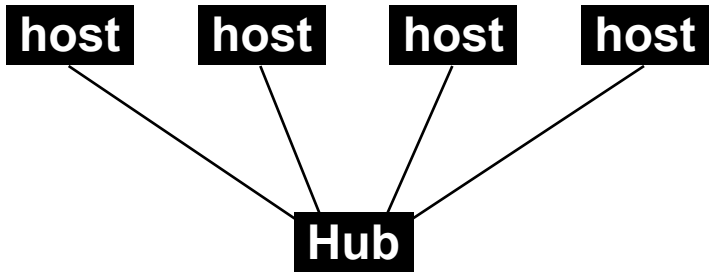
COMP/ELEC 429/556
Introduction to Computer Networks

Scaling Broadcast Ethernet

Some slides used with permissions from Edward W. Knightly, T. S. Eugene Ng, Ion Stoica, Hui Zhang

Recap

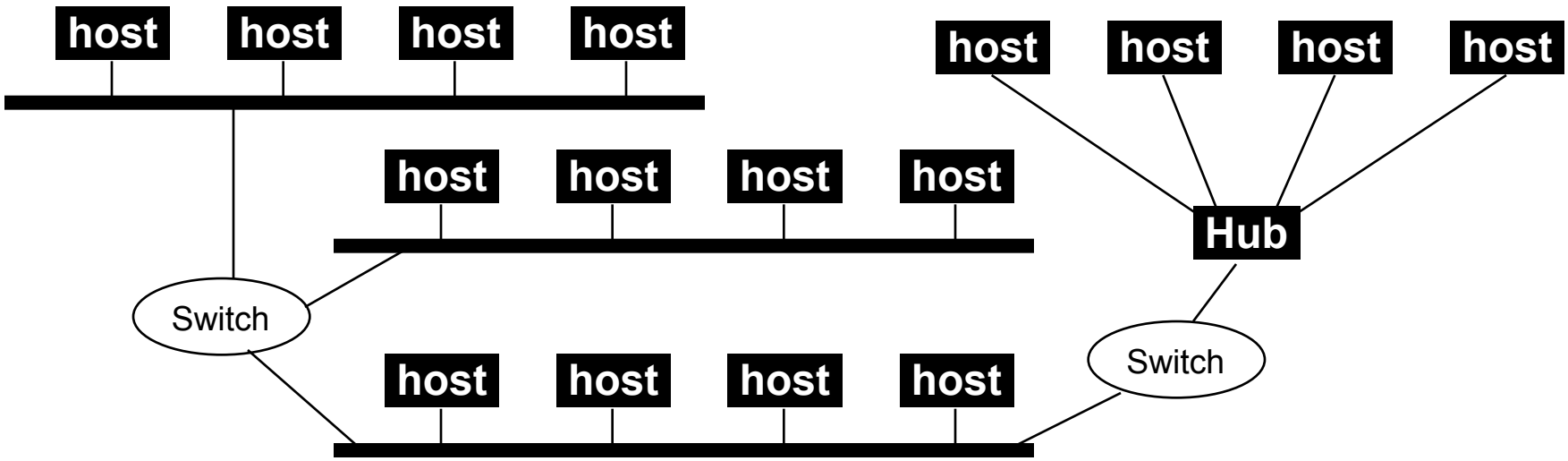
Broadcast technology



Hub (or repeater) emulates a broadcast channel
Easy to add a new host

- Broadcast network is a simple way to connect hosts
  - Everyone hears everything
- Need MAC protocol to control medium sharing
- Problem: Cannot scale up to connect large number of nodes
  - Too many nodes, too many collisions, goodput (throughput of useful data) goes to zero

Need Switching

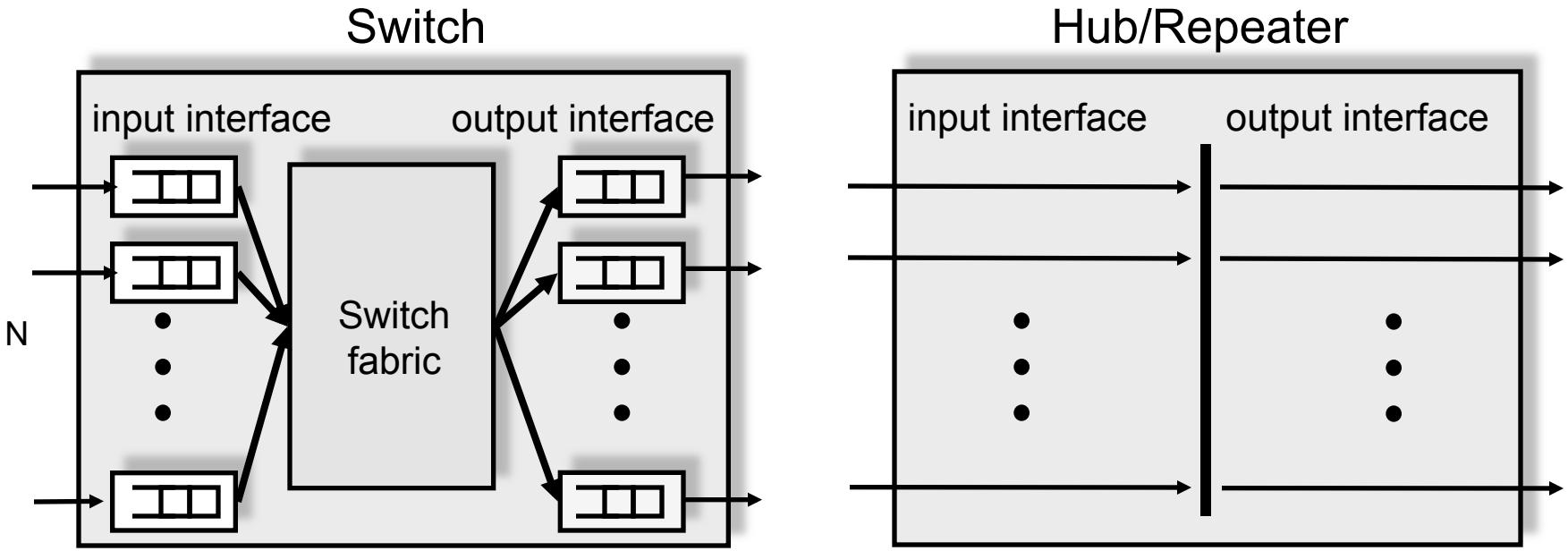


- Switching limits size of collision domains, allows network size to scale up
  - To how big?
  - Will return to this question

Switch

48-port 10Gbps + 4-port 40Gbps switch costs ~$3000

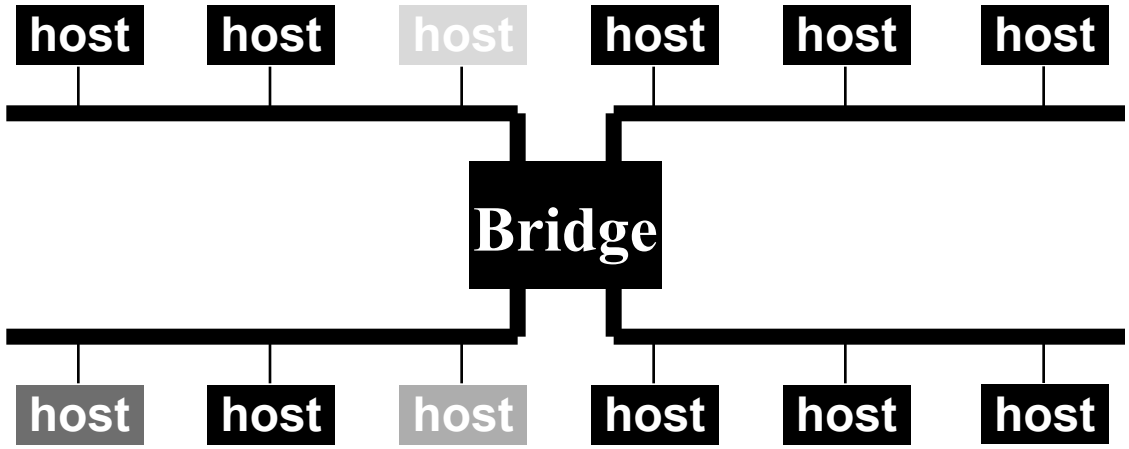
Switch



- Switch has memory buffers to queue packets, reduce loss
- Switch is intelligent: Forward an incoming packet to the correct output interface only
- High performance: Full N x line rate possible

Ethernet Switches are also called Bridges

- Bridges connect multiple broadcast Ethernet segments
  - Only forward packets to the right port
  - Reduce collision domain
- In contrast, hubs rebroadcast packets.



Bridges

- Overall design goal: **Complete transparency**
  - "Plug-and-play"
  - Self-configuring without hardware or software changes
  - Bridges should not impact operation of existing networks

Packet Forwarding

- Each bridge maintains a **forwarding database** with entries

  `< MAC address, port, age>`

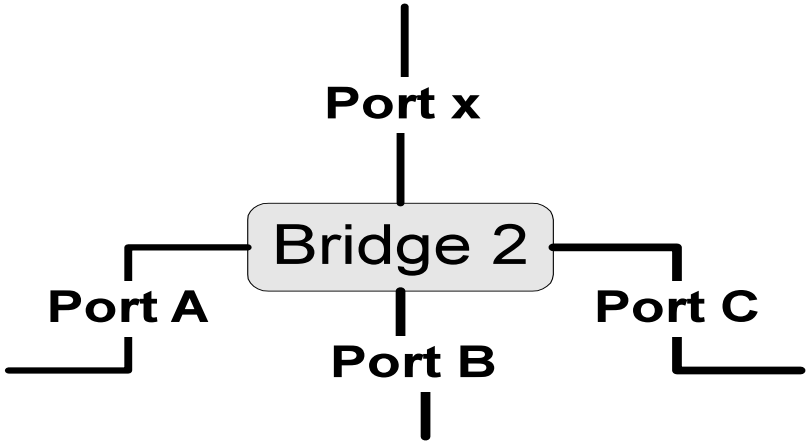
`MAC address:`	Host Ethernet interface address
`port:`	Port number of bridge
`age:`	Aging time of entry

Interpretation:

- A machine with **`MAC address`** lies in direction of the port **`port`** from the bridge. This information is **`age`** time units old.

Packet Forwarding 2

- Assume a packet arrives on port x.



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

Found?

Not found ?

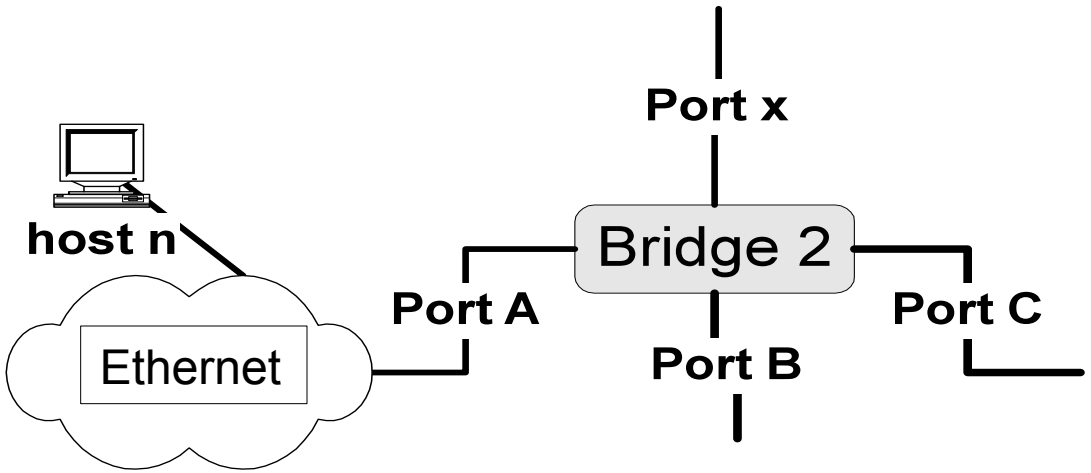
Forward the packet on the appropriate port

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

Address Learning

- The forwarding database is built automatically with a simple heuristic:

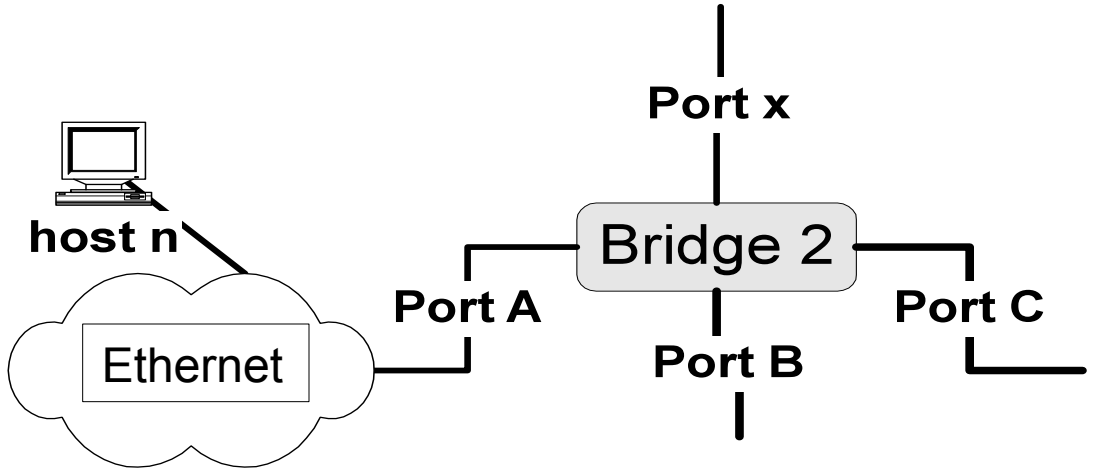
  The source field of a packet that arrives on a port tells which hosts are reachable from this port.



Address Learning 2

Algorithm:

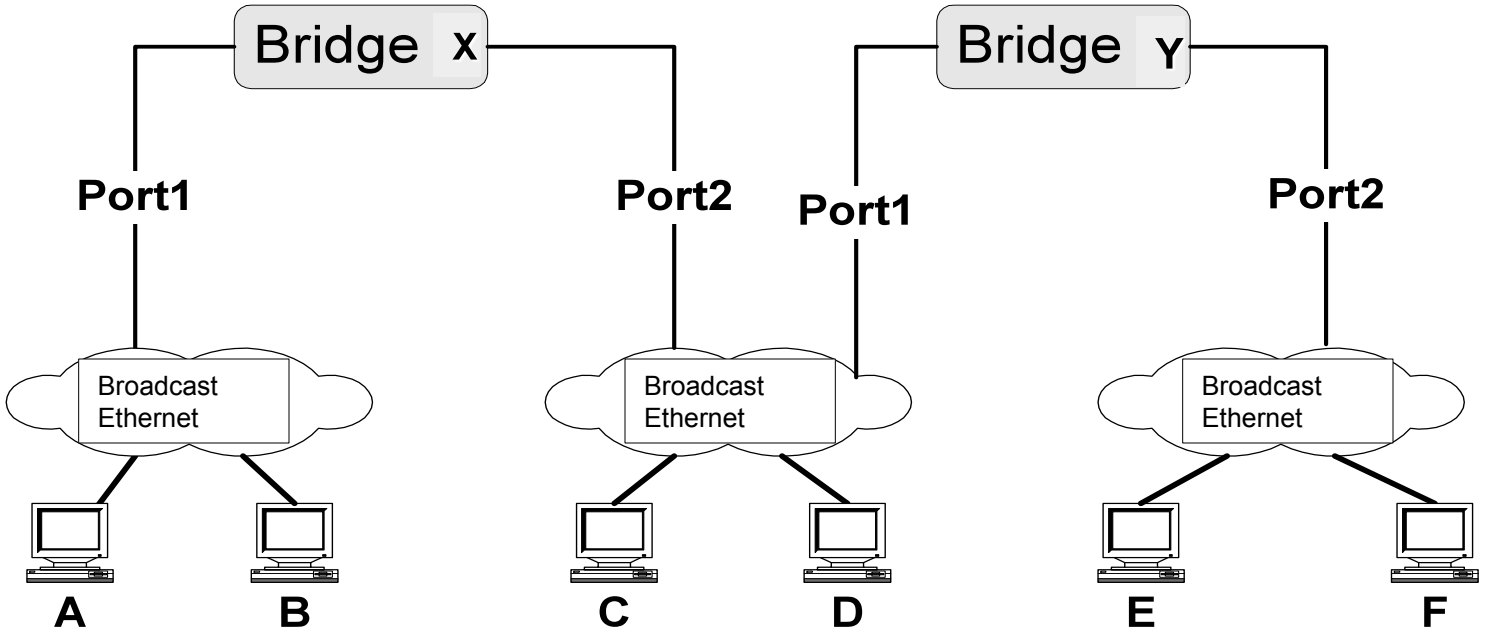
- For each packet received, stores the source address in the forwarding database together with the port where the packet was received.
- An entry is deleted after some time out (default is 15 seconds).



Example

- Consider the following packets:
<Src=A, Dest=F>, <Src=C, Dest=A>, <Src=E, Dest=C>

- What have the bridges learned?



Questions

- What if a host is disconnected from a port and reconnected to a different port in a bridged Ethernet network?
- What are the dangers of flooding packets for unknown destinations?

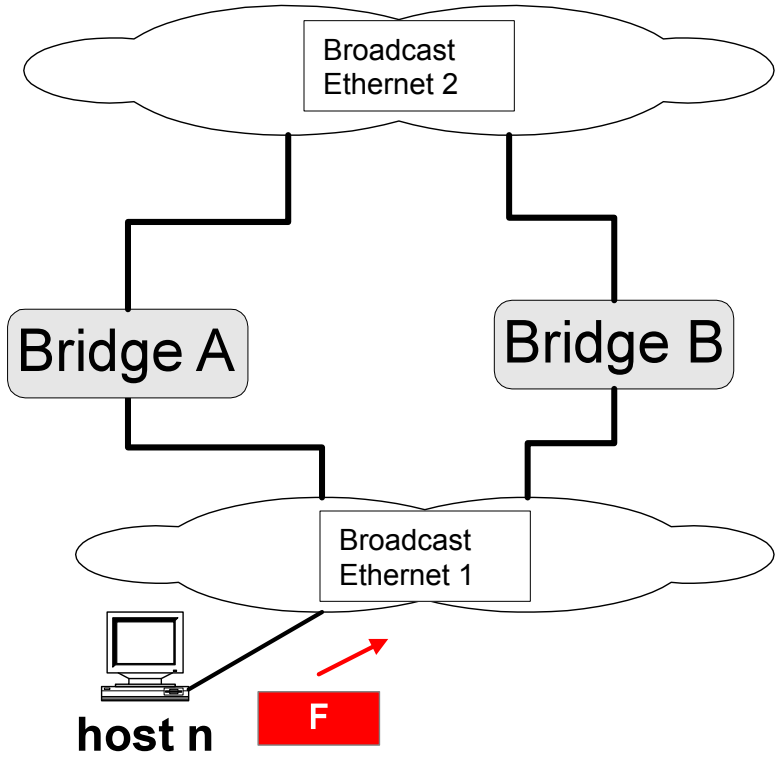
Danger

- Assume *host n* transmits a packet F with unknown destination

What happens?

- Bridges A and B flood the packet to Ethernet 2
- Bridge B sees F on Ethernet 2 (with unknown destination), and copies the packet back to Ethernet 1
- Bridge A does the same
- The copying continues

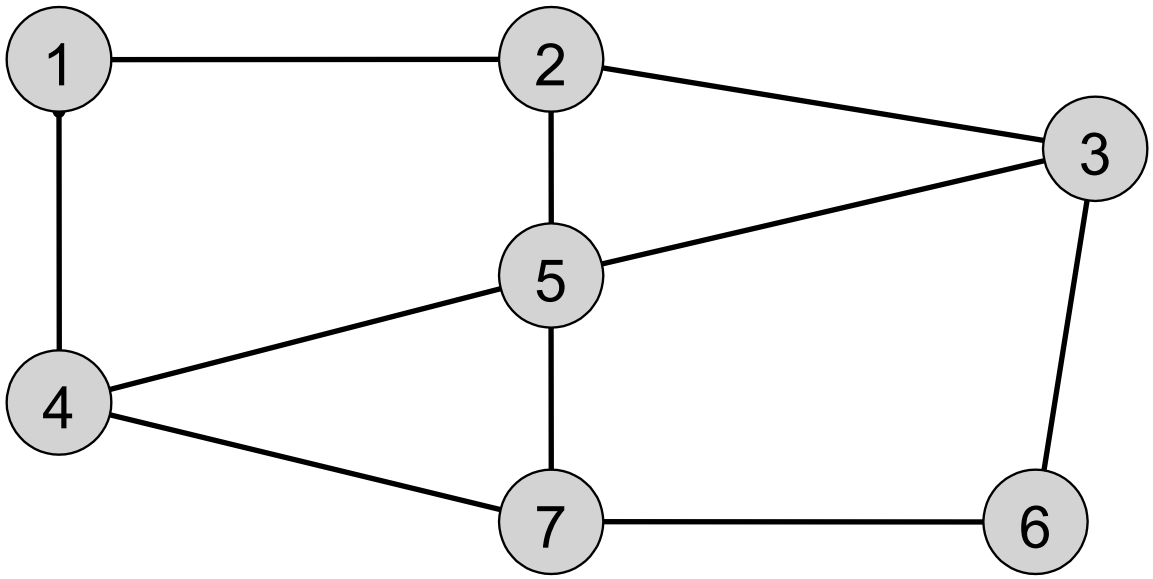
What's the problem? What's the solution?



Spanning Trees

- The solution to the loop problem is to not have loops in the topology
- IEEE 802.1 has an algorithm that builds and maintains a spanning tree in a dynamic environment.
- Bridges exchange messages (Configuration Bridge Protocol Data Unit (BPDU)) to configure the bridge to build the tree.

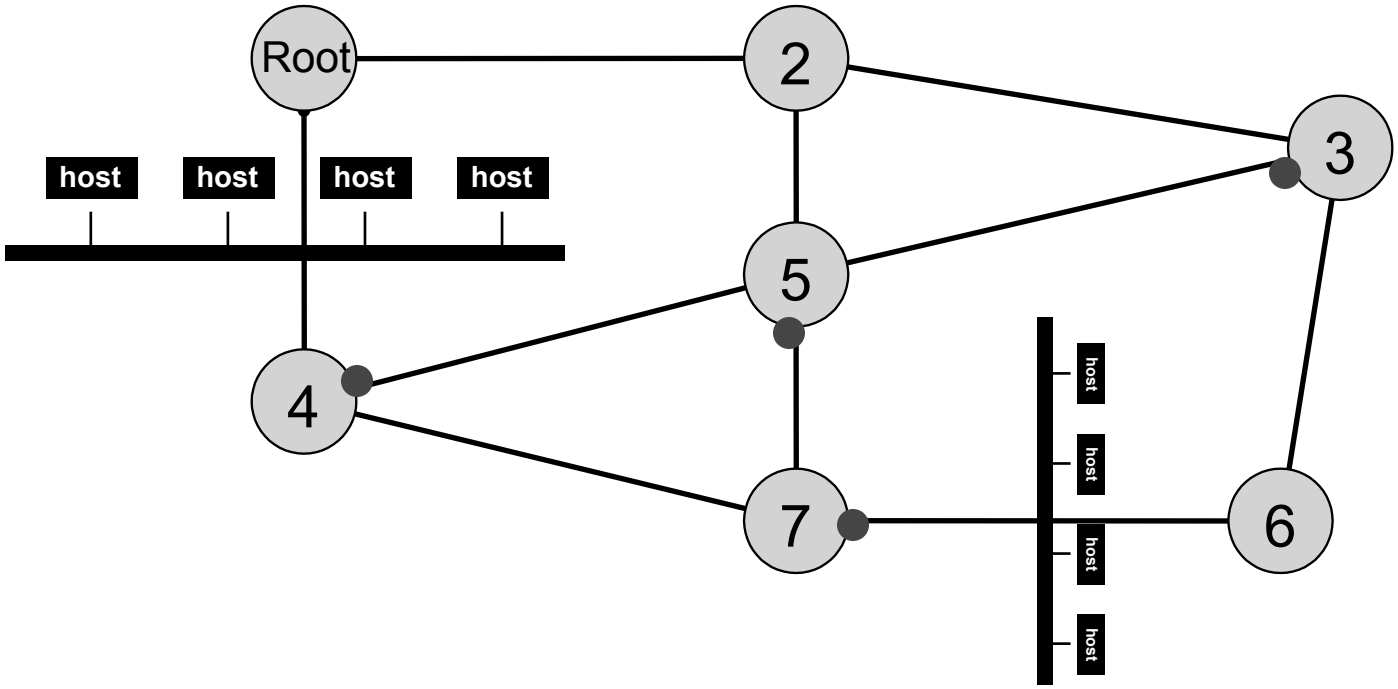
What's a Spanning Tree?



- A subset of edges of a graph forming a tree that spans all the nodes (no cycle)

802.1 Spanning Tree Approach (Sketch)

- Elect a bridge to be the root of the tree
- Every bridge finds least cost path to the root
- Union of these paths become the spanning tree



What do the BPDU messages do?

With the help of the BPDUs, bridges can:

- Elect a single bridge as the **root bridge**.
- Calculate the cost of the least cost path to the root bridge
- Each Ethernet segment can determine a **designated bridge**, which is the bridge with lowest cost 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 least cost path to the root.
- Select ports to be included in the spanning tree.

Concepts

- Each bridge as a unique identifier:

  Bridge ID = <MAC address + priority level>

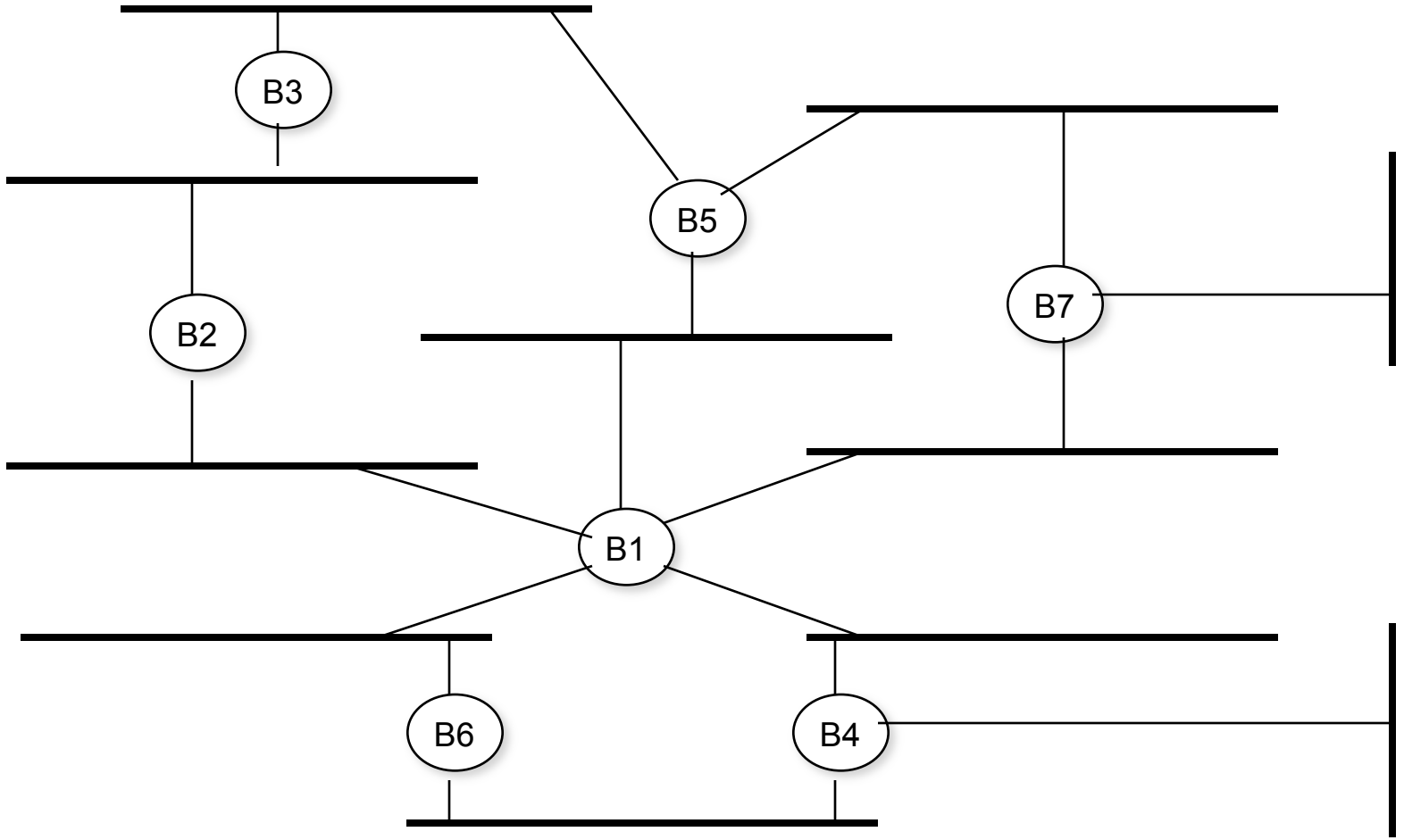
  Note that a bridge has several MAC addresses (one for each port), but only one ID

- 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.
- **Path Cost:** Cost of the least cost path to the root from the port of a transmitting bridge; Assume it is measured in # of hops to the root.
- **Root Port**: Each bridge has a root port which identifies the next hop from a bridge to the root.

Concepts

- **Root Path Cost**: For each bridge, the cost of the least cost path to the root
- **Designated Bridge, Designated Port:** Single bridge/port on a Ethernet segment that provides the least cost path to the root:
  - if two bridges have the same cost, select the one with highest priority (smallest bridge ID)
  - if the least cost bridge has two or more ports on the Ethernet segment, select the port with the lowest identifier
- **Note:** We assume that "cost" of a path is the number of "hops".

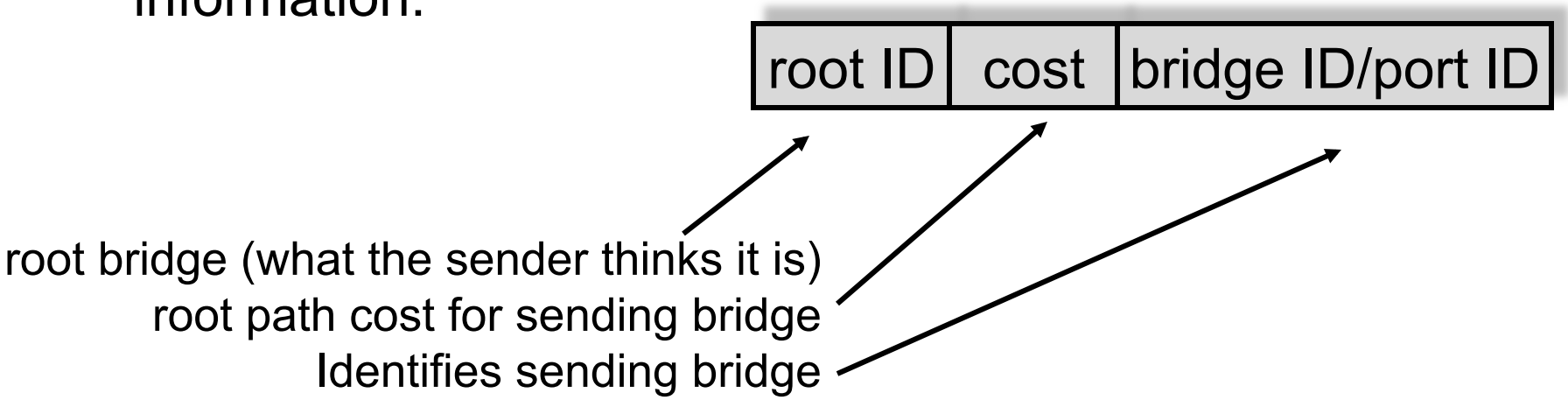
A Bridged Network



Steps of Spanning Tree Algorithm

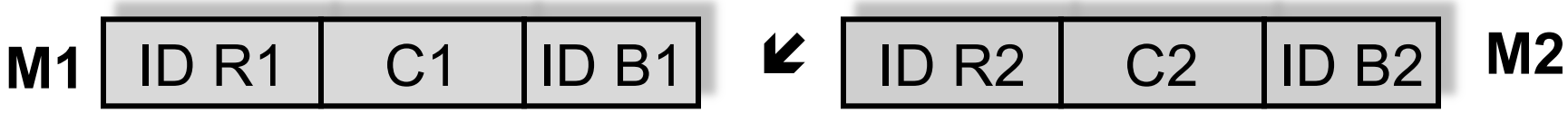
1. **Determine the root bridge**
2. **Determine the root port on all other bridges**
3. **Determine the designated bridge on each Ethernet segment**

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



Ordering of Messages

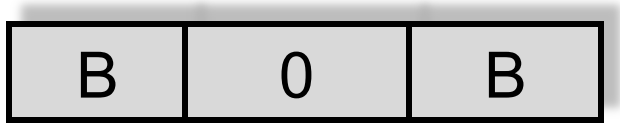
- We can order BPDU messages with the following ordering relation "↙" (let's call it "lower cost"):



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

Determine the Root Bridge

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



- Each bridge looks at the BPDUs received on all its ports and its own transmitted BPDUs.
- Root bridge is the smallest received root ID that has been received so far (Whenever a smaller ID arrives, the root is updated)

Calculate the Root Path Cost
Determine the Root Port

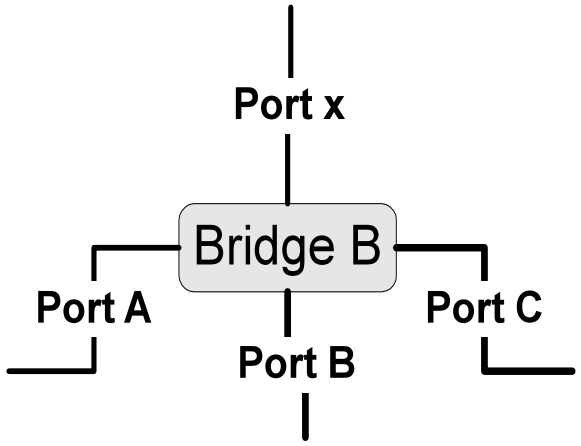
- At this time: A bridge B has a belief of who the root is, say R.
- Bridge B determines the Root Path Cost (Cost) as follows:
  - *If B = R :* Cost = 0.
  - *If B ≠ R:* Cost = {Smallest Cost in any of BPDUs that were received from R} + 1
- **B's root port** is the port from which B received the lowest cost path to R (in terms of relation "↙").
- Knowing R and Cost, B can generate its BPDU (but will not necessarily send it out):

Calculate the Root Path Cost
Determine the Root Port

- At this time: B has generated its BPDU

R	Cost	B

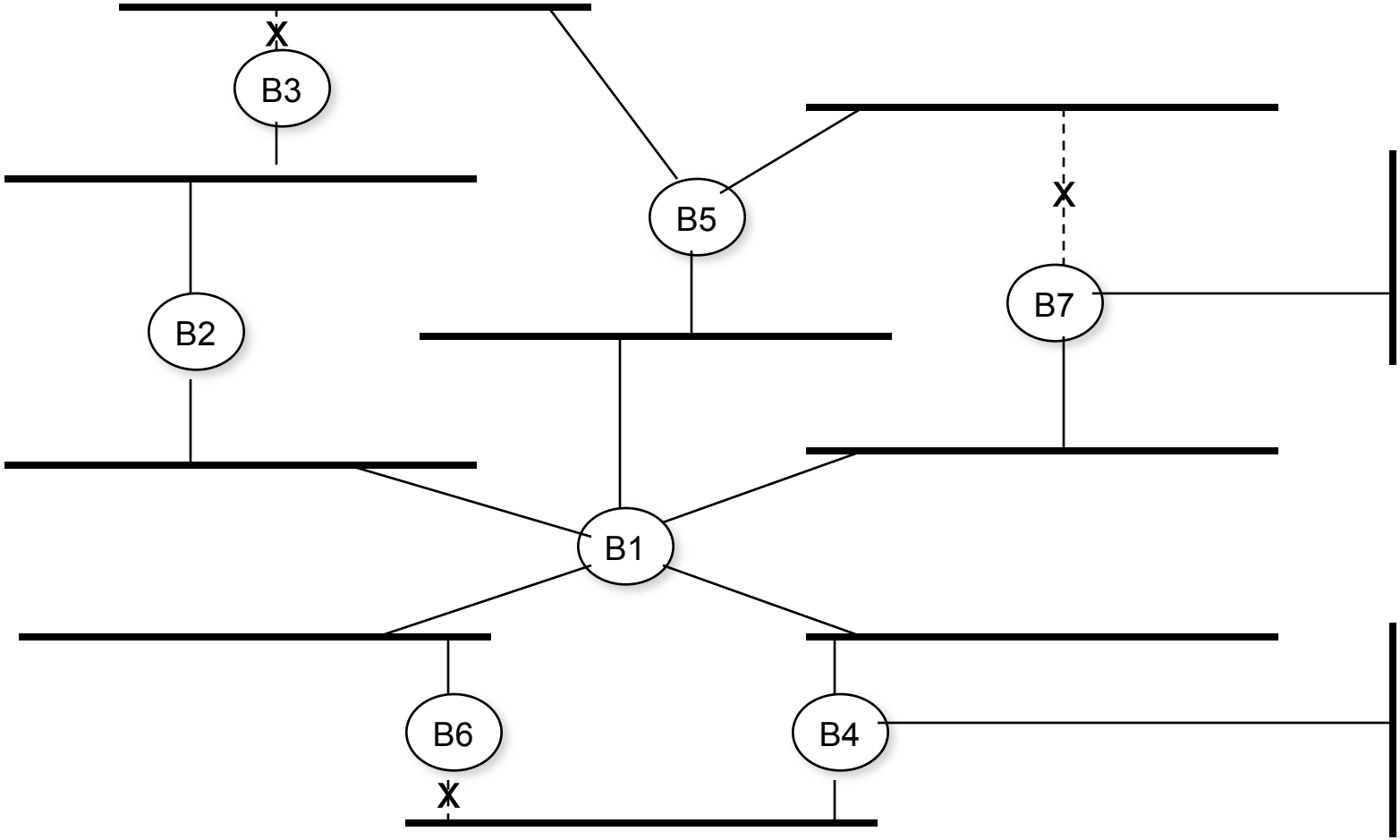
- B will send this BPDU on one of its ports, say **port x**, only if its BPDU is lower (via relation "↙") than any BPDU that B received from port x.
- In this case, B also assumes that it is the **designated bridge** for the Ethernet segment to which the port connects.



Selecting the Ports for the Spanning Tree

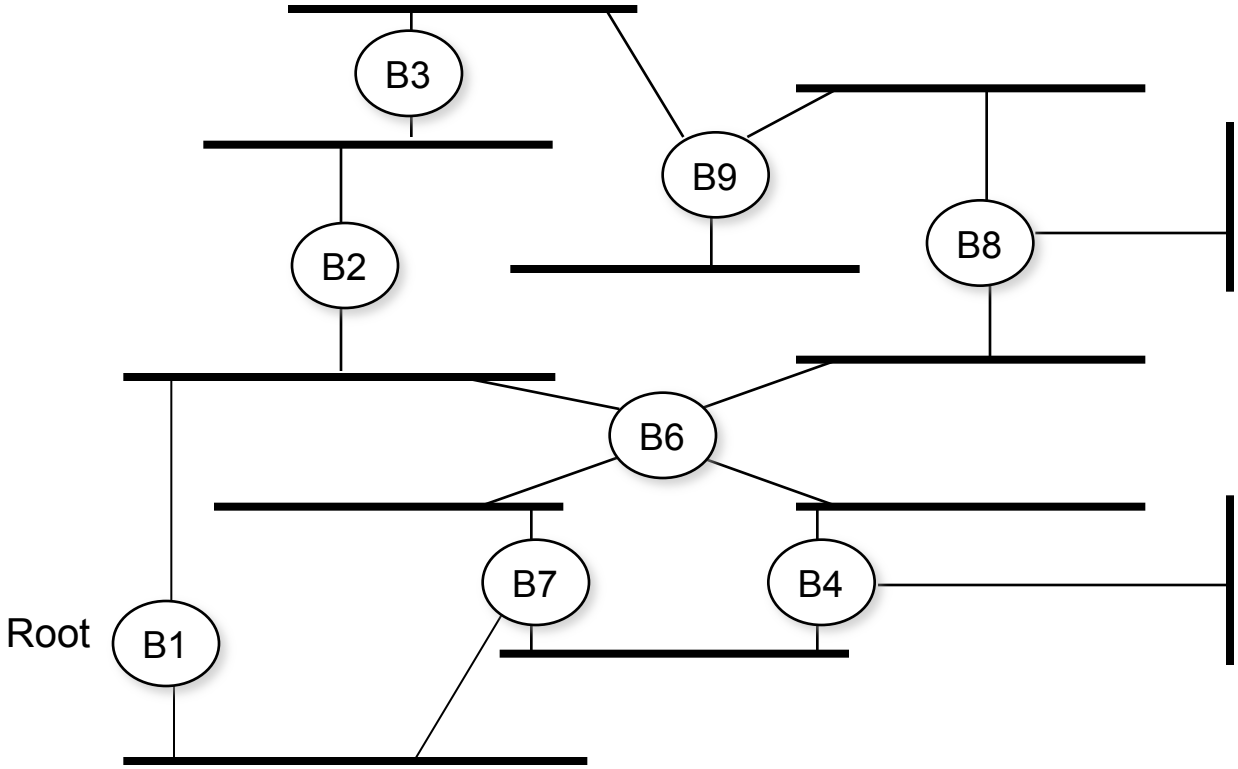
- At this time: Bridge B has calculated the root, the root path cost, and the designated bridge for each Ethernet segment.
- Now B can decide which ports are in the spanning tree:
  - B's root port is part of the spanning tree
  - All ports for which B is the designated bridge are part of the spanning 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)**

A Bridged Network (End of Spanning Tree Computation)

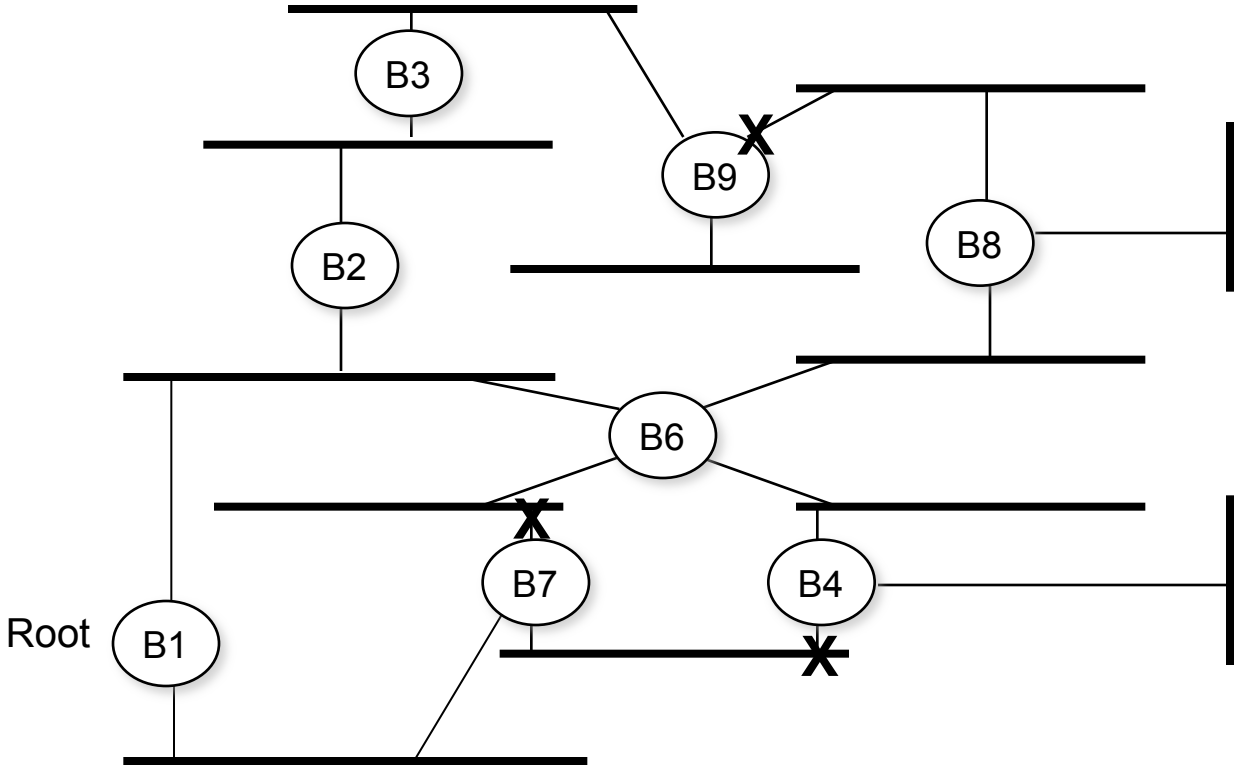


The texbook p.197 is wrong with respect to B3 and B6.

Another example



Another example



Algorhyme by Radia Pearlman

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

Can the Internet be one big bridged Ethernet?

- Inefficient
  - Too much flooding
- Explosion of forwarding table
  - Need to have one entry for every Ethernet address in the world!
- Poor performance
  - Tree topology does not have good load balancing properties
  - Hot spots
- Etc…