

Lecture 8:

Networks to Internetworks

CSE 123: Computer Networks

Aaron Schulman





Lecture 10 Overview

- Bridging & switching
 - ◆ Learning bridges
 - ◆ Spanning Tree
- Internetworking
 - ◆ Routing
 - ◆ Internet Protocol



Learning Bridges Recap

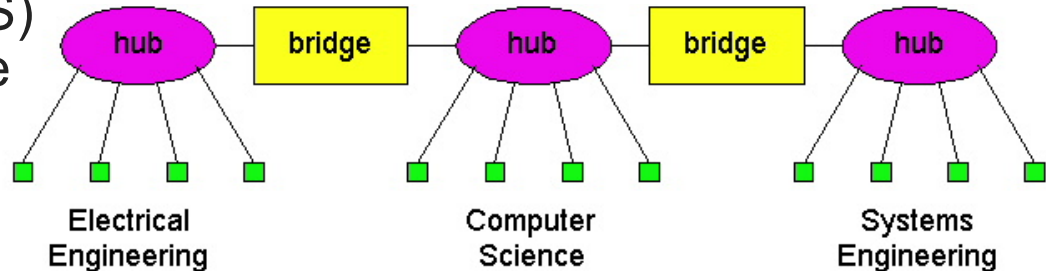
- Each bridge keeps a list mapping link-layer *destination* address to port number
 - ◆ (what are the directions to this destination?)
- This list is populated by looking at the *source* address of each packet it receives on a given port and entering those values in the table
 - ◆ (if a packet from *A* came from port *x*, then packets to *A* should be sent on port *x*)
- If a packet arrives with a destination address not in the table, then send on all ports (except the one it came on)
- Simple, automatic, self healing



Network Topology

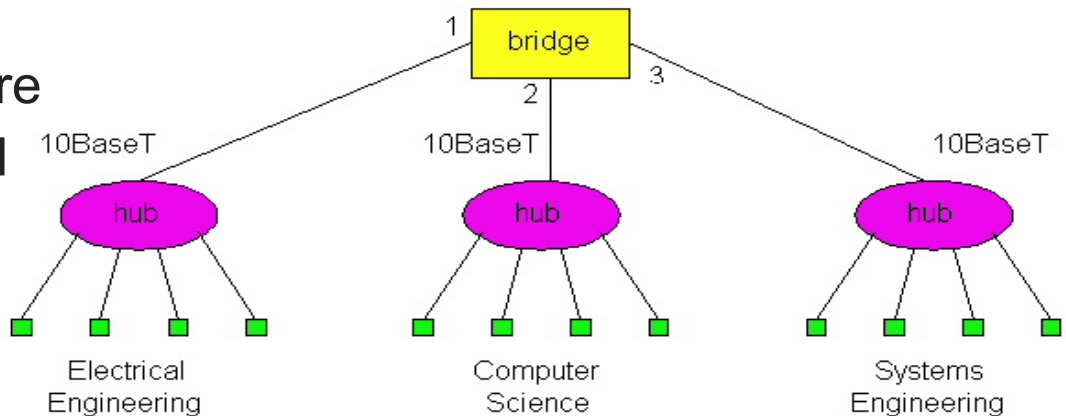
- Linear organization

- ◆ Inter-bridge hubs (e.g. CS) are single points of failure
- ◆ Unnecessary transit (e.g. EE $\leftrightarrow$ SE must traverse CS)



- Backbone/tree

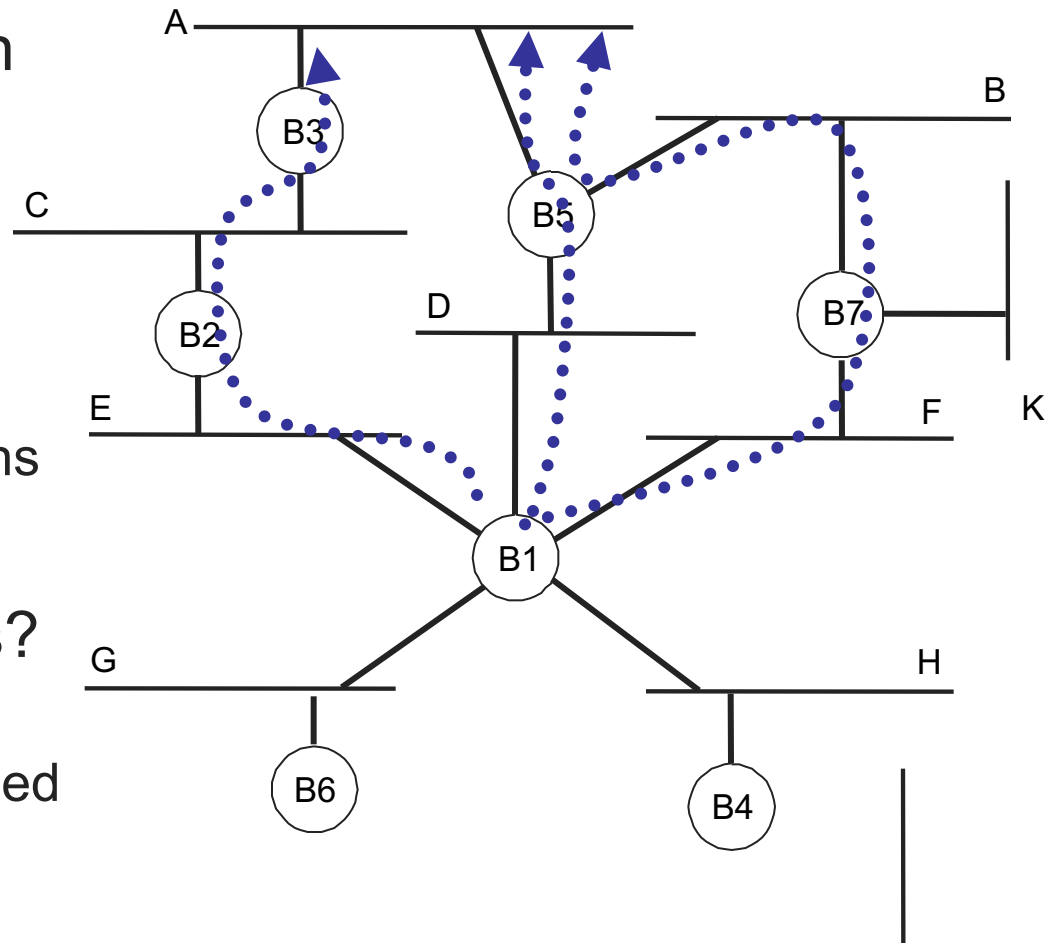
- ◆ Can survive LAN failure
- ◆ Manages all inter-LAN communication
- ◆ Requires more ports





An Issue: Cycles

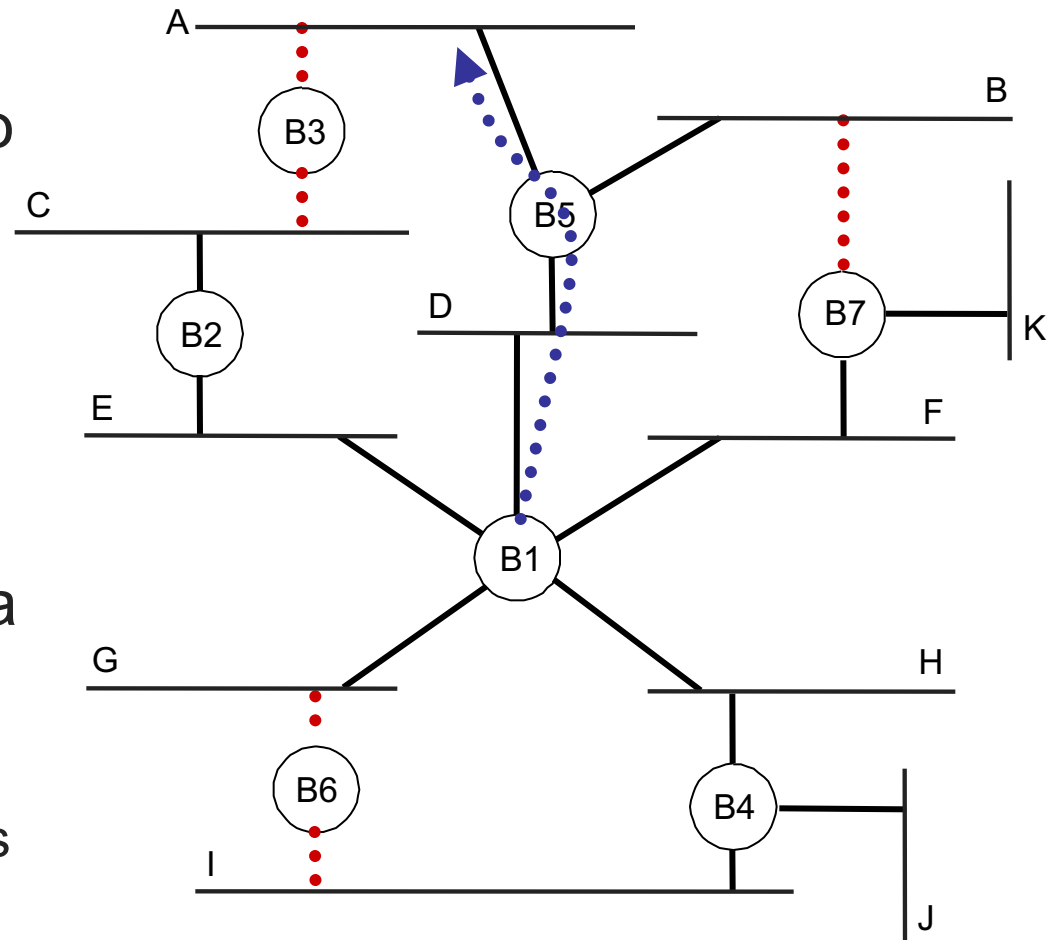
- Learning works well in tree topologies
- But trees are fragile
 - ◆ Net admins like redundant/backup paths
- How to handle Cycles?
 - ◆ Where should *B1* forward packets destined for LAN A?



Spanning Tree



- Spanning tree uses *subset* of bridges so there are no cycles
 - ◆ Prune some ports
 - ◆ Only **one** tree
- Q: How do we find a spanning tree?
 - ◆ Automatically!
 - ◆ Elect root, find paths



Spanning Tree Algorithm



- Each bridge sends periodic configuration messages
 - ◆ (RootID, Distance to Root, BridgeID)
 - ◆ All nodes think they are root initially
- Each bridge updates route/Root upon receipt
 - ◆ Smaller root address is better
 - ◆ Select port with lowest cost to root as “root port”
 - ◆ To break ties, bridge with smaller address is better
- Rebroadcast new config to ports for which we’re “best”
 - ◆ Don’t bother sending config to LANs with better options
 - ◆ Add 1 to distance, send new configs on ports that haven’t told us about a shorter path to the root
- Only forward packets on ports for which we’re on the shortest path to root (prunes edges to form tree)

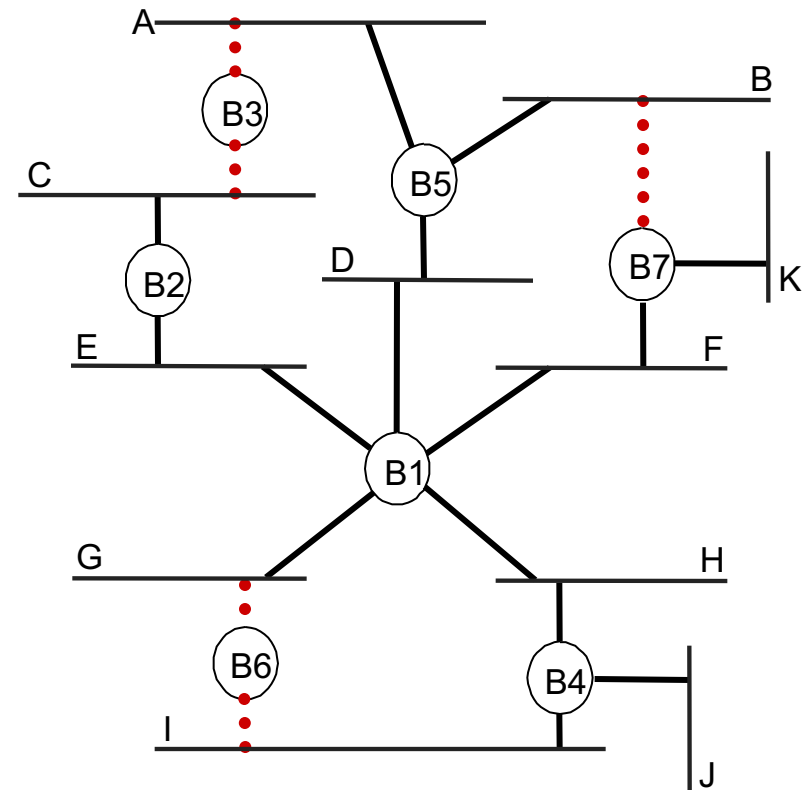
Spanning Tree Example



- Sample messages to and from B3:

1. B3 sends (B3, 0, B3) to B2 and B5
2. B3 receives (B2, 0, B2) and (B5, 0, B5) and accepts B2 as root
3. B3 sends (B2, 1, B3) to B5
4. B3 receives (B1, 1, B2) and (B1, 1, B5) and accepts B1 as root
5. B3 wants to send (B1, 2, B3) but doesn't as its nowhere "best"
6. B3 receives (B1, 1, B2) and (B1, 1, B5) again and again...

Data forwarding is turned off for LAN A





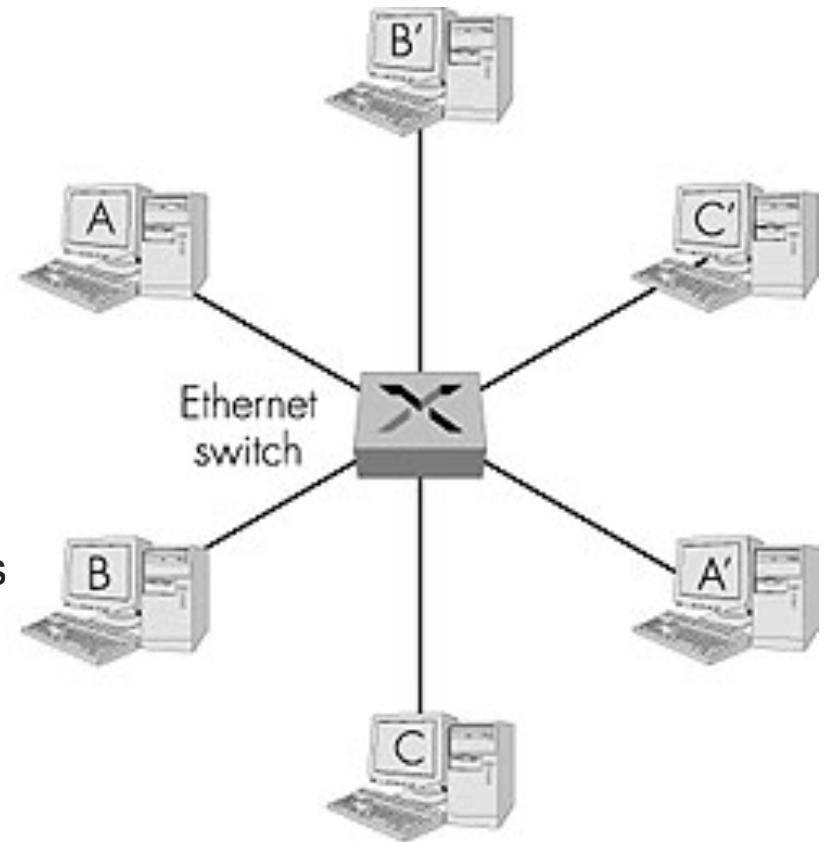
Important Details

- What if root bridge fails?
 - ◆ Age configuration info
 - » If not refreshed for MaxAge seconds then delete root and recalculate spanning tree
 - » If config message is received with a more recent age, then recalculate spanning tree
 - ◆ Applies to all bridges (not just root)
- Temporary loops
 - ◆ When topology changes, takes a bit for new configuration messages to spread through the system
 - ◆ Don't start forwarding packets immediately -> wait some time for convergence

Switched Ethernet



- Hosts directly connected to a bridge
 - ◆ learning + spanning tree protocol
- Switch supports parallel forwarding
 - ◆ A-to-B and A'-to-B' simultaneously
 - ◆ Generally full duplex as well
- **Switch** backplane capacity varies
 - ◆ Ideally, **nonblocking**
 - ◆ I.e., can run at full line rate on all ports
- No longer any shared bus
 - ◆ Each link is its own collision domain
 - ◆ Collision detection largely irrelevant

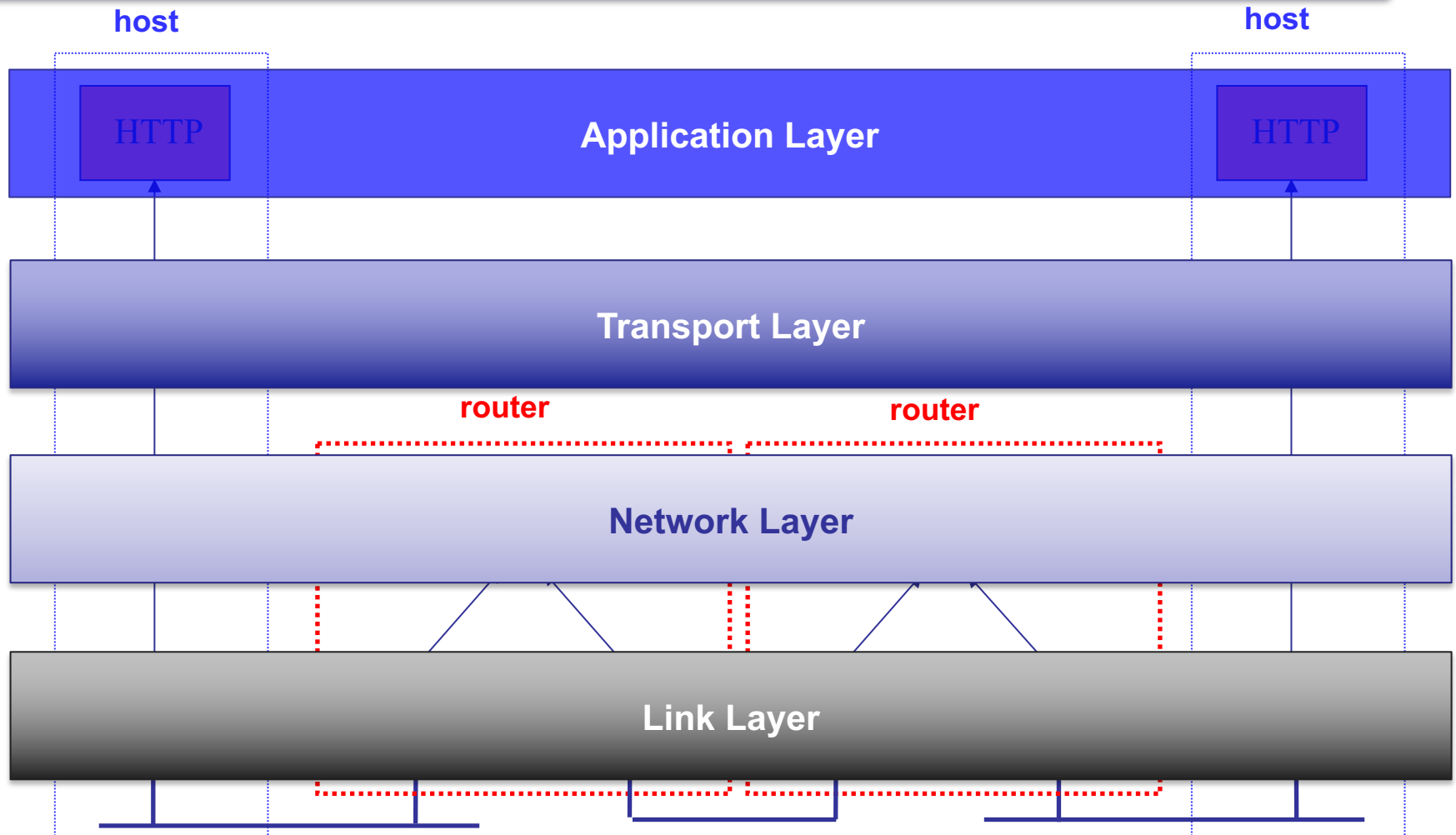




Layer-2 Forwarding

- Create spanning tree across LANs
 - ◆ Learn which ports to use to reach which addresses
- Benefits
 - ◆ Higher link bandwidth (point-to-point links)
 - ◆ Higher aggregate throughput (parallel communication)
 - ◆ Improved fault tolerance (redundant paths)
- Limitations
 - ◆ Requires homogeneous link layer (e.g. all Ethernet)
 - ◆ Harder to control forwarding topology
- What if we want to connect different link layers?

TCP/IP Protocol Stack



For Next Time



- Read 3.2.5-6