CSE160: Computer Networks

Lecture #07 – Bridging LANs

2020-09-17

Professor

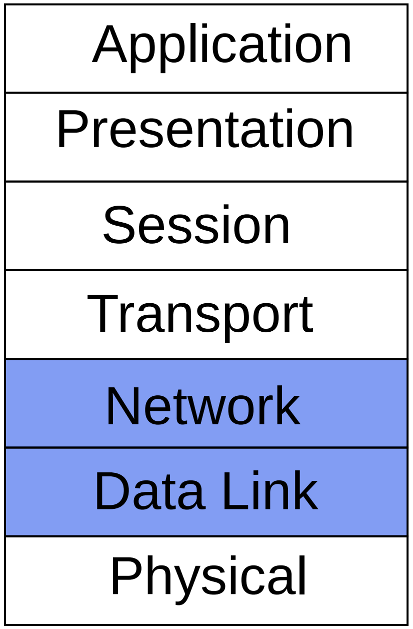
Alberto E. Cerpa

Last Two Times …

- Medium Access Control (MAC) protocols
  - Part of the Link Layer
  - At the heart of Local Area Networks (LANs)
- How do multiple parties share a wire or the air?
  - Random access protocols (CSMA/CD)
  - Contention-free protocols (turn-taking, reservations)
  - Wireless protocols (CSMA/CA and RTS/CTS)
- Scalability problems
  - a << 1 (bandwidth*delay / frame size due to CSMA)
  - THT bounds maximum capacity
  - No more than 256 nodes in practice

This Time -- Switching (a.k.a. Bridging)

- 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

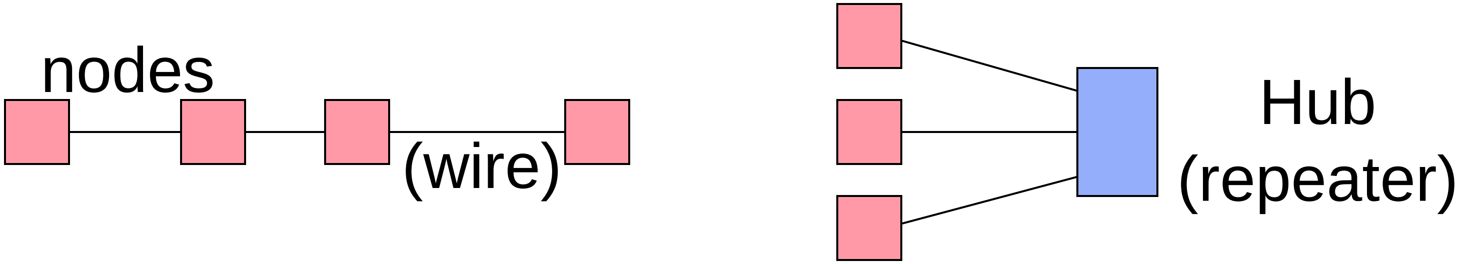
Network

Data Link

Physical

Limits of a LAN

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



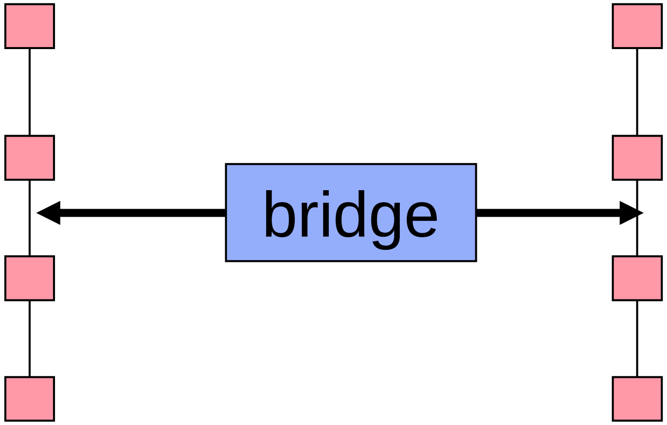
- How do we scale to a larger, faster network?
  - We must be able to interconnect LANs

Switching (a.k.a. Bridging)

- Transferring a packet from one LAN to another LAN
  - Build an “extended LAN”
- Different varieties of switching
  - Packet switched vs. circuit switched
  - Connection vs. Connectionless
- We’ll focus on connectionless, packet switched
  - Ethernet

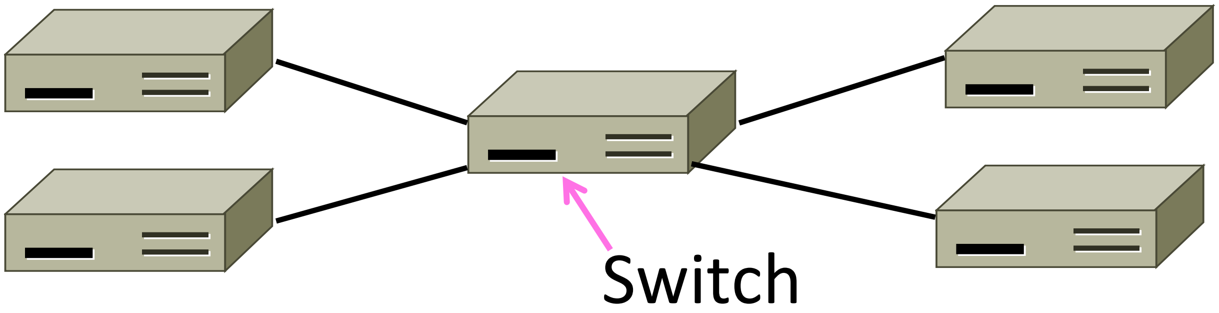
Bridges and Extended LANs

- "Transparently" interconnect LANs with bridge
  - Receive frames from each LAN and forward to the other
  - It performs Medium Access Control to access each LAN
  - Each LAN is its own collision domain; bridge isn't a repeater
  - Could have many ports



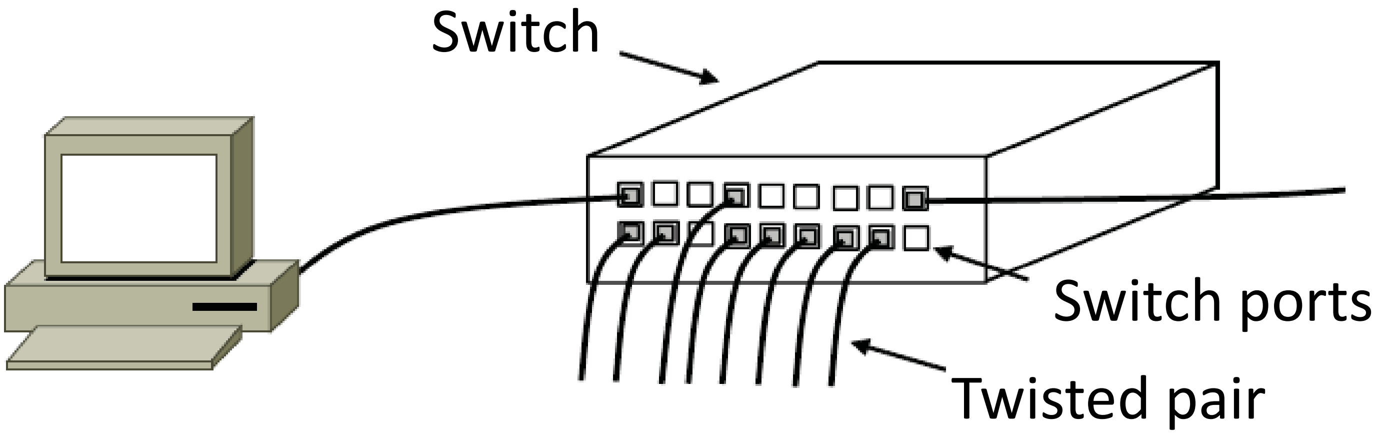
LAN Switches

- How do we connect nodes with a switch instead of multiple access
  - Uses multiple links/wires
  - Basis of modern (switched) Ethernet



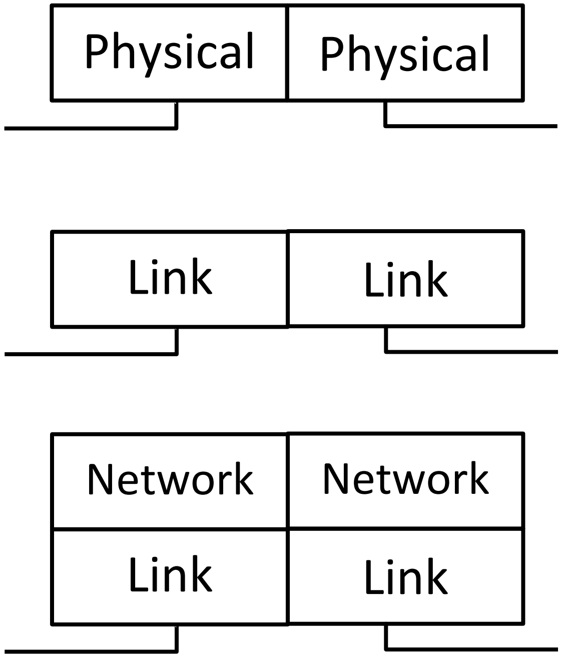
Switched Ethernet

- Hosts are wired to Ethernet switches with twisted pair
  - Switch serves to connect the hosts
  - Wires usually run to a closet



What’s in the box?

- Remember from protocol layers:



Hub, or repeater

Physical | Physical

Switch

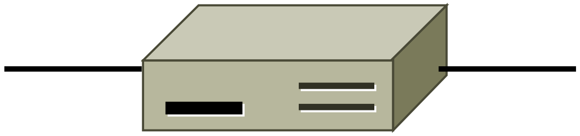
Link | Link

Router

Network | Network

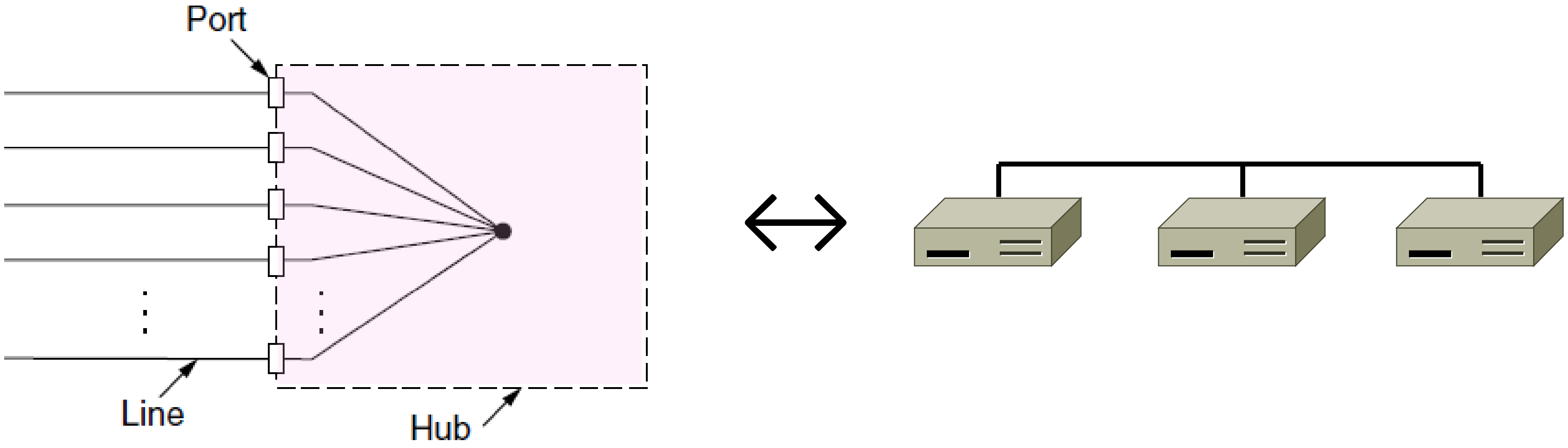
Link | Link

All look like this:



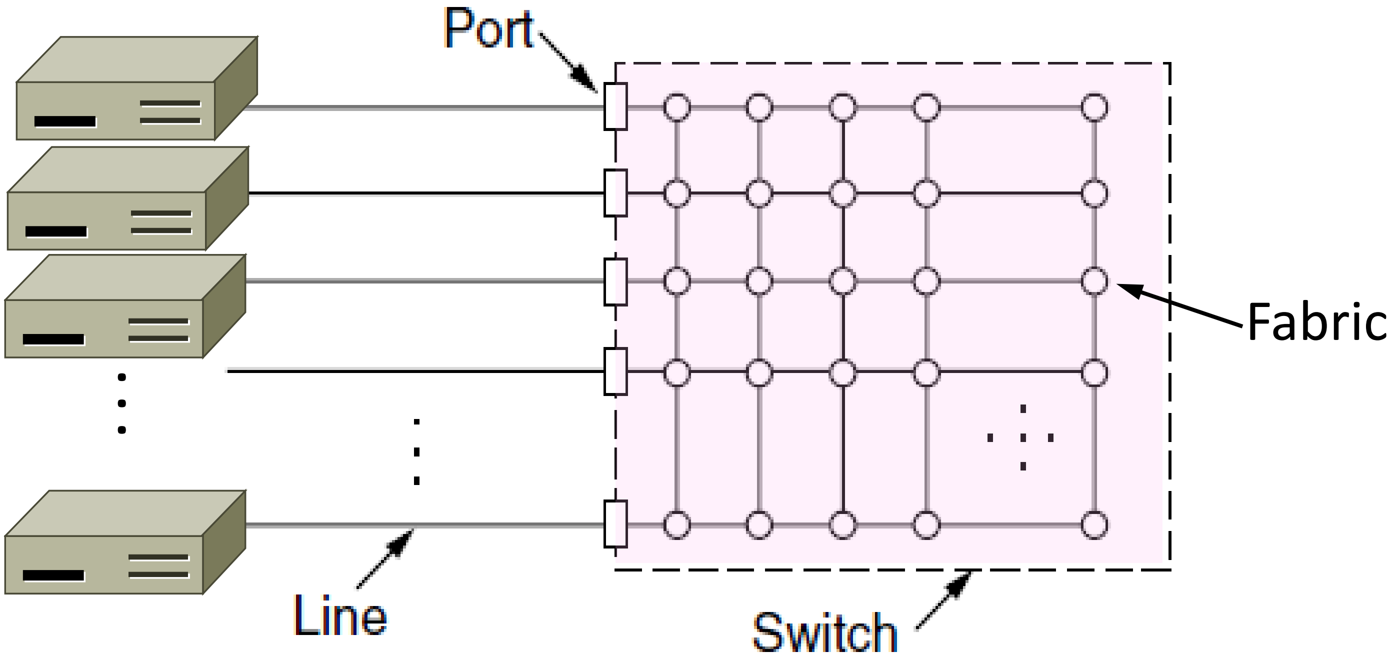
Inside a Hub

- All ports are wired together; more convenient and reliable than a single shared wire



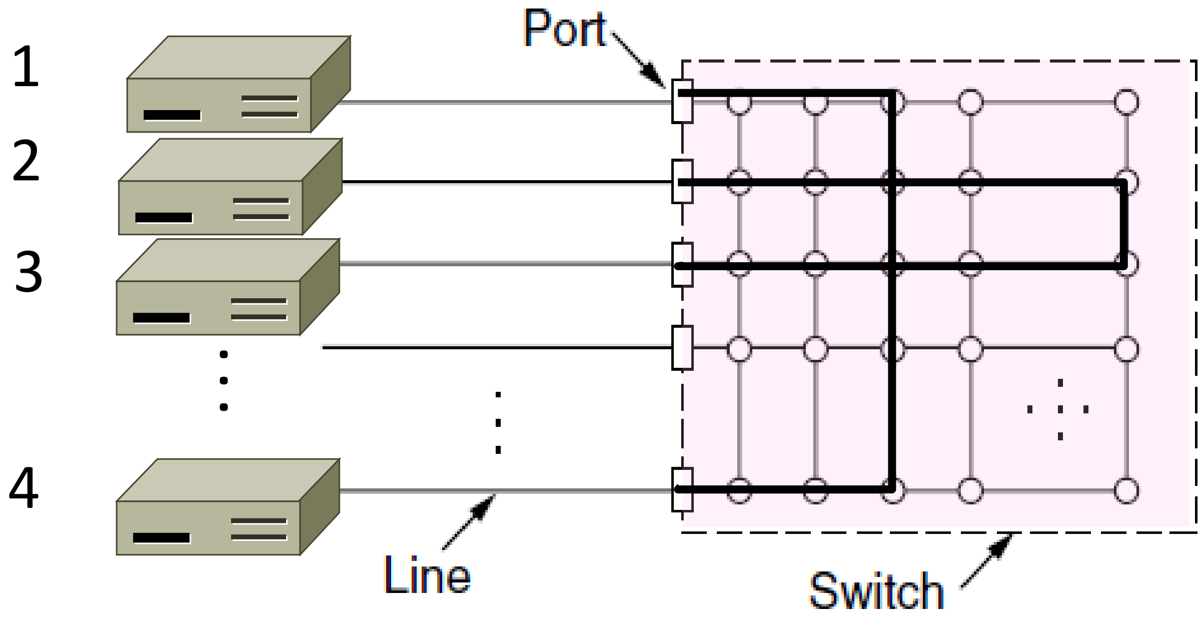
Inside a Switch

- Uses frame addresses to connect input port to the right output port; multiple frames may be switched in parallel



Inside a Switch (2)

- Port may be used for both input and output (full-duplex)
  - Just send, no multiple access protocol



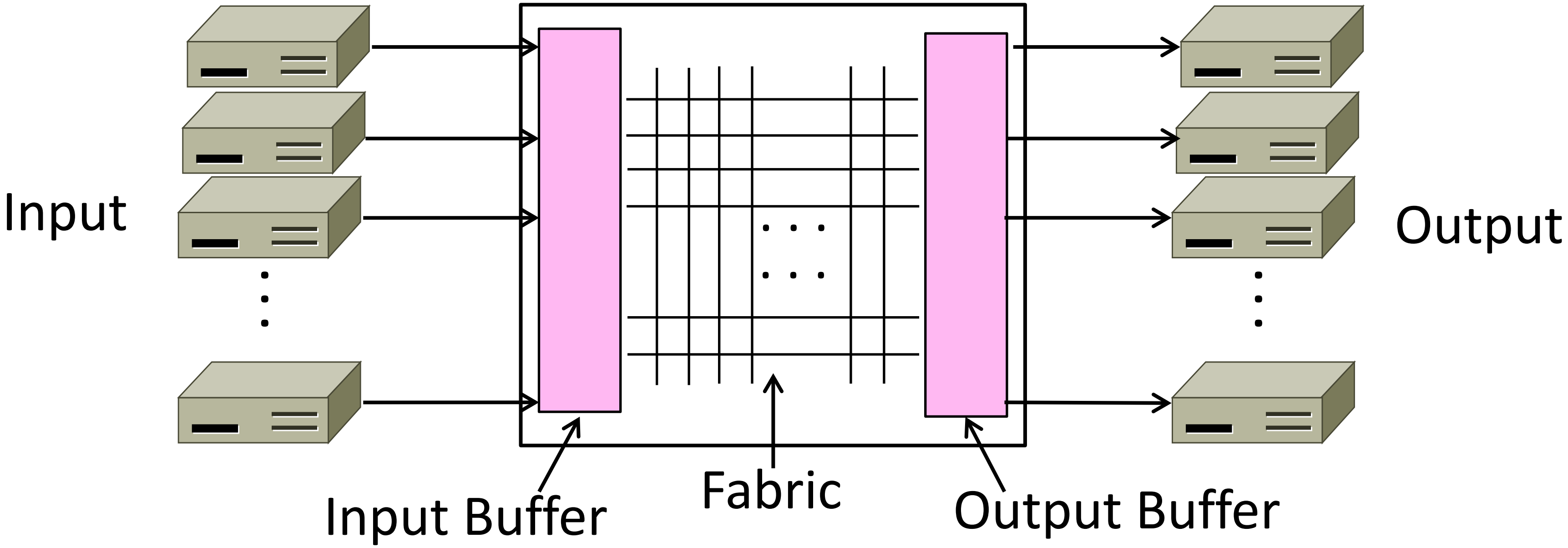
1 → 4

and

2 → 3

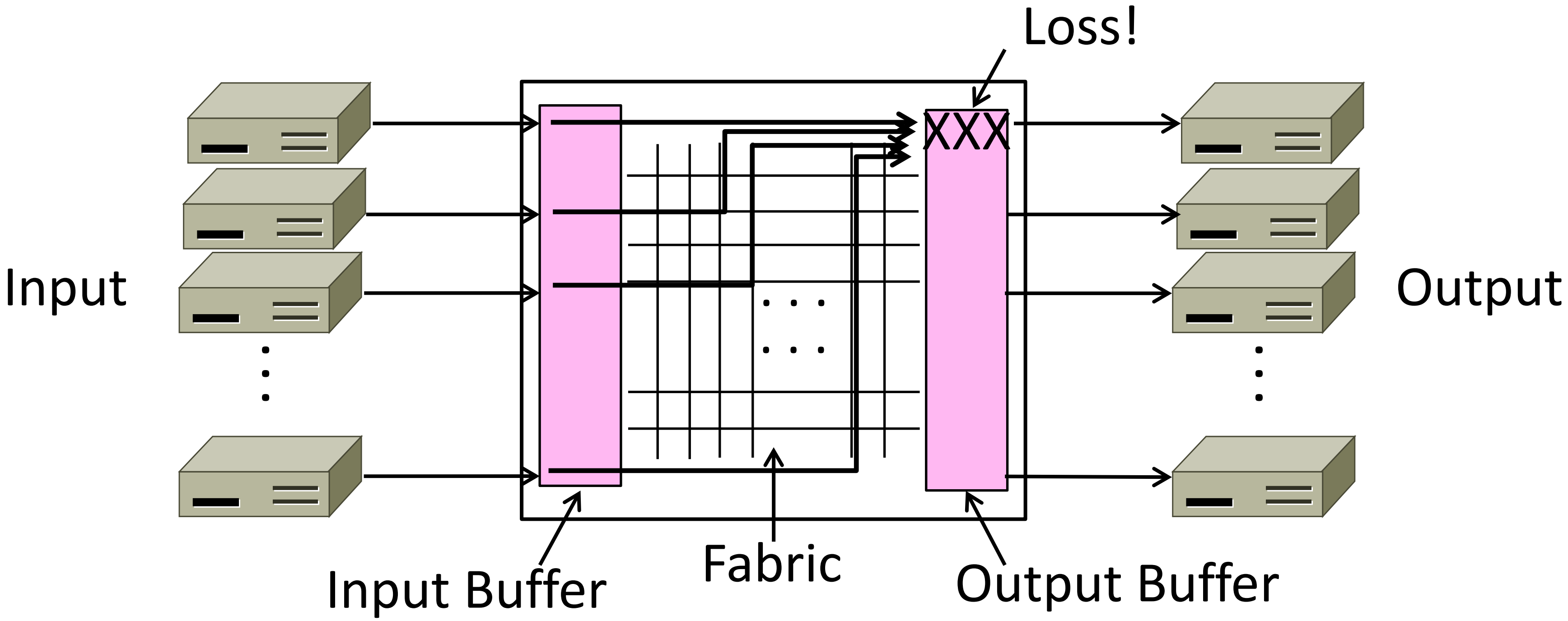
Inside a Switch (3)

- Need buffers for multiple inputs to send to one output



Inside a Switch (4)

- Sustained overload will fill buffer and lead to frame loss



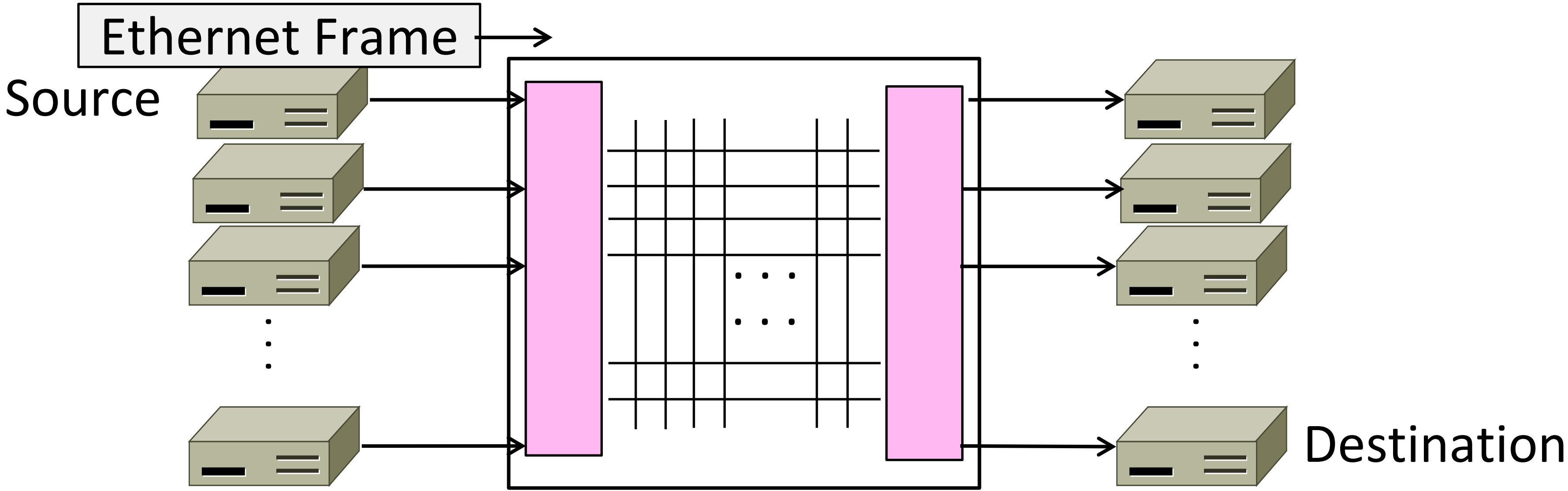
Advantages of Switches

- Switches and hubs have replaced the shared cable of classic Ethernet
  - Convenient to run wires to one location
  - More reliable; wire cut is not a single point of failure that is hard to find

- Switches offer scalable performance
  - E.g., 1 Gbps per port instead of 1 Gbps for all nodes of shared cable/hub

Switch Forwarding

- Switch needs to find the right output port for the destination address in the Ethernet frame. How?
  - Want to let hosts be moved around readily; don't look at IP



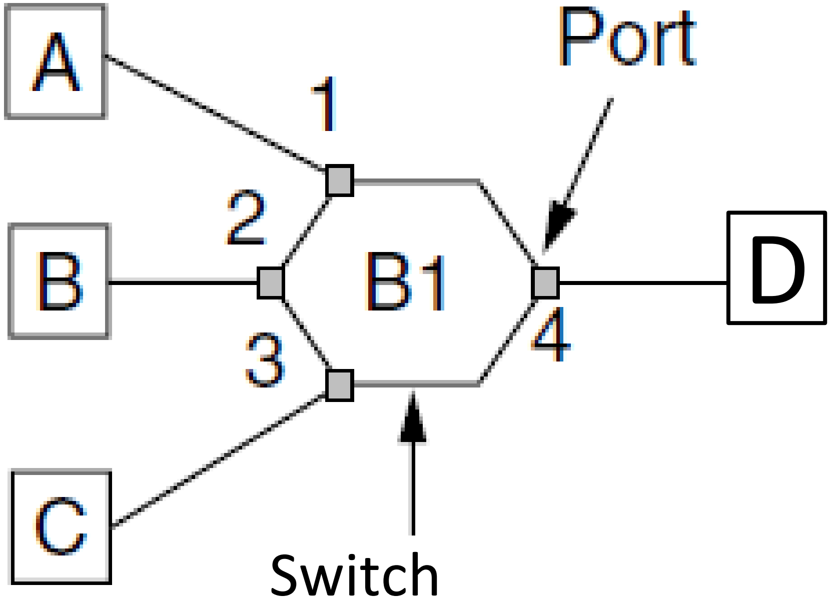
Backward Learning

- Switch forwards frames with a port/address table as follows:

  1. To fill the table, it looks at the source address of input frames
  2. To forward, it sends to the port, or else broadcasts to all ports
  3. Information is aged for robustness

Backward Learning (2)

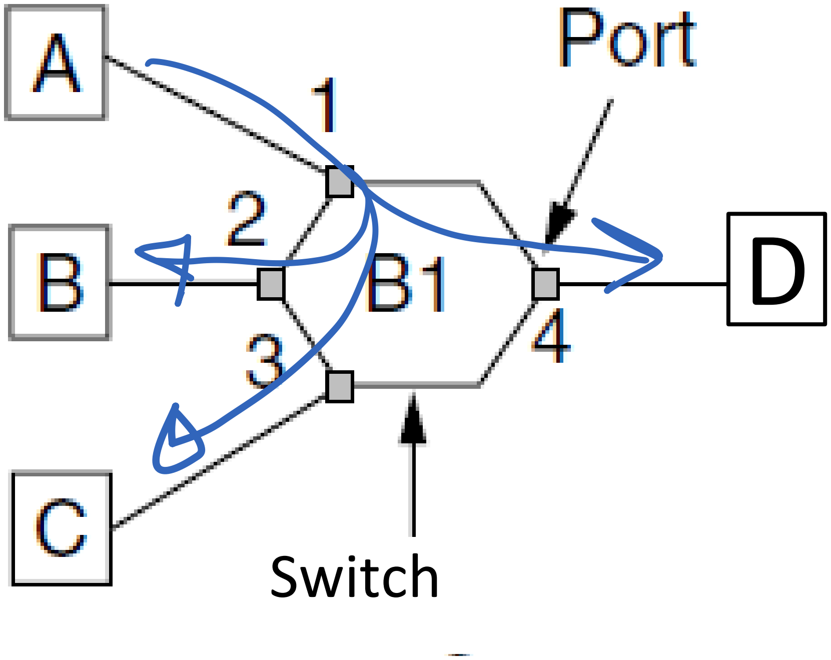
- 1: switch knows nothing, empty table



Address	Port
A	
B	
C	
D	

Backward Learning (3)

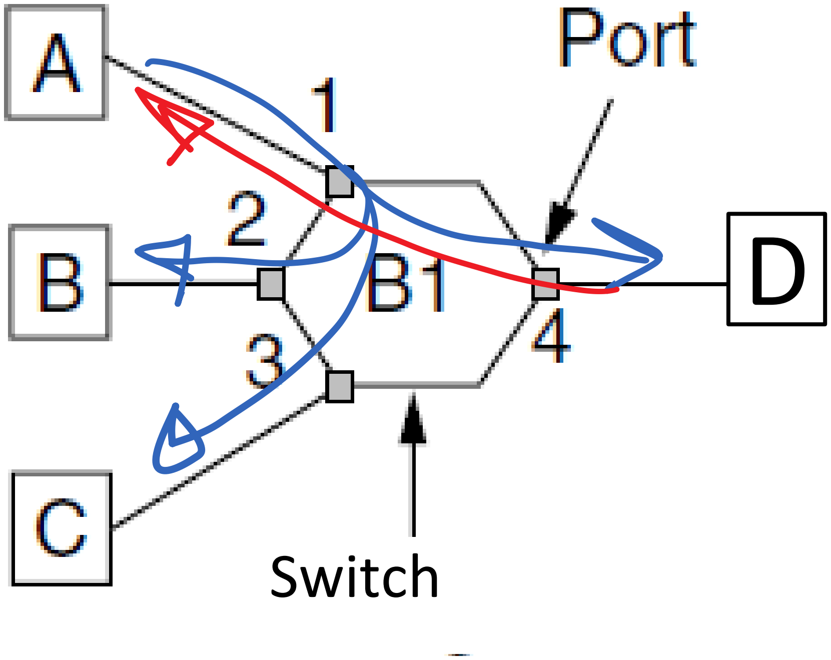
- 2: A sends to D
  - With an empty table, B1 forwards A’s frame to all other ports
  - What does B1 learn?



Address	Port
A	1
B	
C	
D	

Backward Learning (4)

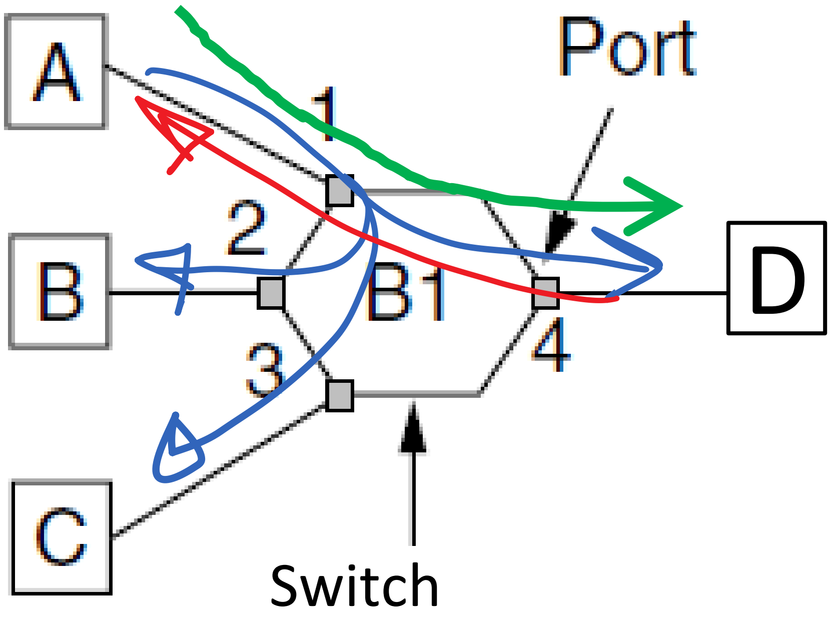
- 3: D sends to A



Address	Port
A	1
B	
C	
D	4

Backward Learning (5)

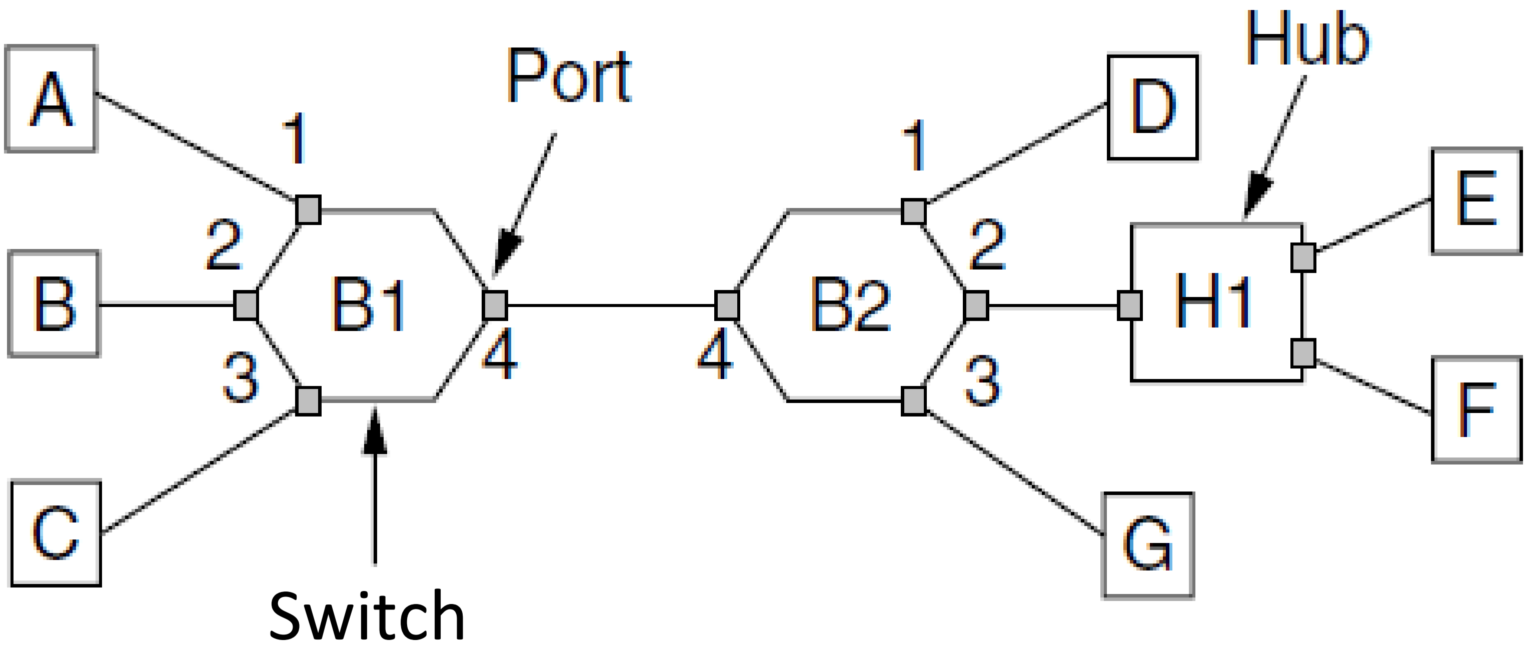
- 4: A sends to D



Address	Port
A	1
B	
C	
D	4

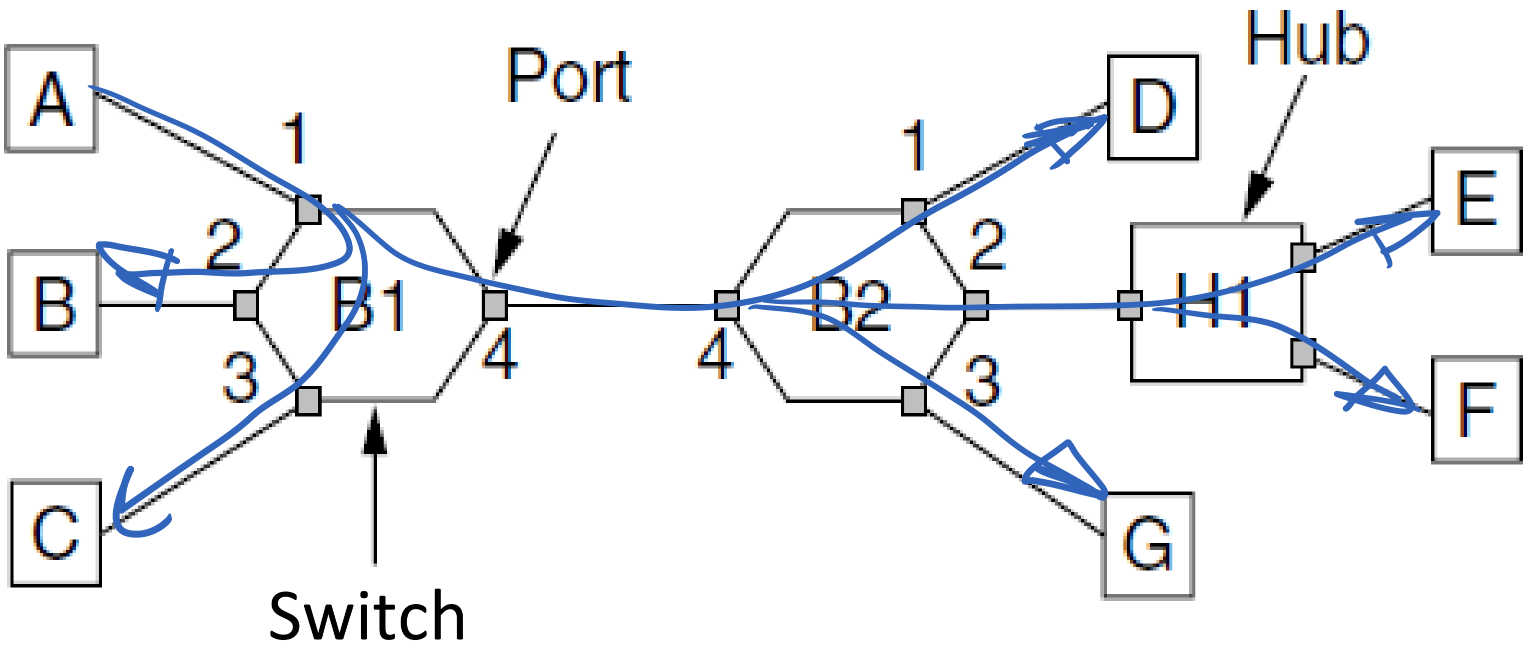
Learning with Multiple Switches

- Just works with multiple switches and a mix of hubs *assuming no loops*, e.g., A send to D then D sends to A



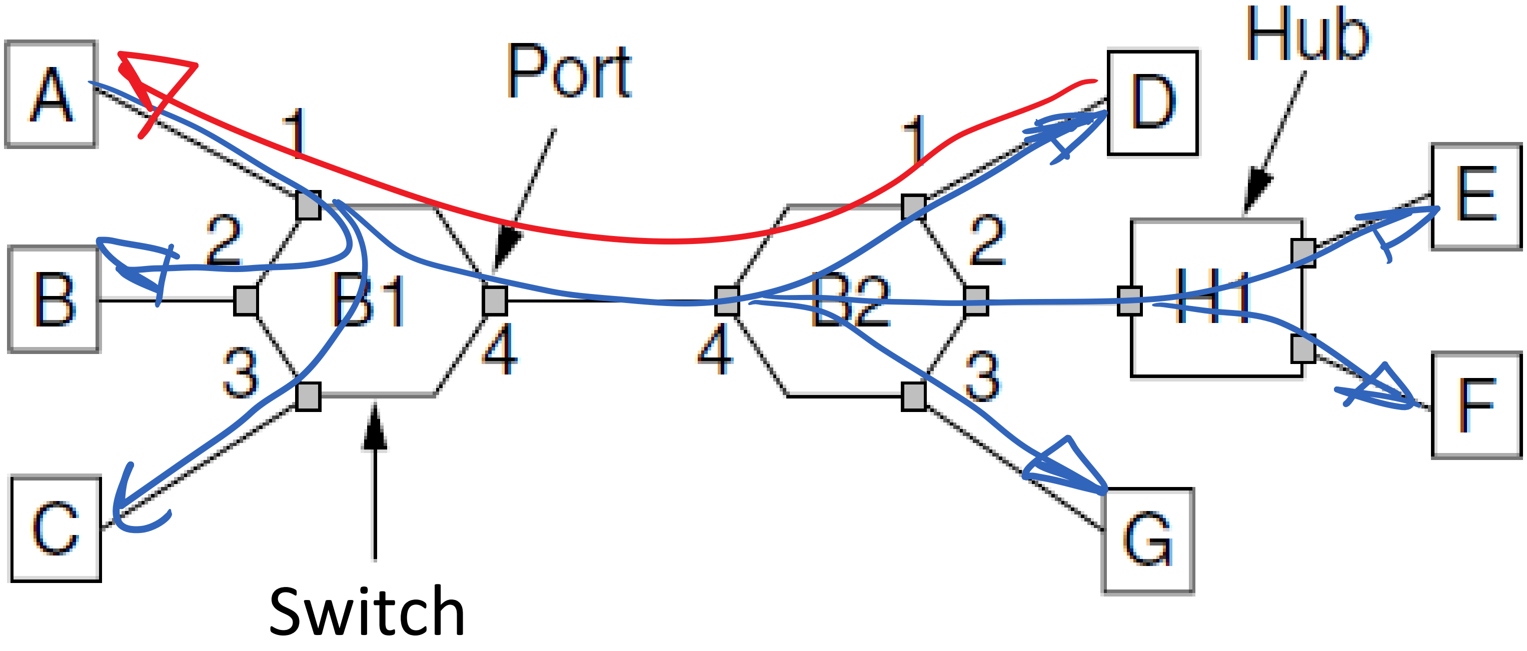
Learning with Multiple Switches (2)

- Just works with multiple switches and a mix of hubs *assuming no loops*, e.g., A send to D then D sends to A



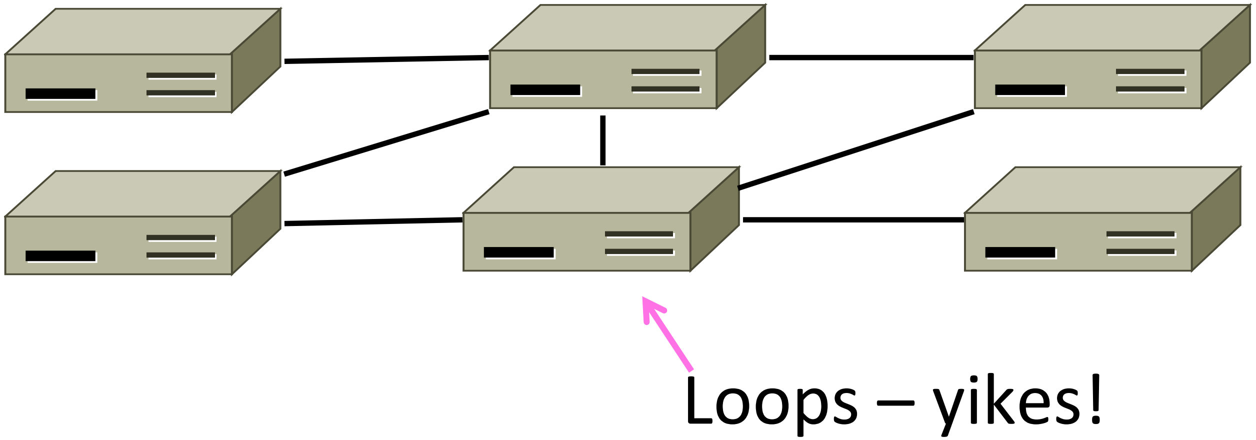
Learning with Multiple Switches (3)

- Just works with multiple switches and a mix of hubs *assuming no loops*, e.g., A send to D then D sends to A



Switched Spanning Tree

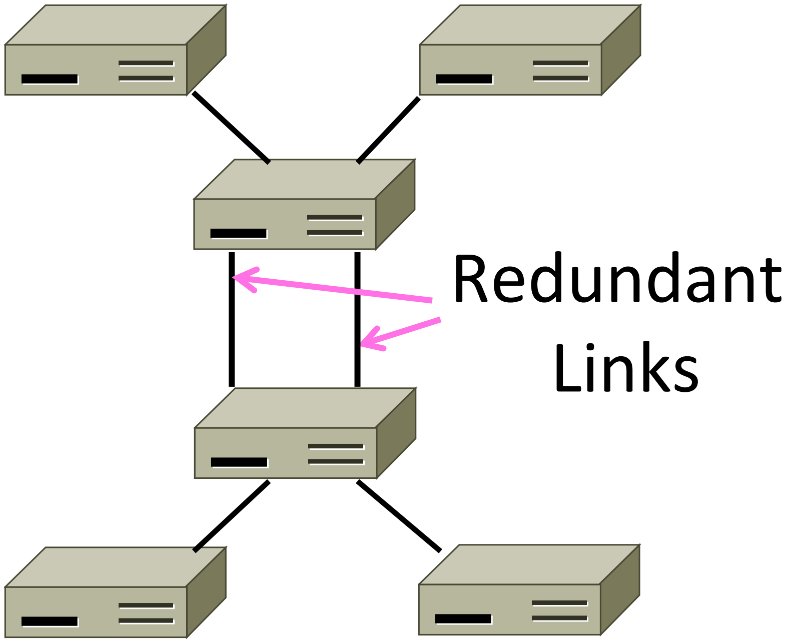
- How can we connect switches in any topology so they just work?
  - Remember learning works only if there are no loops!



Problem – Forwarding Loops

- May have a loop in the topology
  - Redundancy in case of failures
  - Or a simple mistake

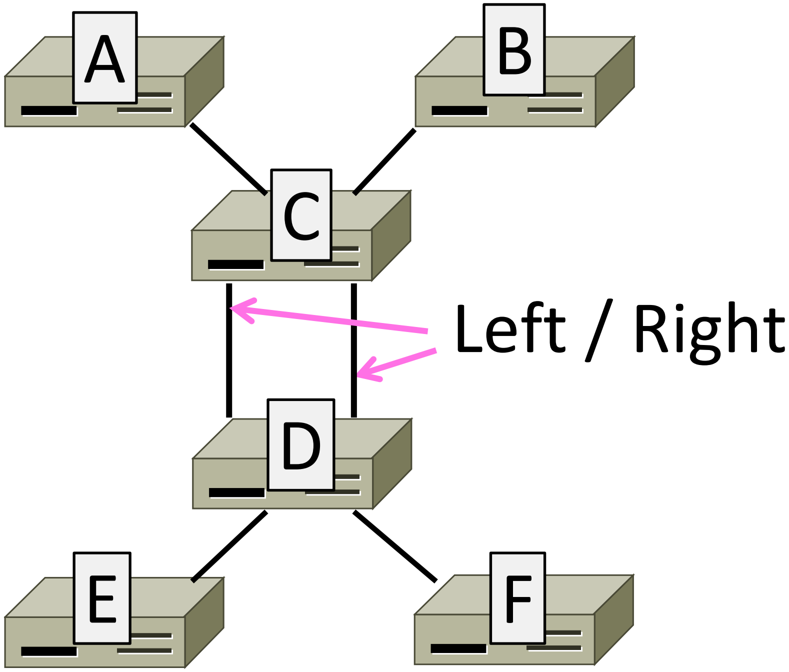
- Want LAN switches to “just work”
  - Plug-and-play, no changes to hosts
  - But loops cause a problem…



Redundant Links

Forwarding Loops (2)

- Suppose the network is started and A sends to F. What happens?
  - A → C → B, D-left, D-right
  - D-left → C-right, E, F
  - D-right → C-left, E, F
  - C-right → D-left, A, B
  - C-left → D-right, A, B
  - D-left → …
  - D-right → …



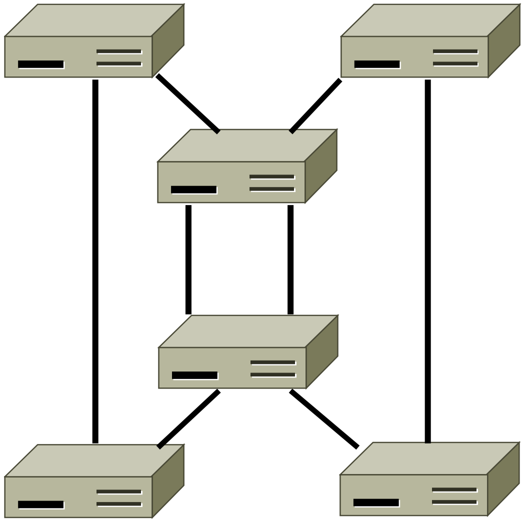
- We get infinite loops!

Spanning Tree Solution

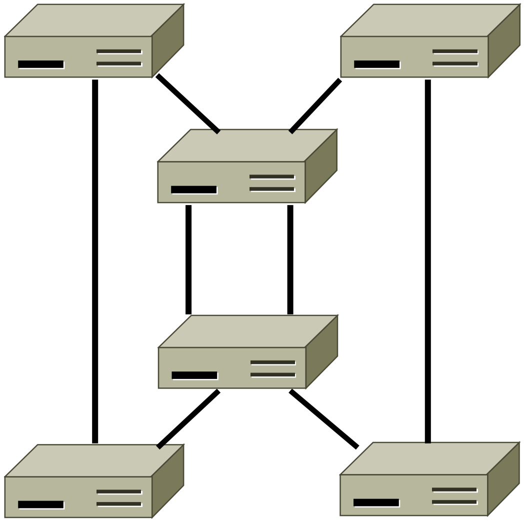
- Switches collectively find a spanning tree for the topology
  - A subset of links that is a tree (no loops) and reaches all switches
  - The switches forward as normal on the spanning tree
  - Broadcasts will go up to the root of the tree and down all the branches

Spanning Tree

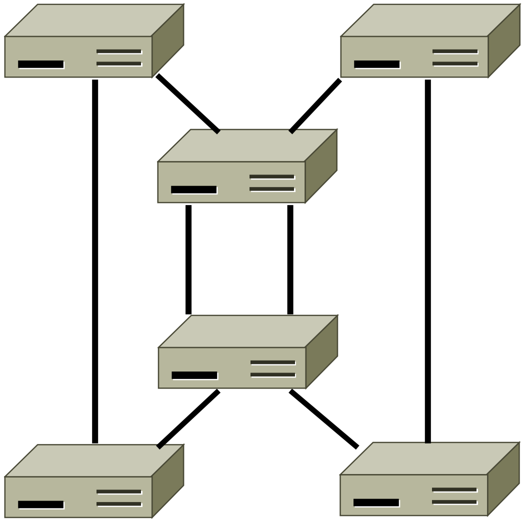
Topology



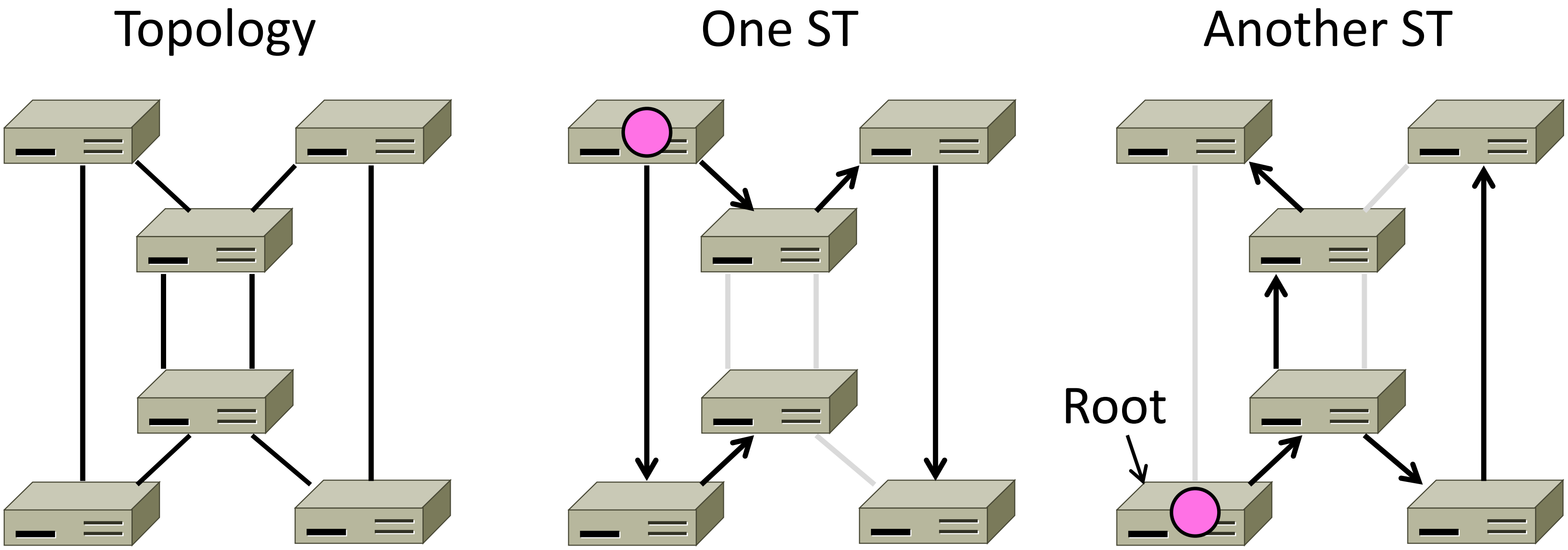
One ST



Another ST



Spanning Tree (2)



Spanning Tree Algorithm

- Rules of the distributed game:
  - All switches run the same algorithm
  - They start with no information
  - Operate in parallel and send messages
  - Always search for the best solution
- Ensures a highly robust solution
  - Any topology, with no configuration
  - Adapts to link/switch failures,…

Radia Perlman (1952–)

- Key early work on routing protocols
  - Routing in the ARPANET
  - Spanning Tree for switches (next)
  - Link-state routing (later)

- Now focused on network security

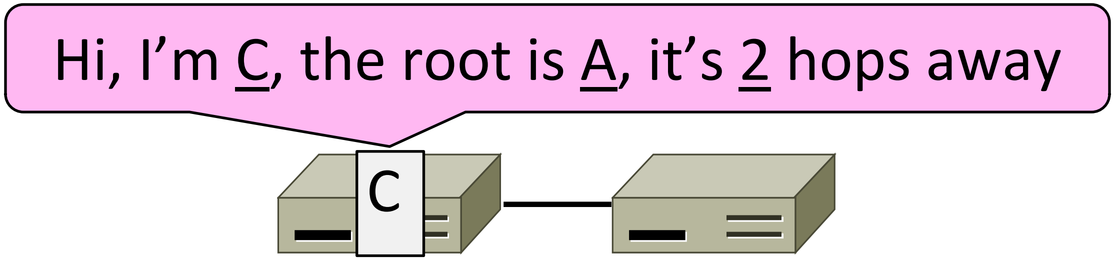
Copyright © 2018 Internet Society

Spanning Tree Algorithm (2)

- Outline:
  1. Elect a root node for the tree (switch with the lowest address)
  2. Grow tree as shortest distances from the root (using lowest address to break distances ties)
  3. Turn off ports for forwarding if they are not on the spanning tree

Spanning Tree Algorithm (3)

- Details:
  - Each switch initially believes it is the root of the tree
  - Each switch sends periodic updates to neighbors with:
    - Its address, address of the root, and distance (in hops) to root
  - Switches favor ports with shorter distances to lowest root
    - Use lowest address as a tie for distances



Hi, I'm C, the root is A, it's 2 hops away

or (C, A, 2)

Spanning Tree Algorithm (4)

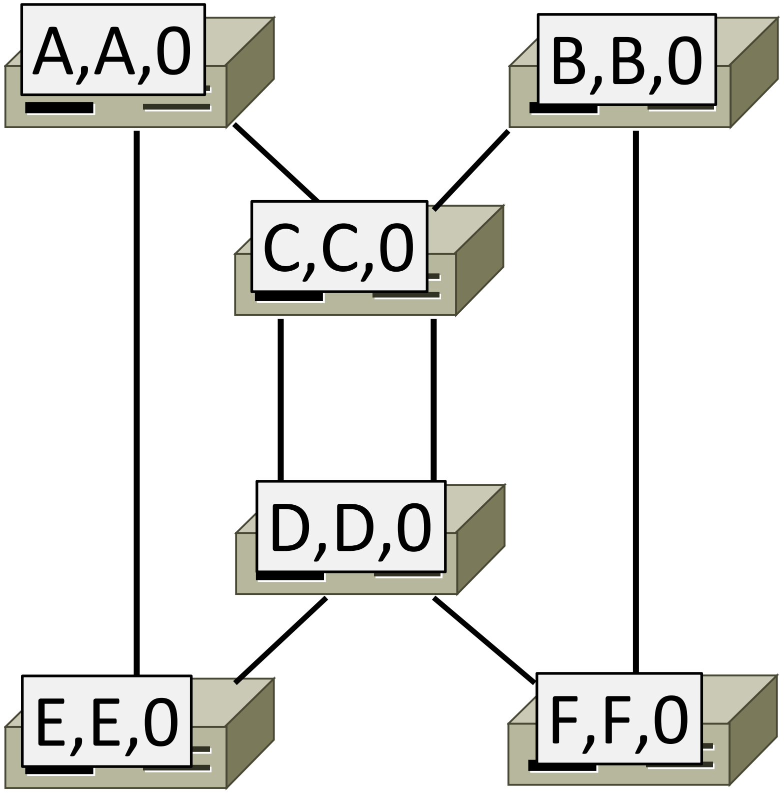
- Switches exchange configuration messages, containing:
  - id for switch sending the message
  - id for what the sending switch believes to be the root
  - distance (hops) from sending switch to root switch
- Each switch records current best configuration message for each port
- Initially, each switch 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
- How do switches exchange messages if they don't know the MAC addresses of other switches?

Spanning Tree Algorithm (5)

- When learn not designated switch/port on LAN, stop forwarding configuration messages
  - in steady state, only designated switch/port that are part of the spanning tree forward configuration messages
- Root switch continues to send configuration messages periodically
- If a switch does not receive config. message after a period of time:
  - assumes topology has changed
  - starts generating configuration messages claiming to be root

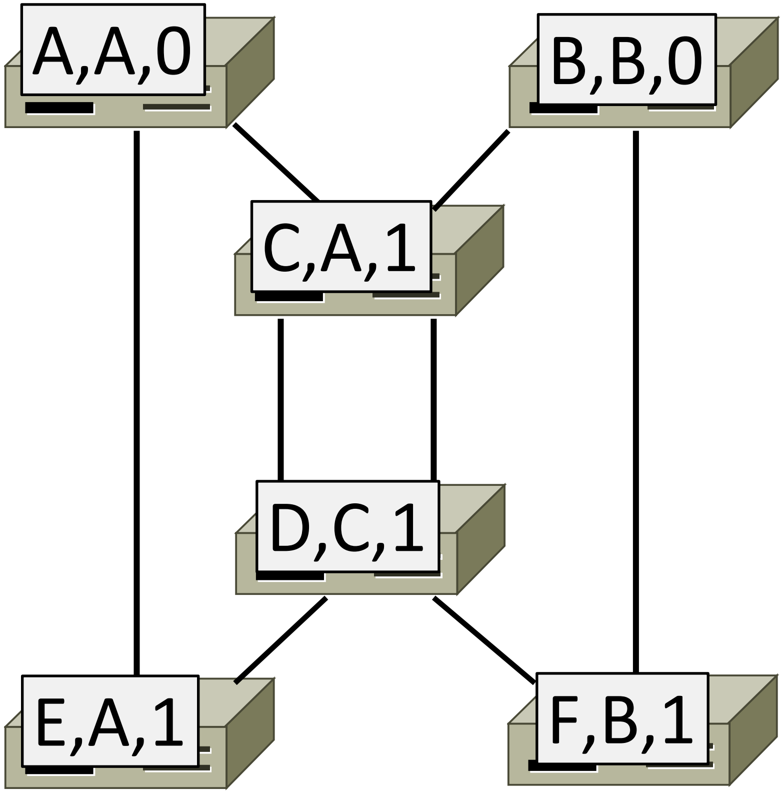
Spanning Tree Example

- 1^{st} round, sending:
  - A sends (A,A,0) to say it is root
  - B, C, D, E, and F do likewise
- 1^{st} round, receiving:
  - A still thinks it is root (A,A,0)
  - B still thinks (B,B,0)
  - C updates to (C,A,1)
  - D updates to (D,C,1)
  - E updates to (E,A,1)
  - F updates to (F,B,1)



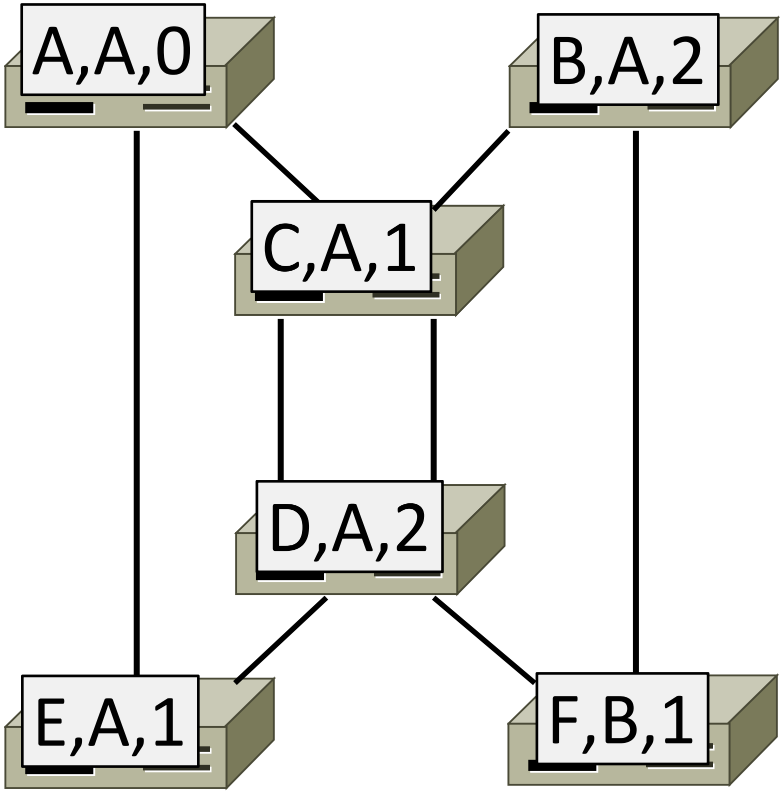
Spanning Tree Example (2)

- 2nd round, sending:
  - Nodes send their updated state
- 2nd round, receiving:
  - A remains (A,A,0)
  - B updates to (B,A,2) via C
  - C remains (C,A,1)
  - D updates to (D,A,2) via C-left
  - E remains (E,A,1)
  - F remains (F,B,1)



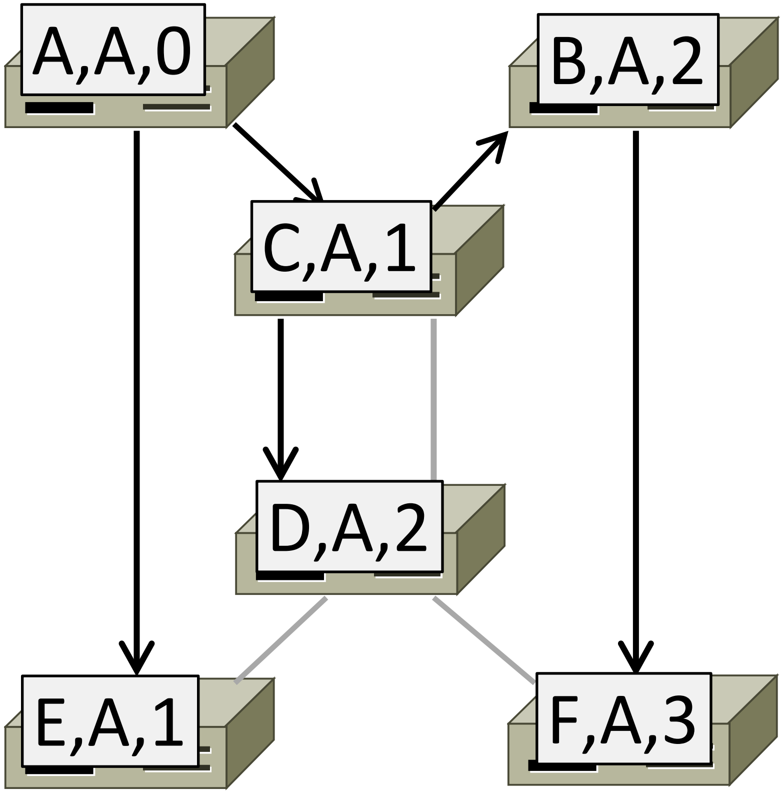
Spanning Tree Example (3)

- 3rd round, sending:
  - Nodes send their updated state
- 3rd round, receiving:
  - A remains (A,A,0)
  - B remains (B,A,2) via C
  - C remains (C,A,1)
  - D remains (D,A,2) via C-left
  - E remains (E,A,1)
  - F updates (F,A,3) via B



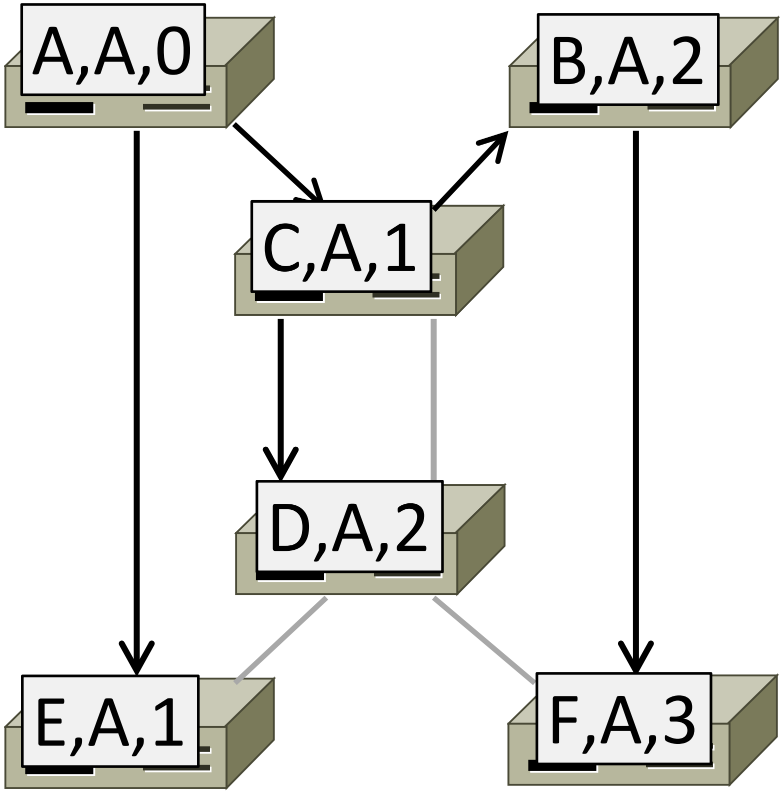
Spanning Tree Example (4)

- 4th round:
  - Steady-state has been reached
  - Nodes turn off forwarding that is not on the spanning tree
- Algorithm continues to run
  - Adapts by timing out information. Why?
  - E.g., if A fails, other nodes forget it, and B will become the new root



Spanning Tree Example (5)

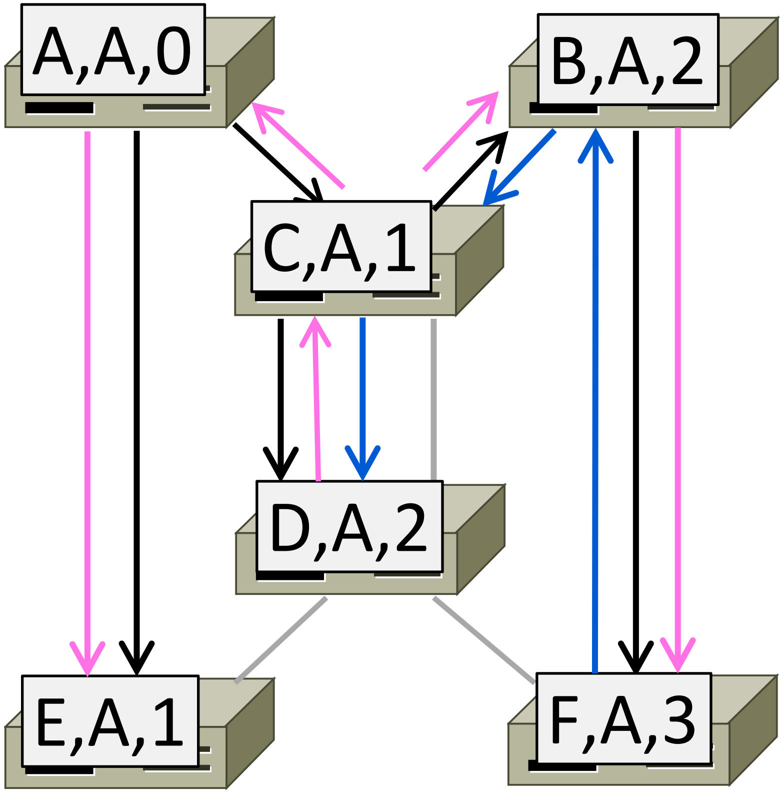
- Forwarding proceeds as usual on the ST
- Initially D sends to F:



- And F sends back to D:

Spanning Tree Example (6)

- Forwarding proceeds as usual on the ST
- Initially D sends to F:
  - D → C-left
  - C → A, B
  - A → E
  - B → F
- And F sends back to D:
  - F → B
  - B → C
  - C → D
  - (hmm, not such a great route)



Algorhyme (Radia Perlman, 1985)

I think that I shall never see
A graph more lovely than a tree.
A tree whose crucial property
Is loop-free connectivity.
A tree that must be sure to span
So packets can reach every LAN.
First, the root must be selected.
By ID, it is elected.
Least-cost paths from root are traced.
In the tree, these paths are placed.
A mesh is made by folks like me,
Then bridges find a spanning tree.

Some other tricky details

- Configuration information is aged
  - Why?
  - If the root fails a new one will be elected
- Reconfiguration is damped
  - Why?
  - Adopt new spanning trees slowly to avoid temporary loops
- More details in the Spanning Tree paper I asked you to read

Limitations of Switches

- LAN switches form an effective small-scale network
  - Plug and play for real!
- Why can’t we build a large network using switches?
  - 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

Key Concepts

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