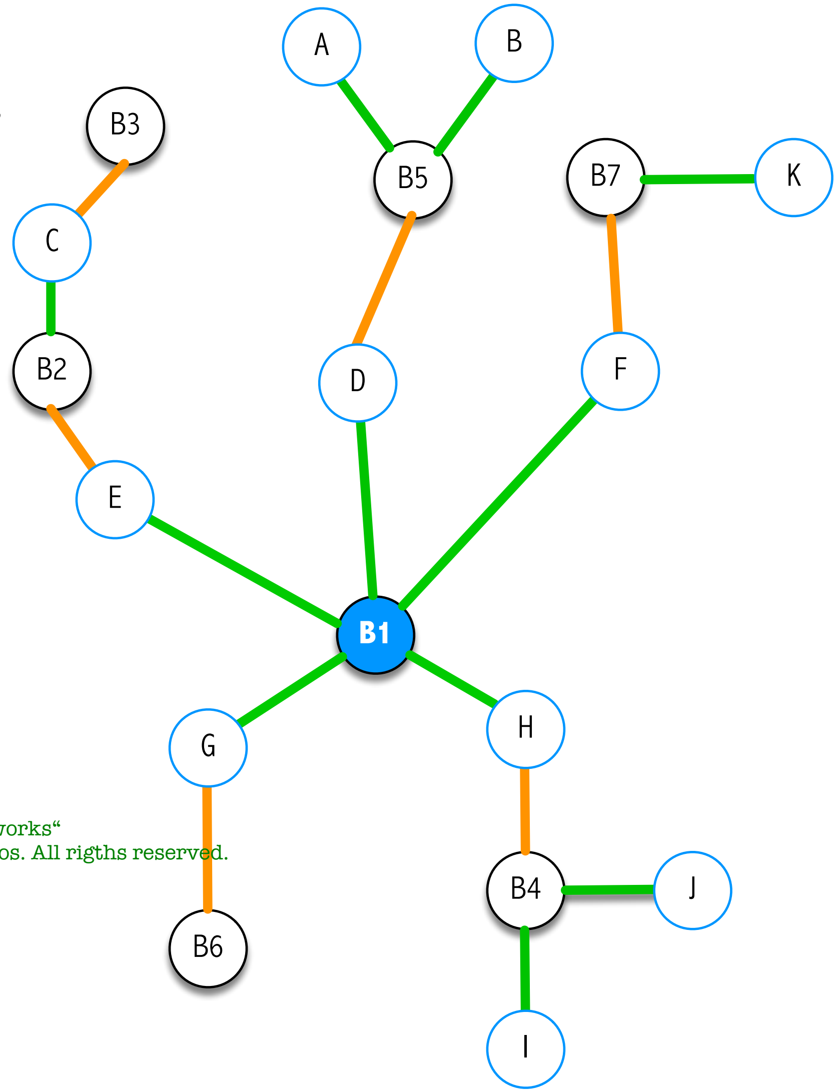
- **Textbook pg. 194 (Fig. 3.10): Extended LAN with loops**
- **Step 3: Designated Bridge at each LAN**
  - LAN A:
    - Cost to root via B3 = 3
    - Cost to root via B5 = 2
      - Choose bridge that is on the least cost path: B5
  - LAN J:
    - Connected to B4 only: Designated bridge is B4
  - LAN B:
    - Cost to root via B5 = 2
    - Cost to root via B7 = 2
      - Break ties by next bridge label, choose lower: B5 < B7, therefore designated bridge at LAN B is B5



Spanning Tree Algorithm

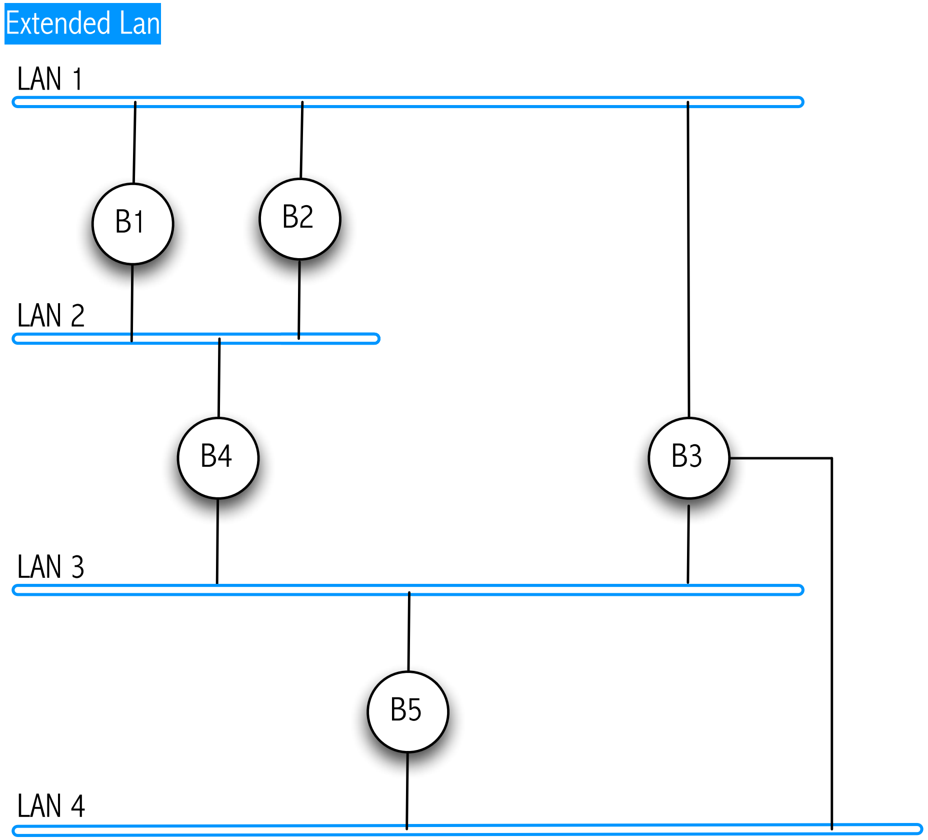
- **Step 4: Obtain the spanning tree**
  - All ports root or designated port result into active ports
    - Ports not RP nor DBP: disabled
  - Spanning tree nodes:
    - Each bridge is a tree node
    - Each LAN is a tree node



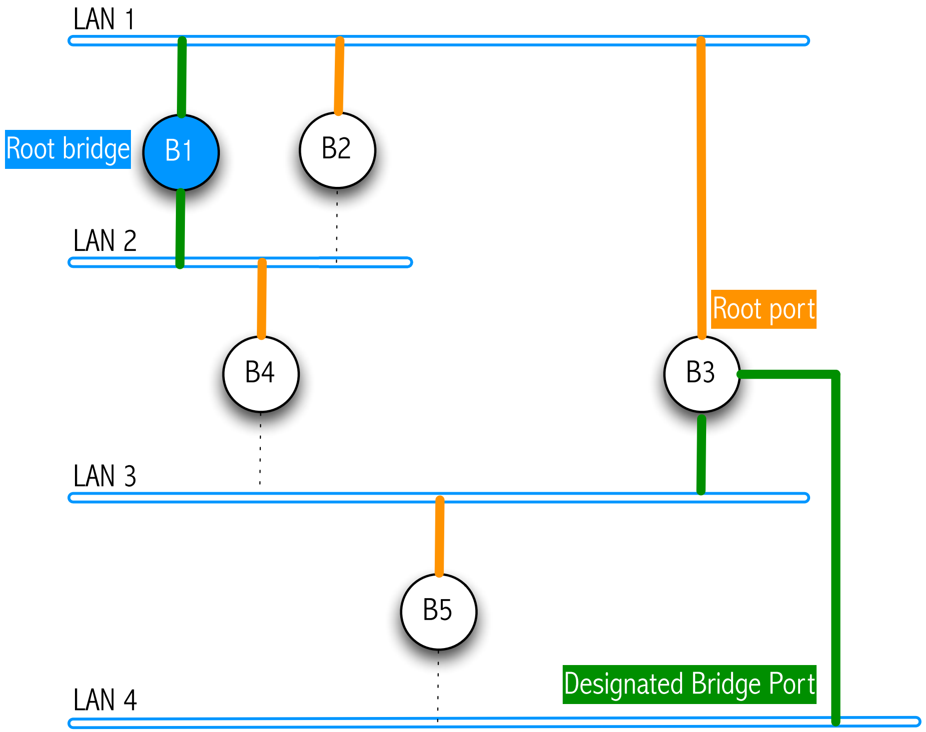


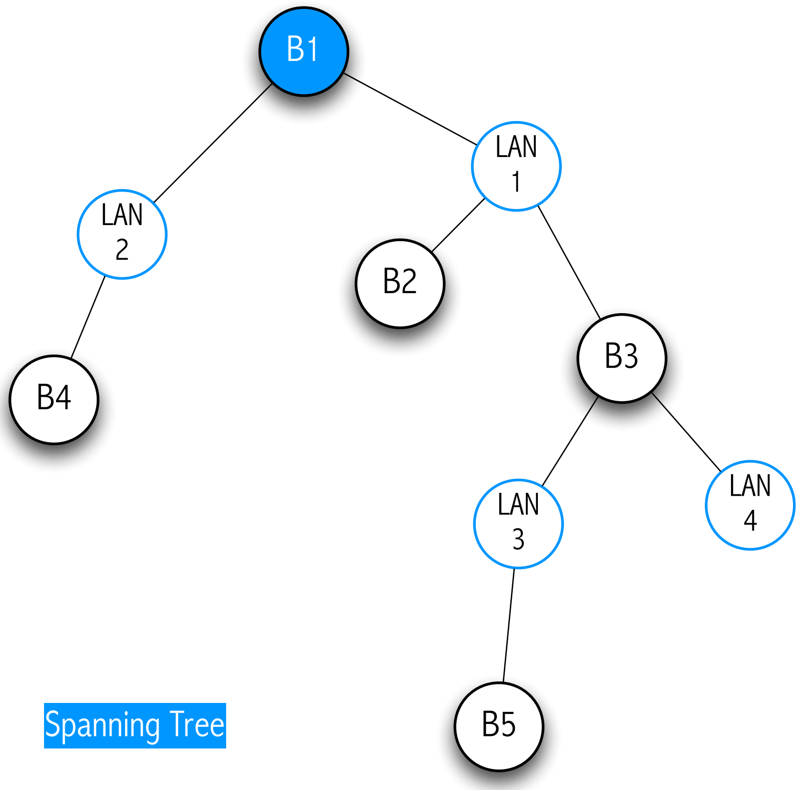
Spanning Tree Algorithm, example

- Obtain the Spanning Tree to the Extended Lan
  - 1. Root bridge
  - 2. RP
  - 3. DBP
  - ST



Spanning Tree Algorithm, example





STP messages

- The Spanning Tree Algorithm in STP is a distributed algorithm
  - It is executed among the switches of an Extended Lan by exchanging STP messages

- Initially each bridge thinks it is the root
  - 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 a 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

STP messages

- 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

STP messages

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

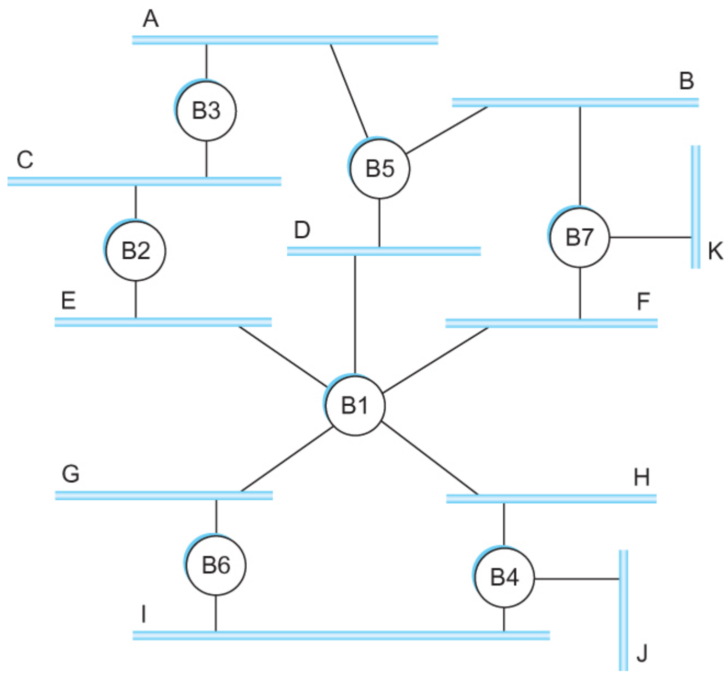
  => a message from a bridge that is closer to the root or equally far from the root but with a smaller id

    - 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

STP messages

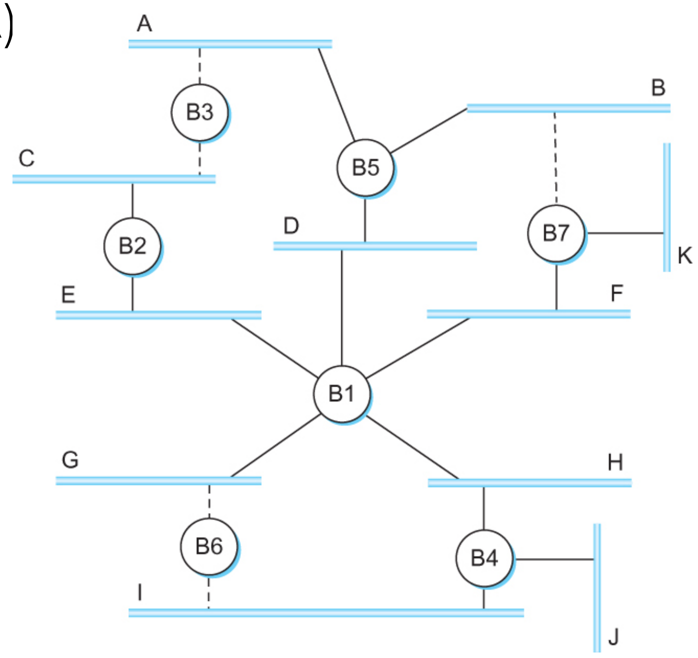
- Consider the situation when the power had just been restored to the building housing the following network



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

STP messages

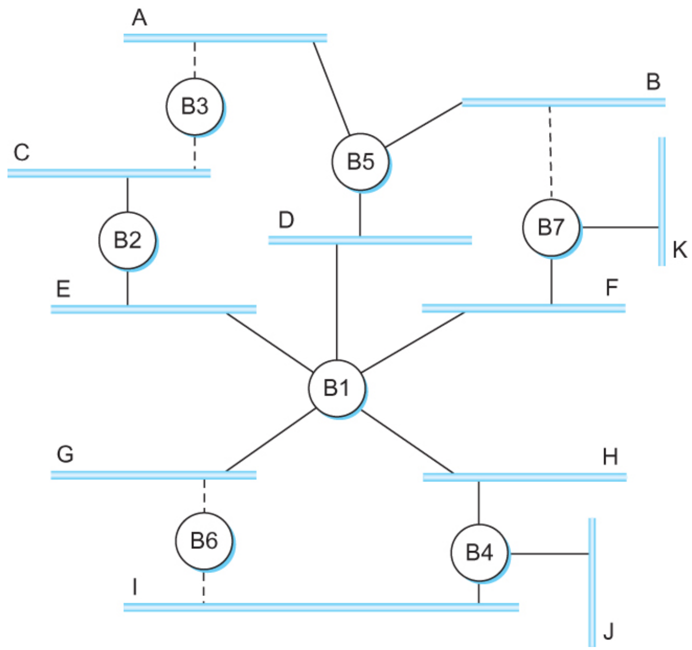
- Denote a configuration message from node X in which it claims to be distance d from the root node Y as (Y, d, X)



- Consider the activity at node B3

STP messages

- 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.
  - 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
    - Spanning tree algorithm does not scale
    - Broadcast does not scale
  - Do not accommodate heterogeneity

Switching and Forwarding

Switch

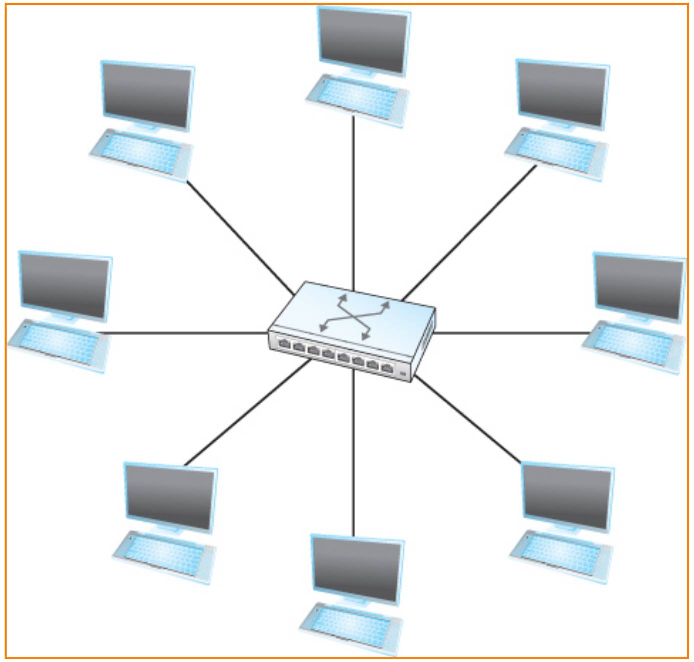
- A **mechanism** that allows us to interconnect links to form a large network (*One network*)

- A multi-input, multi-output **device** which transfers ***frames*** from an input to one or more outputs

Switching and Forwarding

- Point-to-point links

© 2012, Morgan-Kaufmann Pub. Co., Prof. Larry Peterson and Bruce Davie

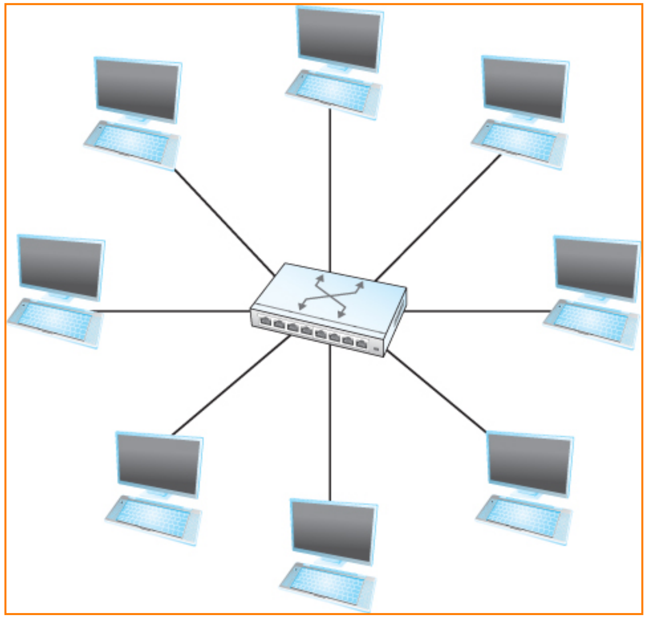


- Topologies:
  - Bus (Ethernet)
  - Ring
  - Star

- Switches implement the **star topology**

Switching and Forwarding

Star topology



- Interconnecting switches
  - By point-to-point links
  - Large networks
- Adding a new host to the network
  - **Not** necessarily means that the hosts already connected will get **worse performance**
  - **By contrast: In a bus topology** adding a new end node generally means worse performance

Switching and Forwarding

Hubs

- CSMA/CD, always

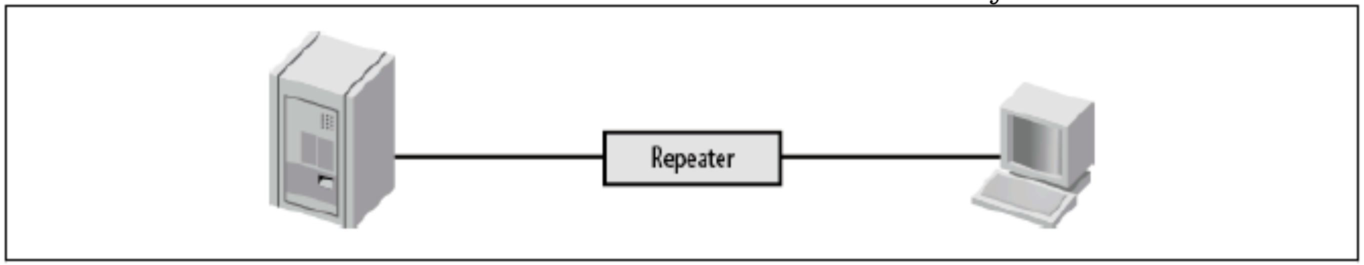


Figure 2-1. Repeater extending a single 10Base-T link

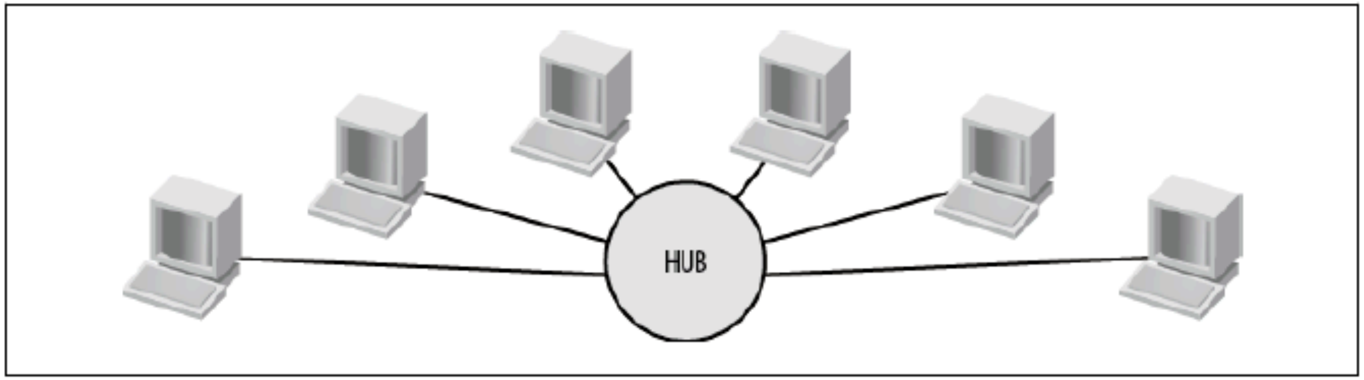


Figure 2-2. Hub connecting multiple hosts to a network

Switching and Forwarding

Hubs

- CSMA/CD, always

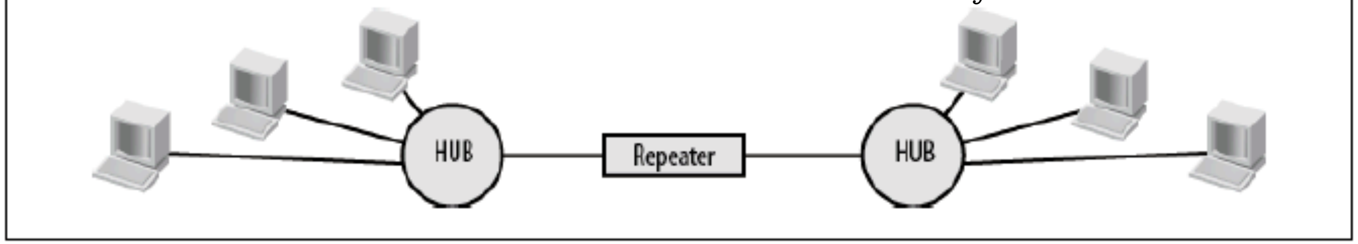


Figure 2-3. Repeater joining hubs

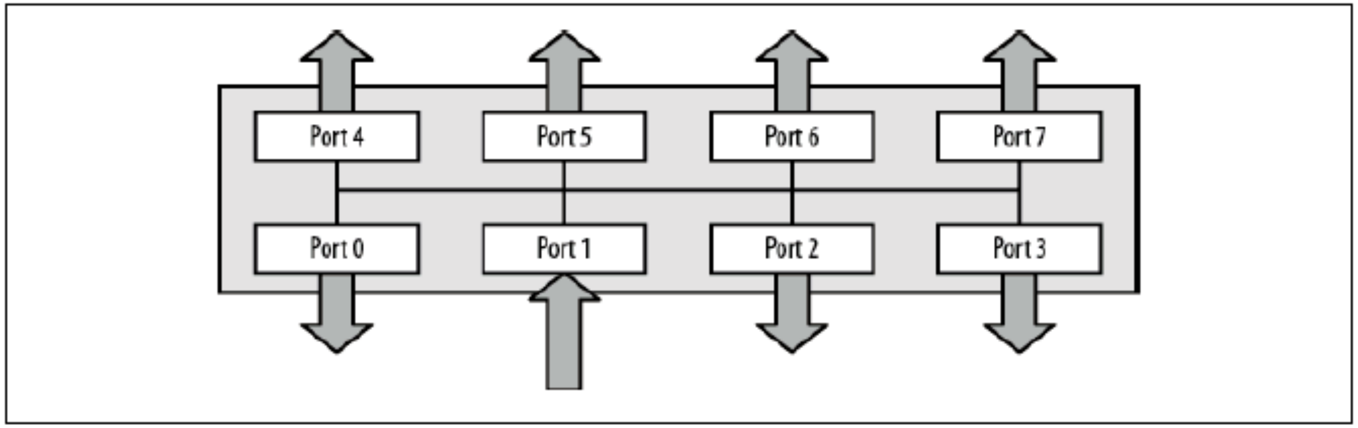


Figure 2-4. Hubs repeat inbound signals to all ports, regardless of type or destination

Switching and Forwarding

Switches

- CSMA/CD, only in half duplex mode

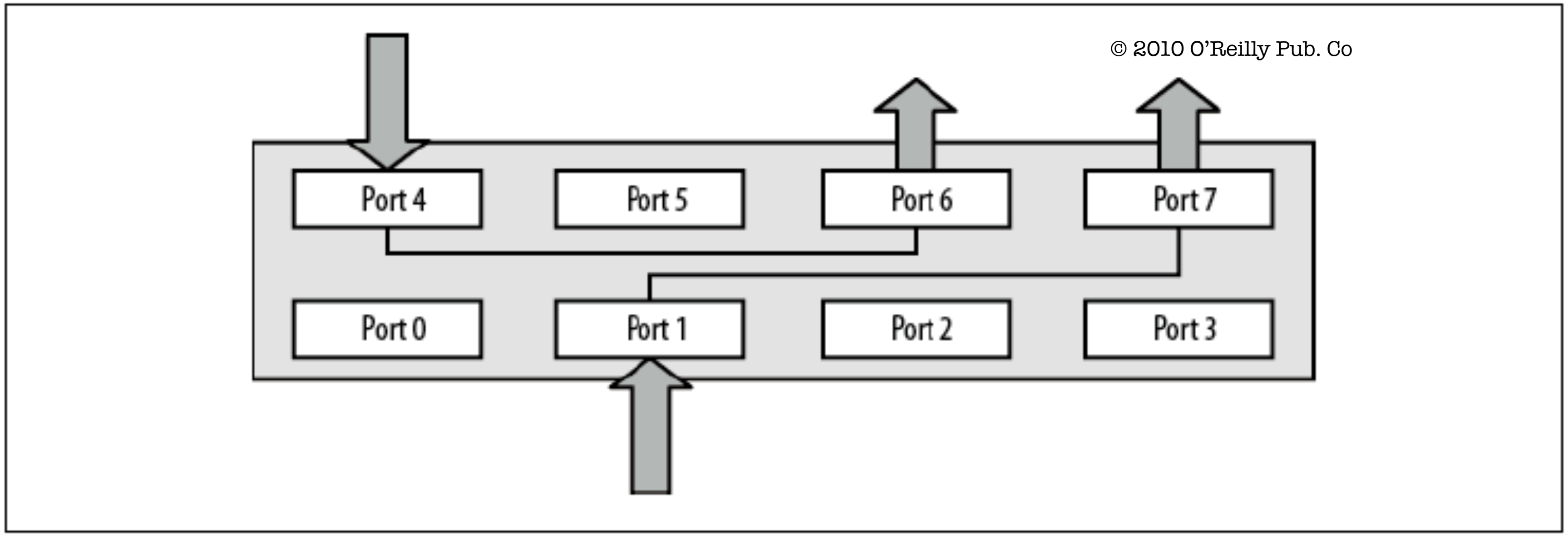
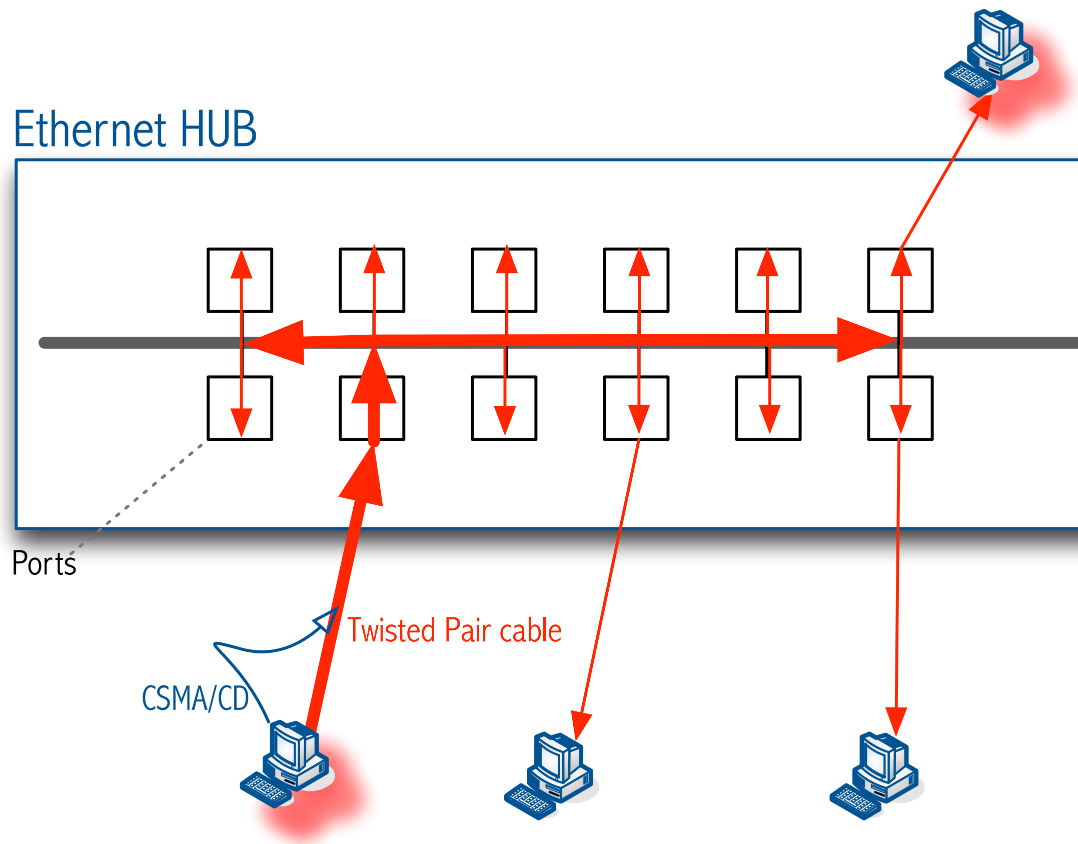


Figure 2-7. A switch forwards frames only to the ports that need to receive them

Switching and Forwarding

10Mbps Hubs

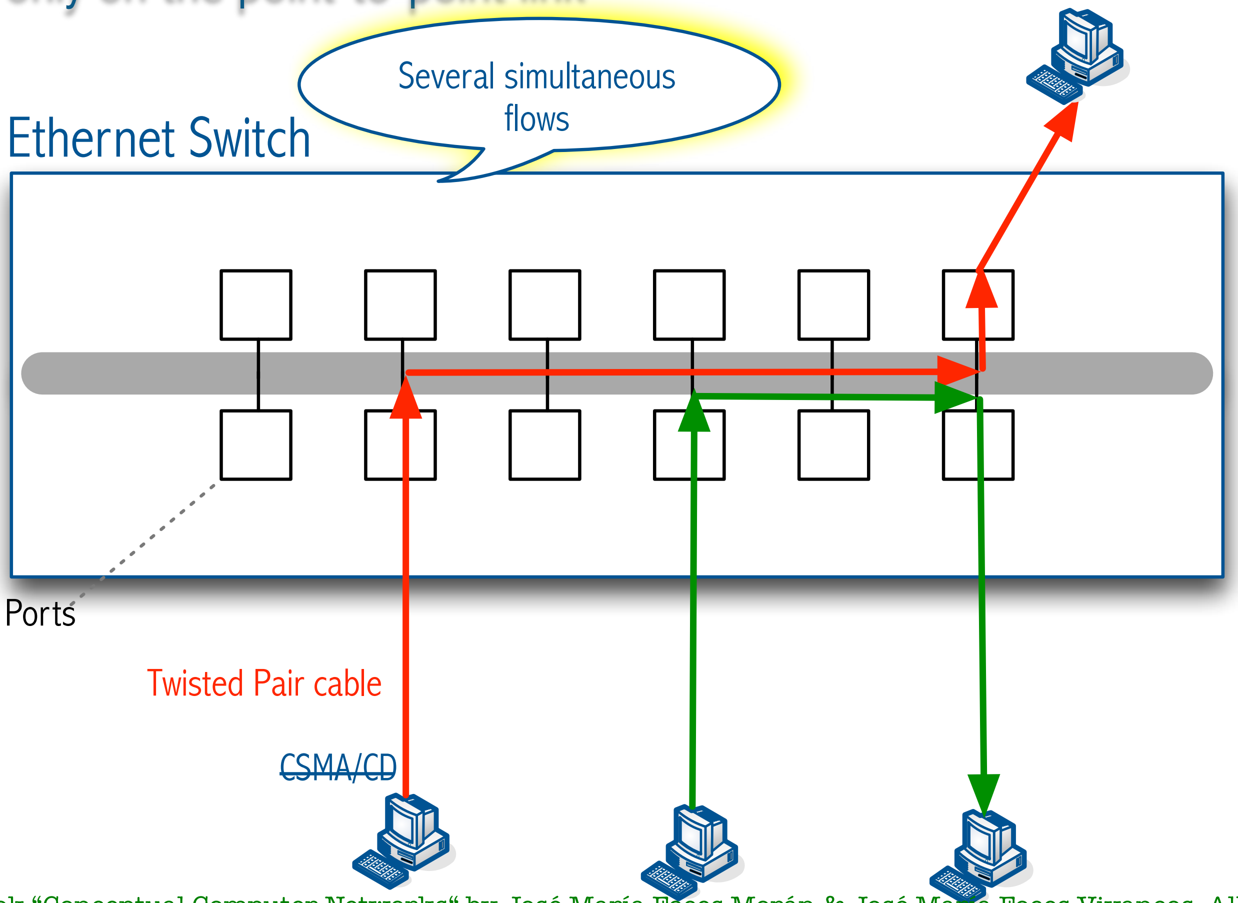
- CSMA/CD, always



Switching and Forwarding

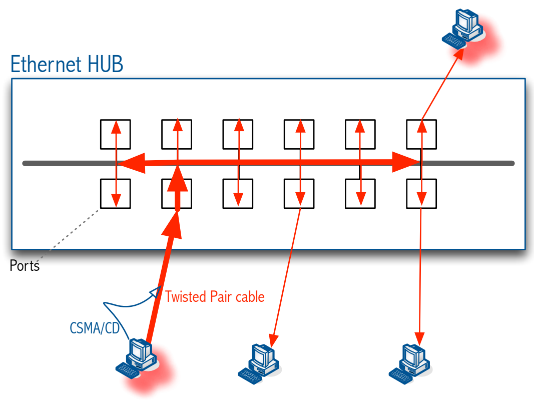
Switches

- CSMA/CD only on the point-to-point link

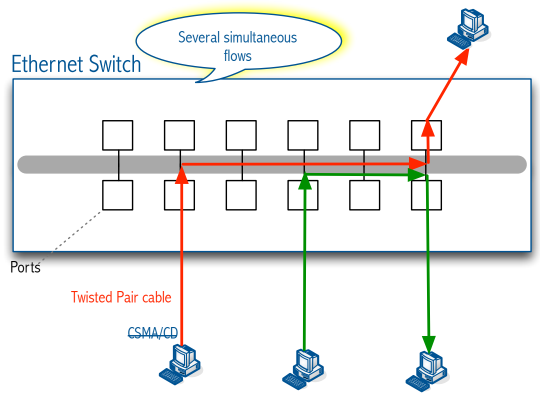


Switching and Forwarding

- Chapter 2
  - Bus Ethernet
    - Maximum throughput is 10Mbps



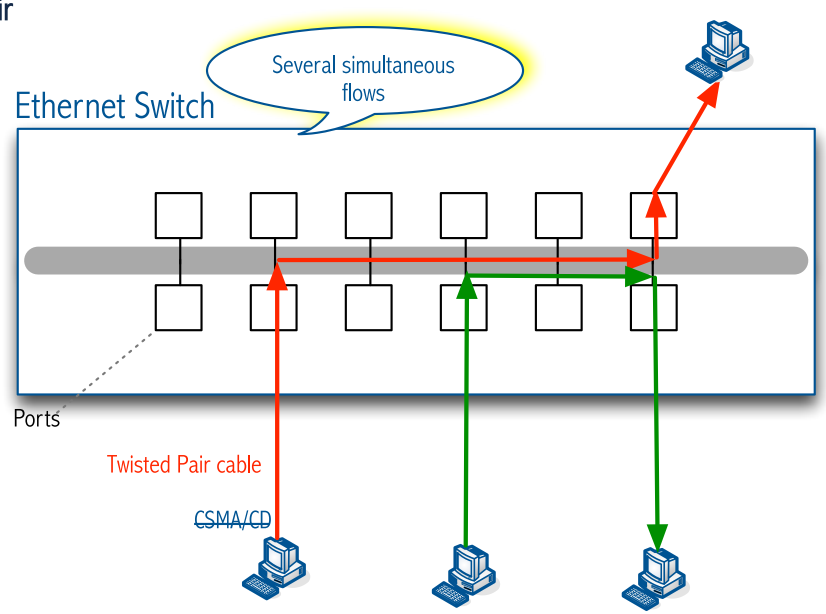
- Chapter 3:
  - **Switched** Ethernet (At 10Mbps, for example)
    - ***Many*** hosts can transmit at 10Mbps **SIMULTANEOUSLY!**
    - **Example 8-port switch**
      - 4 simultaneous transmissions 4 x 10 Mbps = 40 Mbps aggregated throughput



Switching and Forwarding

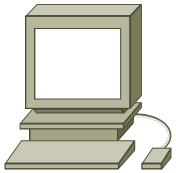
- A switch is connected to a set of links
  - Each link runs the appropriate data link

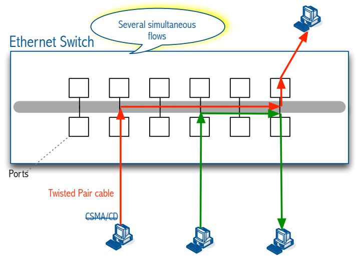
- The job of a SWITCH
  - *Switching:* To receive incoming *frames* on one of its links and to transmit them on some other link
    - Many hosts can transmit at full speed SIMULTANEOUSLY and have their properly forwarded by the switch
    - Normally, UNICAST frames are forwarded by the switch in isolation from the other UNICAST frames being forwarded by it simultaneously
    - Still, the switch can forward BROADCAST traffic by flooding the frames



Switching and Forwarding

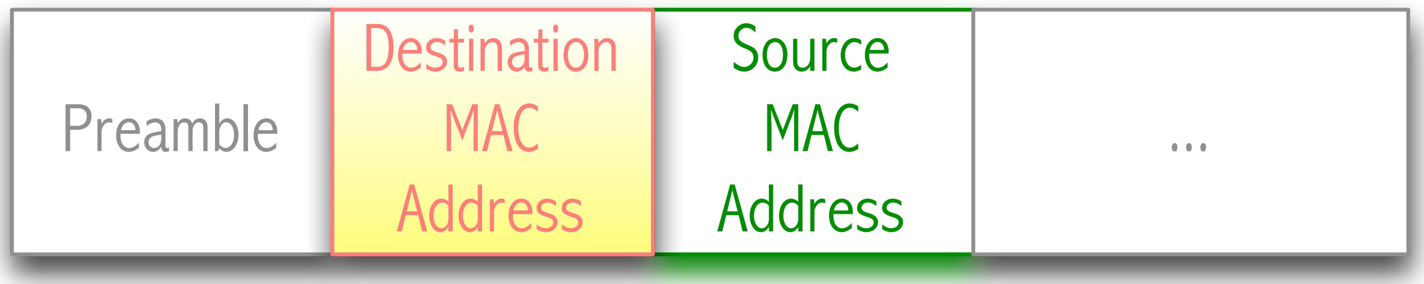
- Assumptions
  - Each end-host adapter has a globally unique Internet address
    - IP address
  - Each network adapter has a globally unique LAN address
    - MAC address (Review Ch.2)
  - Identification of each port
    - A **number**
    - A name
  - Each **bridge** has a unique identification
    - Normally, the MAC of one of its ports



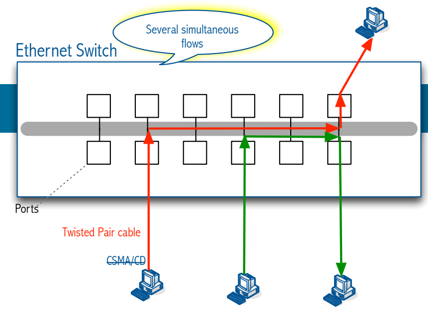


Switching and Forwarding

- Every ***frame*** contains enough information to enable any **switch** to decide how to get it to destination
  - Every *frame* contains the complete destination MAC address



Switching and Forwarding



- SWITCH
  - Which port to place each *frame* on?
    - It looks at the **header** of the ***frame*** for an **identifier** that it uses to make the decision

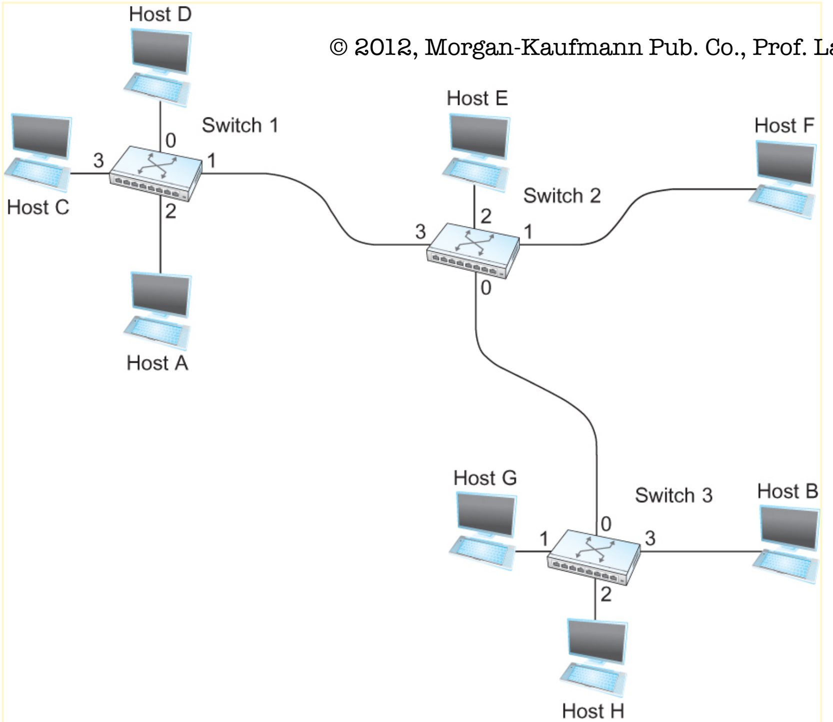
Preamble	Destination MAC Address	Source MAC Address	…

  - Three approaches
    - ***Datagram*** *(Connectionless)*
    - Virtual circuit (Connection-oriented)
    - Source routing, less used

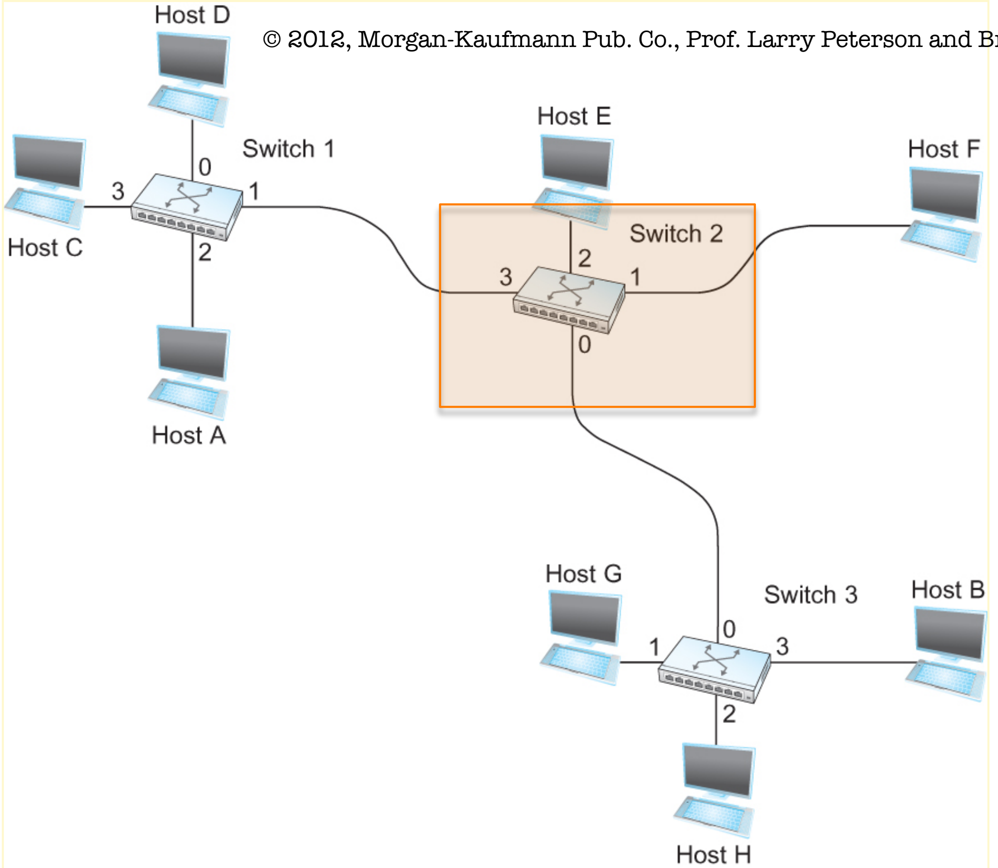
Switching and Forwarding

An example *network* (*ONE NETWORK*)

- To decide how to forward a packet, a switch consults a *forwarding table*



Switching and Forwarding



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

Forwarding Table for Switch 2

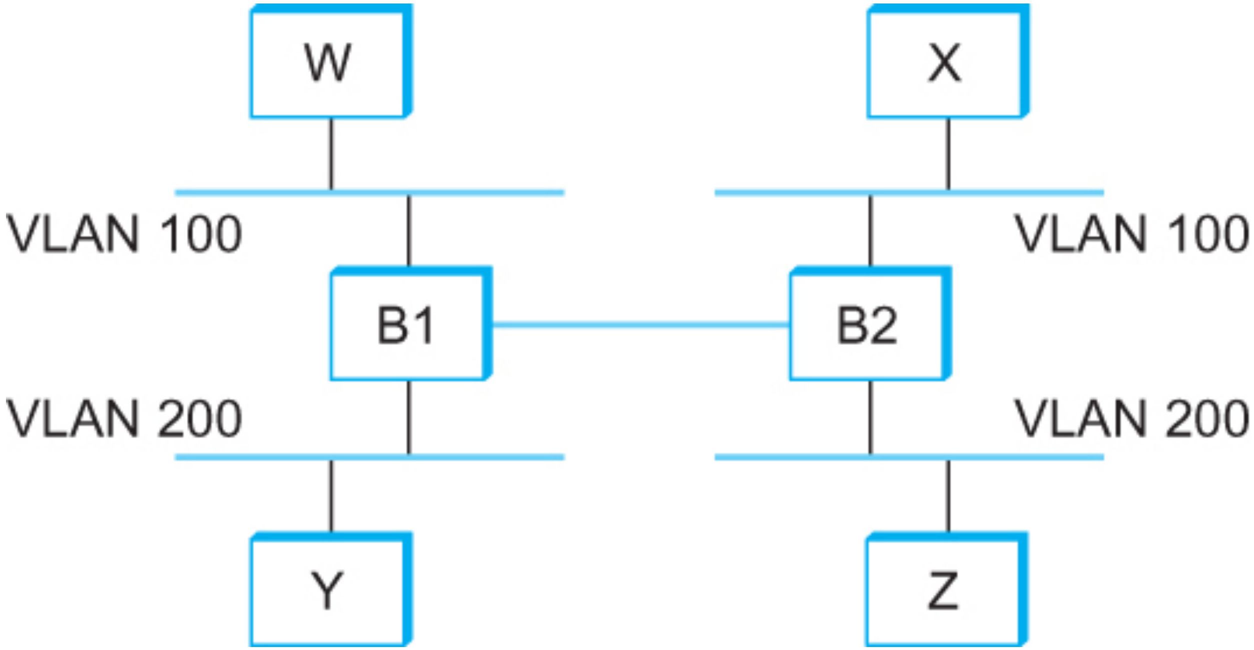
Switching and Forwarding

Connectionless (Datagram) Network

- A host can **send** a packet anywhere at **any time**
- Host sends a packet
  - 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**
  - Two **successive packets** from host A to host B
  - 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**

VLAN

- Virtual LAN



The end