

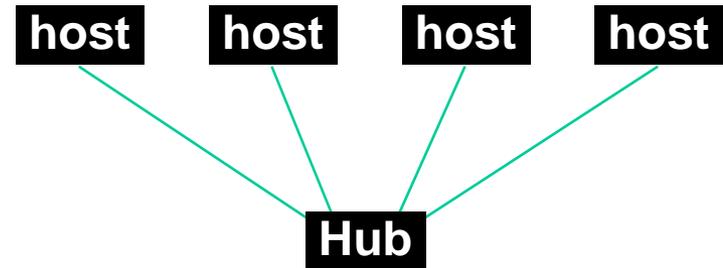
Data Networks

UdS and IMPRS-CS

Lecture 7: Bridging

Recap

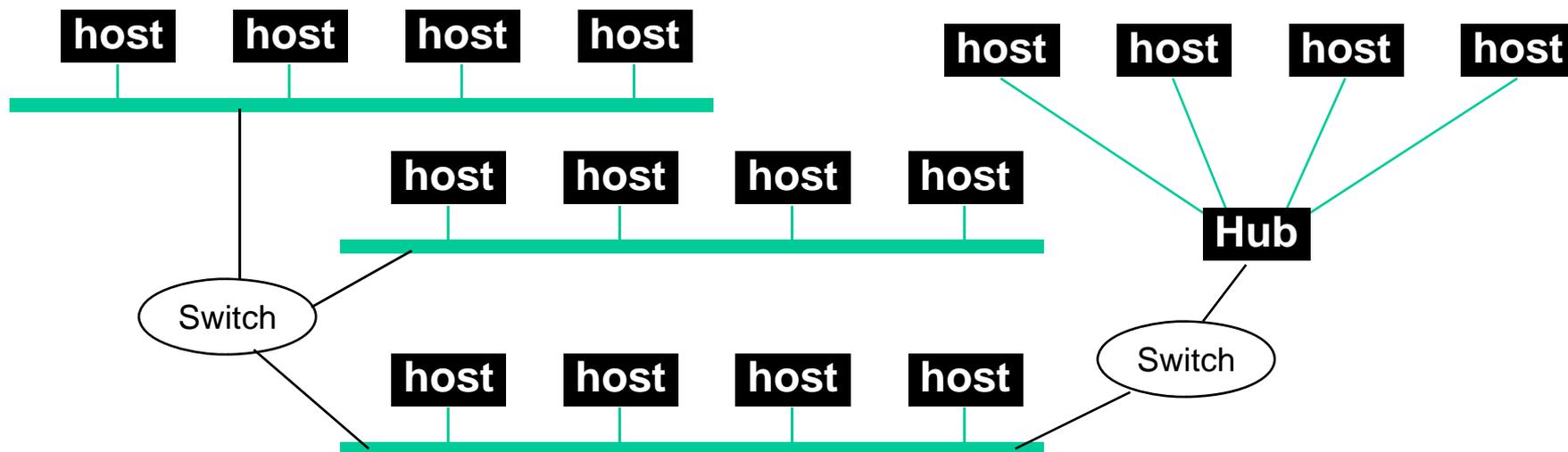
Multiple access technology



Hub emulates a broadcast channel
Easy to add a new host

- Multiple access 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 numbers of nodes
 - Too many nodes share bandwidth of channel, too many collisions, goodput (throughput of useful data) goes to zero

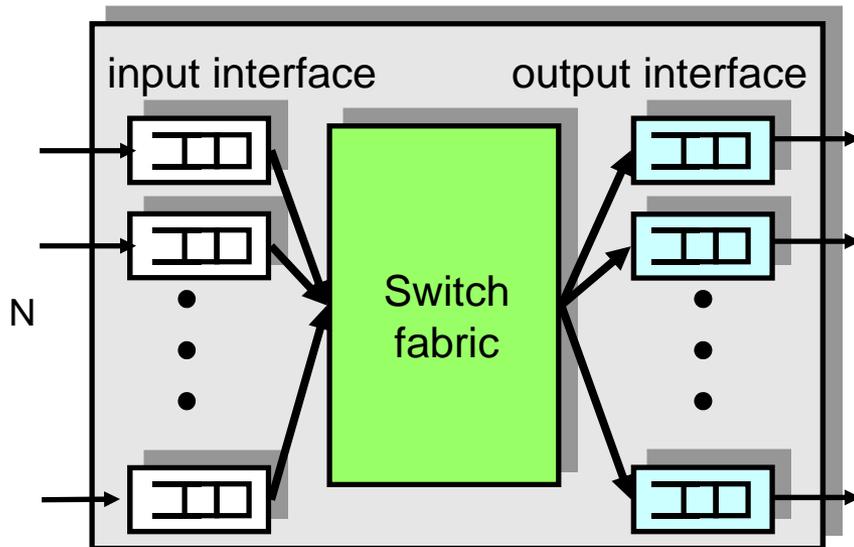
Need Switching Techniques



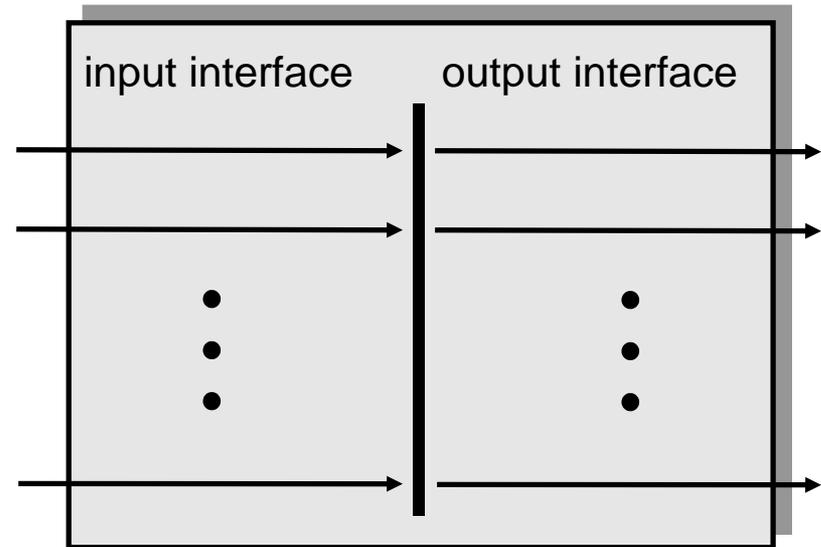
- Switching limits size of collision domains, allows network size to scale up
 - To how big? Can Internet be one big switched Ethernet?
 - Will return to this question
- Switches are more complex than hubs
 - Intelligence, memory buffers, high performance

Switch

Switch

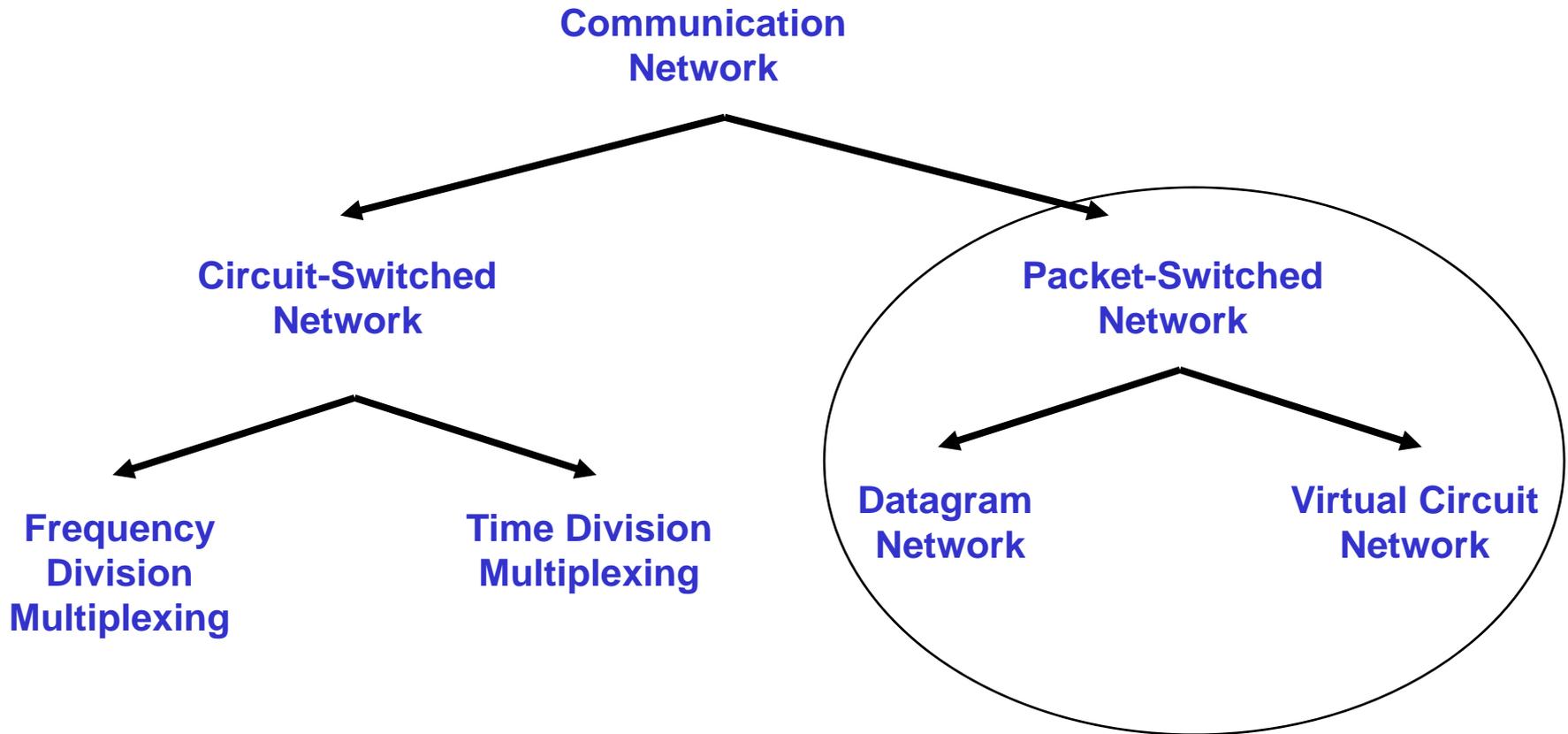


Hub



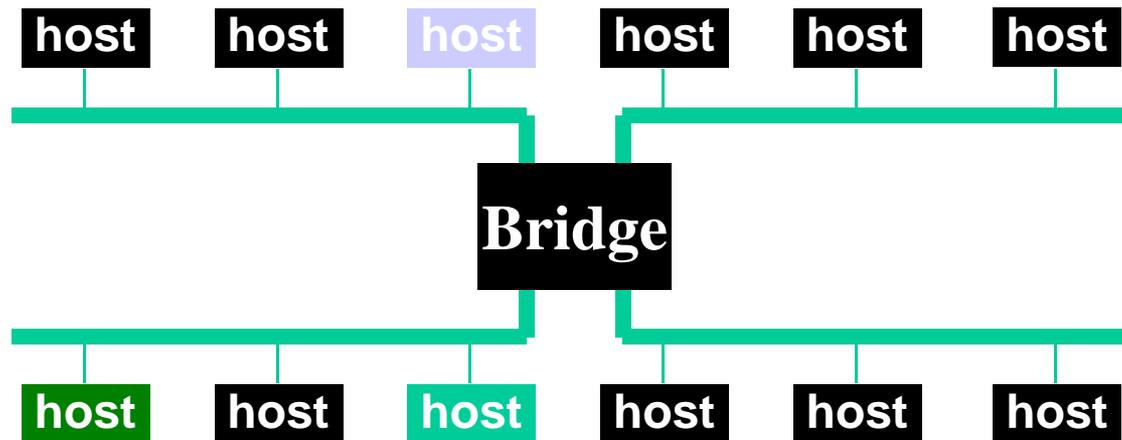
- Switch has memory buffers to queue packets, reduce loss
- Switch is intelligent: Forward an incoming packet to the correct output interface only
- High performance: Full $N \times$ line rate possible

Taxonomy of Networks



Building Large LANs Using Bridges

- Bridges connect multiple IEEE 802 LANs at layer 2
 - Datagram packet switching
 - Only forward packets to the right port
 - Reduce collision domain
 - Increase available bandwidth
- In contrast, hubs rebroadcast packets.



Transparent Bridges

- Overall design goal: **Complete transparency**
 - “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 table** with entries
< **MAC address**, **port**, **age**>

MAC address:	host address
port:	port number of bridge
age:	age of entry

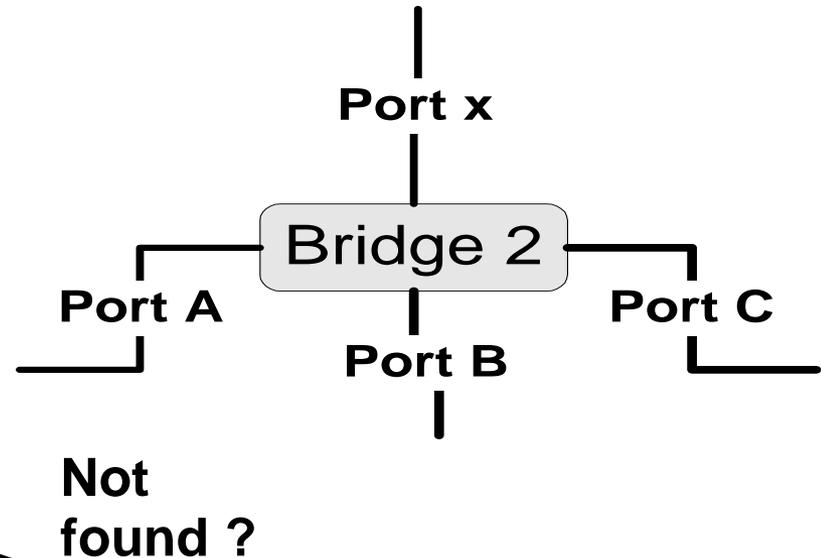
interpretation:

- 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 frame arrives on port x.

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



Found?

Not found ?

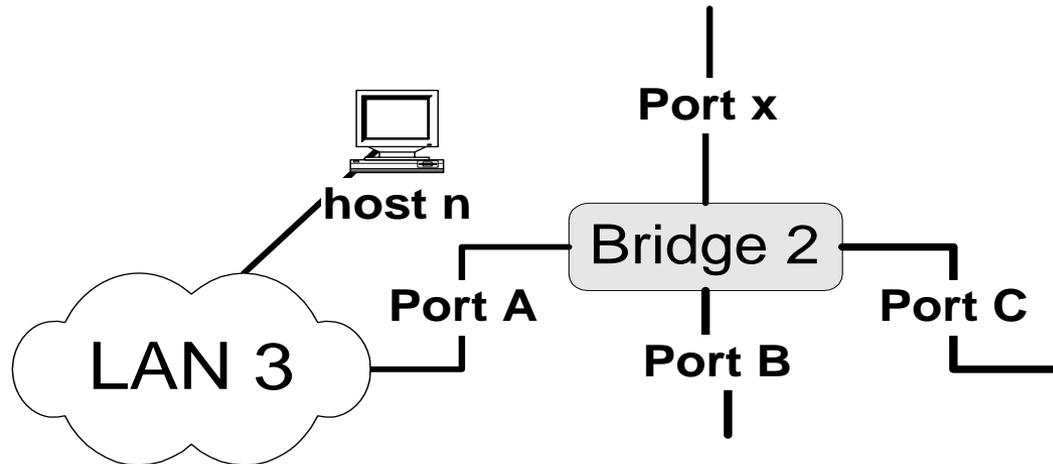
Forward the frame on the appropriate port

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

Address Learning

- In principle, the forwarding table could be initialized statically (static routing)
- In the 802.1 bridge, the process is automated using a simple heuristic:

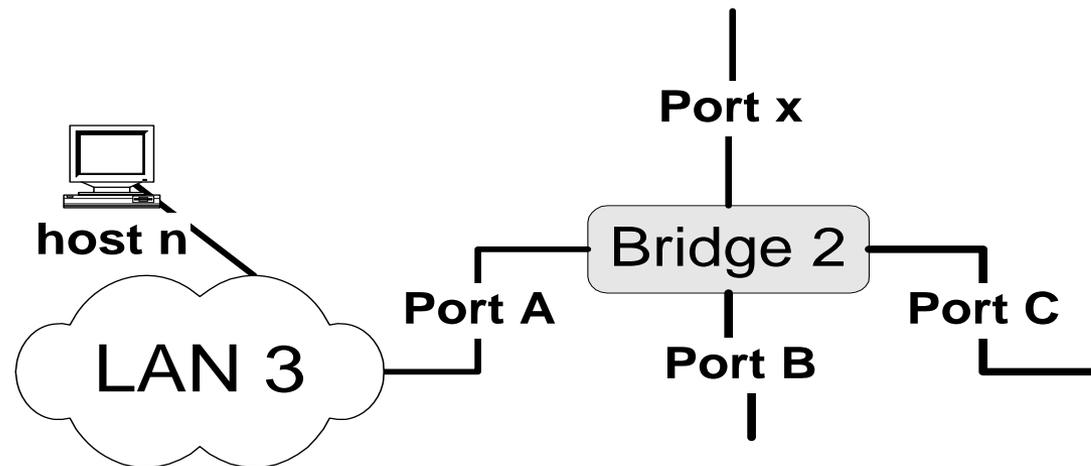
The source field of a frame that arrives on a port tells which hosts are reachable from that port.



Address Learning 2

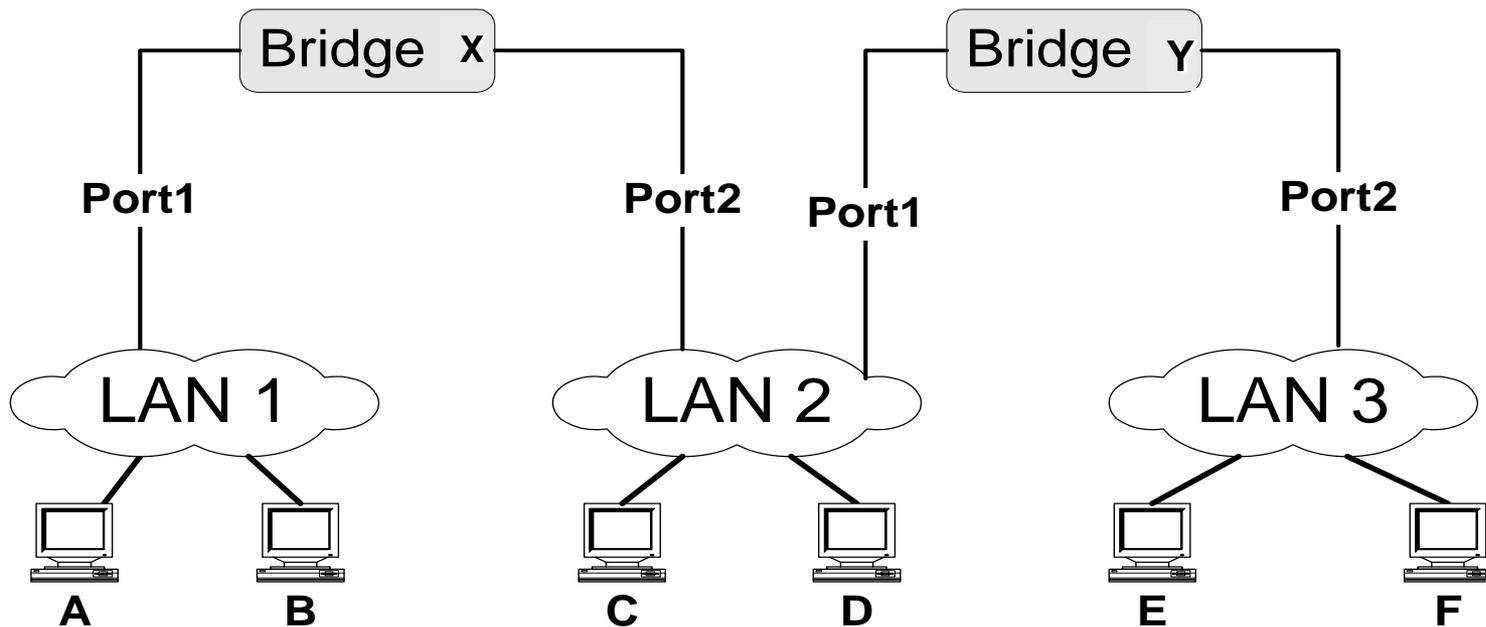
Algorithm:

- For each frame received, switch stores the source address in the forwarding table, together with the port on which the frame was received.
- An entry is deleted after some time when not refreshed (default is 15 seconds).



Example

- Consider the following packets: (Src=source, Dst=destination)
<Src=A, Dst=F>, <Src=C, Dst=A>, <Src=E, Dst=C>
- What happens?



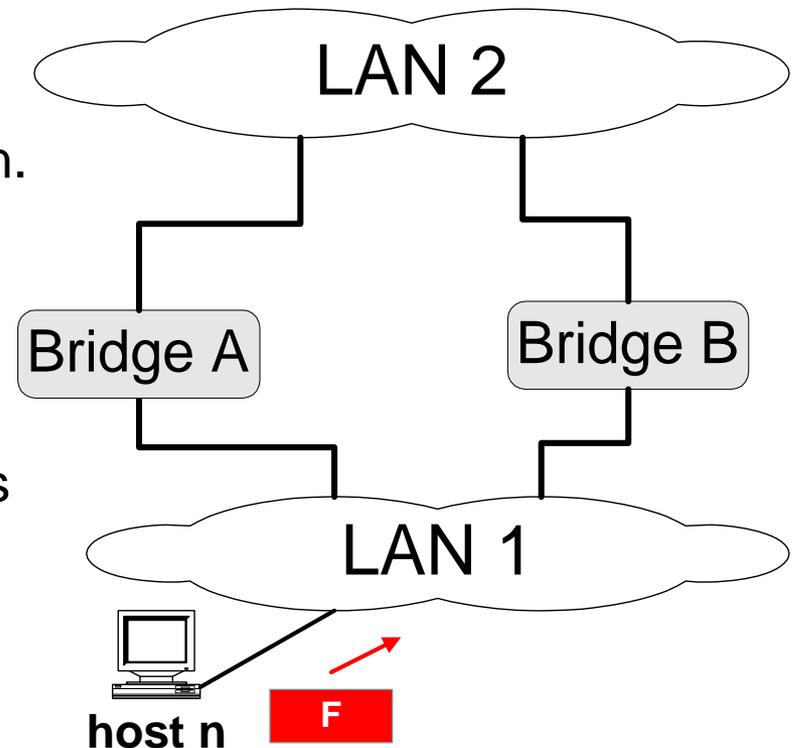
Loops

- Consider two LANs connected by two bridges.
- Assume *host n* is transmitting a frame *F* with unknown destination.

What happens?

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

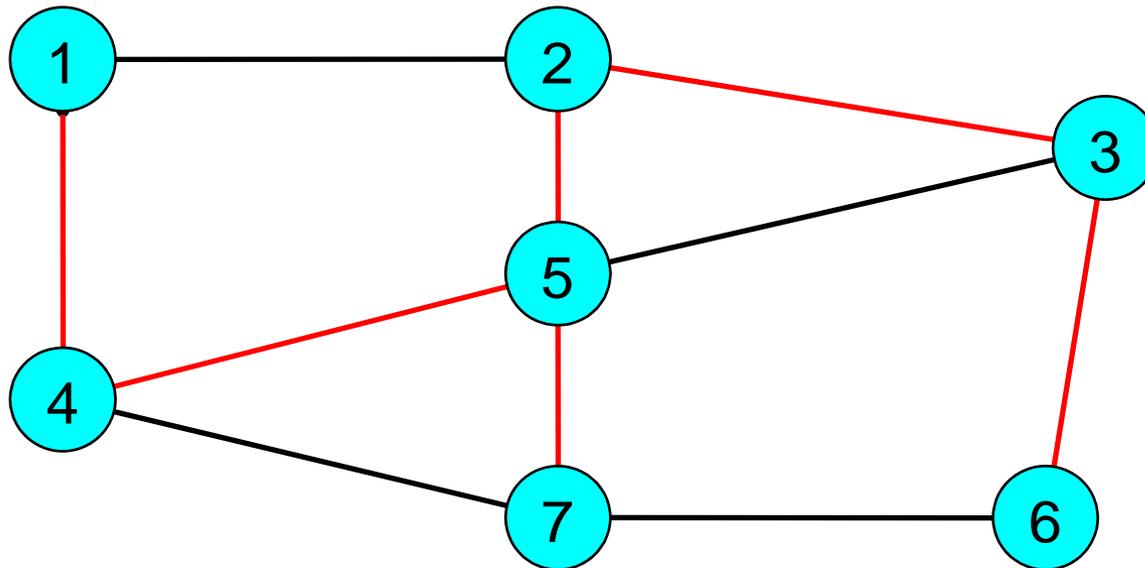
What's the problem?



Spanning Trees

- The solution to the loop problem is to not allow loops in the forwarding 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 maintain the tree.

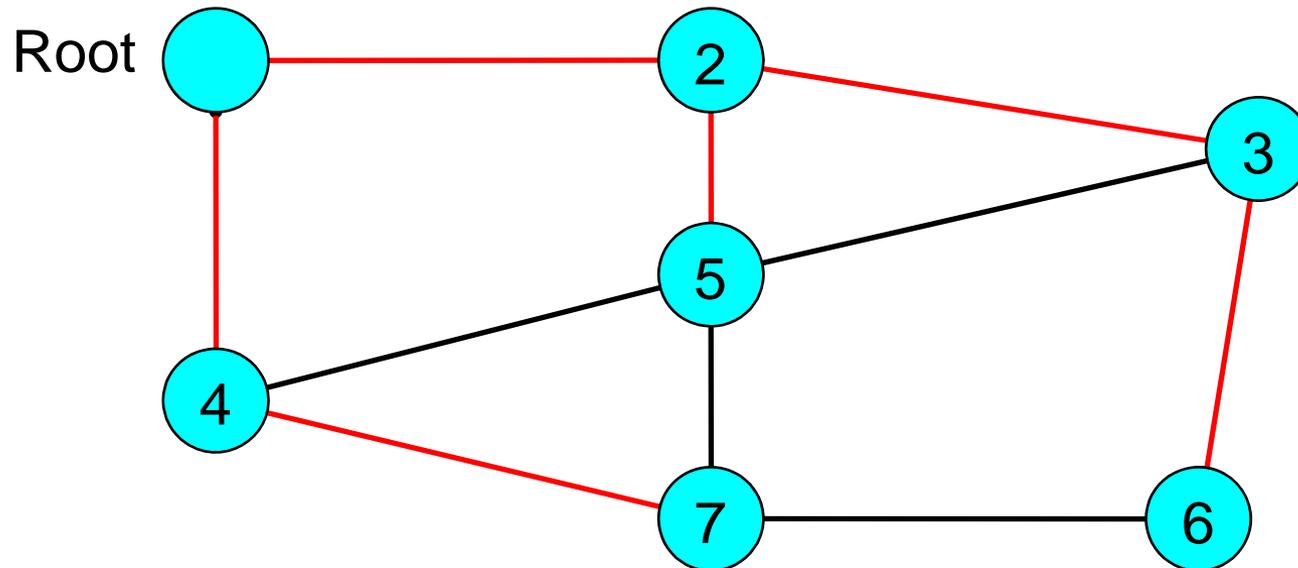
What's a Spanning Tree?



- A subset of edges of a graph that spans all the nodes without creating any cycle (i.e., a tree)

802.1 Spanning Tree Approach (Sketch)

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



What do the BPDUs messages do?

With the help of the BPDUs, bridges can:

- 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.

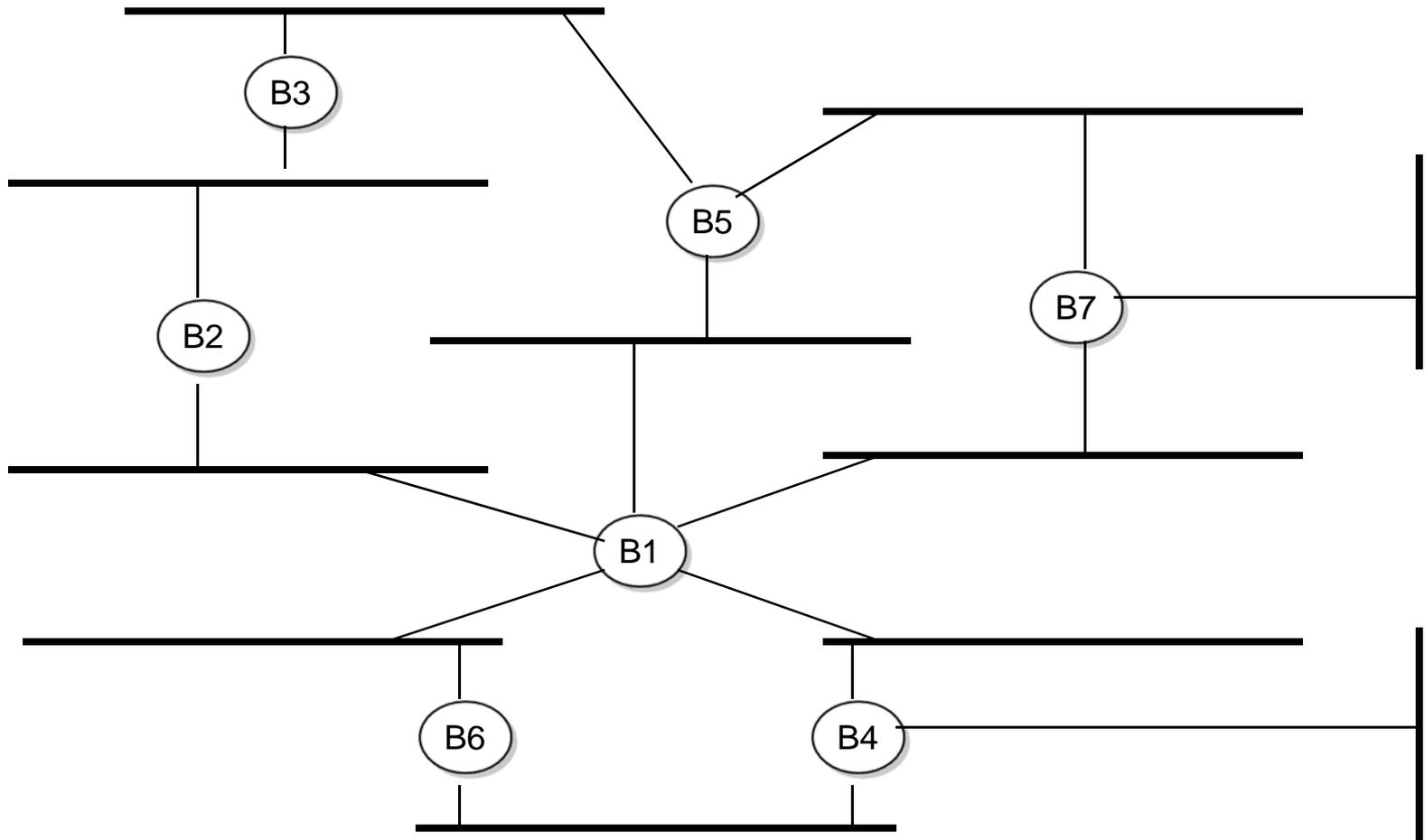
Concepts

- Each bridge as 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.
- **Path Cost:** Cost of the least cost path to the root from the port of a transmitting bridge; Assume it is measured in # of forwarding 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 min-cost path to the root
- **Designated Bridge, Designated Port:** Single bridge on a LAN that provides the minimal cost path to the root for this LAN:
 - if two bridges have the same cost, select the one with highest priority (smallest bridge ID)
 - if the min-cost bridge has two or more ports on the LAN, 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 LAN**

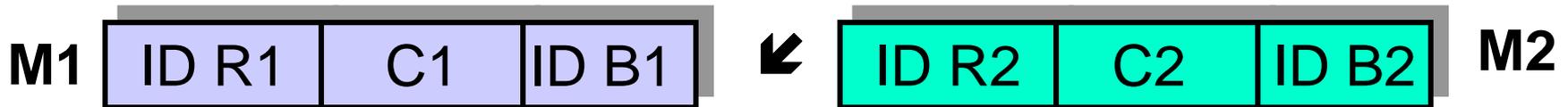
- Each bridge is sending out BPDUs that contain the following information:



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

Ordering of Messages

- We can order BPDUs with the following ordering relation “ \blacktriangleleft ” (let’s call it “lower cost”):



If $(R1 < R2)$

M1 \blacktriangleleft M2

elseif $((R1 == R2) \text{ and } (C1 < C2))$

M1 \blacktriangleleft M2

elseif $((R1 == R2) \text{ and } (C1 == C2) \text{ and } (B1 < B2))$

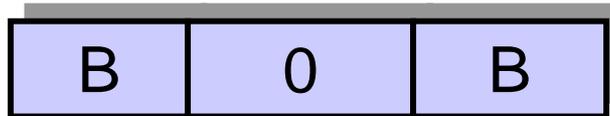
M1 \blacktriangleleft M2

else

M2 \blacktriangleleft M1

Determine the Root Bridge

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



- BPDUs are flooded throughout the network.
- 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 $B \neq R$* : Cost = {Least Cost of any BPDU received from R} + 1
- **B's root port** is the port from which B received the lowest cost path BPDU from R (in terms of relation "↙").
- Knowing R and Cost, B can generate its BPDU (but will not necessarily send it out):

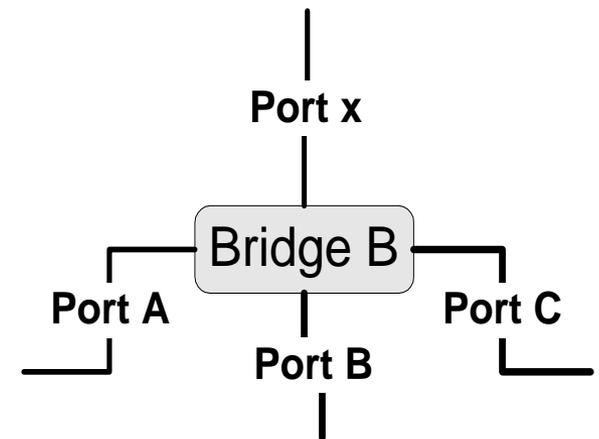


Calculate the Root Path Cost Determine the Root Port

- At this time: B has generated its BPDU



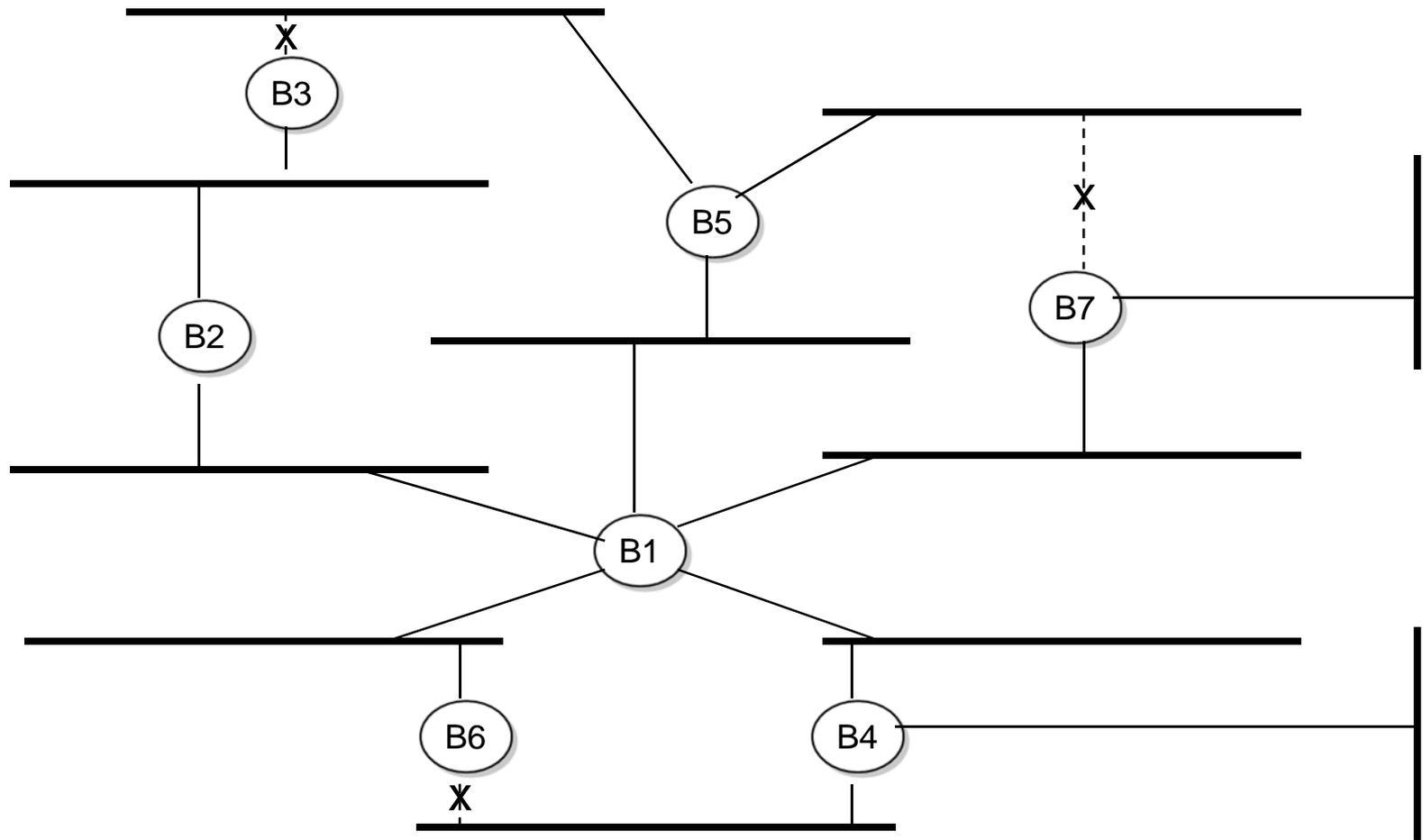
- B will send this BPDU on one of its ports, say **port x**, only if its BPDU is lower (via relation “ \blacktriangleleft ”) than any BPDU that B received from port x.
- In this case, B also assumes that it is the **designated bridge** for the LAN to which the port connects.
- Received BPDUs eventually time out



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)**

A Bridged Network (End of Spanning Tree Computation)



Ethernet Switches

- Bridges make it possible to increase LAN capacity.
 - Packets are no longer broadcasted - they are only forwarded on selected links
 - Adds a switching flavor to the broadcast LAN
- Ethernet switch is a special case of a bridge: each bridge port is connected to a single host.
 - Can make the link full duplex (really simple protocol!)
 - Simplifies the protocol and hardware used (only two stations on the link) – no longer full CSMA/CD
 - Can have different port speeds on the same switch
 - Unlike in a hub, packets can be stored

Can the Internet be One Big Switched 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...

