

# CSE 461: Introduction to Computer Communications Networks

## Autumn 2006

### Module 4

### Bridging LANs

John Zahorjan  
zahorjan@cs.washington.edu  
534 Allen Center

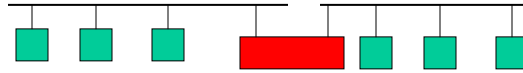
## This Module: Bridging / Switching

- Focus:
  - What to do when one shared LAN isn't big enough?
- Interconnecting LANs
  - Bridges and LAN switches
  - A preview of the Network layer

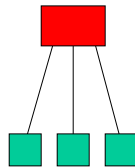
Application
Presentation
Session
Transport
Network
Data Link
Physical

## Terminology / Pictures are a little confusing

Original Ethernet (repeater)



Modern Ethernet  
(Hub)



Not talking more  
about these today

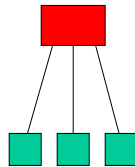
10/9/2006

CSE461 06au

## Instead, we'll be talking about these



Bridges  
Switches



10/9/2006

CSE461 06au

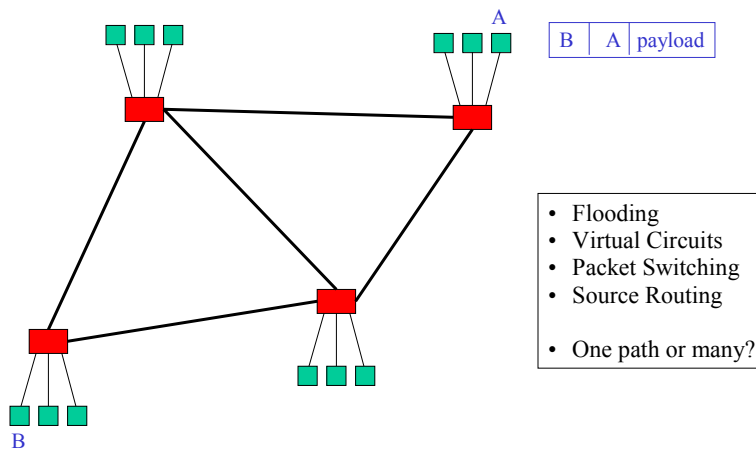
## The Common Theme: Limits of a LAN

- 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
  - Can't pass all packets by every host
    - Bridges/switches must make sensible choices about which outgoing links to place packets on
- For the system architectures we're most interested in, some packet buffering will take place
  - Store and forward

10/9/2006

CSE461 06au

## Some Choices

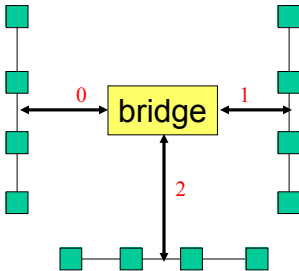


10/9/2006

CSE461 06au

## First Realization: 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

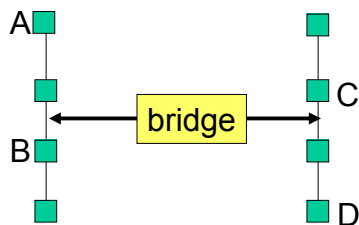


10/9/2006

CSE461 06au

## Learning Bridges

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

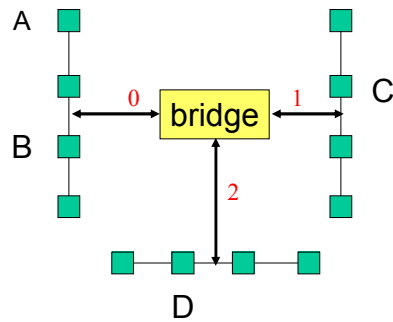


- How does the bridge know?
  - Learn who is where by observing source addresses and prune
  - Send
  - Forward using destination address; age for robustness

10/9/2006

CSE461 06au

## An Example



A → B  
 B → A  
 C → D  
 A → D  
 D → C

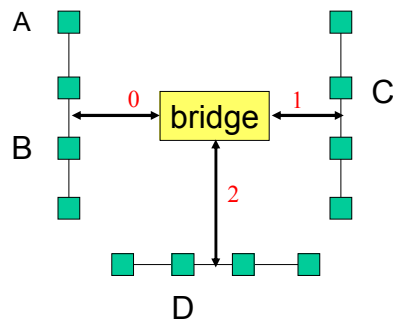
Forwarding Table

Dest. Address	Interface

10/9/2006

CSE461 06au

## After the Four Packets Have Been Sent



A → B : 1,2  
 B → A :  
 C → D : 0,2  
 A → D : 1,2  
 D → C : 1

Forwarding Table

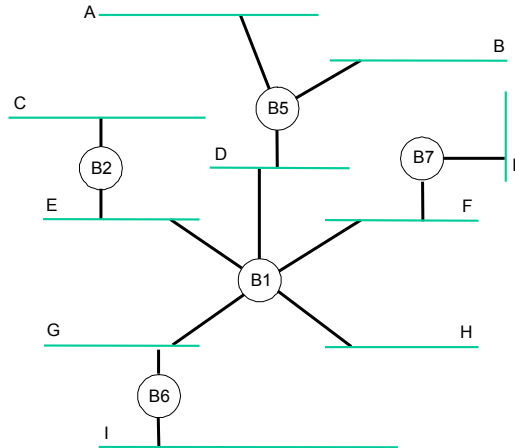
Dest. Address	Interface
A	0
B	0
C	0
D	2

10/9/2006

CSE461 06au

## Why stop at one bridge?

Why not just keep doing this forever?

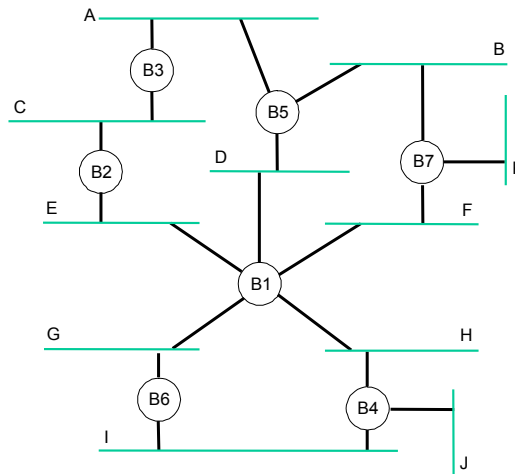


10/9/2006

CSE461 06au

## What's wrong with this picture?

- Redundancy added for fault tolerance
- Redundancy added by mistake
- Either way, what goes wrong?

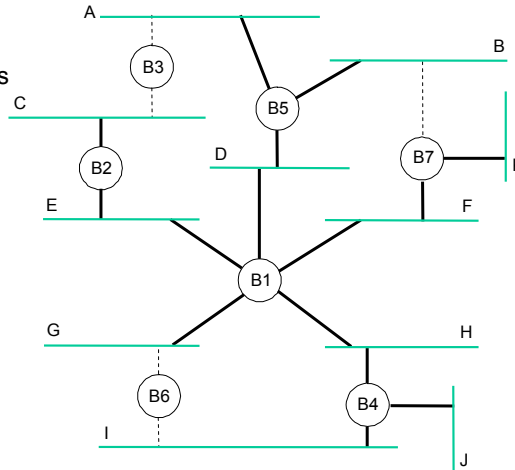


10/9/2006

CSE461 06au

## Spanning Tree Example

- Spanning tree uses select bridge ports so there are no cycles
  - Prune some ports
  - Only one tree
- Q: How do we find a spanning tree?
  - Automatically with a distributed algorithm

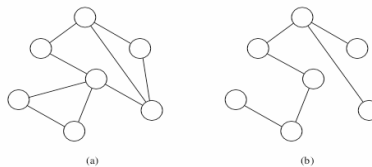


10/9/2006

CSE461 06au

## Spanning Tree

- Compute ST with a bridge as *root* such that
  - Root forwards onto all of its outgoing ports
  - Other bridges forward TO the root if a packet is coming from a bridge further from the root, else they forward away from the root
    - Packet traversal: forwards (UP)\* then (DOWN\*)



10/9/2006

CSE461 06au

## Spanning tree vs. learning

- Once the spanning tree is in place...
  - the bridge uses the regular learning algorithm to figure out which ports to forward / flood packet on
- Job of spanning tree algorithm is to disable some ports to eliminate cycles

10/9/2006

CSE461 06au

## 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
    - <http://www1.cs.columbia.edu/~ji/F02/ir02/p44-perlman.pdf>
- Outline: Goal is to turn some bridge ports off
  1. Elect a root node of the tree (lowest address)
  2. Grow tree as shortest distances from the root (using lowest address to break distance ties)
    - All done by bridges sending periodic configuration messages over ports for which they are the "best" path
    - Then turn off ports that aren't on "best" paths

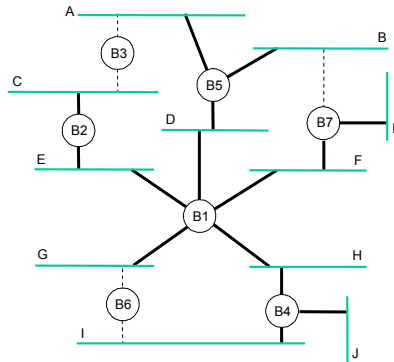
10/9/2006

CSE461 06au



## Algorithm Overview

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



10/9/2006

CSE461 06au

## Algorithm continued

- Bridges exchange configuration messages, containing:
  - id for bridge sending the message
  - id for what the sending bridge believes to be the 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 messages
  - instead, forward root's configuration message
    - incrementing distance field by 1
  - in steady state, only root generates configuration messages

10/9/2006

CSE461 06au

## Algorithm More...

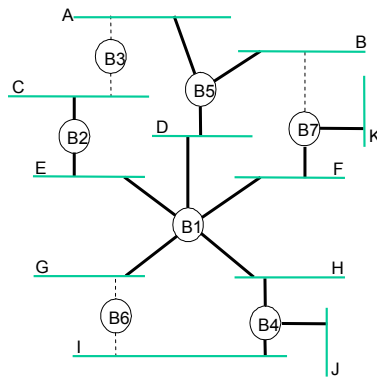
- When learn not designated bridge on LAN, stop forwarding configuration messages
  - in steady state, only designated bridges forward configuration messages
- Root bridge continues to send configuration messages periodically
- If a bridge does not receive config. message after a period of time:
  - assumes topology has changed
  - starts generating configuration messages claiming to be root

10/9/2006

CSE461 06au

## Algorithm Example

- Message format:
  - (root, dist-to-root, sending bridge)
- Sample messages sequences 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 could send (B1, 2, B3) but doesn't as its nowhere "best"
    - B2 and B5 are better choices.
    - so B3 is NOT a designated bridge
  6. B3 receives (B1, 1, B2) and (B1, 1, B5) again ... stable
    - B3 turns off data forwarding to LANs A and C



10/9/2006

CSE461 06au

## Some other tricky details

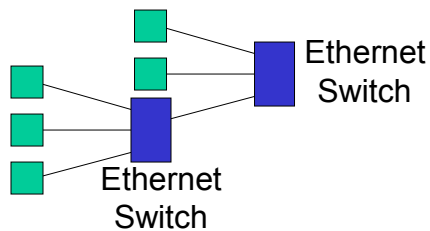
- Configuration information is aged
  - If the root fails a new one will be elected
- Reconfiguration is damped
  - Adopt new spanning trees slowly to avoid temporary loops

10/9/2006

CSE461 06au

## LAN Switches

- LAN switches are multi-port bridges
  - Modern, high performance form of bridged LANs
  - Looks like a hub, but frames are switched, not shared
  - Every host on a separate port, or can combine switches



10/9/2006

CSE461 06au

## Limitations of Bridges/Switches

- LAN switches form an effective small-scale network
  - Plug and play for real!
- Why can't we build a large network using bridges?
  - Little control over forwarding paths
  - Size of bridge forwarding tables grows with number of hosts
  - Broadcast traffic flows freely over whole extended LAN
  - Spanning tree algorithm limits reconfiguration speed
  - Poor solution for connecting LANs of different kinds

10/9/2006

CSE461 06au

## 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

10/9/2006

CSE461 06au