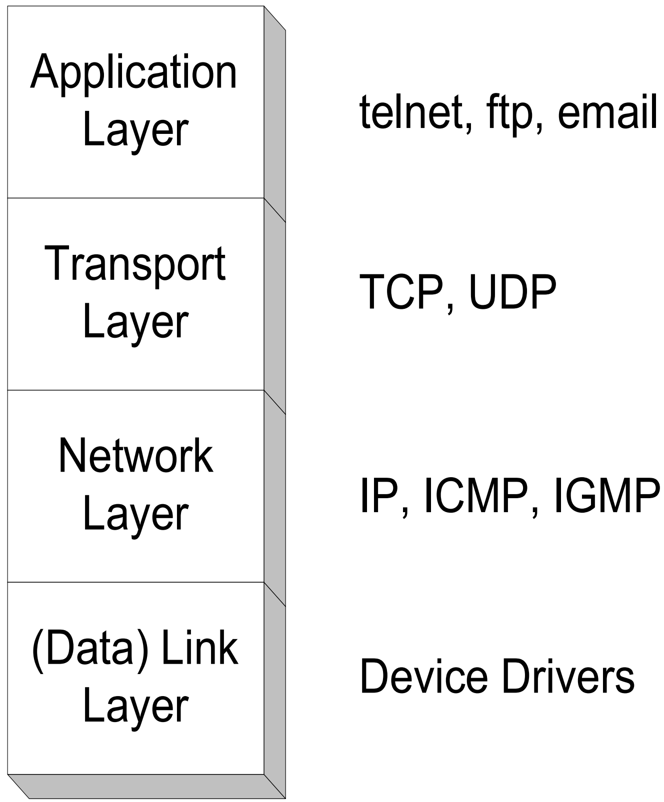
Ethernet Introduction

ECE 50863 – Computer Network Systems

Layered Protocol Architecture



Ethernet

- Most successful Local Area Network (LAN) technology
- Originally developed in the mid 1970s
- Standardization: IEEE 802.3 standard
- Earlier 10 Mbps, then 100Mbps (Fast Ethernet), and 1Gbps (Gigabit Ethernet)

Ethernet Frame Format

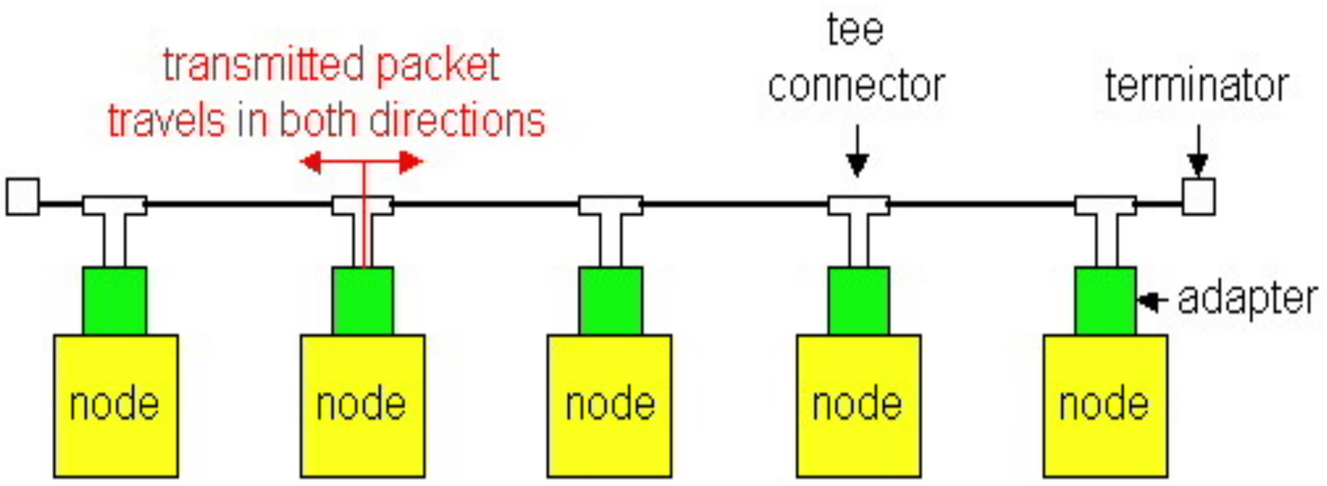
8	6	6	2			4
Preamble	**Dest**	**Source**	**Type**	**Data**	**Pad**	**CRC**

- Preamble marks the beginning of the frame.
  - Also provides clock synchronization
- Source and destination are 48 bit IEEE MAC addresses.
  - Flat address space
  - Globally unique: 24-bits reserved for vendor
- Type field is a demultiplexing field.
  - What network layer (layer 3) should receive this packet?
  - Format modified slightly in the 802.3 standard
- Cyclic Redundancy Check (CRC) for error checking.
- Data Field: At least 46 bytes, at most 1500 bytes

MAC vs. IP addresses

- MAC addresses: 6 bytes. Used by Layer 2 (e.g., Ethernet)
- IP addresses: 4 bytes. Used by Layer 3 (IP Layer)
- MAC: Flat Address space
- IP: Hierarchical address space.
  - Ensures scalability of Internet routing
- Analogy:
  - MAC address: like Social Security Number
  - IP address: like postal address
- Portability:
  - MAC address: portable: can move LAN card from one LAN to another
  - IP address NOT portable: depends on to which network attached

Ethernet: Broadcast medium 1



Ethernet Address Recognition

- Each frame contains destination address
- All hosts receive a transmission
- Host discards any frame addressed to another host
- Important: interface hardware, not software, checks address

- Packet can be sent to:
  - Single destination (unicast)
  - All stations on network (broadcast)
  - Subset of stations (multicast)

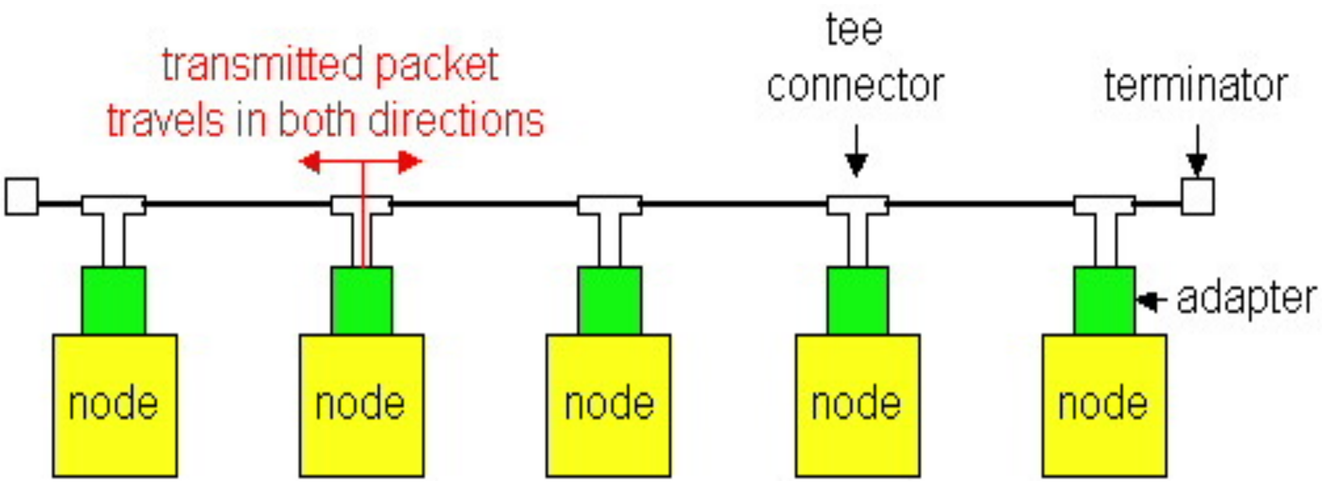
- All 1's: Broadcast address
- First bit 1, but not broadcast address: multicast address

- Promiscuous mode: Host can choose to accept all packets even if not destined to it

Ethernet: CSMA/CD Algorithm

ECE 50863 – Computer Network Systems

Ethernet: Broadcast medium 2



How do we prevent collisions (multiple hosts transmitting together?)

Multiple Access Protocols

- Distributed algorithm that determines how hosts should coordinate transmission.

- Key Objectives:
  - Efficiency
  - Fairness
  - Distributed: no single coordinator node to regulate things

- Multiple approaches possible
  - Example: Token ring
    - Pass a token around, and only allow a host to transmit if it has a token.

Ethernet's Approach

- When node has packet to send, goes ahead and transmits
  - no *a priori* coordination among nodes
- Two or more transmitting nodes -> "collision"
- Resolve collision using **Random access MAC protocol**
- This specifies:
  - when to transmit
  - how to detect collisions
  - how to recover from collisions

Evolution of Ethernet's approach

Aloha

Developed in the 1970s for a packet radio network

Slotted Aloha

Improvement: Start transmission only at fixed times (slots)

CSMA

CSMA = Carrier Sense Multiple Access

Improvement: Start transmission only if no transmission is ongoing

CSMA/CD

CD = Collision Detection

Improvement: Stop ongoing transmission if a collision is detected (e.g. Ethernet)

CSMA/CD Algorithm (used in Ethernet)

- Sense for carrier.
- If carrier present, wait until carrier ends.
  - Sending would force a collision and waste time
- Send packet and sense for collision.
- If no collision detected, consider packet delivered.
- Otherwise, abort immediately, perform “exponential back off” and send packet again.
  - Start to send at a random time picked from an interval
  - Length of the interval increases with every retransmission

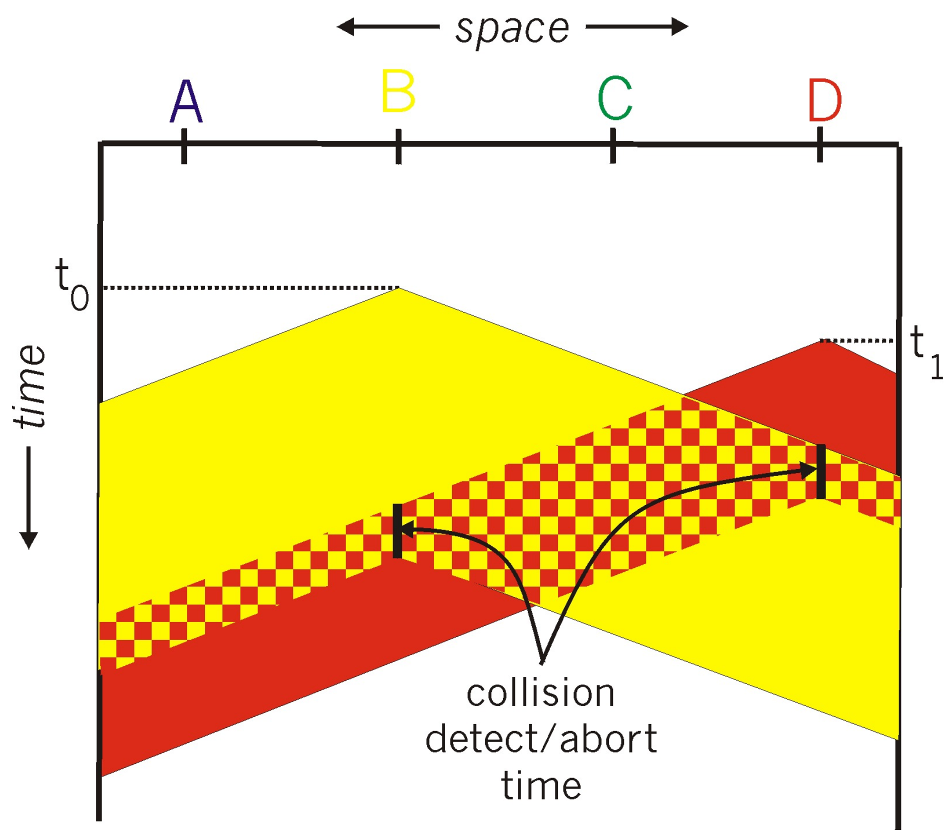
Exponential Backoff Algorithm

- Ethernet uses an exponential backoff algorithm to determine when a station can retransmit after a collision
- Helps adjust dynamically to the load on the system. Repeated collision => system highly loaded => less aggressive in retransmitting

Algorithm:

- After first collision wait 0 or 1 time units
  - Time unit => standard specified, 51.2 microseconds for 10Mbps Ethernet.
- After i-th collision, wait a random number between 0 and 2^i-1 time units
- Do not increase random number range, if i=10
- Give up after 16 collisions

CSMA/CD collision detection

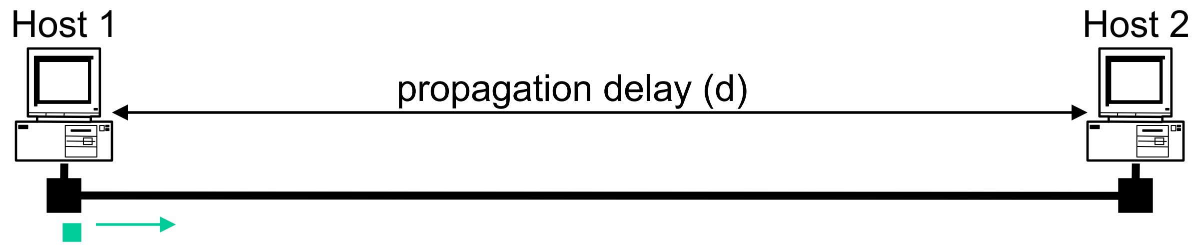


Minimum frame Size

- Why put a minimum frame size?
- Give a host enough time to detect collisions
- In Ethernet, minimum frame size = 64 bytes (two 6-byte addresses, 2-byte type, 4-byte CRC, and 46 bytes of data)
- If host has less than 46 bytes to send, the adaptor pads (adds) bytes to make it 46 bytes
- What is the relationship between minimum frame size and the length of the LAN?

Minimum Frame Size (more) 1

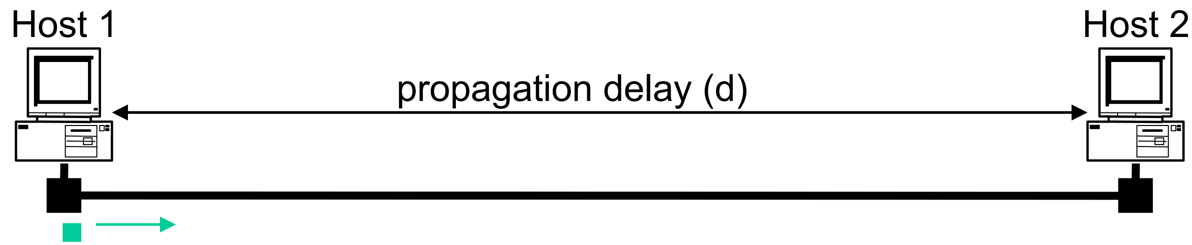
a) Time = t; Host 1 starts to send frame



Question: When is the latest that Host 2 could start transmitting data which would result in a collision with Host 1?

Minimum Frame Size (more) 2

a) Time = t; Host 1 starts to send frame

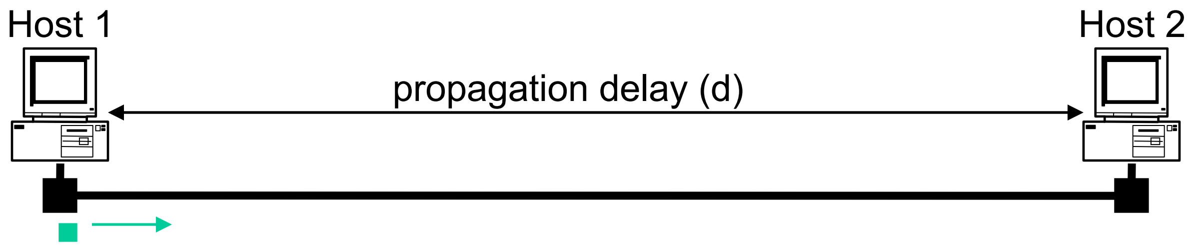


Question: When is the latest that Host 2 could start transmitting data which would result in a collision with Host 1?

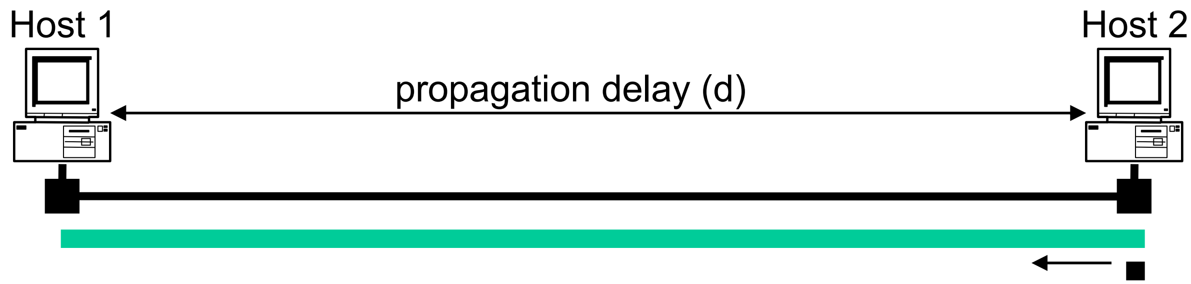
Answer: t+d, because after that point Host 2 would sense Host 1's signal (i.e., sense carrier), and will not start transmitting.

Minimum Frame Size (more) 3

a) Time = t; Host 1 starts to send frame



b) Time = t + d; Host 2 starts to send a frame just before it hears from host 1's frame

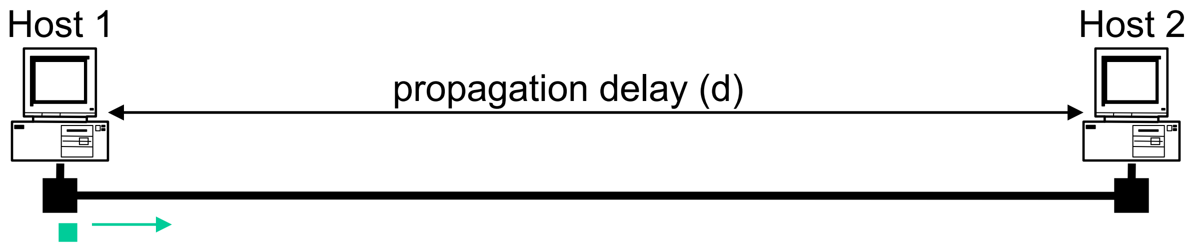


Question: If Host 2 started transmitting at t+d, when would Host 1 sense collision?

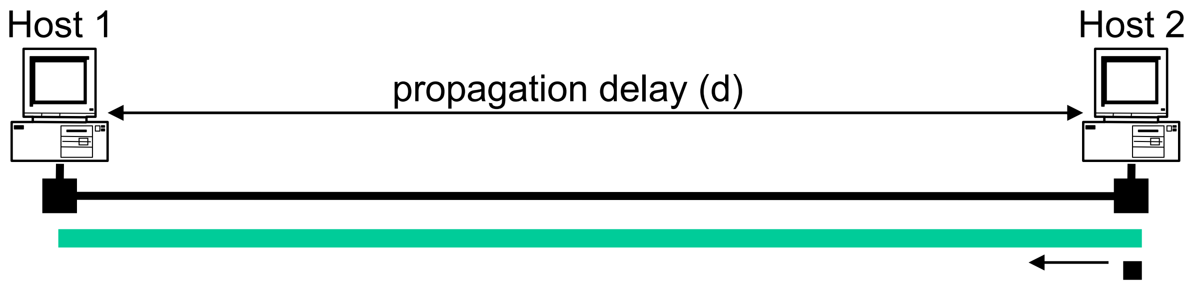
Answer: t+2d

Minimum Frame Size (more) 4

a) Time = t; Host 1 starts to send frame



b) Time = t + d; Host 2 starts to send a frame just before it hears from host 1's frame



Question: If Host 2 started transmitting at t+d, when would Host 1 sense collision?

Answer: t+2d

Question: What is the minimum time that Host 1 should keep transmitting to ensure it can detect collision?

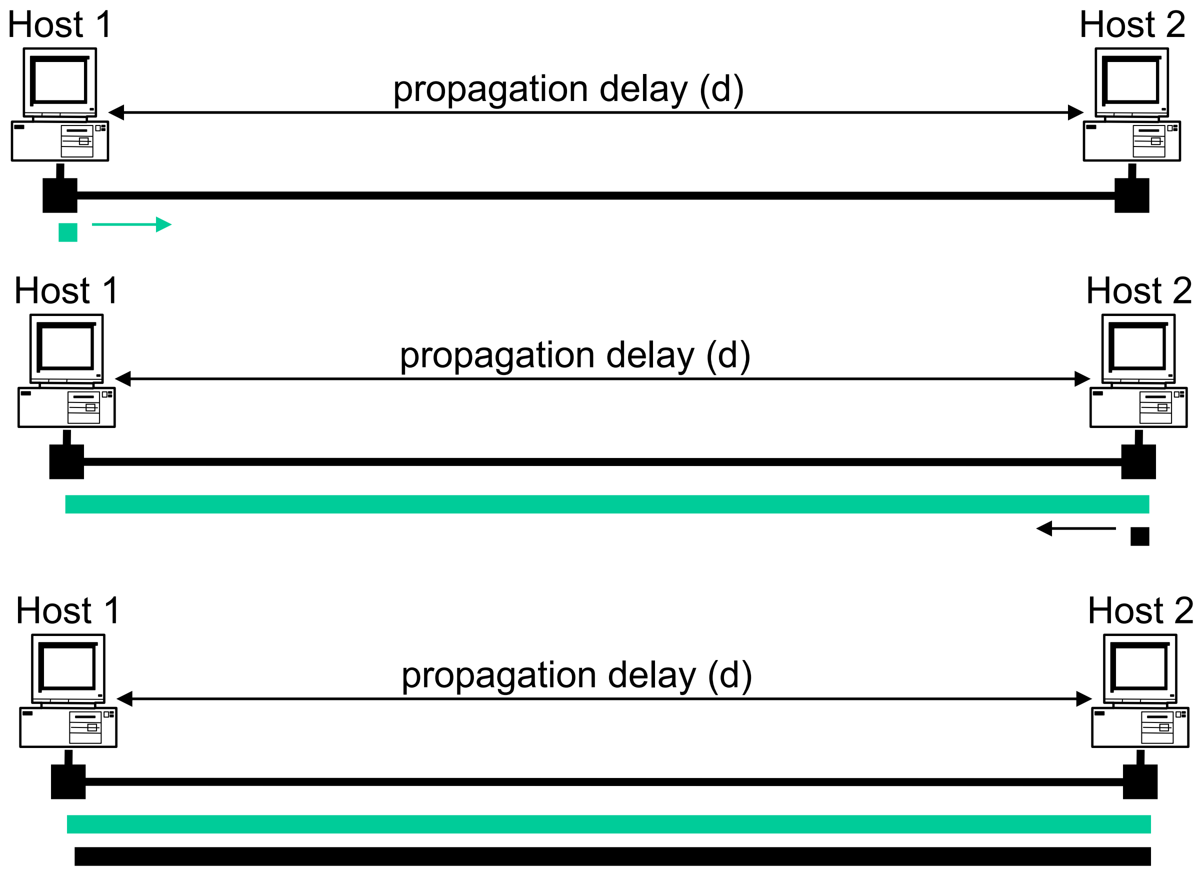
Answer: 2d

Minimum Frame Size (more) 5

a) Time = t; Host 1 starts to send frame

b) Time = t + d; Host 2 starts to send a frame just before it hears from host 1's frame

c) Time = t + 2*d; Host 1 hears Host 2's frame → detects collision



Host 1 must not finish transmission before Host 2's signal seen

Deriving Minimum Frame Size relation

- Host 1 must transmit for at least time 2d

MinFrameSize/bandwidth > 2 * d

MinFrameSize/bandwidth > 2 * (LAN-length)/(propagation-speed)

Rather than increase minimum frame size with bandwidth, Ethernet reduces the max permissible length

LAN length <

(MinFrameSize)*(propagation-speed)/(2*bandwidth)

= (8*64b)*(2*10^8mps)/(2*10^7 bps)

= 5.12 km

Ethernet Interconnects: Types of Interconnects

ECE 50863 – Computer Network Systems

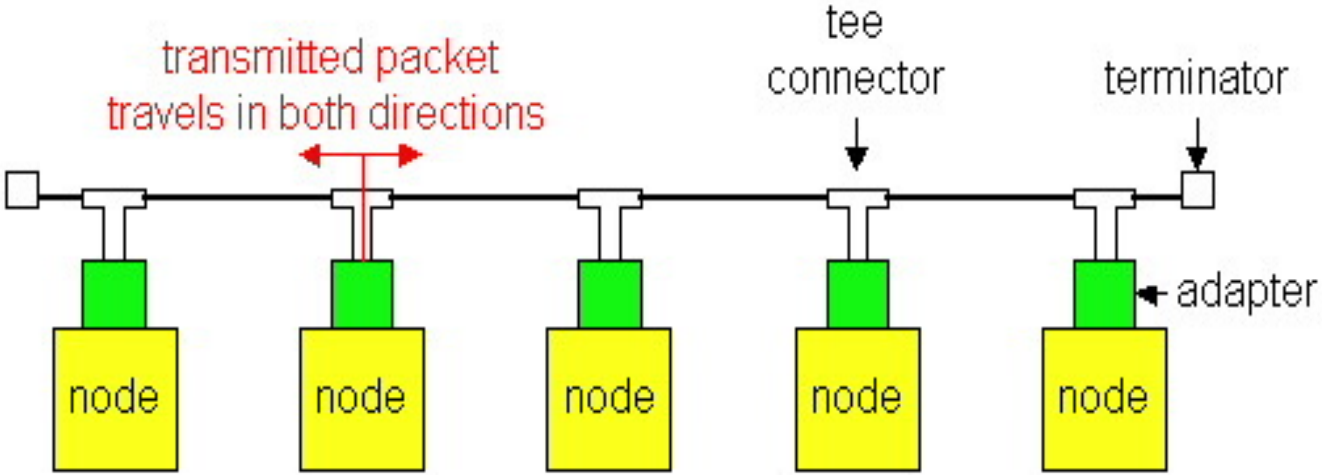
Building large Ethernet networks

- Discussion so far:
  - Single broadcast cable, lots of hosts connected to it
- Does not work well under high load
  - Limitation on the physical distance between the nodes
- How do we get a large campus connected?

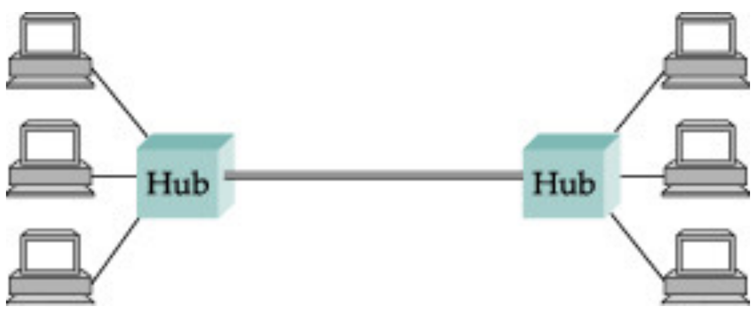
Classes of Ethernet Interconnects

- Repeaters/Hubs:
  - Dumb physical layer device that forwards digital signals
- Bridges/switches:
  - More intelligent devices that forward data only to hosts needing them

Diagram



transmitted packet
travels in both directions
tee
connector
terminator
adapter
node
node
node
node
node



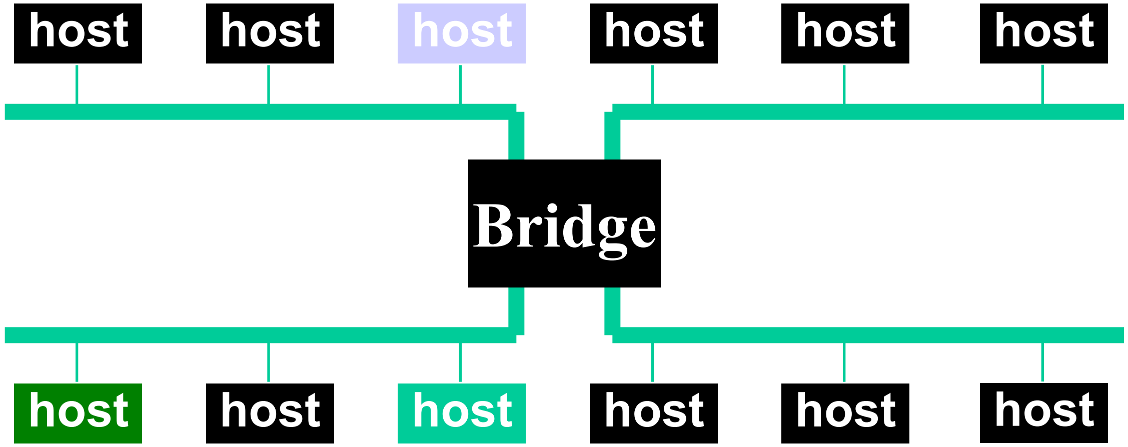
Hub
Hub

Collision Domains

- Collision Domain:
  - Data transmitted by host reaches all other hosts.
  - All hosts compete for access to same link, and only one can transmit at any given time.
- Hosts on a single Ethernet segment are in the same collision domain.
- Hosts separated by repeaters/hubs are also in the same collision domain
- Hosts separated by switches/bridges are in DIFFERENT collision domains.

Building Larger LANs: Bridges

- Bridges connect multiple IEEE 802 LANs at layer 2.
  - Only forward packets to the right port
  - Reduce collision domain compared with single LAN



Transparent Bridges

- Overall design goal:
  - “Plug-and-play”
  - Self-configuring without hardware or software changes
  - Bridges should not impact operation of existing LANs

- Three parts to transparent bridges:
  (1) Forwarding of Frames
  (2) Learning of Addresses
  (3) Spanning Tree Algorithm

Frame Forwarding

- Each bridge maintains a **forwarding database** with entries
  `< MAC address, port, age>`

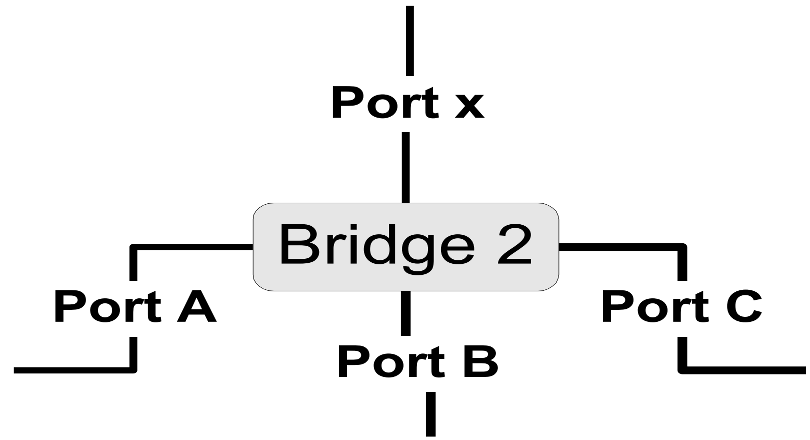
`MAC address:`	host name or group address
`port:`	port number of bridge
`age:`	aging time of entry

with interpretation:

- a machine with **`MAC address`** lies in direction of the **`port`** number from the bridge. The entry is **`age`** time units old.

Frame Forwarding 2

- Assume a MAC frame arrives on port x.



Search if MAC address of destination is listed for ports A, B, or C.

Found?

Forward the frame on the appropriate port

Not found ?

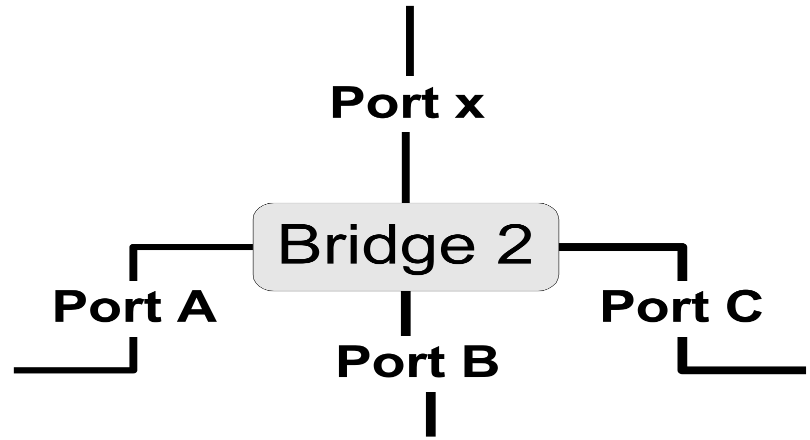
Flood the frame, i.e., send the frame on all ports except port x.

Ethernet Interconnects: Bridge Learning Algorithm

ECE 50863 – Computer Network Systems

Recap: Frame Forwarding 2

- Assume a MAC frame arrives on port x.



Search if MAC address of destination is listed for ports A, B, or C.

Found?

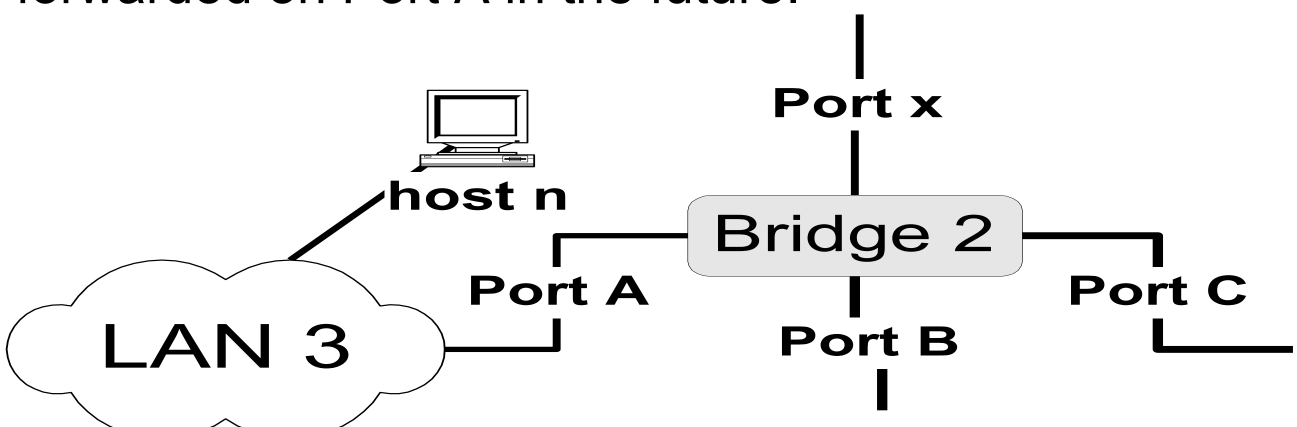
Forward the frame on the appropriate port

Not found ?

Flood the frame, i.e., send the frame on all ports except port x.

Address Learning

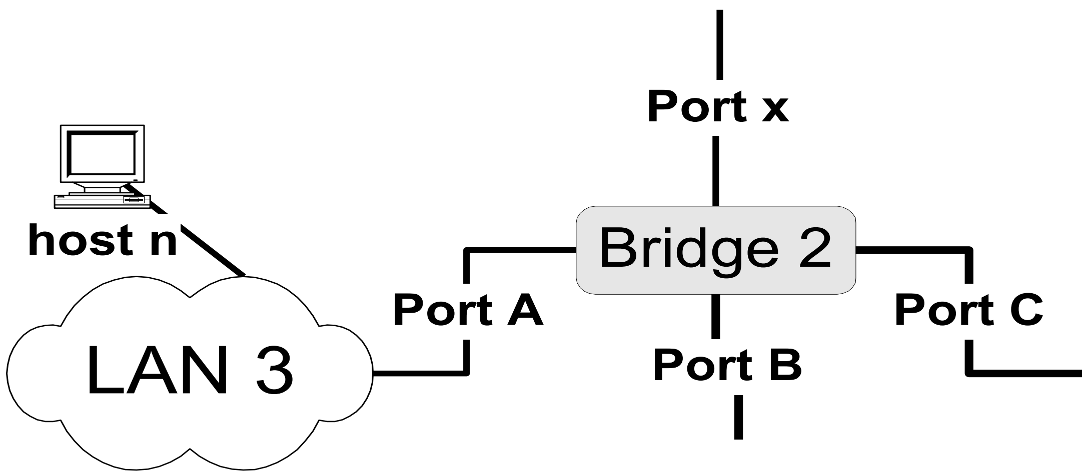
- One approach: manually create tables
  - Need to be adjusted each time a new host is added, or host is moved around in the network.
- Learning algorithm: automatically learn addresses
  - The source field of a frame that arrives on a port tells the switch which hosts are reachable from this port
  - If a packet from source n to destination d arrives on port A, Bridge 2 learns that packets with destination n should be forwarded on Port A in the future.



Address Learning 2

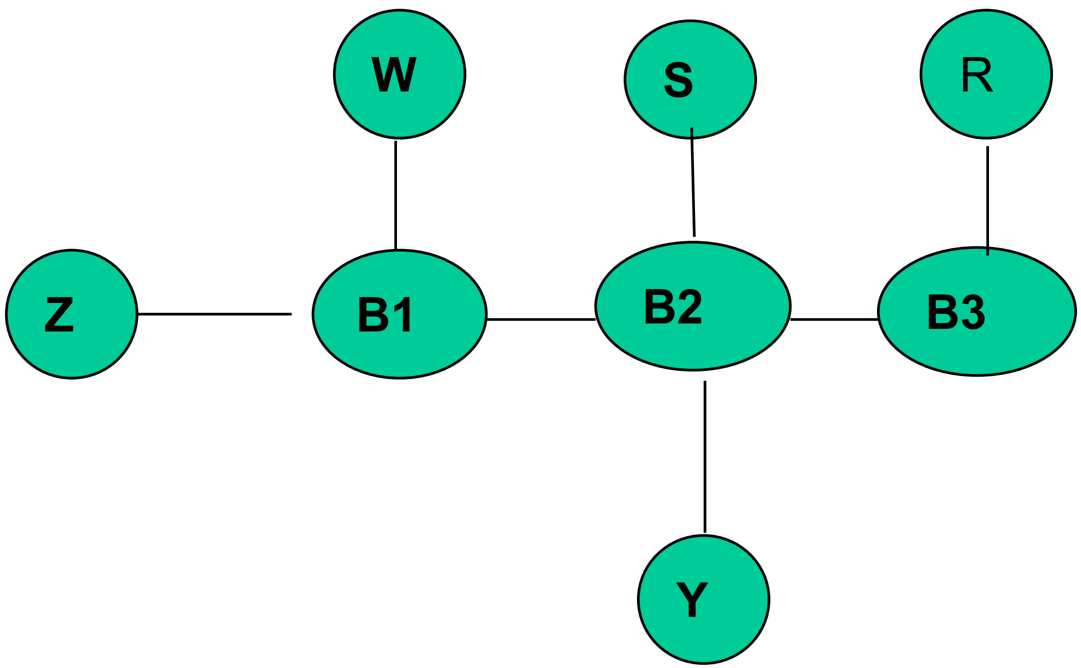
Algorithm:

- For each frame received, the bridge stores the source field in the forwarding database together with the port where the frame was received.
- All entries are deleted after some time



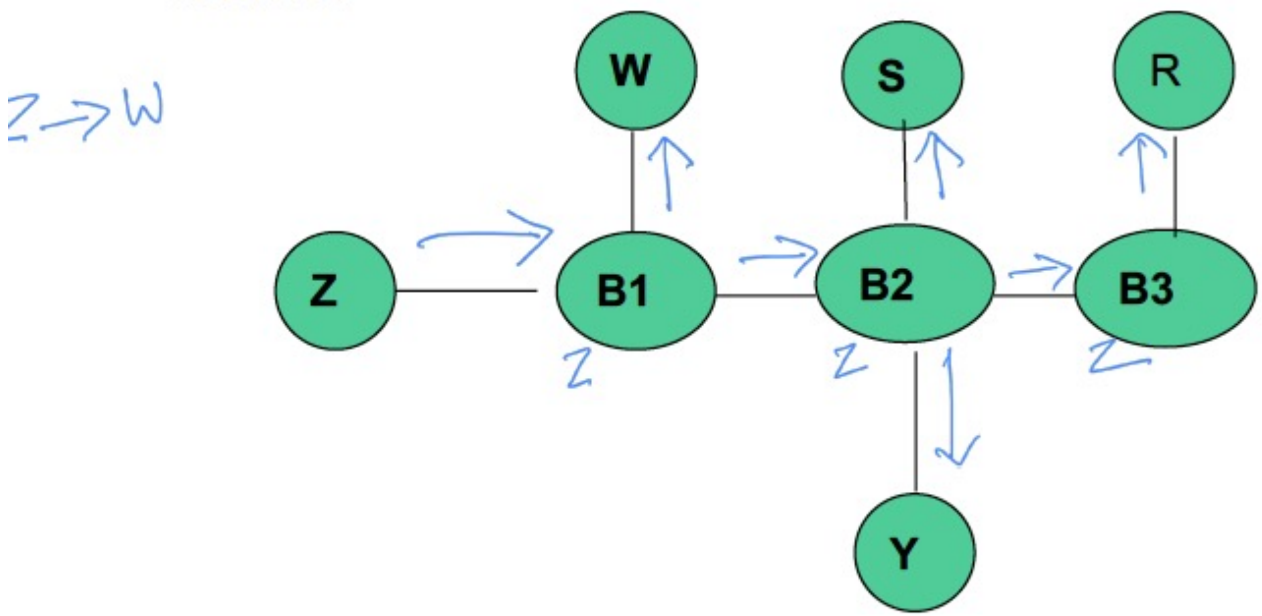
Example 1

- B1, B2, B3 are bridges; Z, W, S, R, Y are hosts.
- Bridges start with empty tables.
- Z sends a packet to W; Next S sends to Z. Simulate what happens at each step, and what each bridge learns.



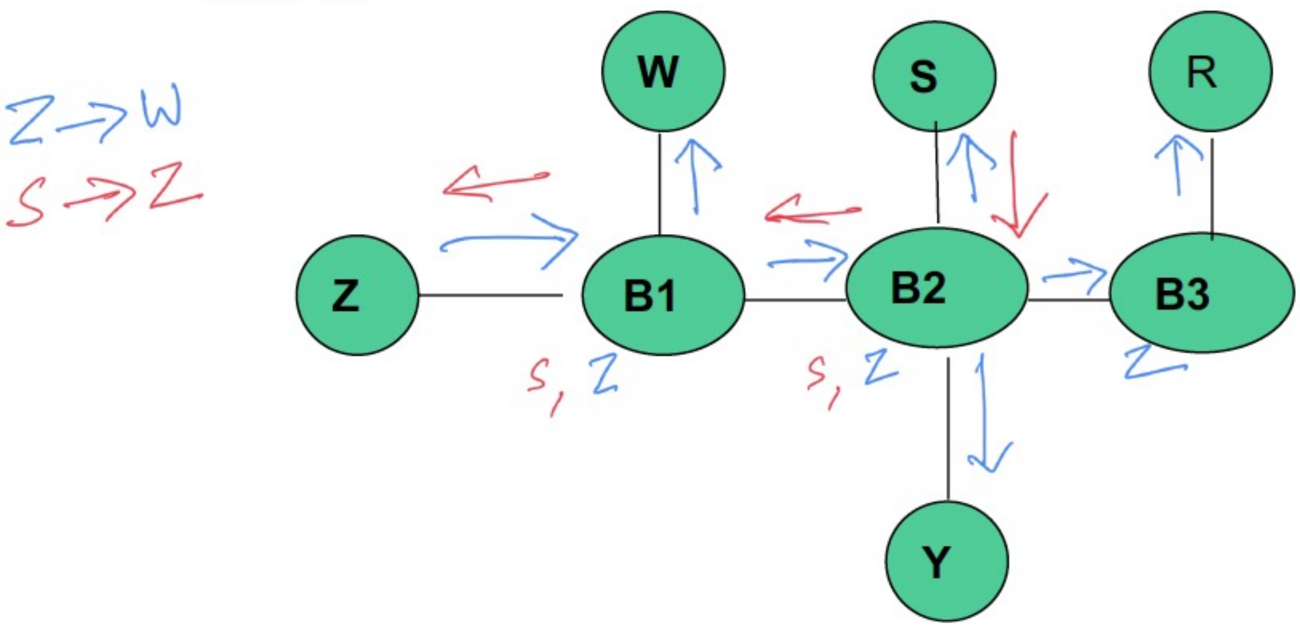
Example

- B1, B2, B3 are bridges; Z, W, S, R, Y are hosts.
- Bridges start with empty tables.
- Z sends a packet to W; Next S sends to Z. Simulate what happens at each step, and what each bridge learns.



Example

- B1, B2, B3 are bridges; Z, W, S, R, Y are hosts.
- Bridges start with empty tables.
- Z sends a packet to W; Next S sends to Z. Simulate what happens at each step, and what each bridge learns.



Ethernet Interconnects: Spanning Tree Algorithm

ECE 50863 – Computer Network Systems

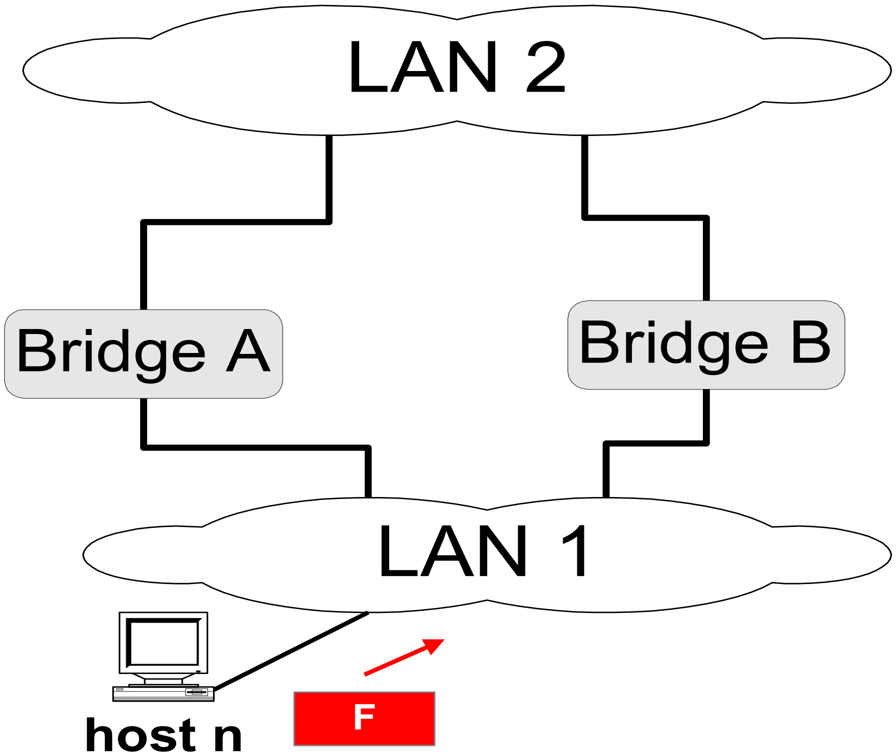
Issue with bridges in topologies with loops 1

- Consider the two LANs that are connected by two bridges.
- Assume *host n* is transmitting a frame F to destination *d* for which entries are unavailable.

What happens?

- Bridges A and B flood the frame to LAN 2.
- Bridge B sees F on LAN 2 (with destination *d*), and copies the frame back to LAN 1
- Bridge A does the same.
- The copying continues

Where's the problem? What's the solution ?



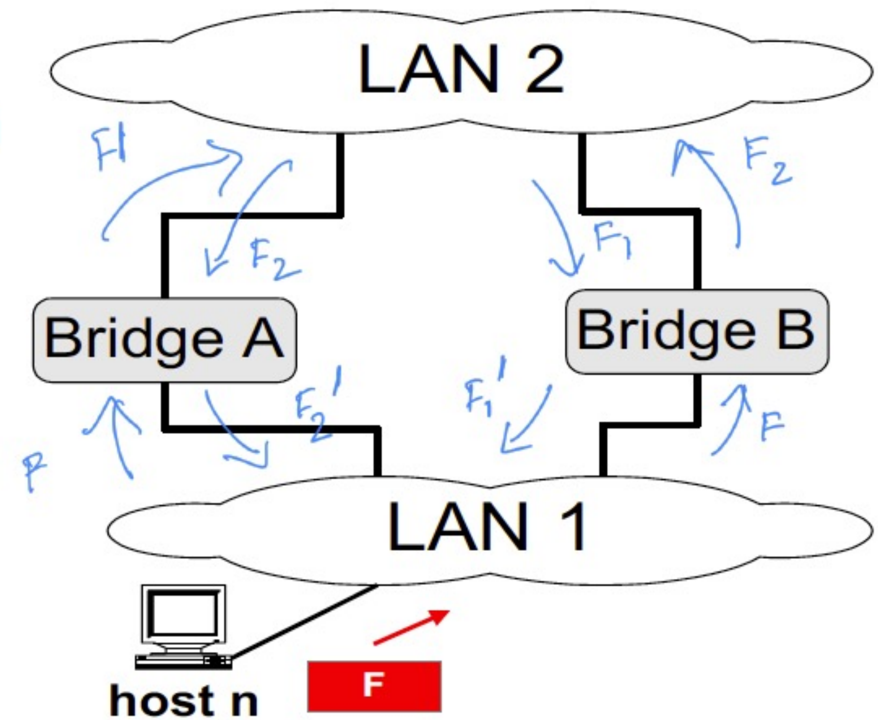
Issue with bridges in topologies with loops

- Consider the two LANs that are connected by two bridges.
- Assume *host n* is transmitting a frame F to destination *d* for which entries are unavailable.

What happens?

- Bridges A and B flood the frame to LAN 2.
- Bridge B sees F on LAN 2 (with destination *d*), and copies the frame back to LAN 1
- Bridge A does the same.
- The copying continues

Where's the problem? What's the solution ?



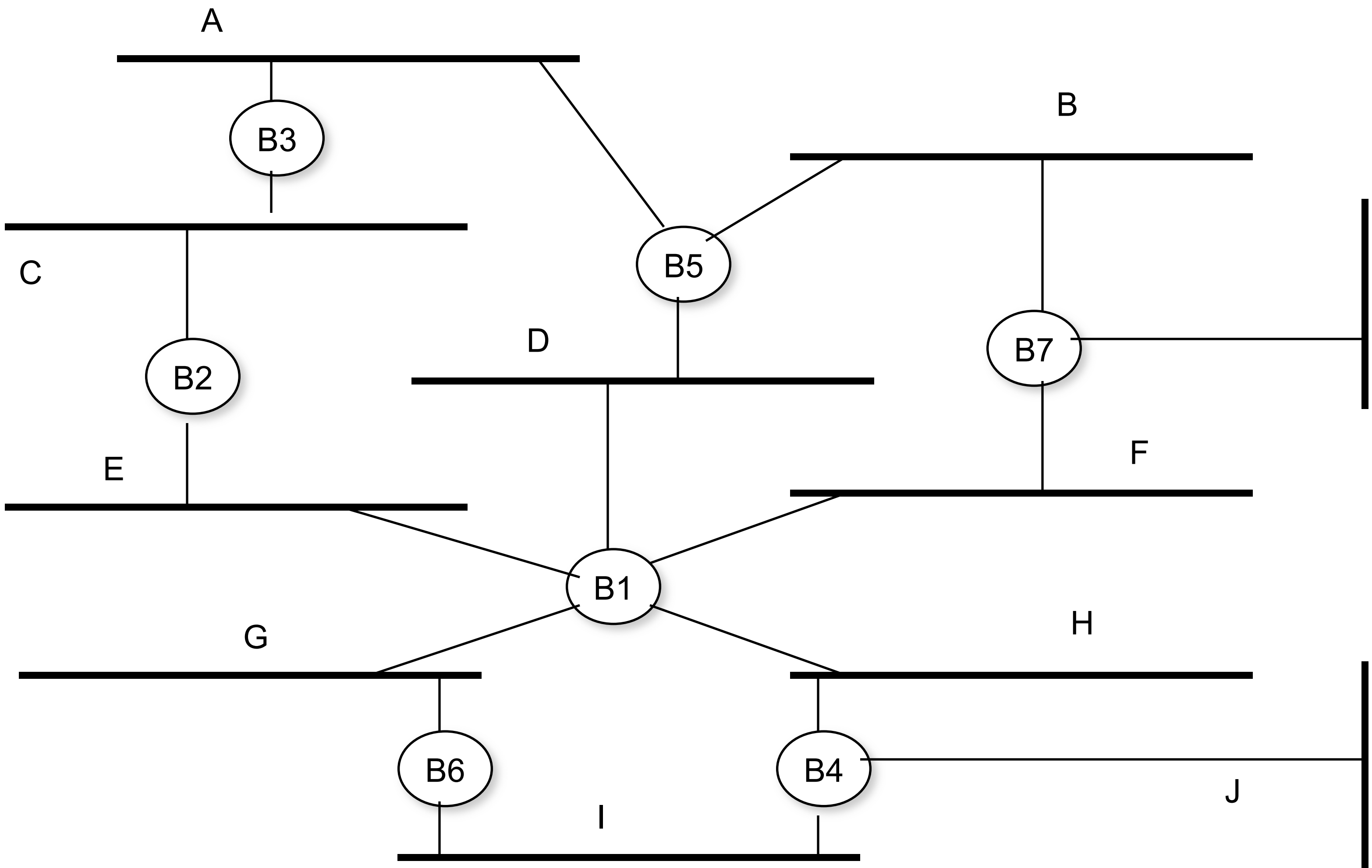
Spanning Trees

- Eliminate loops in the topology
- Network itself may have redundancy to handle failures
- Spanning Tree algorithm disables links temporarily
  - May reenable links if failures occur.
- Start by discussing “centralized” version of algorithm
  - Full topology information is available
  - Later discuss “distributed version”

Connections to "graph theory"

- Ideas connected to branch of mathematics called "graph theory"
- Graph:
  - Set of vertices (nodes), and edges that connect pairs of nodes.
- Spanning Tree of a graph
  - Has all the vertices of the original graph, and a subset of edges
  - Resulting graph is connected, but does not have loops.
- Many possible spanning trees for a graph.

A Bridged Network 1



Key ideas

- Elect a single bridge as the **root bridge**.
- Calculate the distance of the shortest path to the root bridge
- Each LAN can determine a **designated bridge**, which is the bridge closest to the root. The designated bridge will forward packets towards the root bridge.
- Each bridge can determine a **root port**, the port that gives the best path to the root.
- Select ports to be included in the spanning tree

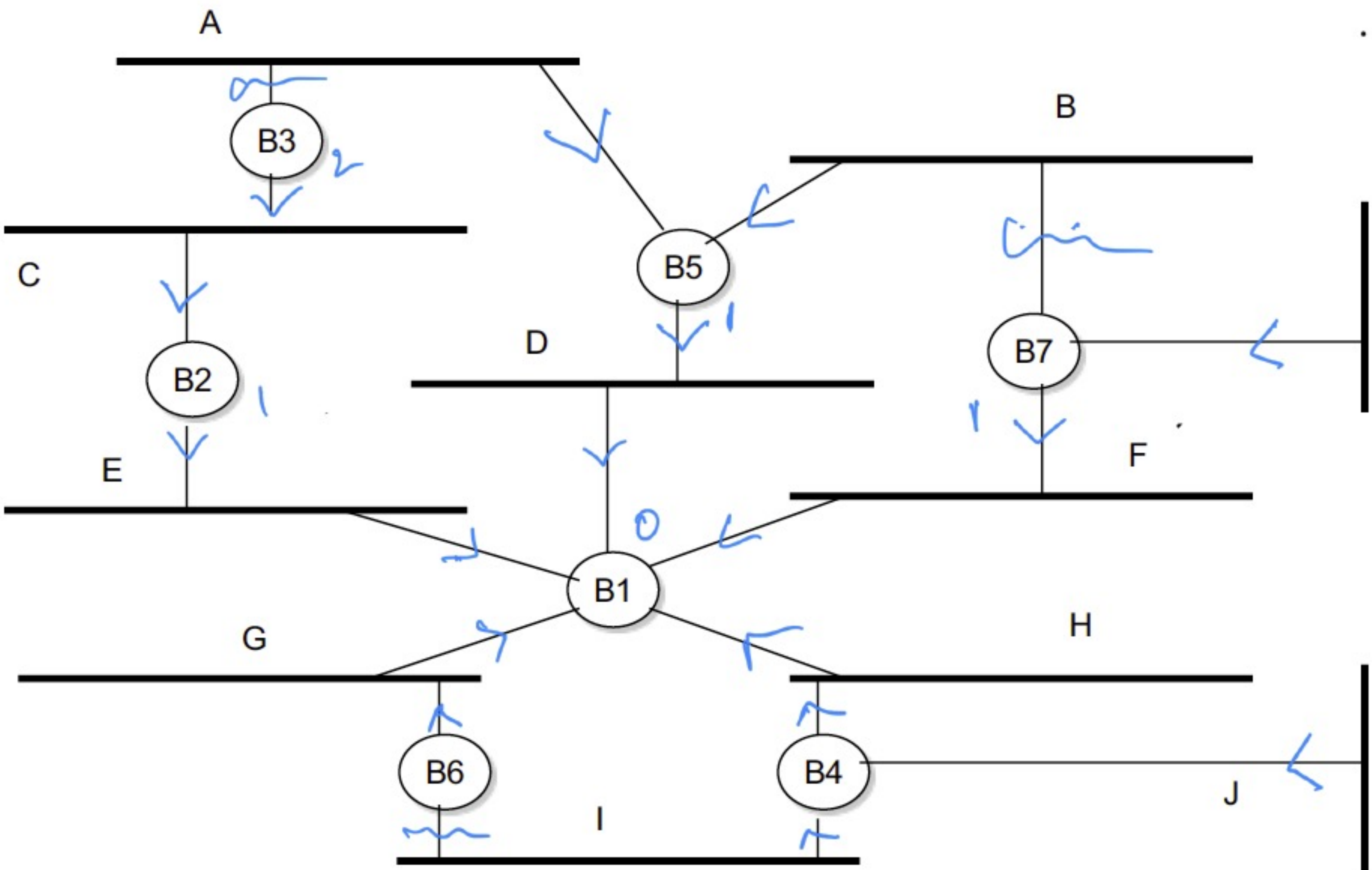
Concepts

- Each bridge has a unique identifier: Bridge ID
- Each port within a bridge has a unique identifier (port ID).

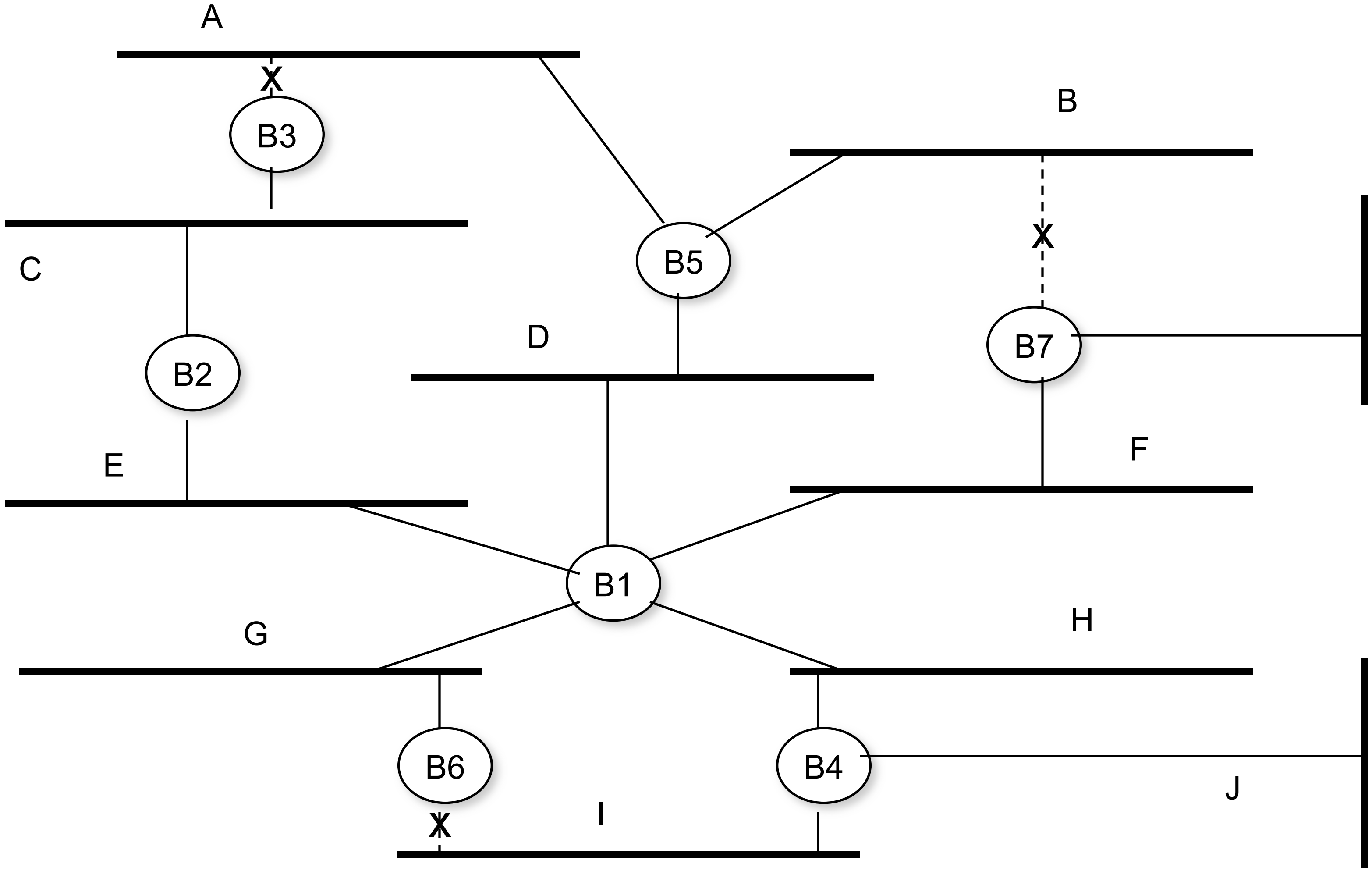
- **Root Bridge:** The bridge with the lowest identifier is the root of the spanning tree.
- **Root Port**: Each bridge has a root port which identifies the next hop from a bridge to the root.
- **Root Path Cost**: For each bridge, the cost of the min-cost path to the root, measured in #Hops to the root.
- **Designated Bridge**
  - Bridge on LAN that provides minimal cost path to root for this LAN
  - if two bridges have the same cost, select one with lower identifier
- **Designated Port:**
  - If a bridge is the designated bridge for a LAN, the appropriate port is the designated port.

- Bridge disables all ports which are neither root ports, nor designated ports.

A Bridged Network



A Bridged Network (End of Spanning Tree Computation) 1



Ethernet Interconnects: Spanning Tree Algorithm: Distributed Version

ECE 50863 – Computer Network Systems

Centralized Vs. Distributed

- So far: centralized version of spanning tree algorithm.
  - Central entity has full picture of entire topology
  - In reality, each bridge has very limited information
- Need for Distributed algorithms
  - Bridges exchange messages with each other
    - Referred to as “Bridge Protocol Data Unit (BPDU)”
  - Messages enable them to learn information needed to make the right decisions.
- Topology may not be correct initially, eventually gets to the right one.

Steps of Spanning Tree Algorithm

1. Determine the root bridge

2. Determine the root port on all other bridges

3. Determine the designated port on each LAN

- Each bridge sends out BPDUs that contain the following information:

root ID	cost	bridge ID/port ID

root bridge (what the sender thinks it is)
root path cost for sending bridge
Identifies sending bridge

Initialization and Operation

- Initially, all bridges assume they are the root bridge.
- Each bridge B sends messages of this form on all its ports:

B	0	B

- Upon receiving message on a port
  - Bridge checks if received message is “better” than best recorded for that port (initially its own message is best).
  - If “better”, it discards old information for that port, and saves new information, adding 1 to the distance field.

Ordering of Messages

- We can order messages with the following ordering relation “**>**“ (let’s call it “lower cost” or “better”):

M1	ID R1	C1	ID B1	>	ID R2	C2	ID B2	**M2**

If (R1 < R2)

M1 > M2

elseif ((R1 == R2) and (C1 < C2))

M1 > M2

elseif ((R1 == R2) and (C1 == C2) and (B1 < B2))

M1 > M2

else

M2 > M1

Determining Root Bridge, Root Port

- If bridge receives message from lower bridge id:
  - It knows it is not the root.
  - It updates its belief of who the root is, say R.
- Bridge B determines the Root Path Cost (Cost) as follows:
  - *If B = R :* Cost = 0.
  - *else:* Cost = {Smallest Cost in any of BPDUs that were received from R} + 1
- **B's root port** is the port from which B received the lowest cost path to R (in terms of relation "**>**").
- Knowing R and Cost, B can generate its BPDU (but will not necessarily send it out):

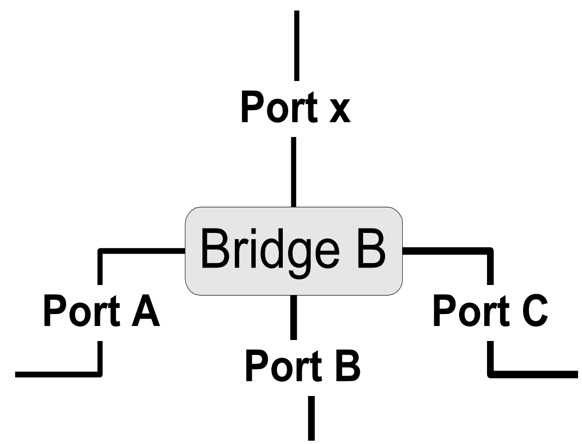
R	Cost	B

Calculate the Root Path Cost
Determine the Root Port

- At this time: B has generated its message

R	Cost	B

- B will send this BPDU on one of its ports, say **port x**, only if its message is lower (via relation "**>**") than any message that B received from port x.
- In this case, B also assumes that it is the **designated bridge** for the LAN to which the port connects.



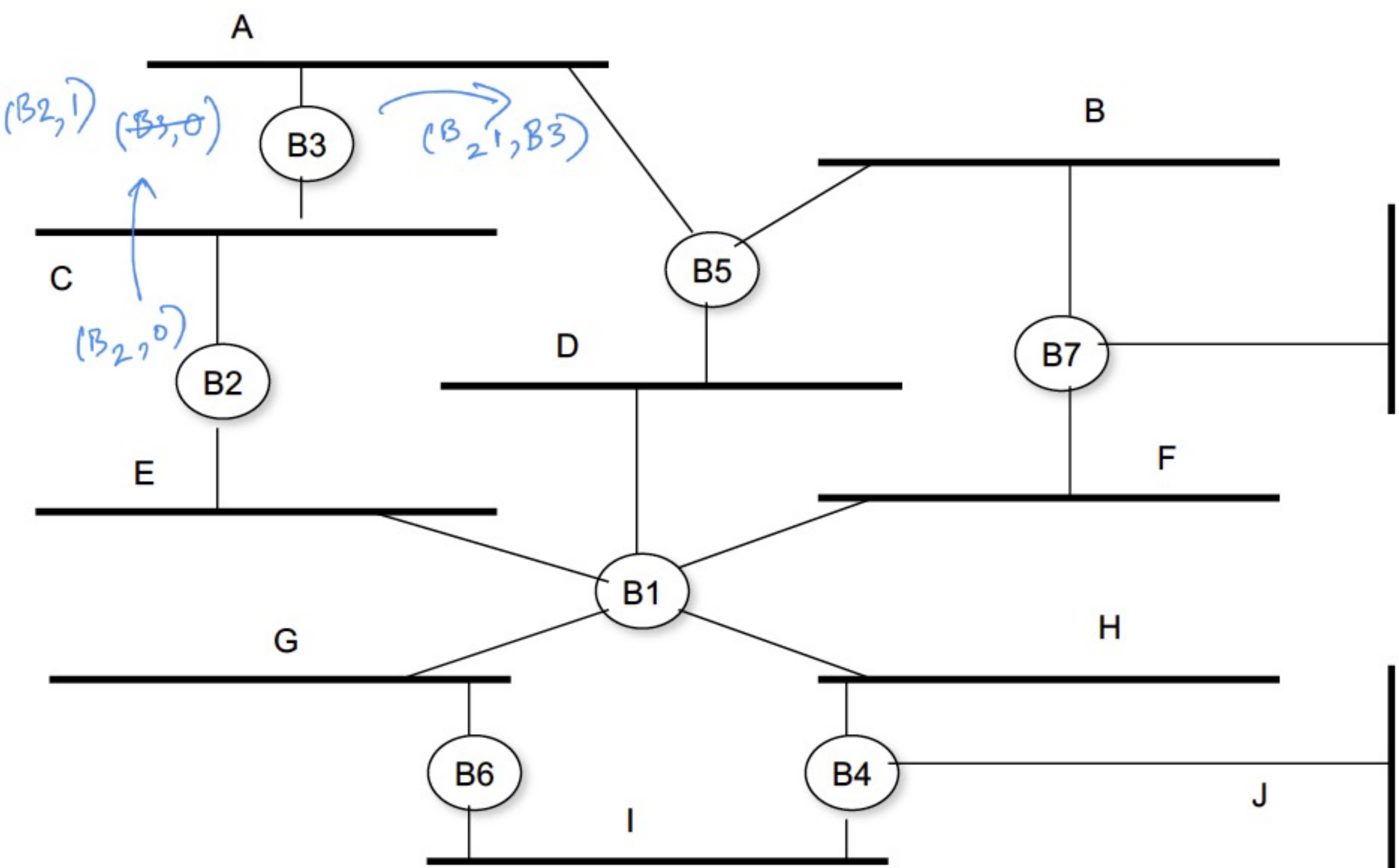
Selecting the Ports for the Spanning Tree

- At this time: Bridge B has calculated the root, the root path cost, and the designated bridge for each LAN.
- Now **B can decide which ports are in the spanning tree:**
    - B's root port is part of the spanning tree
    - All ports for which B is the designated bridge are part of the spanning tree.
- B's ports that are in the spanning tree will forward packets **(=forwarding state)**
- B's ports that are not in the spanning tree will not forward packets **(=blocking state)**

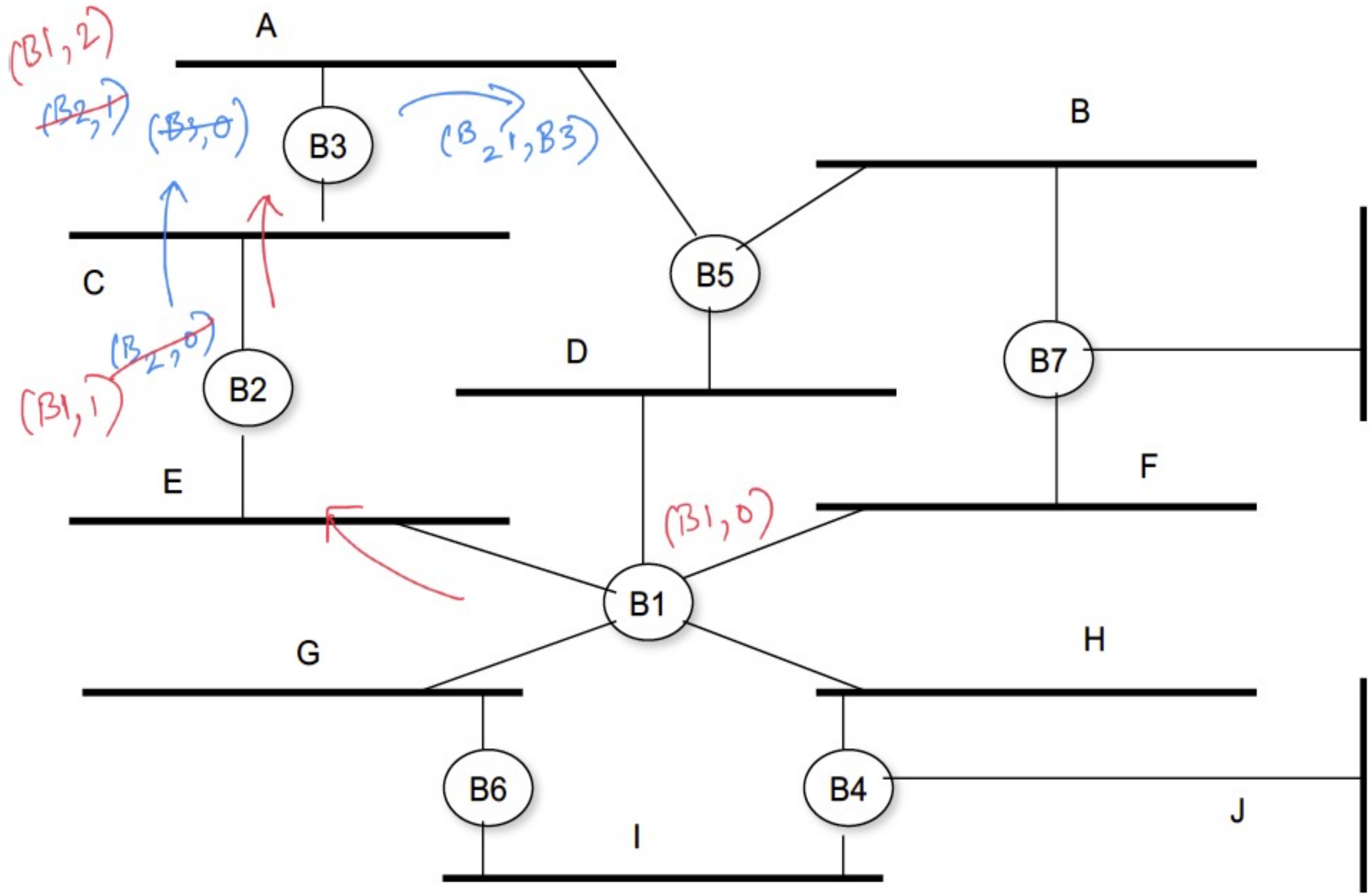
Example 4

- B3 rcvs (B2,0,B2). What happens?
- Makes B2 root. Send what to B5?
  - Sends (B2,1,B3) to B5
- Meanwhile B2 accepts B1 as root
  - Sends (B1,1,B2) to B3
- B3 updates its root to: B1
- Meanwhile B5 accepts B1 as root
  - Sends (B1,1,B5) to B3
- B3: no change to value of root

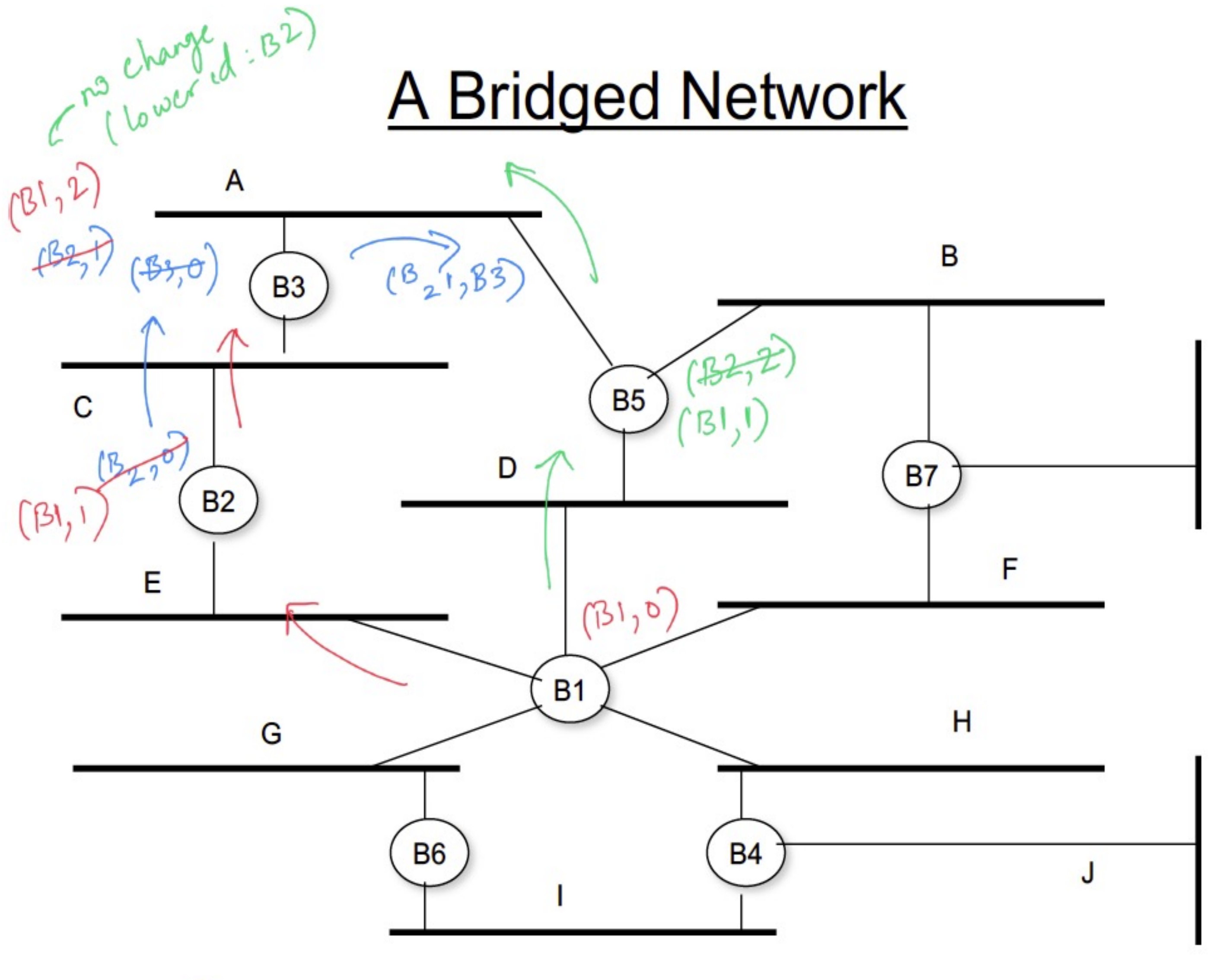
A Bridged Network



A Bridged Network



A Bridged Network



A Bridged Network (End of Spanning Tree Computation) 3