CS 181AG
Lecture 3

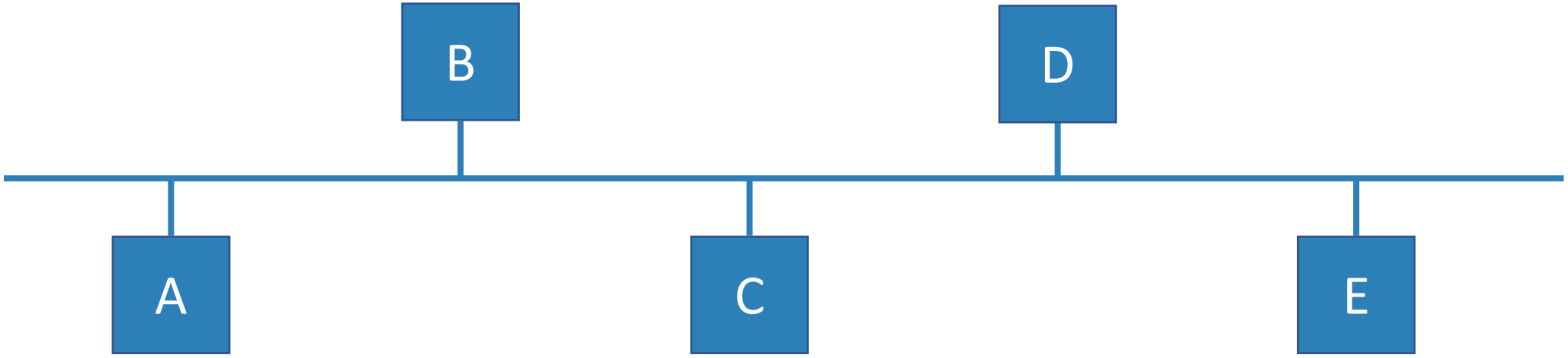
LAN Extensions

Arthi Padmanabhan

Sep 7, 2022

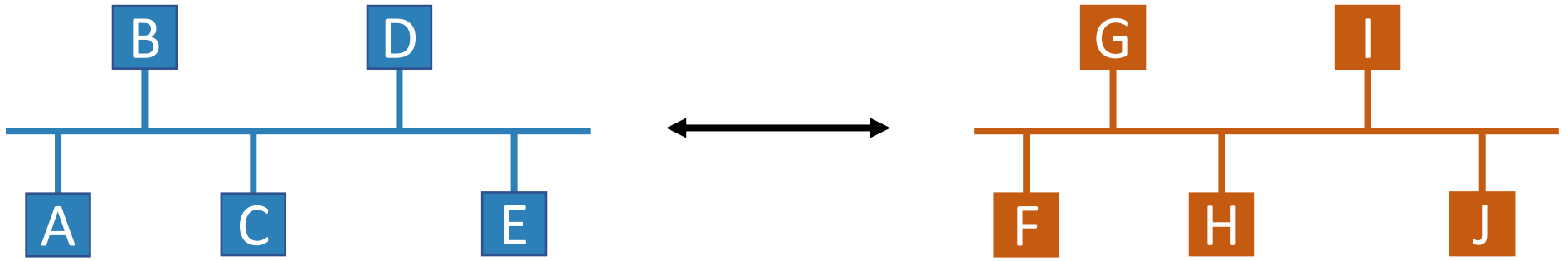
Last Time

- How do multiple nodes share a single wire?



Today

- Last time: How do multiple nodes share a single wire?
- This time: moving beyond one wire
  - How do we extend LANs to support more nodes?



Limitations of One Wire

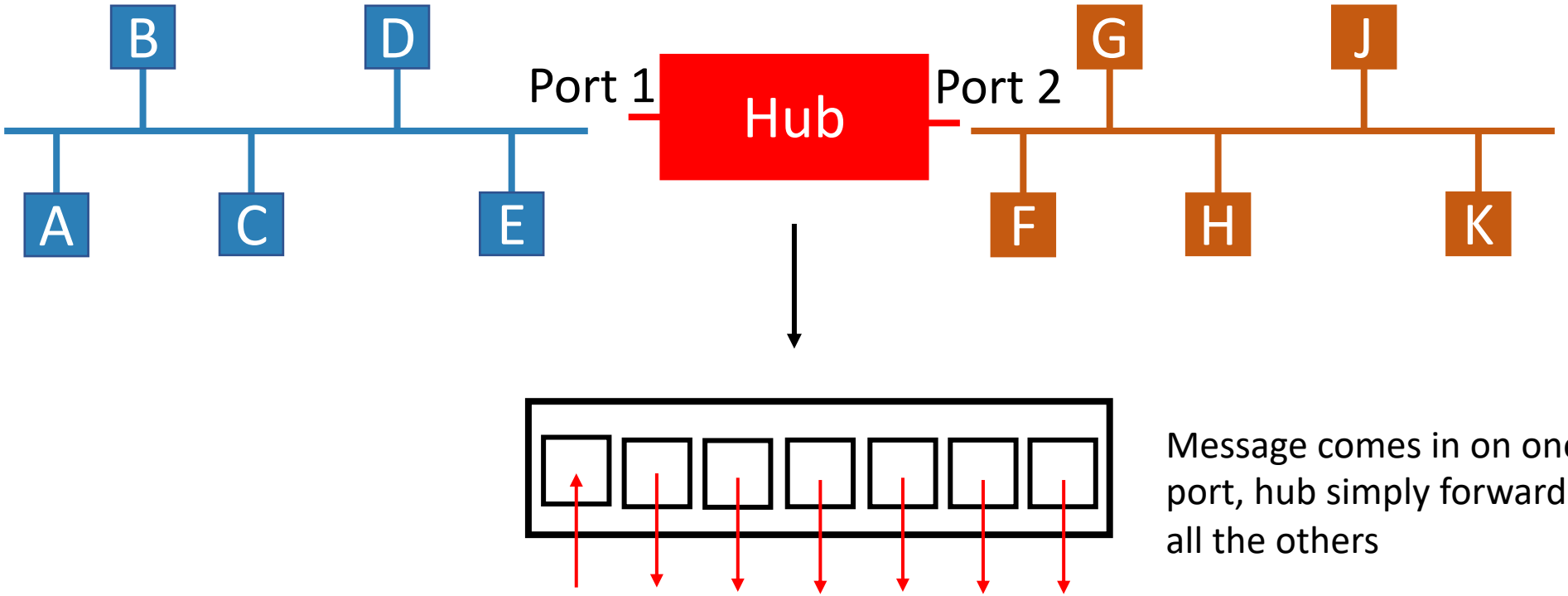
- Signal can travel limited distance
- More hosts = more collisions

Simplest Solution

- Repeater: Physical layer device that repeats/amplifies the signal to the other wire
- Hub: repeater that can connect more than 2 wires

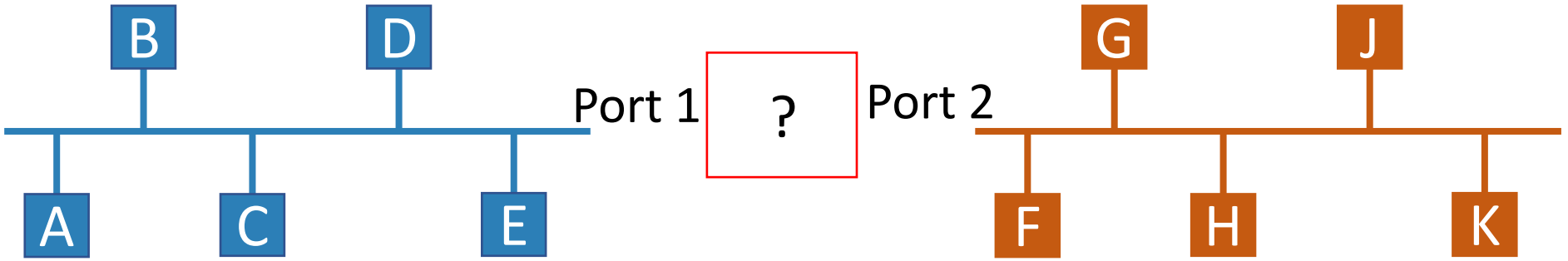
Simplest Solution: Repeaters/Hubs

- Repeater/Hub: Physical layer device that repeats/amplifies the signal to the other wire(s)



Can we make it smarter?

- Store and forward capability!



Bridge

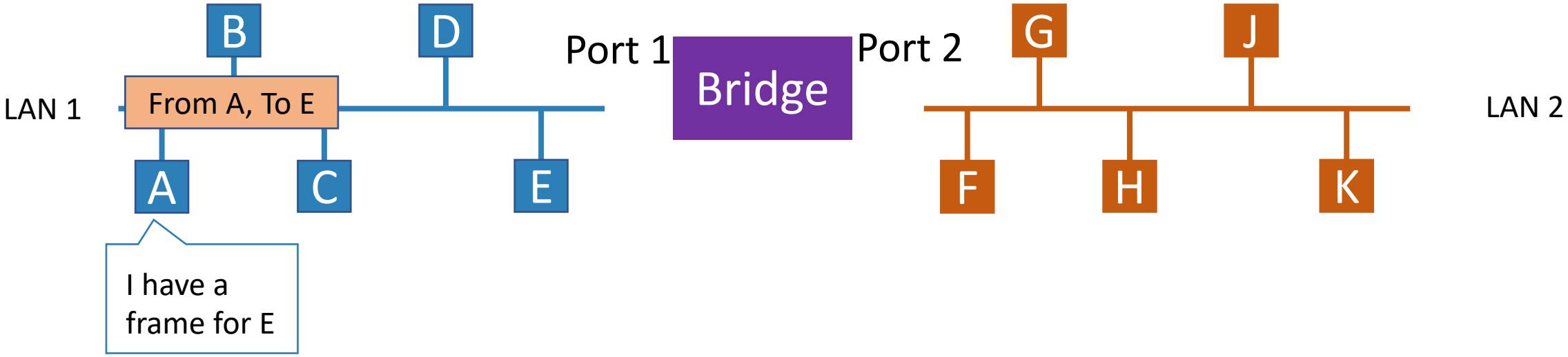
- Store and forward device that buffers entire packet before sending
- Uses CSMA/CD (acts like a host) on each LAN

Bridges

- Store and forward device that buffers entire packet before sending
- Uses CSMA/CD (acts like a host) on each LAN
- Key value add: selective forwarding
  - If A sends to J, flood the message to LAN 2
  - If A sends message to B, message does not need to be flooded to LAN 2
  - How does bridge know where each destination resides?

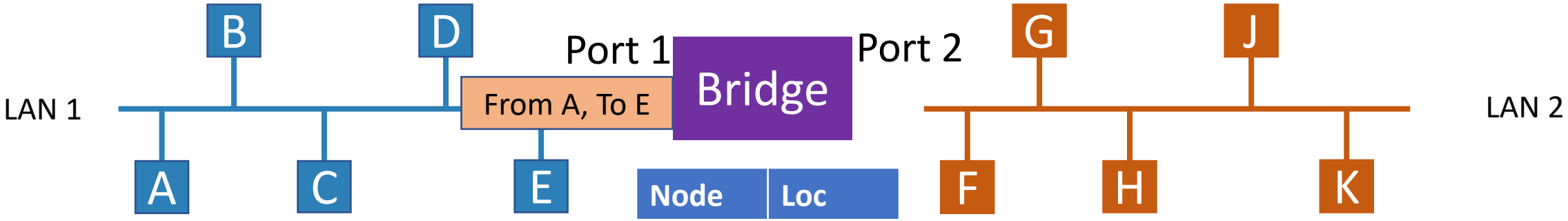
Learning Bridges

- Configuring locations of each address by hand is slow and inflexible
- Instead, learn through the source address where each address resides



Learning Bridges

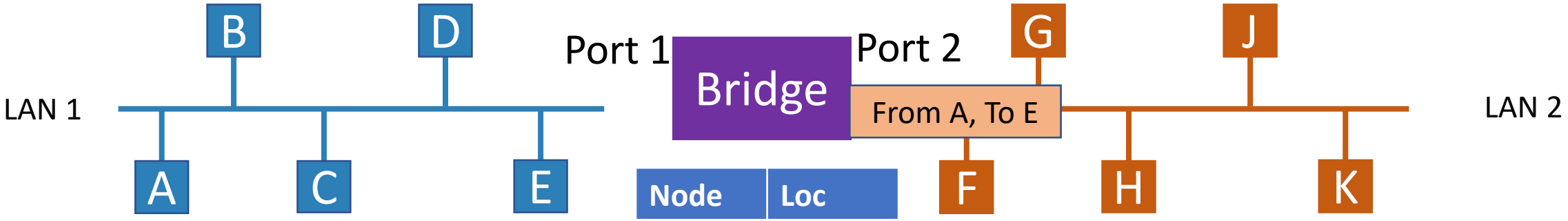
- Configuring locations of each address by hand is slow and inflexible
- Instead, learn through the source address where each address resides



Node	Loc
A	Port 1

Learning Bridges

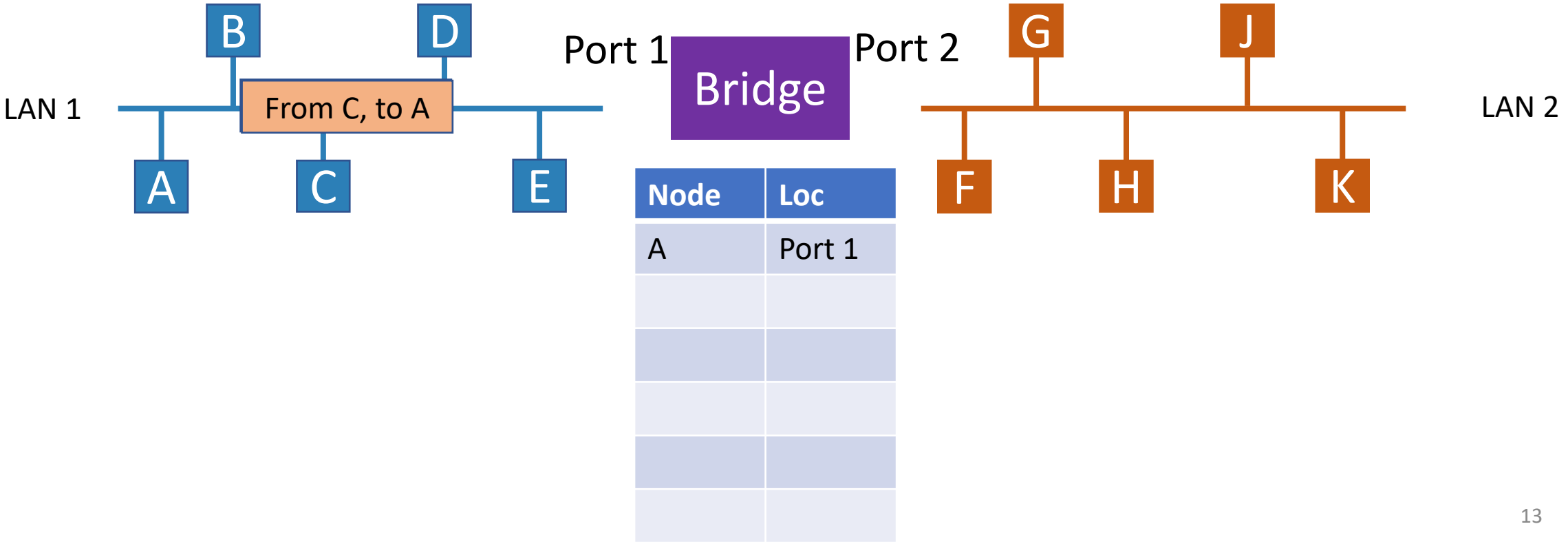
- Configuring locations of each address by hand is slow and inflexible
- Instead, learn through the source address where each address resides



Node	Loc
A	Port 1

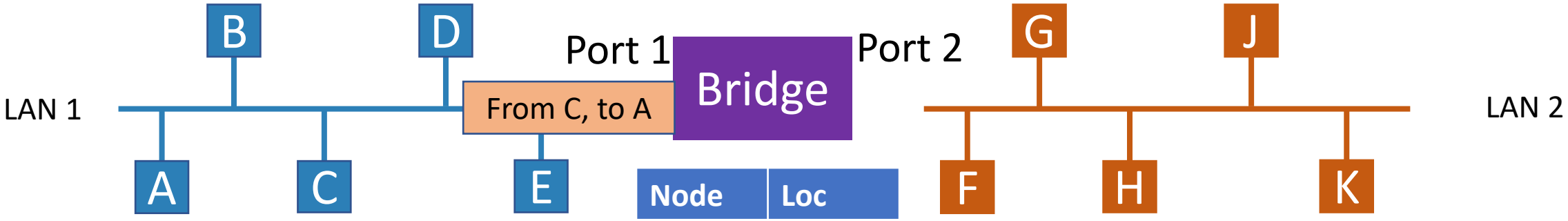
Learning Bridges

- Configuring locations of each address by hand is slow and inflexible
- Instead, learn through the source address where each address resides



Learning Bridges

- Configuring locations of each address by hand is slow and inflexible
- Instead, learn through the source address where each address resides



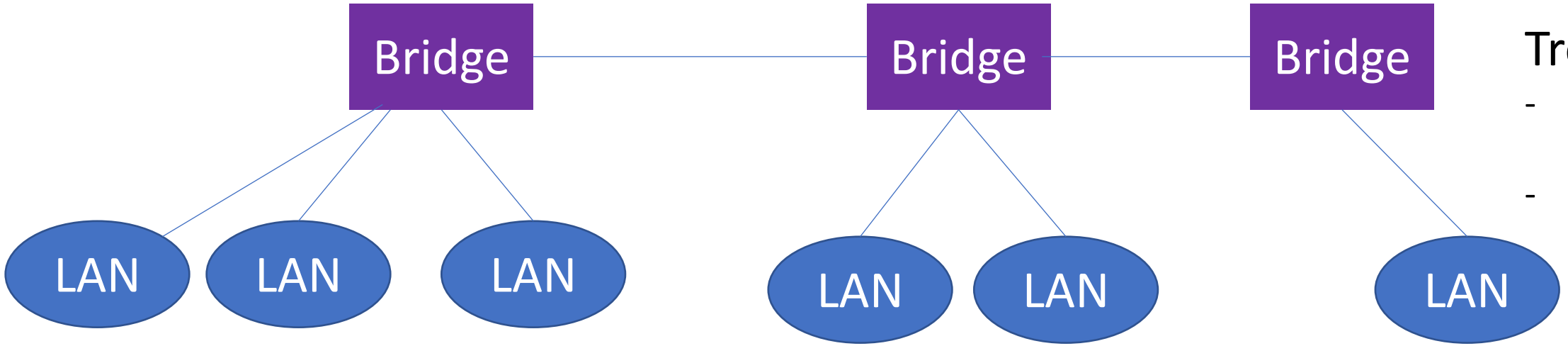
Node	Loc
A	Port 1
C	Port 1

Bridge: I know that A is on Port 1. I don't need to send this to Port 2

Network Topology

Linear

- Each bridge and LAN is single point of failure

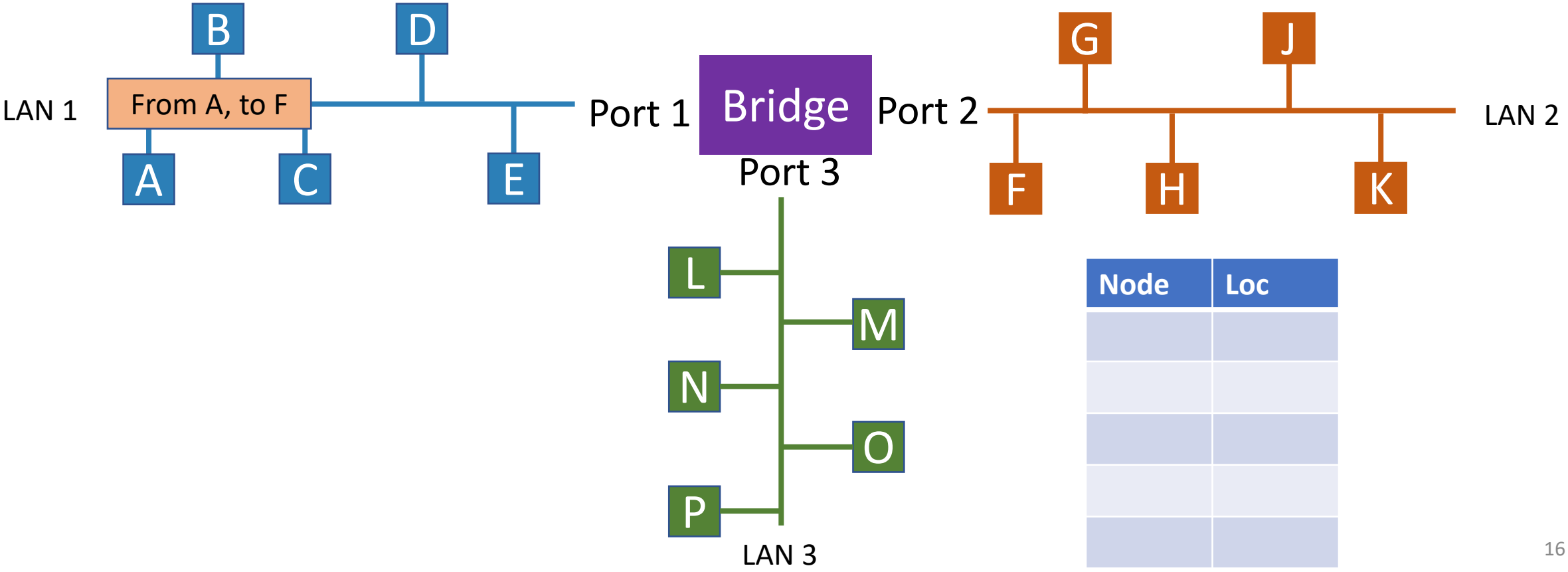


Tree

- Can survive LAN failure
- Requires more ports

Multiport Bridge

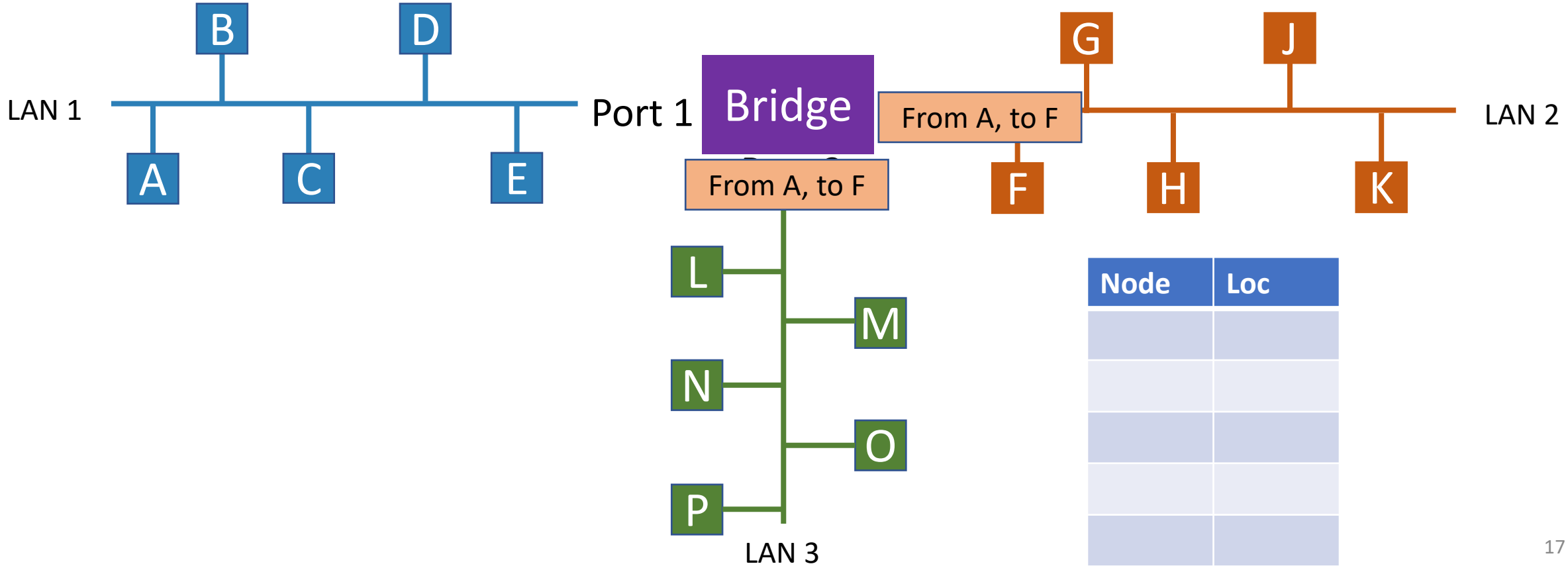
- Learning works the same way over multiple ports



Node	Loc

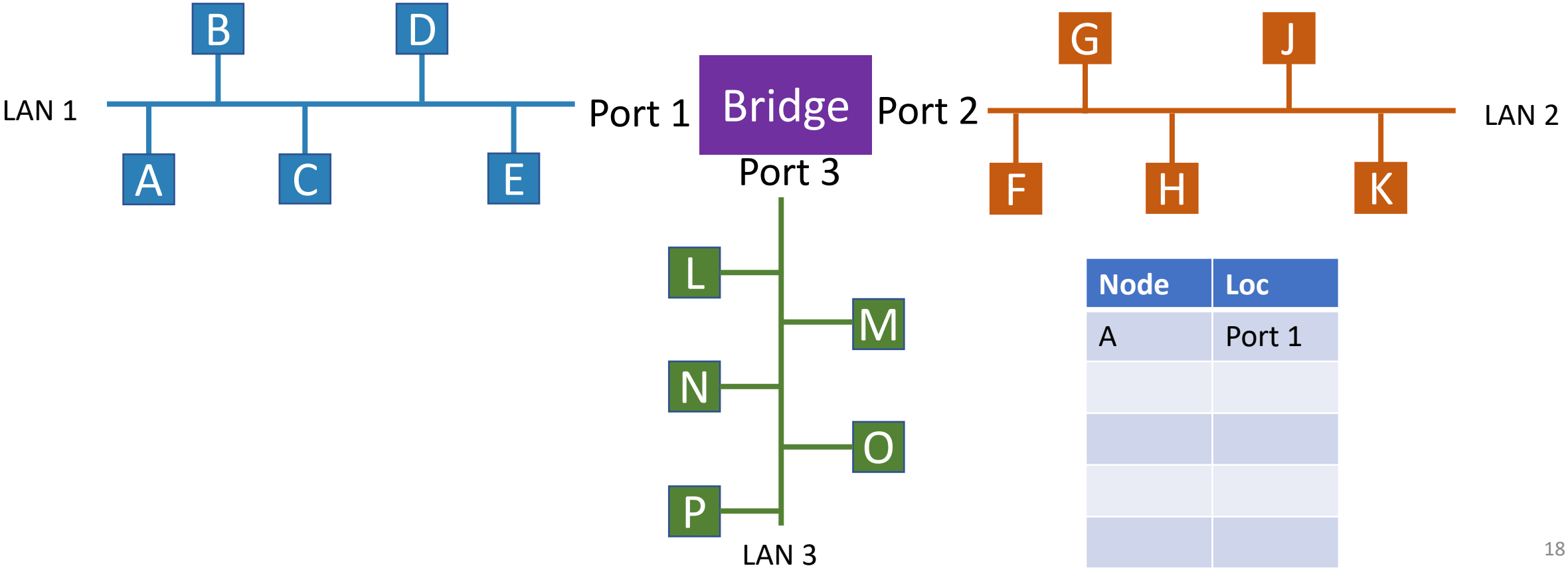
Multiport Bridge

- Learning works the same way over multiple ports



Multiport Bridge

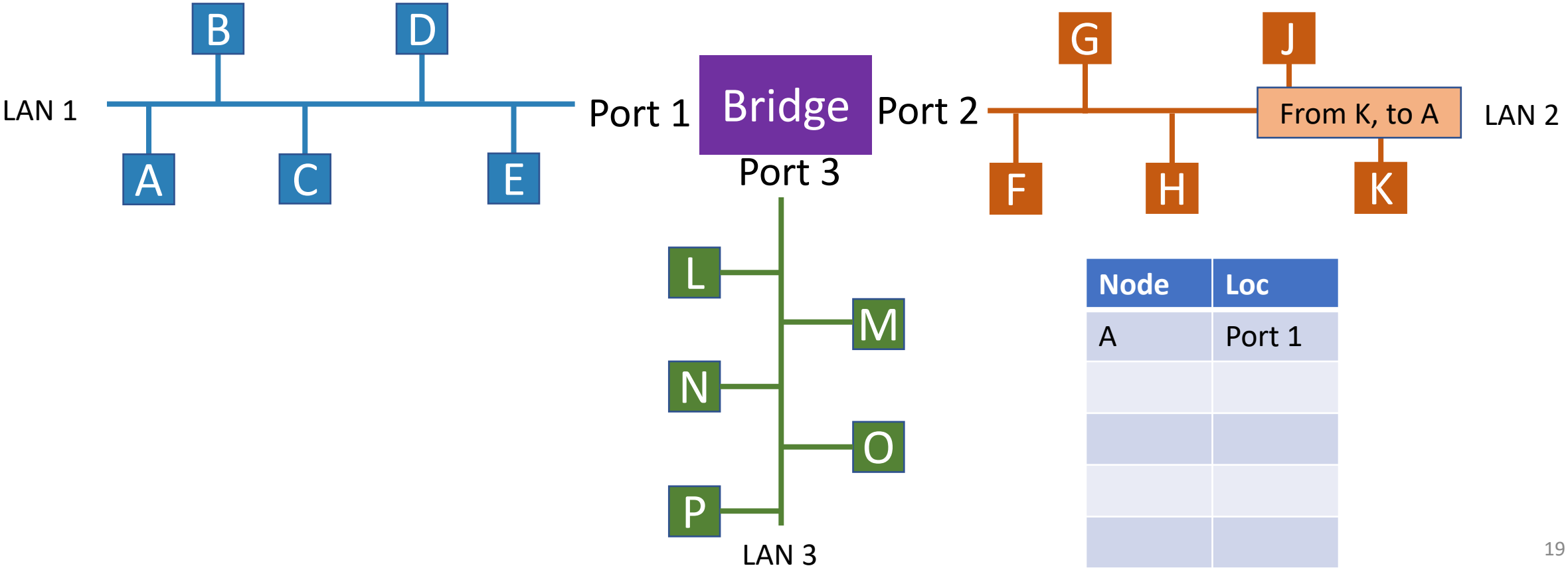
- Learning works the same way over multiple ports



Node	Loc
A	Port 1

Multiport Bridge

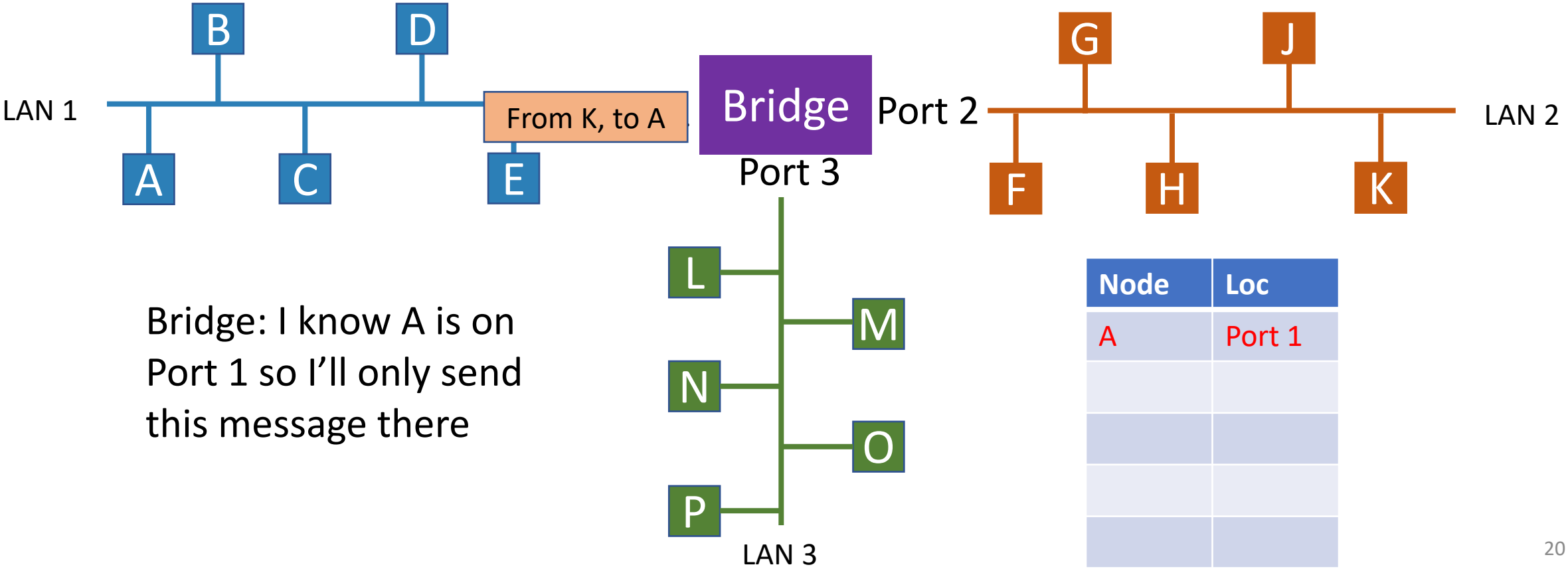
- Learning works the same way over multiple ports



Node	Loc
A	Port 1

Multiport Bridge

- Learning works the same way over multiple ports

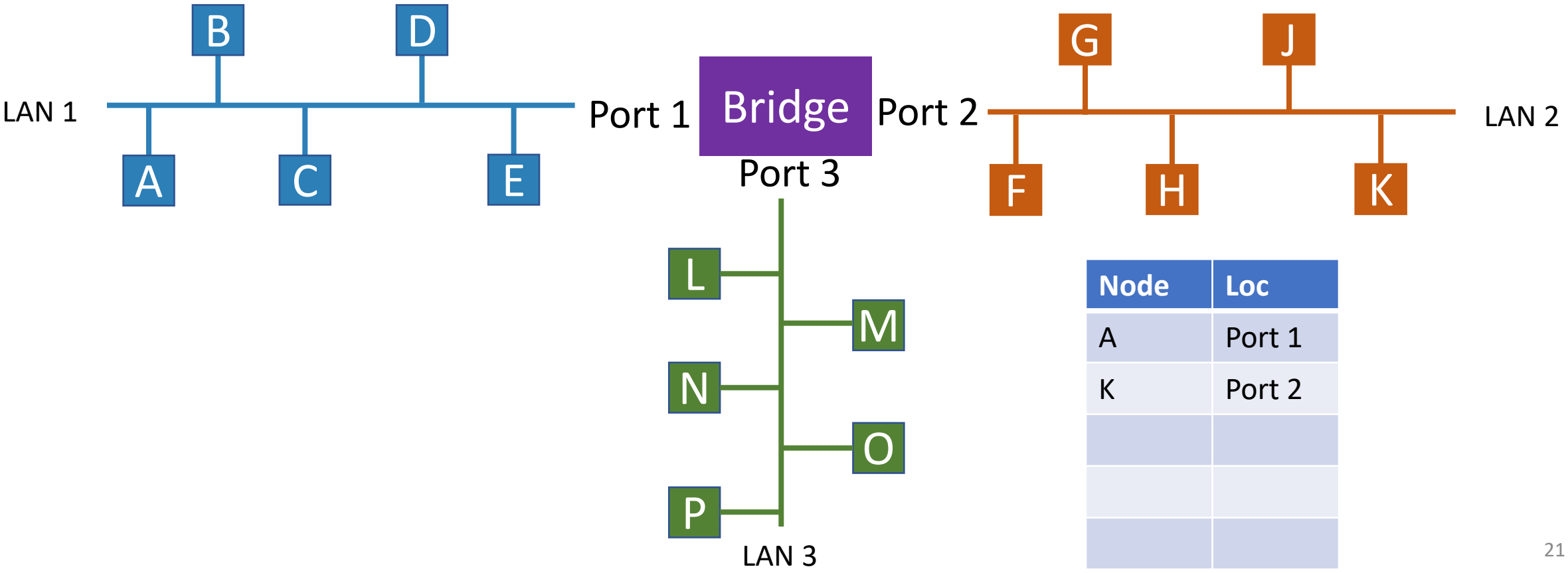


Bridge: I know A is on Port 1 so I'll only send this message there

Node	Loc
A	Port 1

Multiport Bridge

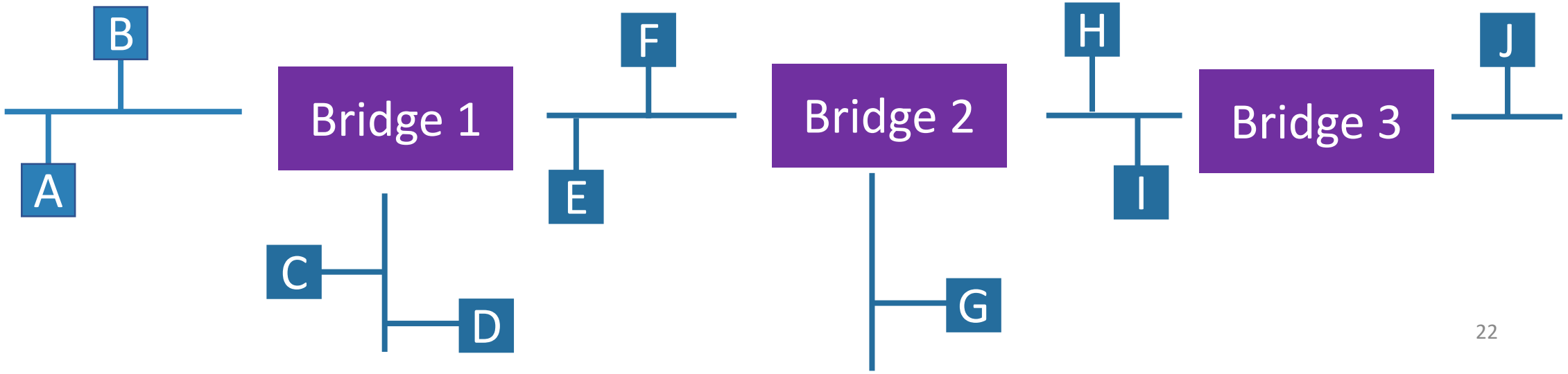
- Learning works the same way over multiple ports



Node	Loc
A	Port 1
K	Port 2	

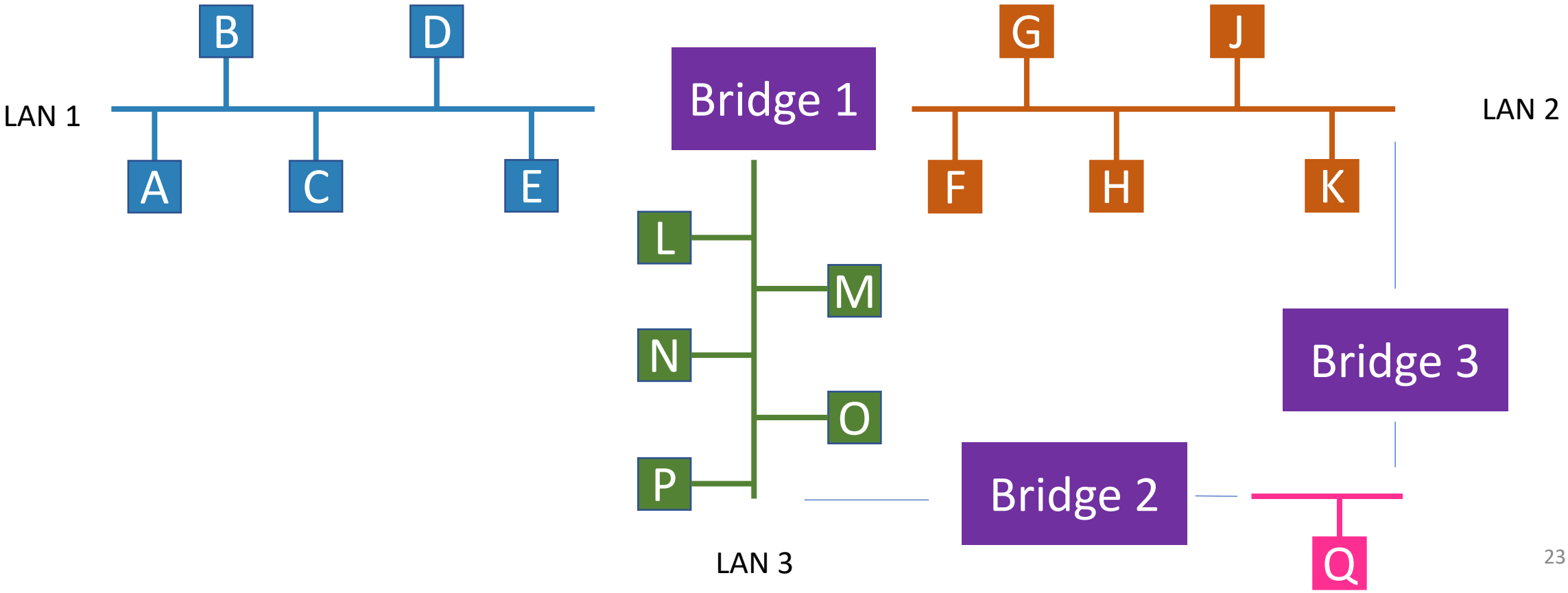
Practice Problem

- List all the nodes that hear each of the following messages. Assume all forwarding bases start empty
  1. A sends to B
  2. J sends to B
  3. H sends to A.
  4) J sends to H
  5) G sends to H
  6) A sends to G



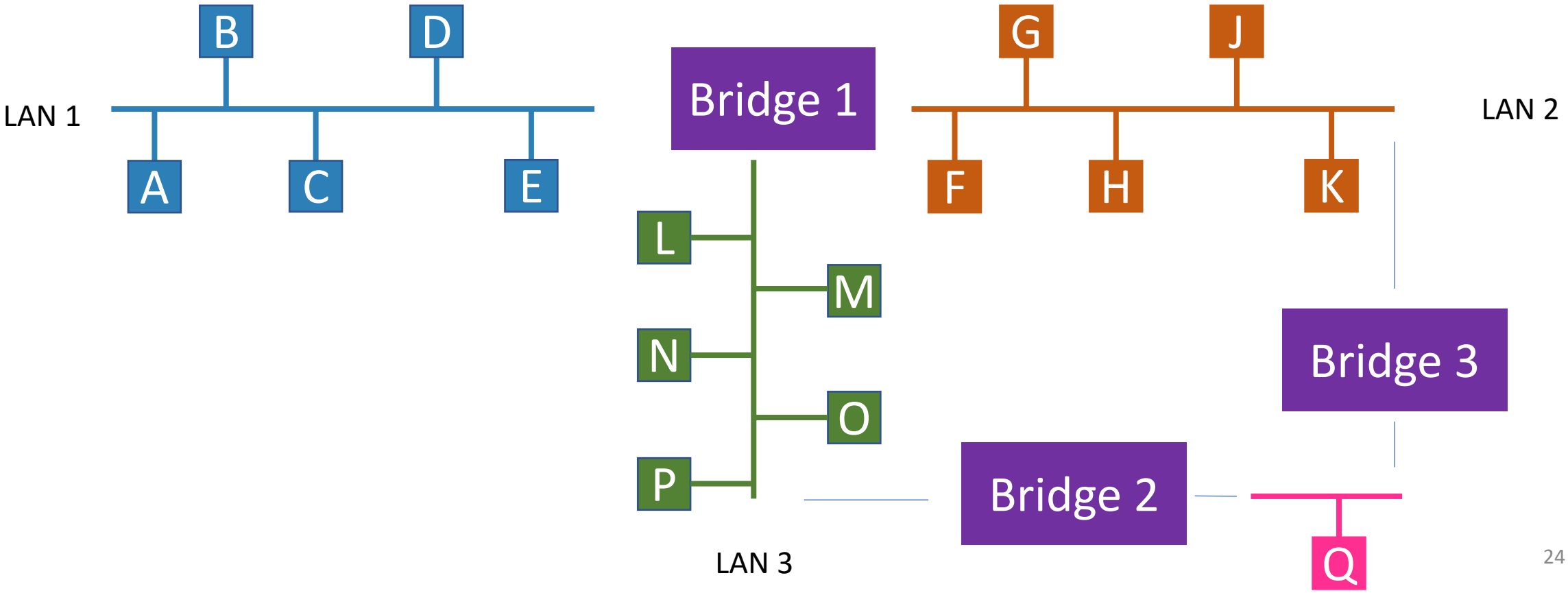
Cycles

- What happens if a cycle is added?



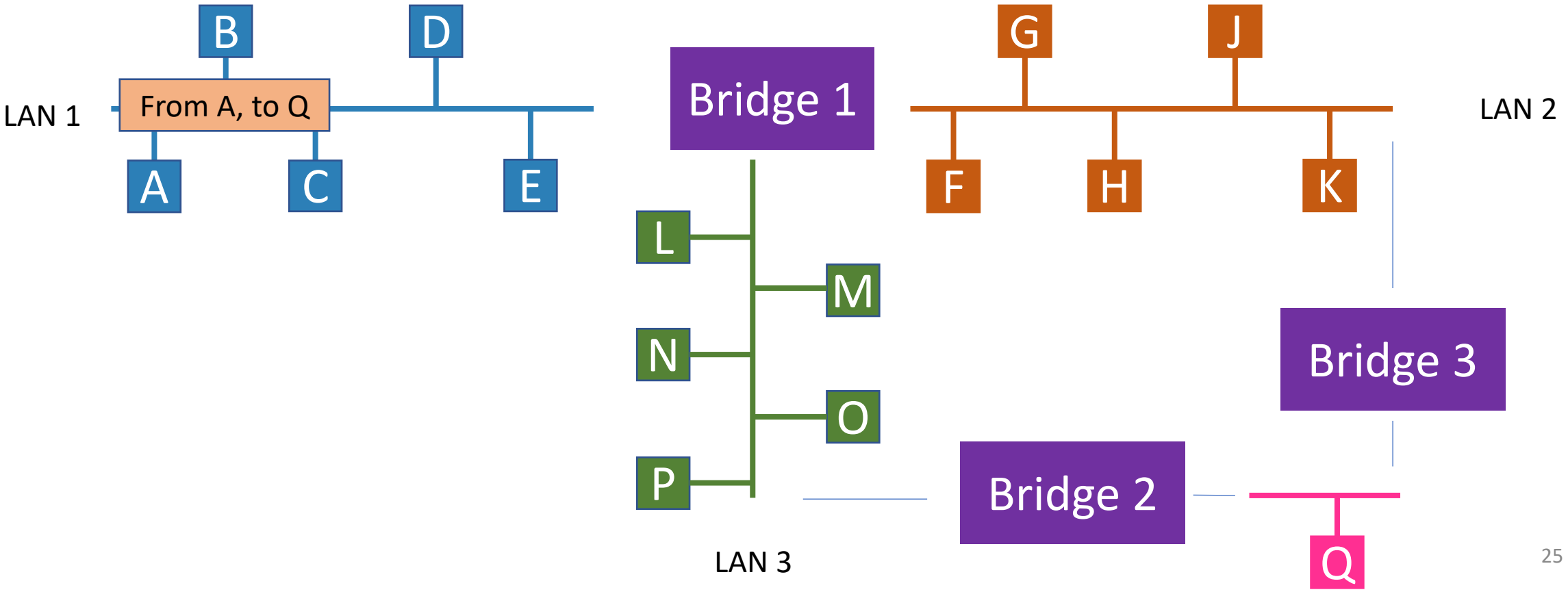
Cycles

- What happens if a cycle is added?



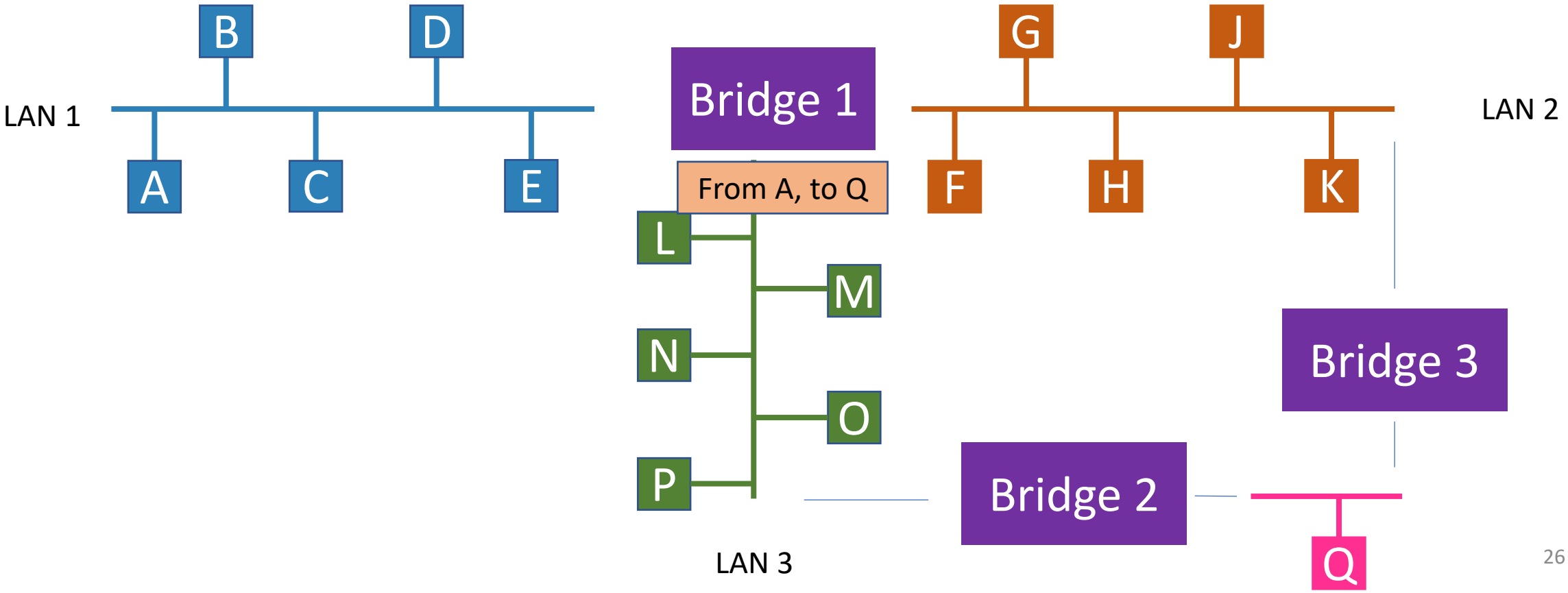
Cycles

- What happens if a cycle is added?



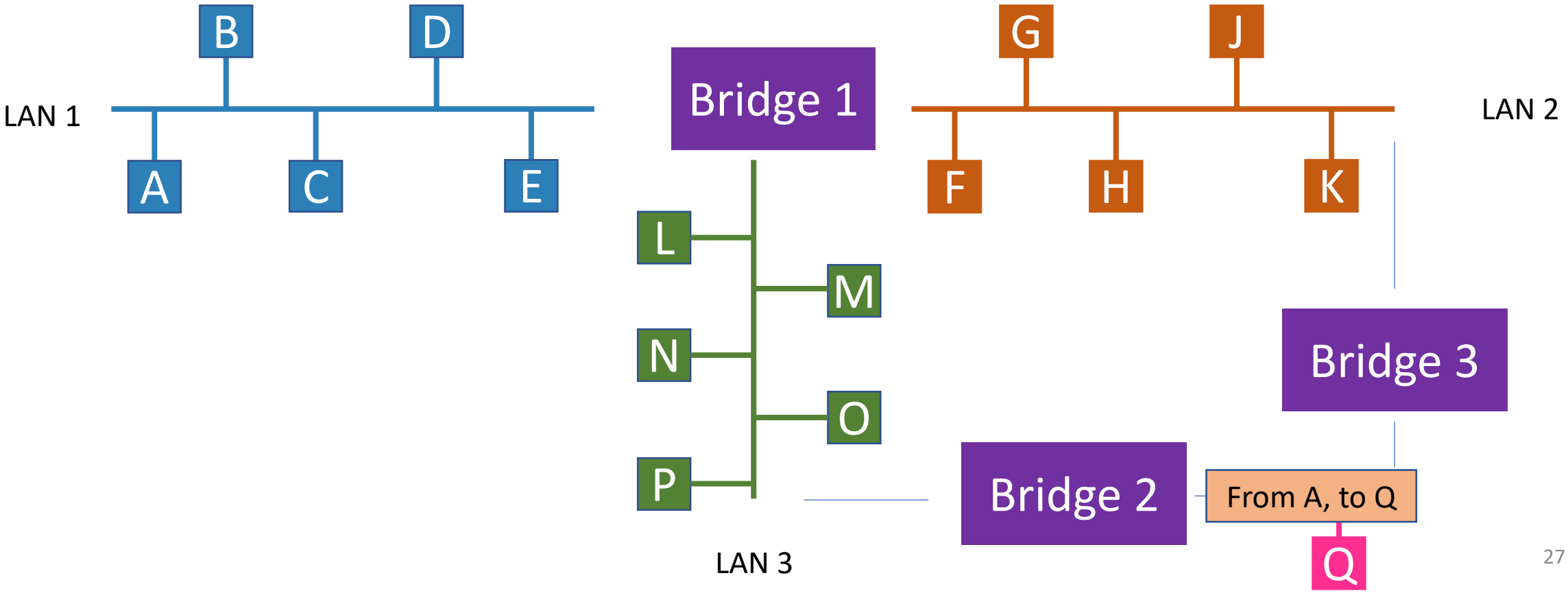
Cycles

- What happens if a cycle is added?



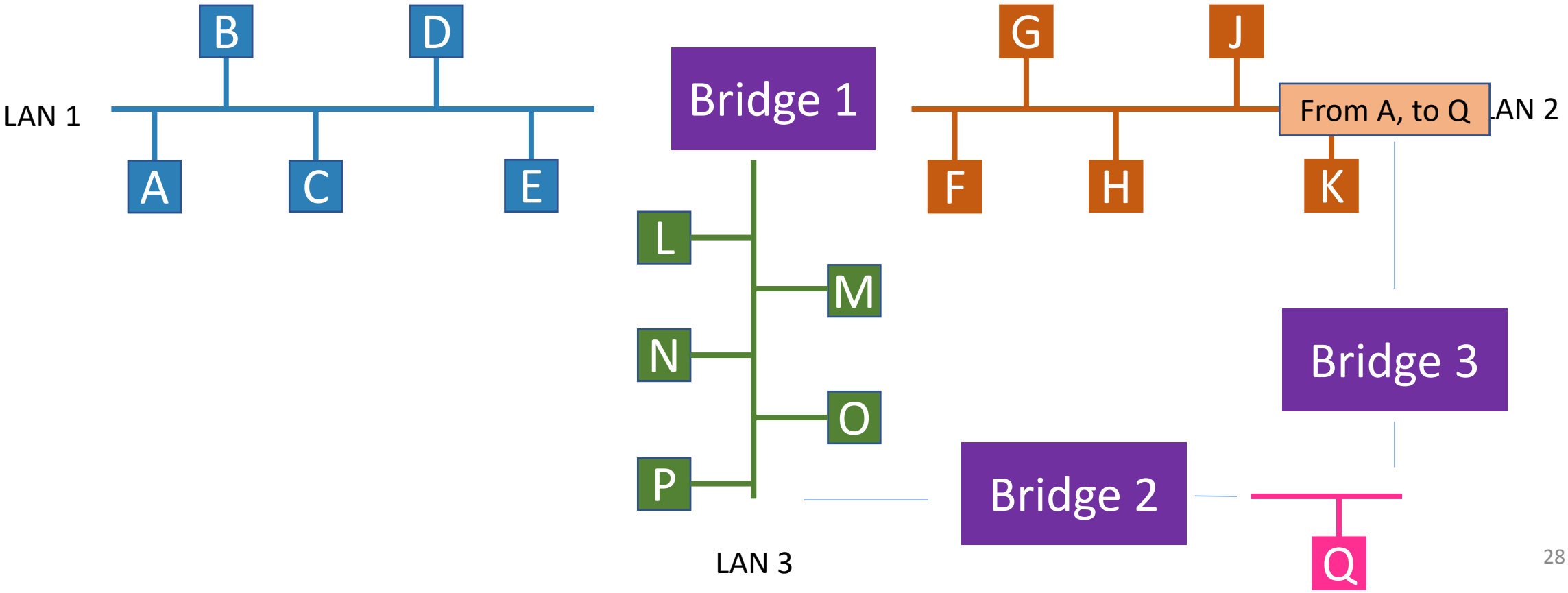
Cycles

- What happens if a cycle is added?



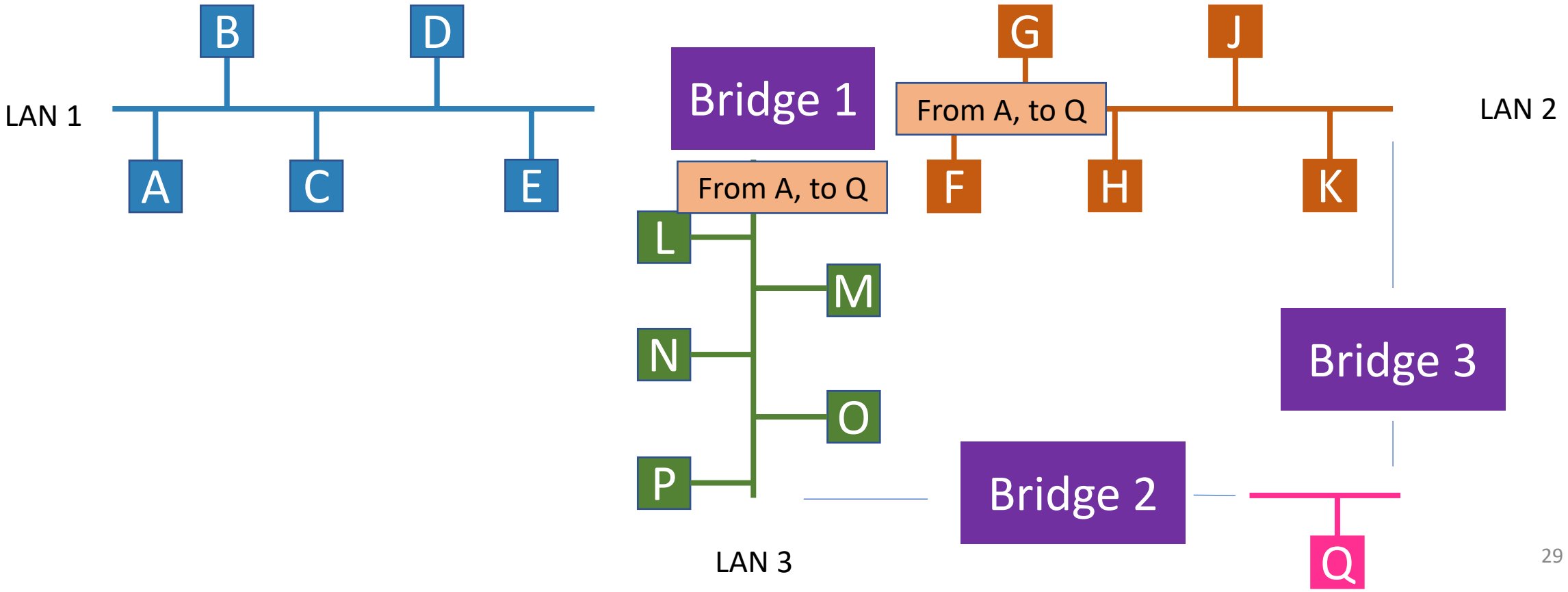
Cycles

- What happens if a cycle is added?



Cycles

- What happens if a cycle is added?

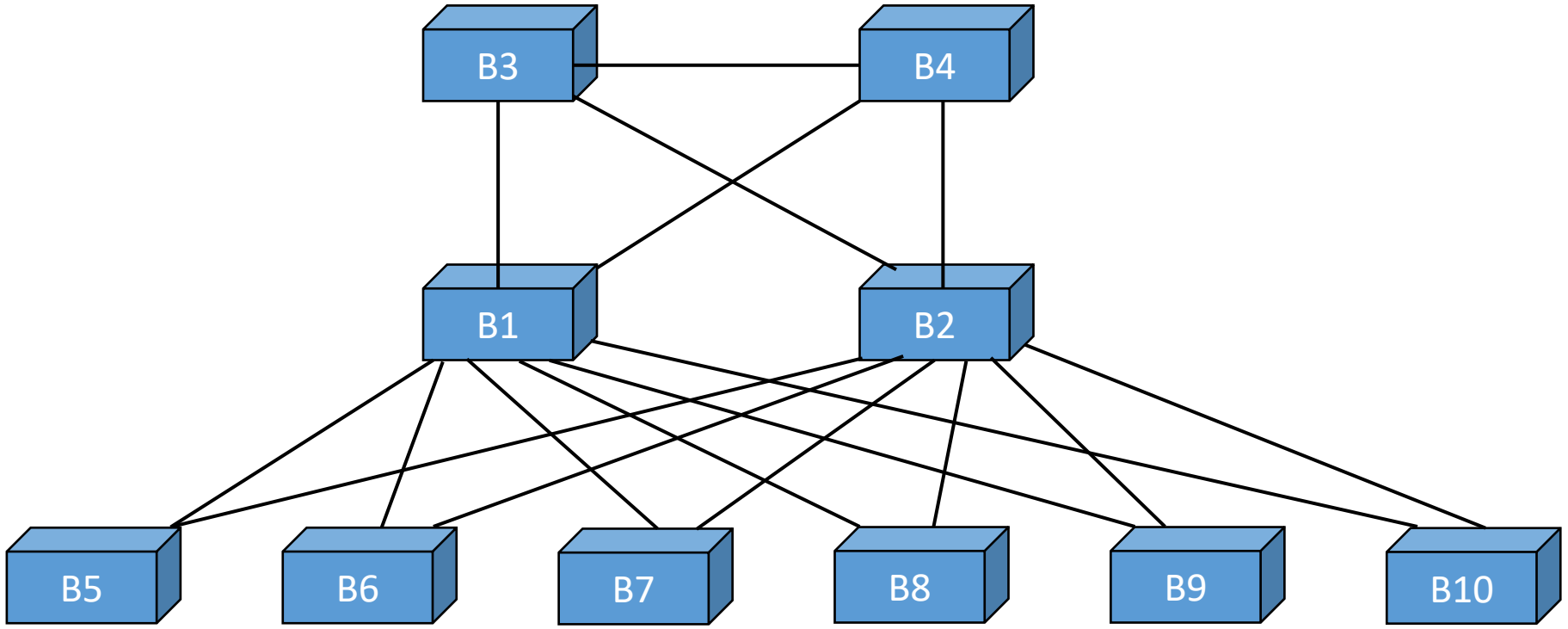


Cycles

- What happens if a cycle is added?
  - Broadcast storm!
- How do we solve it?
  - Turn off some bridges' ports – Spanning Tree Protocol

Spanning Tree Protocol

- Spanning tree uses a subset of bridge ports so there are no cycles



Spanning Tree Protocol

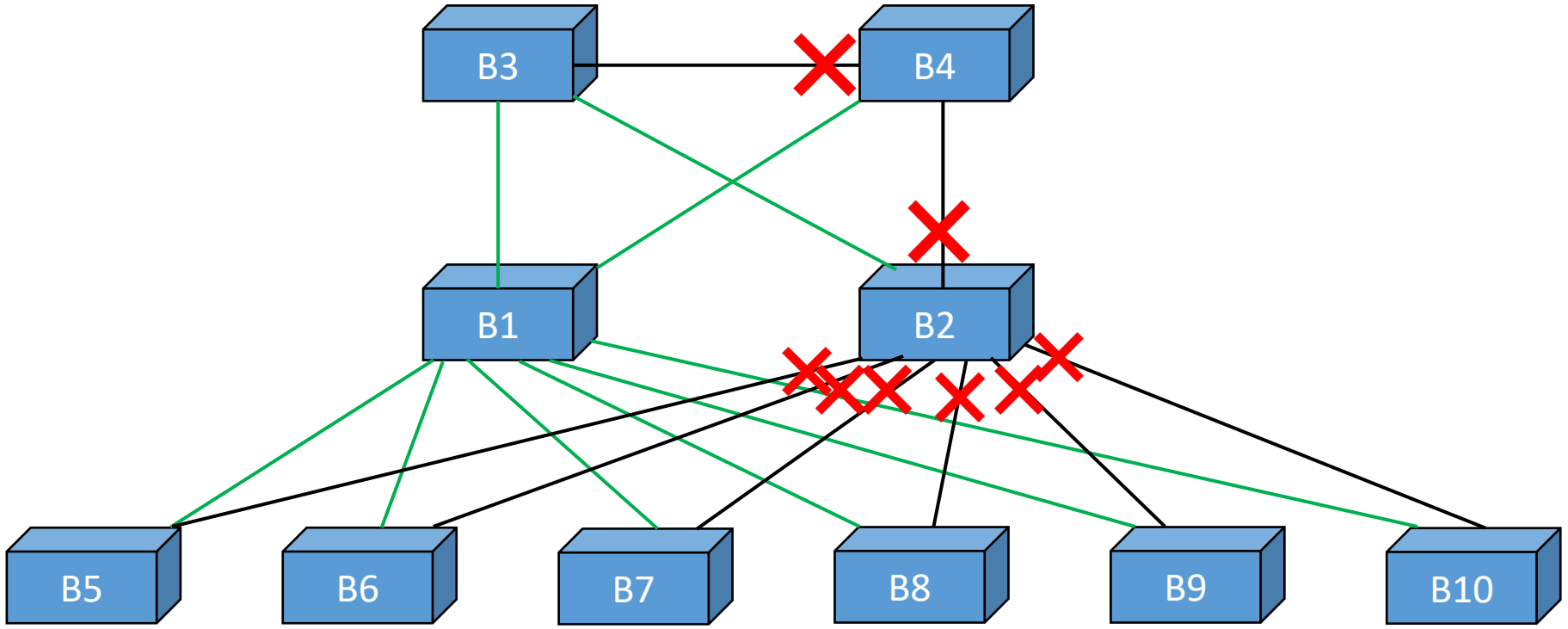
1) Elect a root (lowest Bridge ID)
2) Find root ports
3) Select one designated port (DP) per segment
4) Block all remaining ports

Spanning Tree Protocol

1) Elect a root (lowest Bridge ID)
   - Bridge ID: priority and MAC address
2) Find root ports along shortest paths to root, breaking ties with MAC address of next bridge
3) Select one designated port (DP) per segment
   - If one side is root port, other is designated port
   - Otherwise, choose one DP and one ND based on same criteria (shortest path, break ties with lowest MAC address)
4) Block all non-designated ports

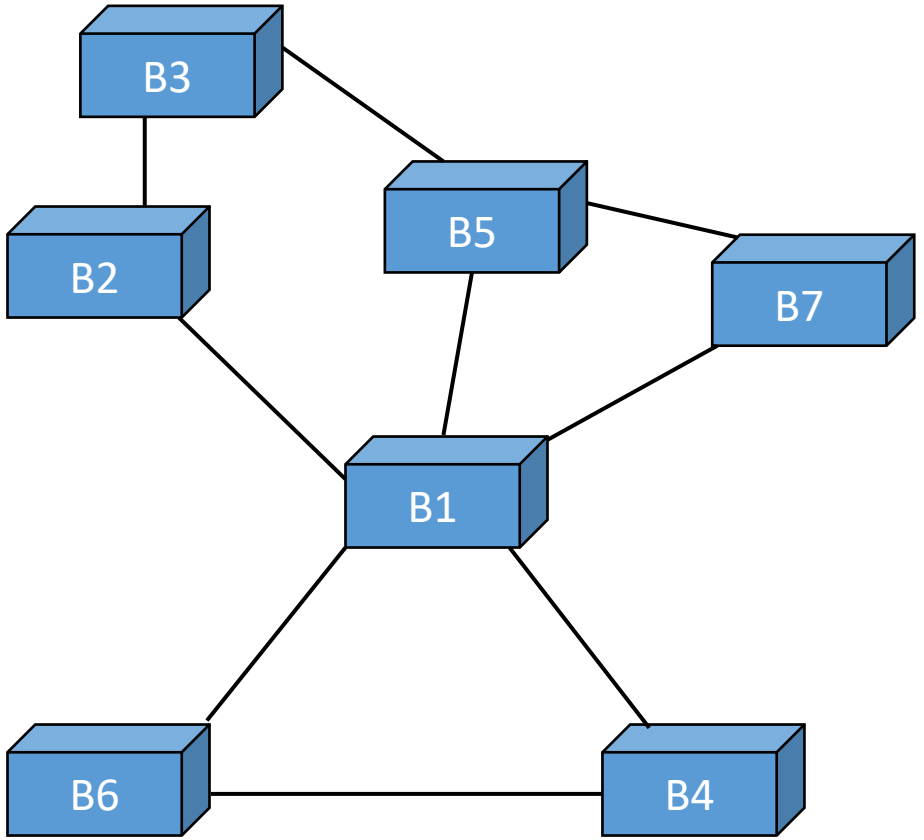
Spanning Tree Protocol

- Spanning tree uses a subset of bridge ports so there are no cycles



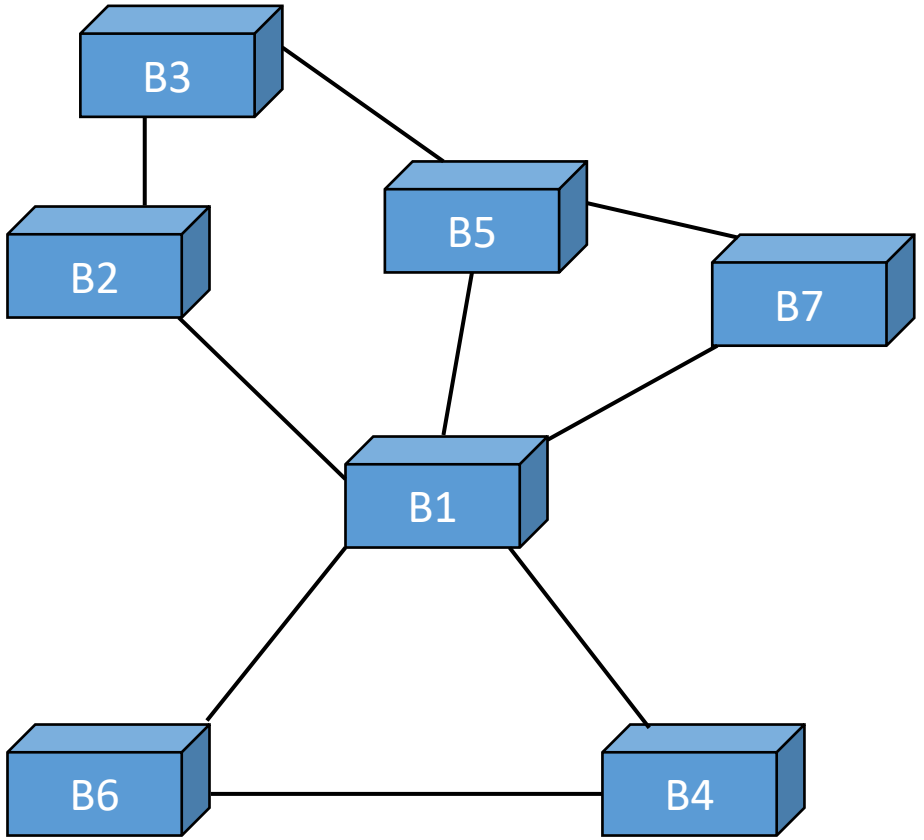
Practice Problem

- Label all ports according to STP rules and decide which ports to turn off



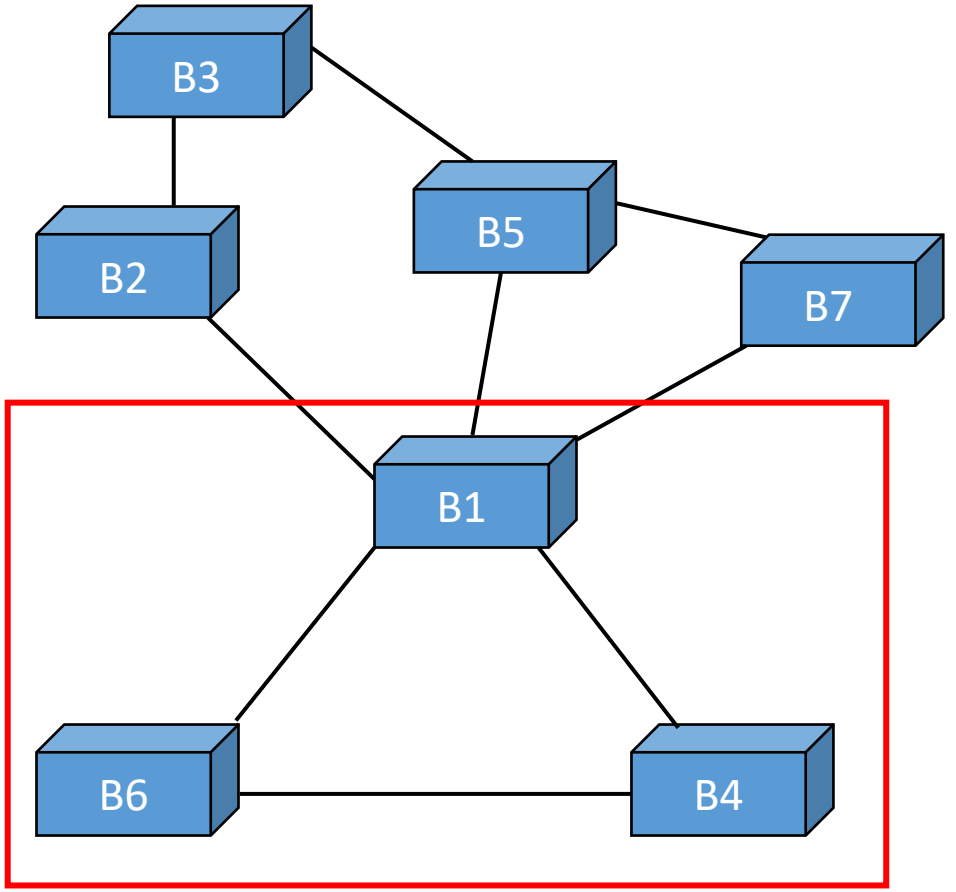
Distributed Communication Details

- Each bridge sends messages (BPDUs) to all others that it's connected to:
- "I am Bridge B6, the root bridge is B1, and I'm 1 hop away"
- (B6, B1, 1)



Distributed Communication Details

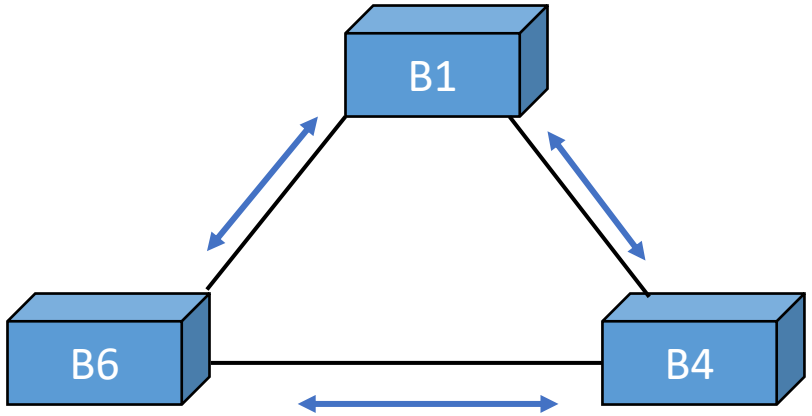
- Each bridge starts by thinking it's the root



Distributed Communication Details

- Each bridge starts by thinking it's the root

Me	Root	Hops
B1	B1	0



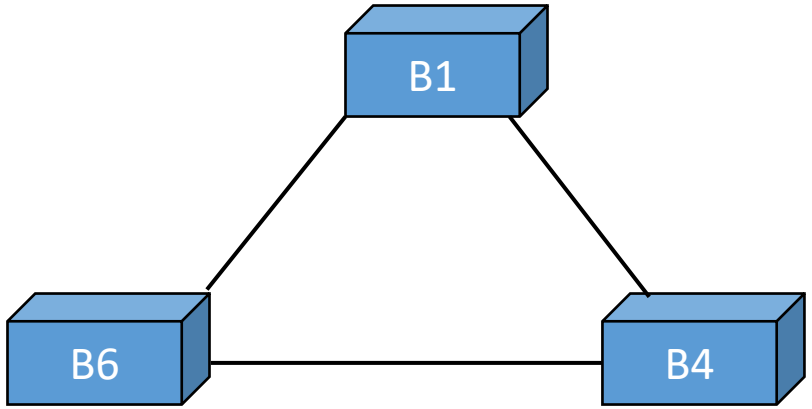
Me	Root	Hops
B6	B6	0

Me	Root	Hops
B1	B6	0

Distributed Communication Details

- Each bridge starts by thinking it's the root
- Each bridge updates route/Root upon receiving others' messages
  1. Better root
  2. Lower cost to get there
  3. MAC address

Me	Root	Hops
B1	B1	0



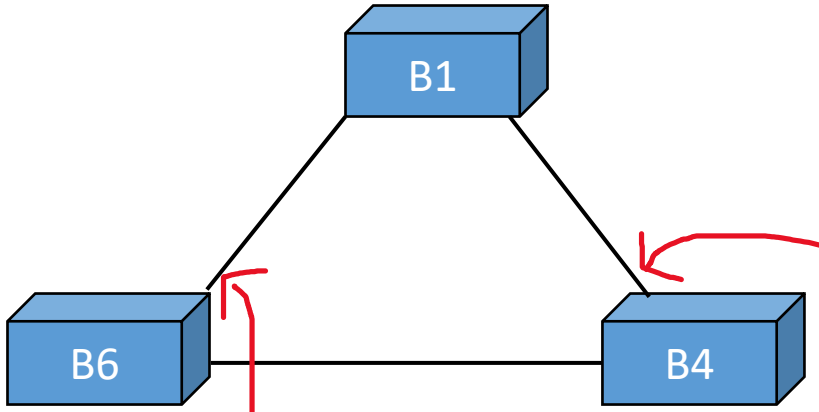
Me	Root	Hops
B6	B6	0

Me	Root	Hops
B4	B4	0

Distributed Communication Details

- Each bridge starts by thinking it's the root
- Each bridge updates route/Root upon receiving others' messages

Me	Root	Hops
B1	B1	0

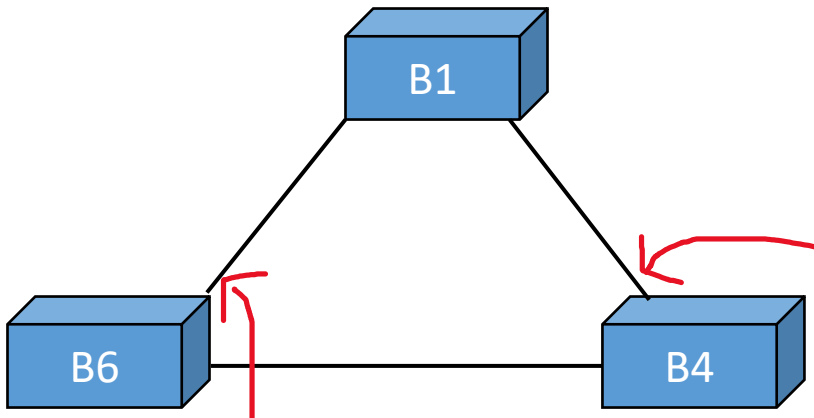


Me	Root	Hops	RP
B6	B1	1	

Me	Root	Hops	RP
B4	B1	1	

Distributed Communication Details

- Each bridge starts by thinking it's the root
- Each bridge updates route/Root upon receiving others' messages
- Rebroadcast only to the non-root ports where we're best



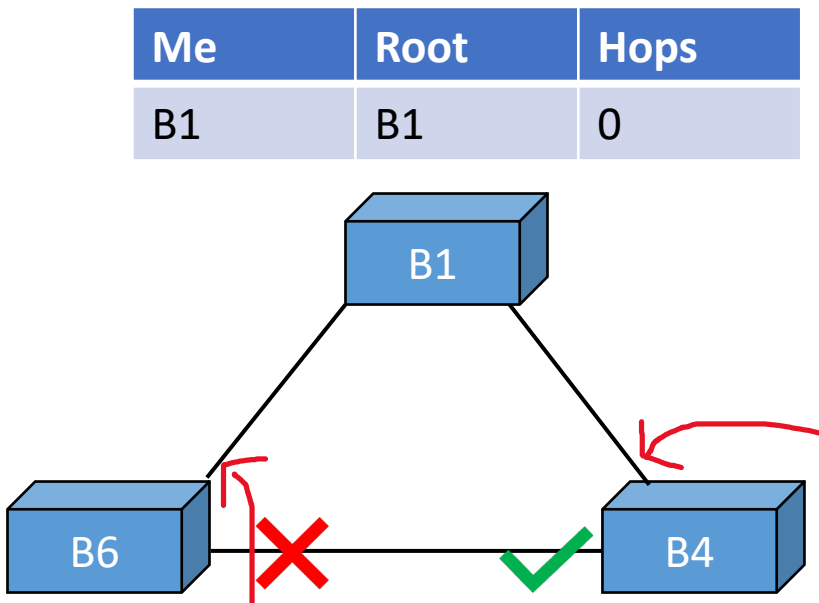
Me	Root	Hops
B1	B1	0

Me	Root	Hops	RP
B6	B1	1	

Me	Root	Hops	RP
B4	B1	1	

Distributed Communication Details

- Each bridge starts by thinking it's the root
- Each bridge updates route/Root upon receiving others' messages
- Rebroadcast only to the non-root ports where we're best – turn off others



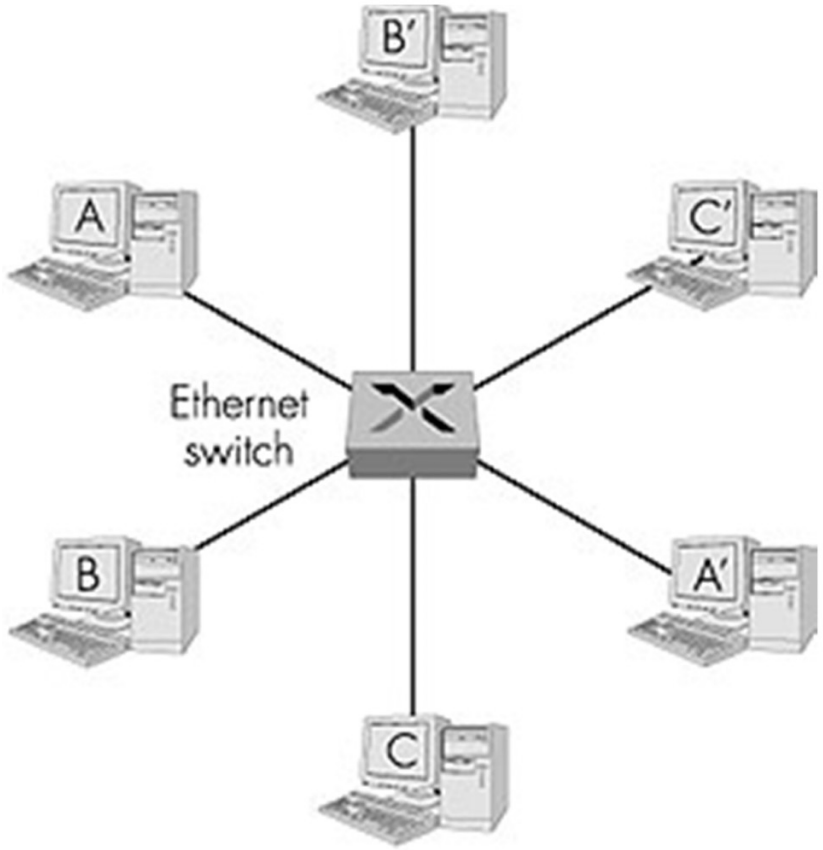
Me	Root	Hops
B1	B1	0

Me	Root	Hops	RP
B6	B1	1	

Me	Root	Hops	RP
B4	B1	1	

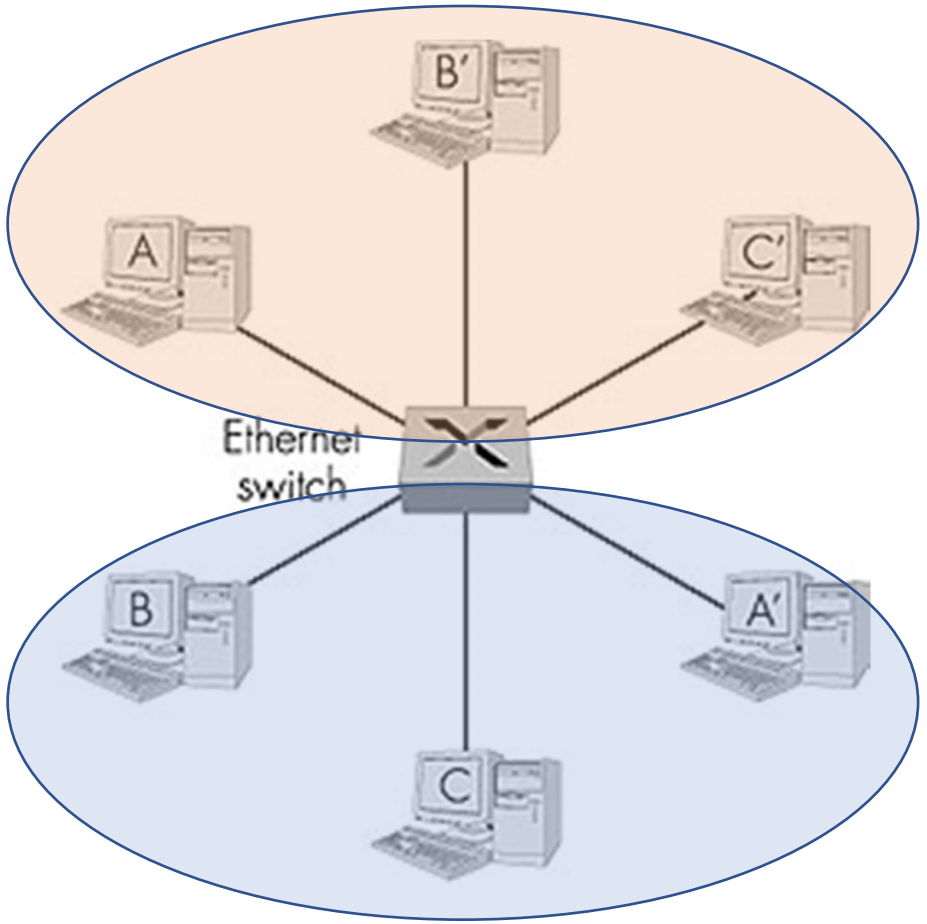
Switched Ethernet

- Ethernet now works in a point-to-point manner – nodes directly connects to bridge
- **Switch** (conceptually interchangeable with bridge) forwards packets to the correct port, and only one host resides at that port
- Switch supports parallel forwarding (A can send to B while B sends to A)
- No need for shared medium access (CSMA, etc) anymore

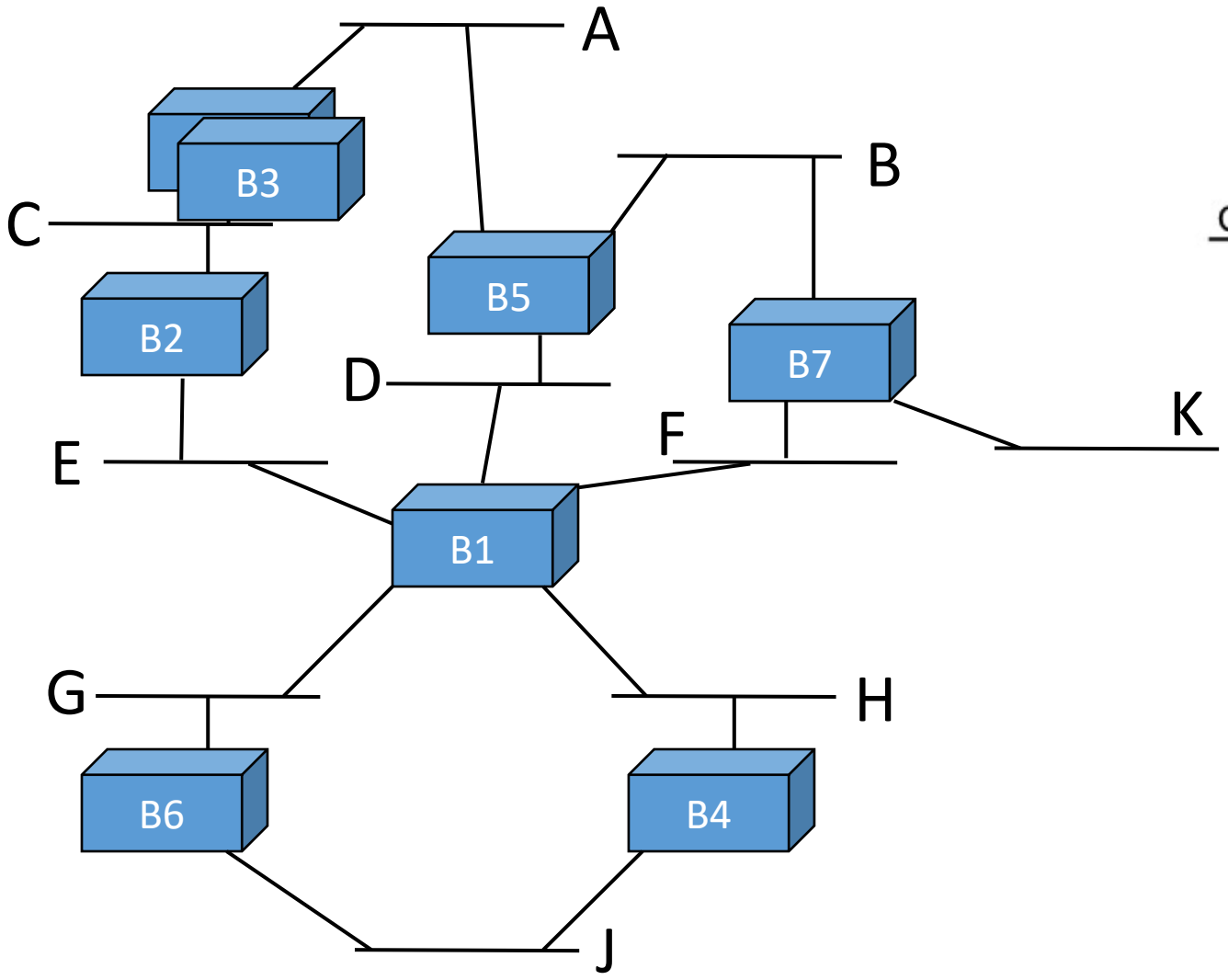


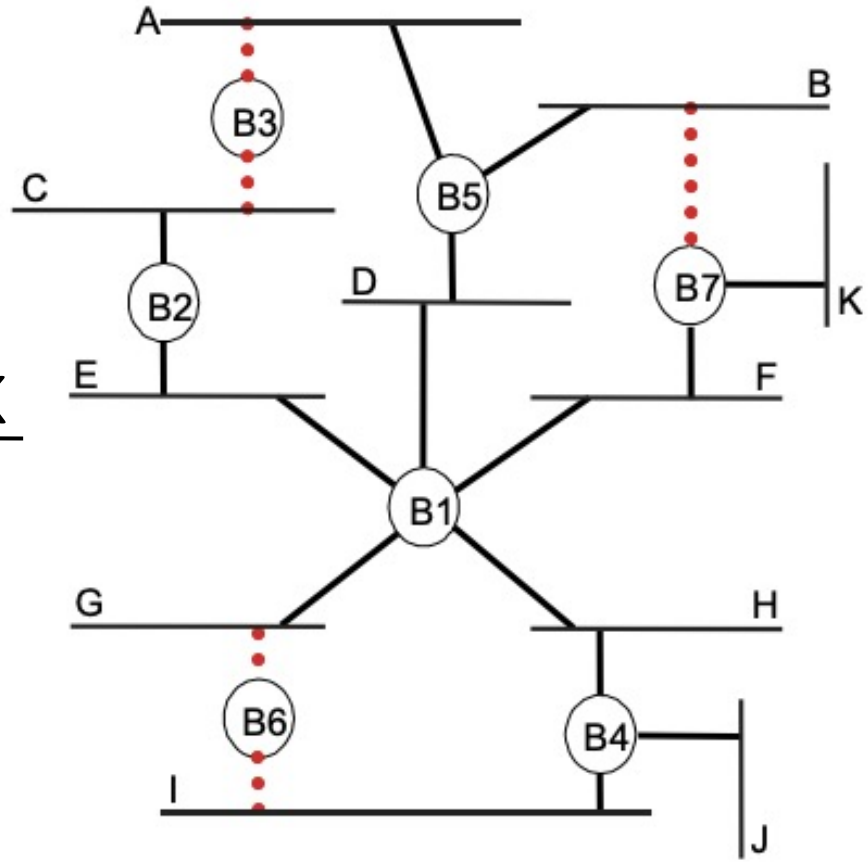
VLANs

- Create virtual LANs to group certain nodes together
- Efficient if you want to send to the same group of nodes consistently
- Ex/ computer science and physics departments are on the same LAN



Spanning Tree Protocol





Spanning Tree Protocol