

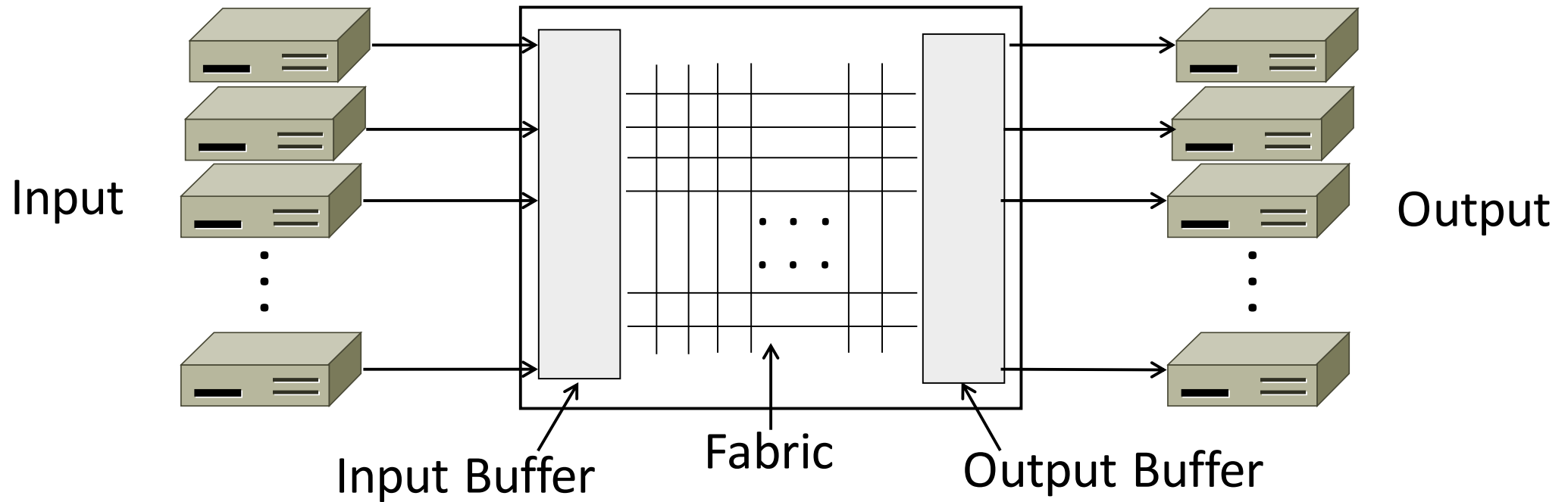
Link Layer

(continued)

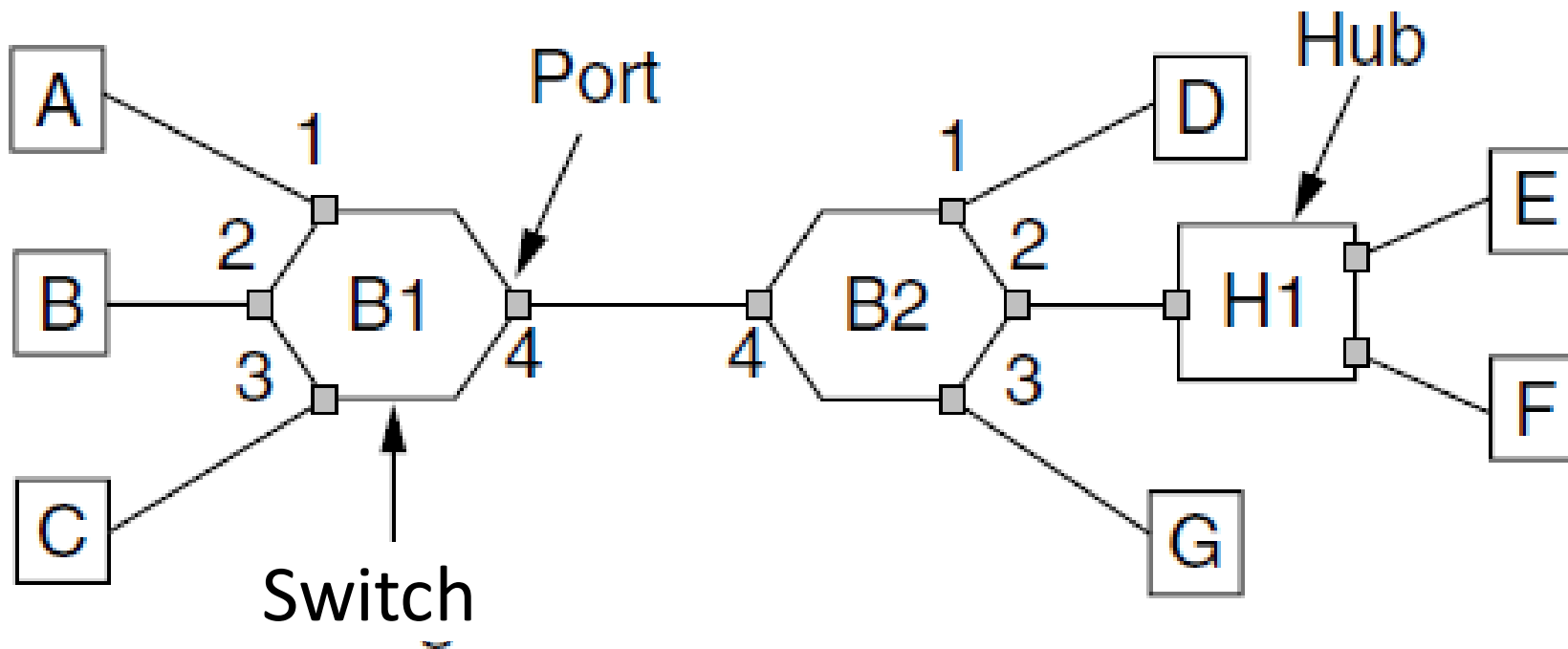
Topics

1. Framing
 - Delimiting start/end of frames
2. Error detection and correction
 - Handling errors
3. Retransmissions
 - Handling loss
4. Multiple Access
 - 802.11, classic Ethernet
5. **Switching**
 - **Modern Ethernet**

Reminder, Switch

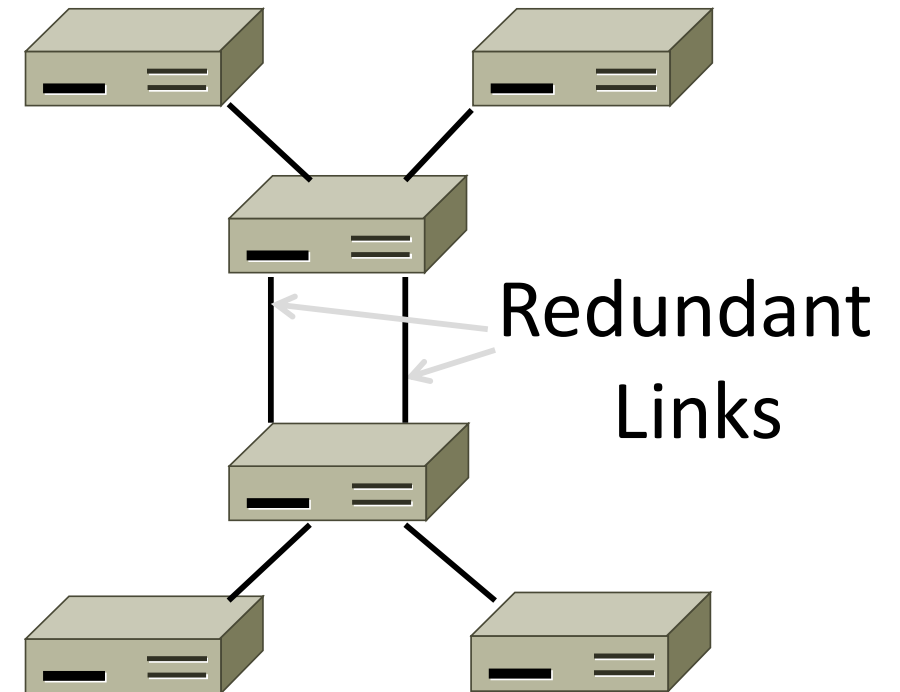


Learning with Multiple Switches



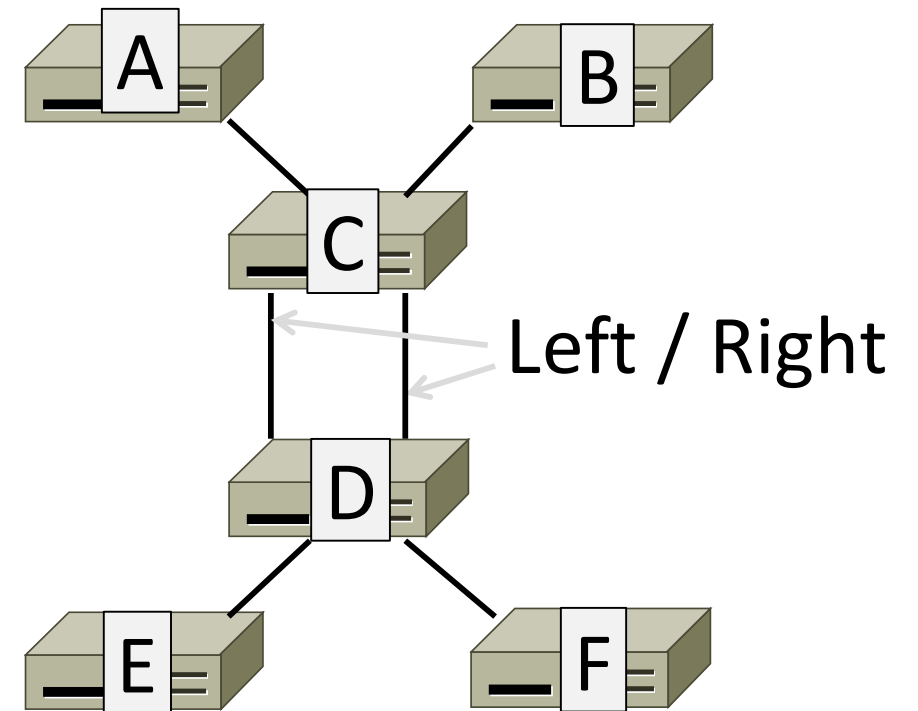
Problem – Forwarding Loops

- 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 ...



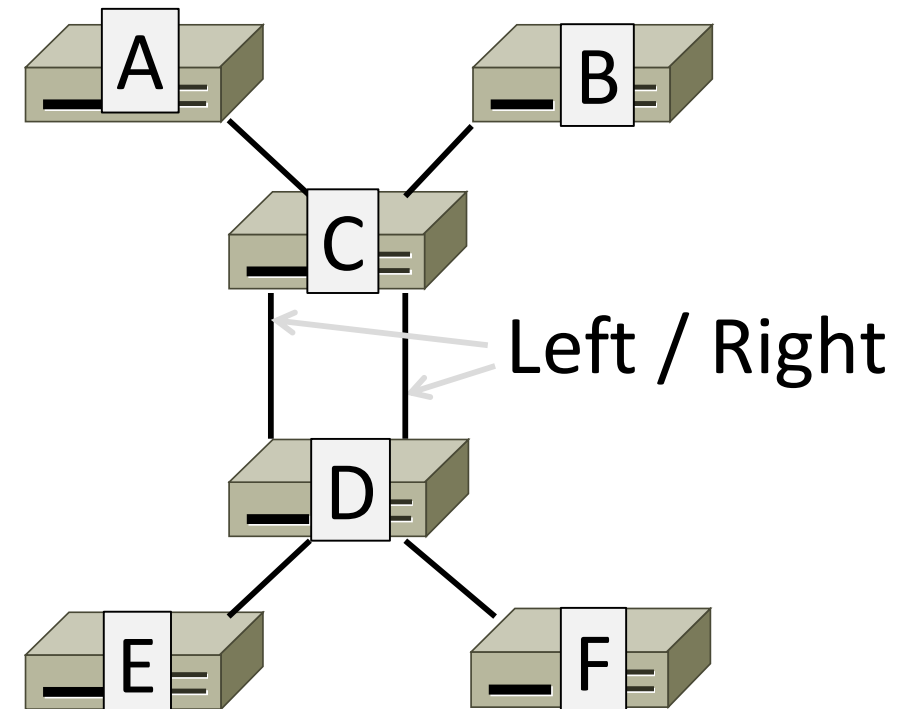
Forwarding Loops

- Suppose the network is started and A sends to F. What happens?



Forwarding Loops (3)

- 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 → ...
- *Why not just remember what you've seen and throw away duplicates?*

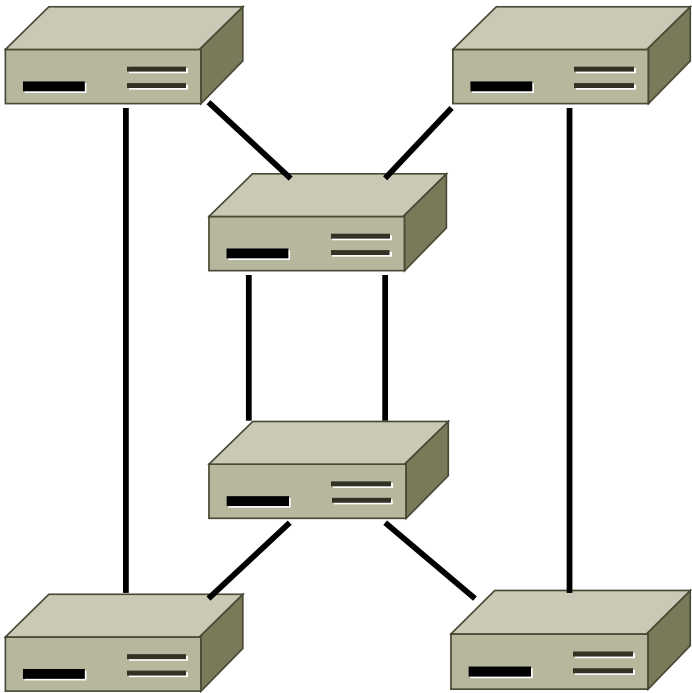


Spanning Tree Solution

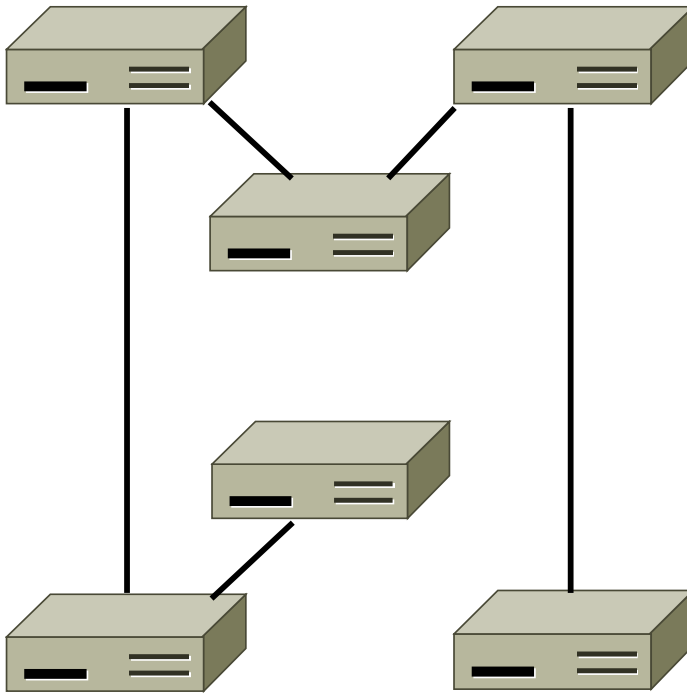
- Switches collectively find a spanning tree for the topology
 - A subset of links that is a tree (no loops) and reaches all switches
 - They switch forward as normal on the spanning tree
 - Broadcasts will go up to the root of the tree and down all the branches

Spanning Tree

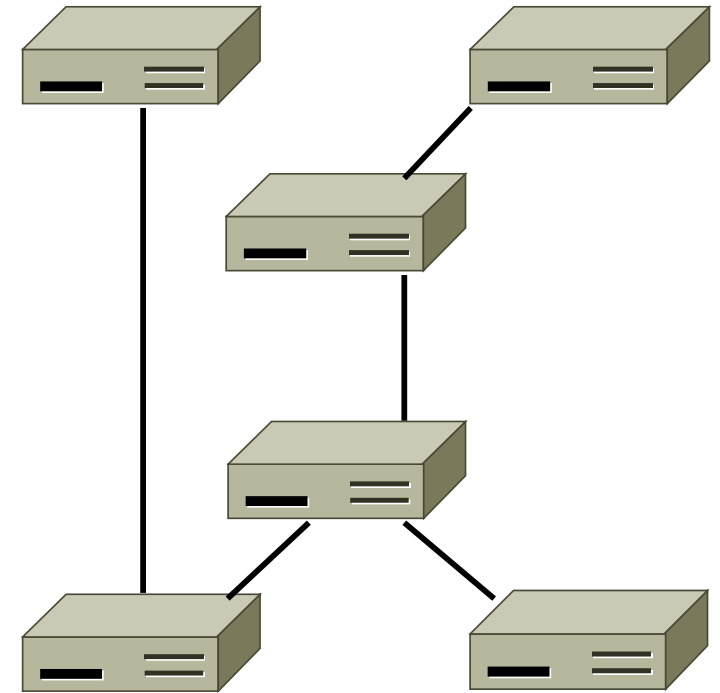
Topology



One ST



Another ST



Spanning Tree 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



Spanning Tree Algorithm (2)

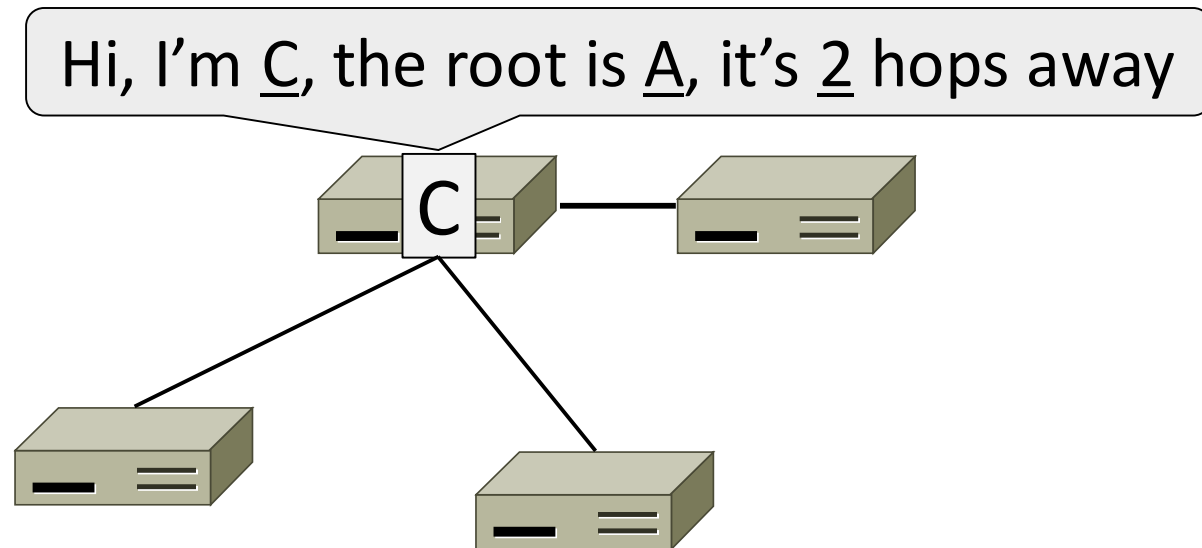
- Overview:

1. Elect a root node of the tree (switch with the lowest address)
2. Grow tree as shortest distances from the root (using lowest address to break distance ties)
3. Turn off ports for forwarding if they aren't on the spanning tree

Spanning Tree Algorithm

- Details:
 - Each switch initially believes it is the root of the tree
 - Each switch sends periodic updates to neighbors with:
 - Its address
 - address of who it believes is the root
 - its distance (in hops) to that root
 - Switches favors
 - Lower numbered switches as roots
 - Routers with shorter distances to root
 - Uses lowest address as a tie for distances

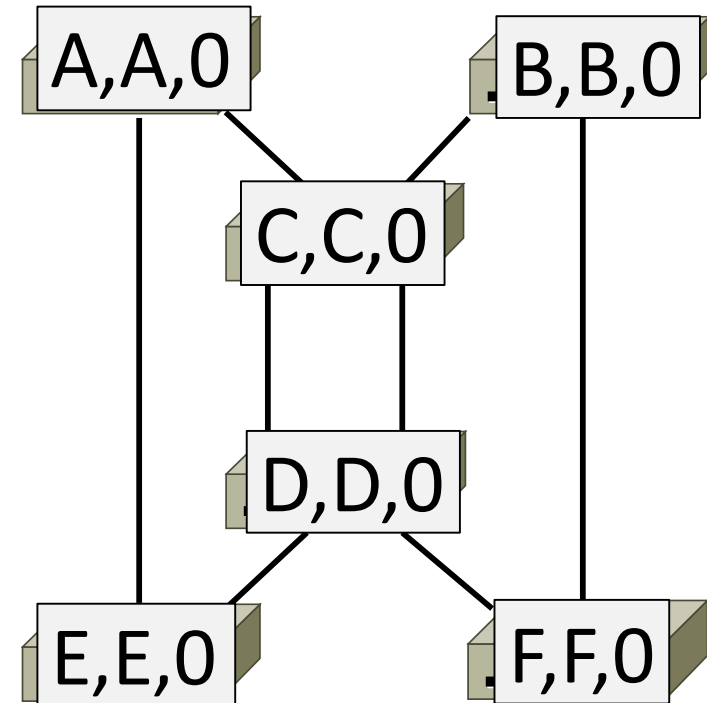
Spanning Tree Algorithm



or (C, A, 2)

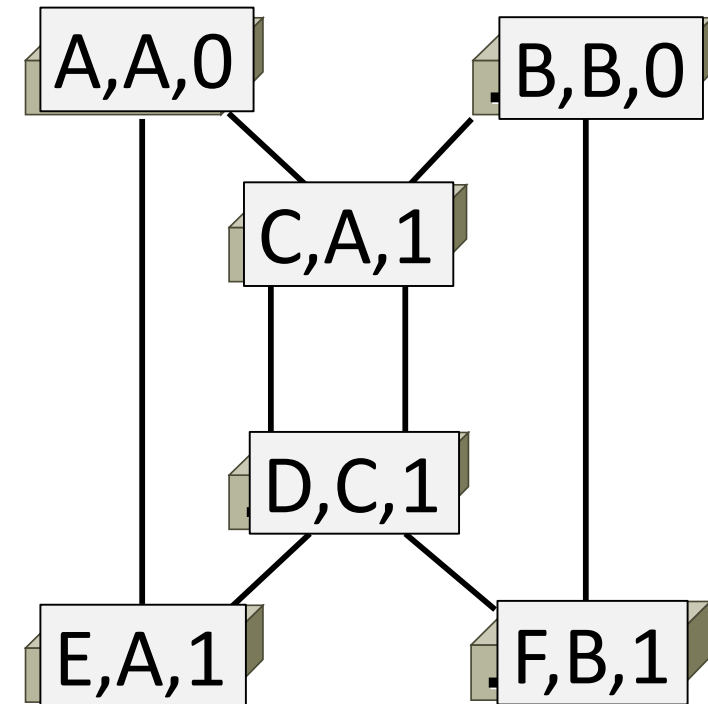
Spanning Tree Example

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



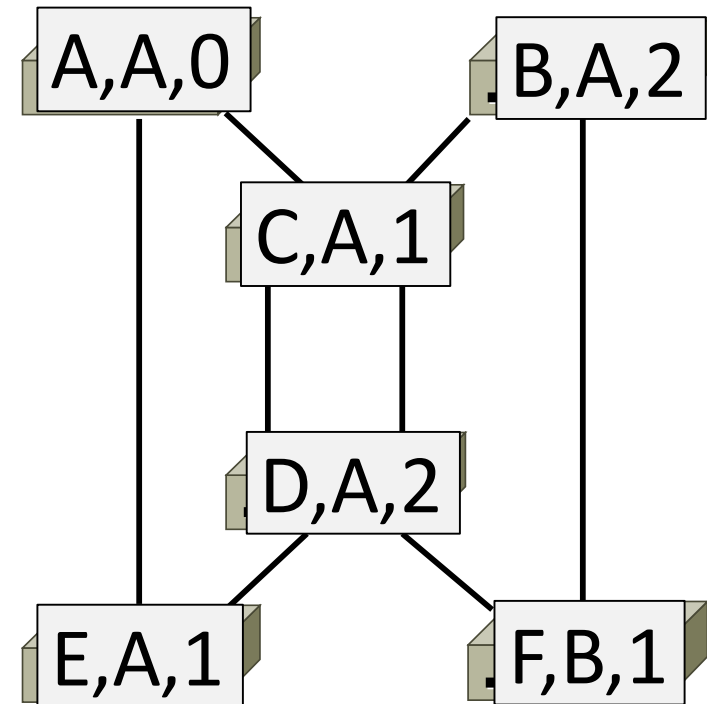
Spanning Tree Example (2)

- 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
 - E remains (E, A, 1)
 - F remains (F, B, 1)



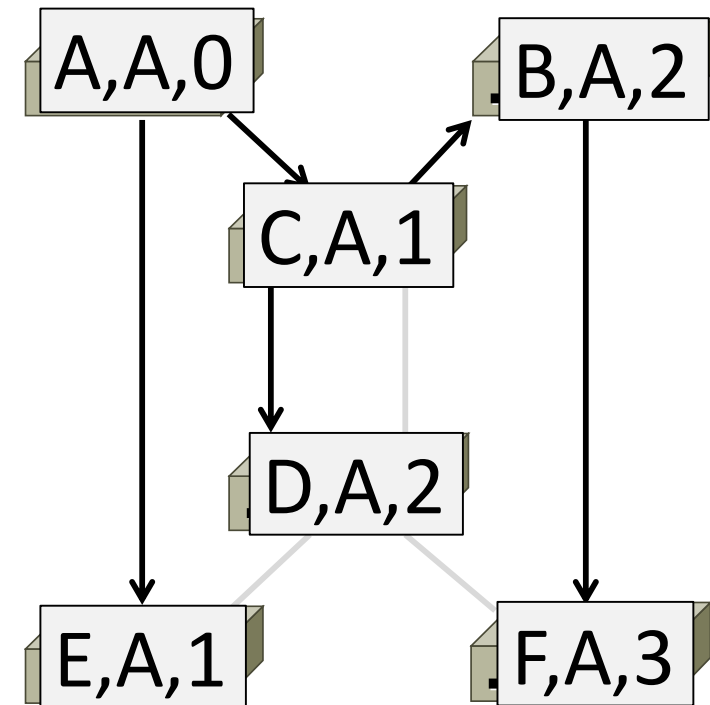
Spanning Tree Example (3)

- 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 to (F, A, 3) via B



Spanning Tree Example (4)

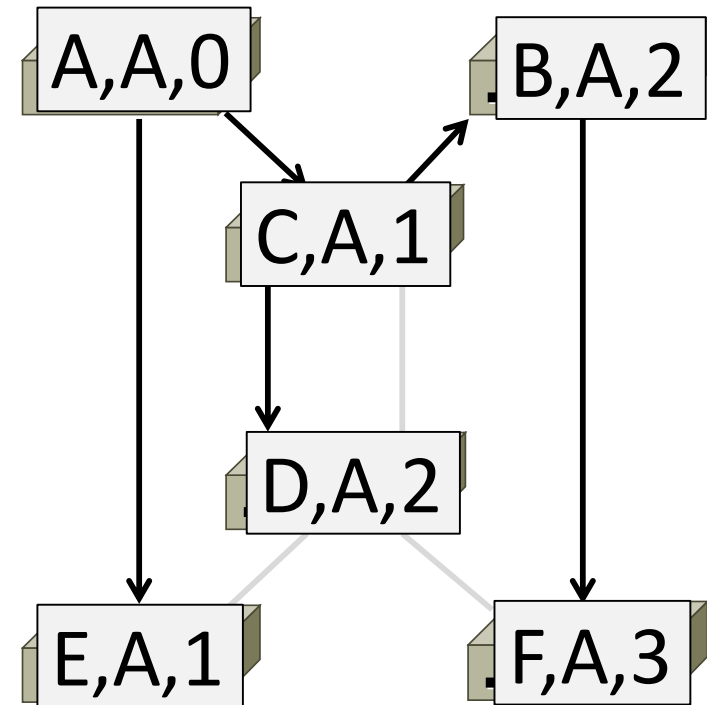
- 4th round
 - Steady-state has been reached
 - Nodes turn off forwarding that is not on the spanning tree
- Algorithm continues to run
 - Adapts by timing out information
 - E.g., if A fails, other nodes forget it, and B will become the new root



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:
 - $D \rightarrow C$ -left
 - $C \rightarrow A, B$
 - $A \rightarrow E$
 - $B \rightarrow F$
- And F sends back to D:
 - $F \rightarrow B$
 - $B \rightarrow C$
 - $C \rightarrow D$(hm, not such a great route)

