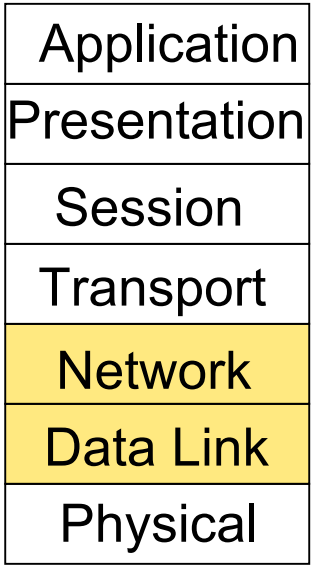
CSE/EE 461
Connecting LANs

Readings: pp 165-192, 271-299

SWITCHING

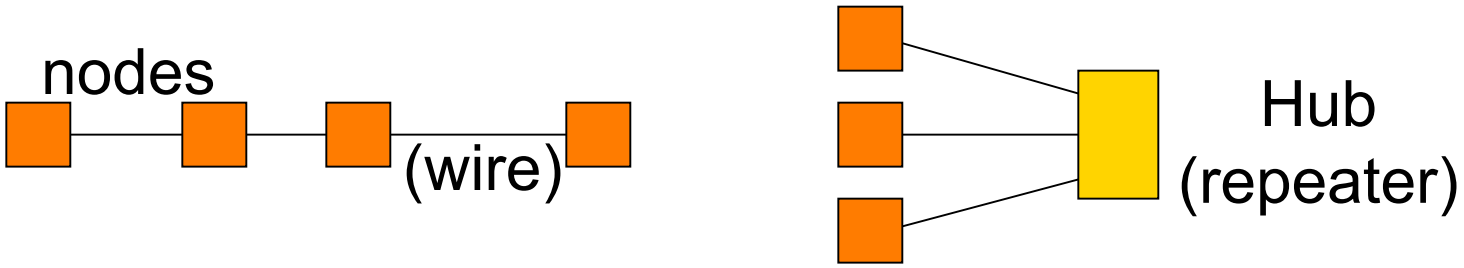
- Focus:
  - What to do when one shared LAN isn't big enough?

- Interconnecting LANs
  - Bridges and LAN switches
  - A preview of network routing.



Limits of a LAN

- One shared LAN can limit us in terms of:
  - Distance
  - Number of nodes
  - Performance



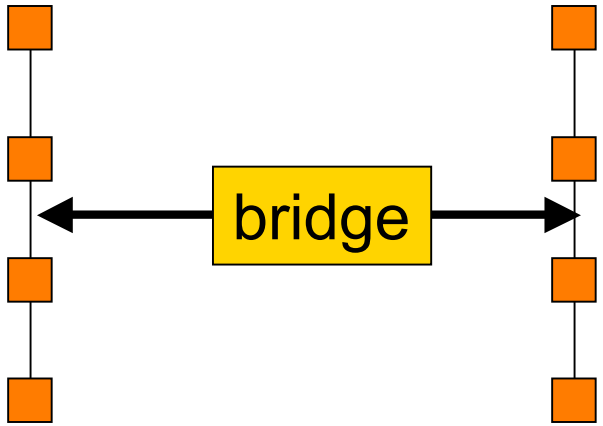
- How do we scale to a larger, faster network?
  - We must be able to interconnect LANs

SWITCHING

- Xferring a packet from one network to another
- Packet switched vs. circuit switched
- Connection vs. Connectionless
- Contention vs. Congestion

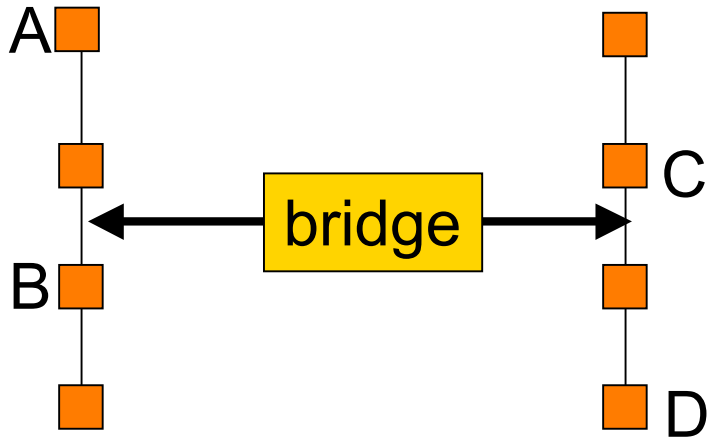
Bridges and Extended LANs

- "Transparently" interconnect LANs with bridge
  - Receive frames from each LAN and forward to the other
  - Each LAN is its own collision domain; bridge isn't a repeater
  - Could have many ports or join to a remote LAN



Backward Learning Algorithm

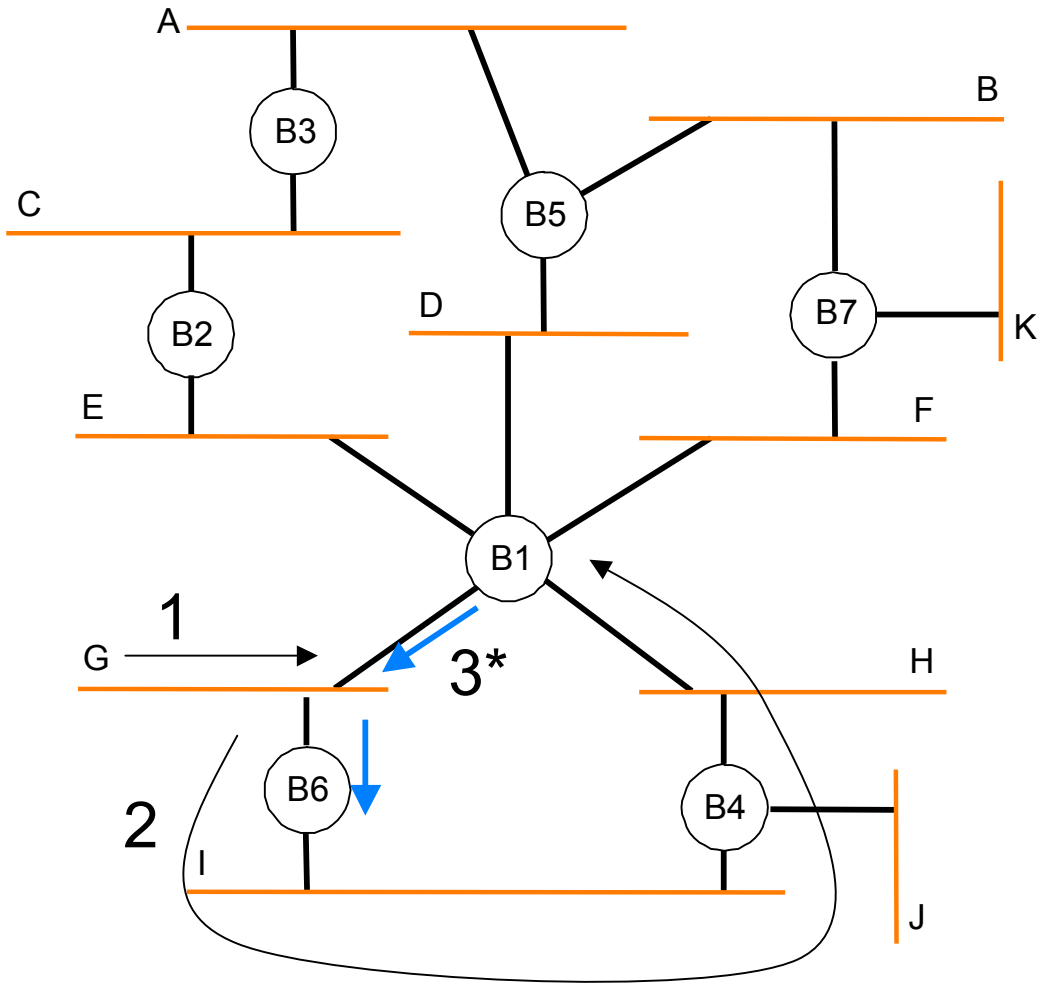
- To optimize overall performance:
  - Shouldn't forward A→B or C→D, should forward A→C and D→B



- How does the bridge know?
  - Learn who is where by observing source addresses and prune
  - Forward using destination address; age for robustness

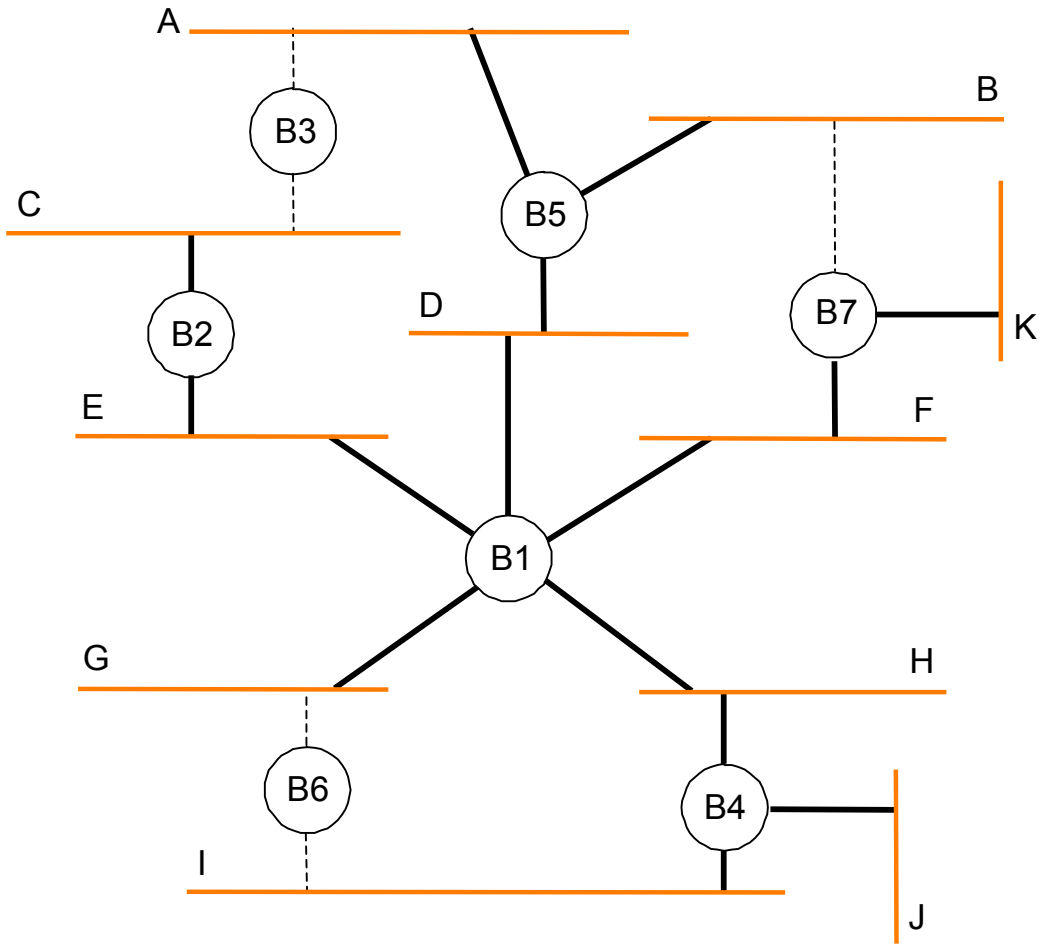
Why stop at one bridge?

- But to avoid loops we must forward only on select bridge ports!
- The Spanning Tree algorithm does this
- It is separate from backward learning



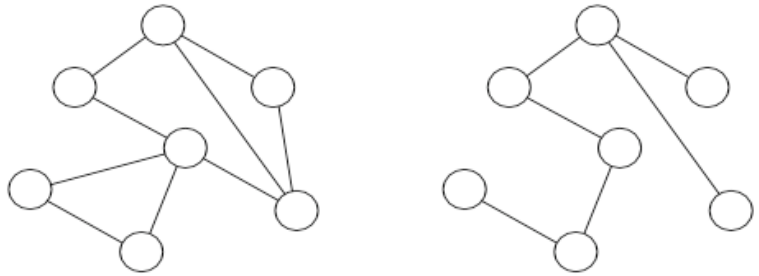
Spanning Tree Example

- Spanning tree uses select bridges so there are no cycles
- Only one tree
  - Prune some ports
    - Switch turns off output link to a common LAN
- Q: How do we find a spanning tree?
  - Manually?
  - Automatically



Spanning Tree

- Compute ST with
  - *Single root* bridge such that
    - Root forwards onto all of its outgoing ports
  - *Designated bridge* per LAN such that
    - Only the designated bridge forwards packets
- Can be used with backward learning

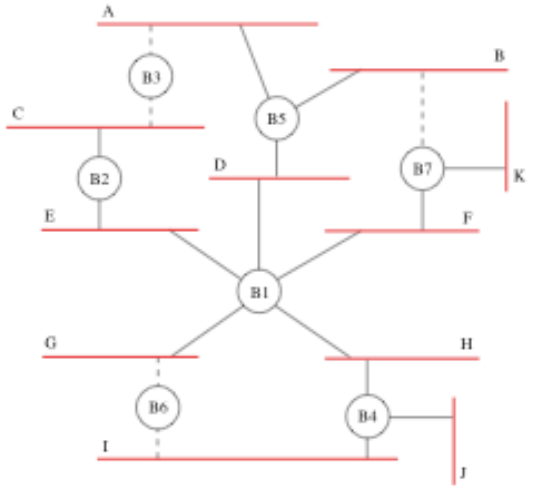


Spanning Tree Algorithm

- Distributed algorithm to compute spanning tree
  - Robust against failures, needs no organization
  - Developed by Radia Perlman at DEC
    - IEEE 802.1 spec
- Outline: Goal is to turn some bridge ports off
  - Each bridge starts off thinking (and saying)
    - I am the root.
    - I am the best path to the root.
  - Over time, it may learn that it is not the root, but continues to believe (and say) it is on the best path.
  - Eventually, it may learn that it is not even on the best path
    - So it stops announcing that it is.
      - Shuts down outgoing link.
- No Host Involvement Required

Algorithm Goal

- Each bridge has a unique id (e.g., B1, B2, B3)
- Select bridge with smallest id as root
- Select bridge on each LAN that is closest to the root as that LAN's designated bridge (use id to break ties)
- Each bridge forwards frames over each LAN for which it is the designated bridge



Algorithm Implementation

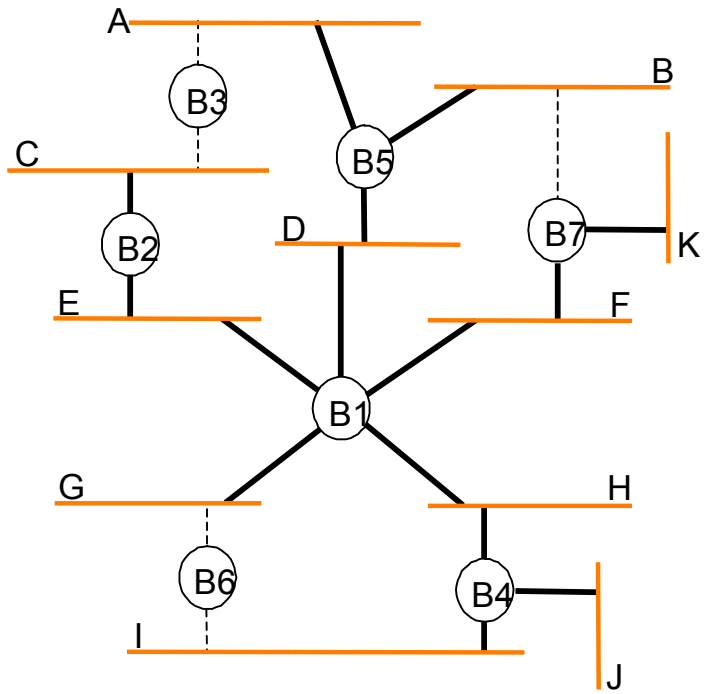
- Bridges exchange configuration messages
  - id for bridge sending the message
  - id for what the sending bridge believes to be root bridge
  - distance (hops) from sending bridge to root bridge
- Each bridge records current best configuration message for each port
- Initially, each bridge believes it is the root
- When learn not root, stop generating configuration message
  - in steady state, only root generates configuration messages

Algorithm More…

- When learn not designated bridge, stop forwarding configuration messages
  - in steady state, only designated bridges forward configuration messages
- Root bridge continues to send configuration messages periodically
- If any given bridge does not receive configuration message after a period of time, starts generating configuration messages claiming to be to be the root

Algorithm Example

- Message format:
  (root, dist-to-root, bridge-to-root)
- Sample messages sequences to and from B3:
  1. B3 receives (B2, 0, B2) from B2
  2. B3 accepts B2 as root (2 < 3)
  3. B3 sends (B2, 1, B3) to B5
  4. B2 accepts B1 as root and sends (B1, 1, B2) to B3
  5. B5 accepts B1 as root and sends (B1, 1, B5) to B3
  6. B3 accepts B1 as root
     a. Sees B2 as "closer"
     b. Shuts down outgoing link to LAN A.
     c. Sees B5 as "closer"
     d. Shuts down outgoing link to LAN B.



Some other tricky details

- Configuration information is aged
  - If the root fails a new one will be elected
- Reconfiguration rate is damped
  - Adopt new spanning trees slowly to avoid temporary loops

Limitations of Bridges/Switches

- Little control over forwarding paths
  - Closer may not mean closer.
- Spanning tree algorithm limits reconfiguration speed
- True Broadcast traffic flows freely over whole extended LAN
- Size of bridge forwarding tables grows with number of hosts
- All explains why can't we build a large network using bridges

Key Concepts

- We can overcome LAN limits by interconnection
  - Bridges and LAN switches
  - But there are limits to this strategy …
- Next Topic: Routing and the Network layer
  - How to grow large and really large networks