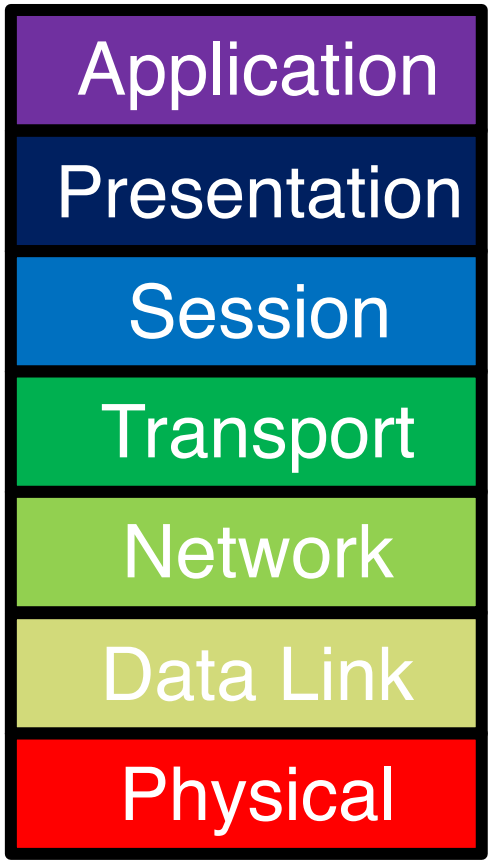
Cristina Nita-Rotaru

CS4700/5700: Network fundamentals

Bridging.

1: Bridging

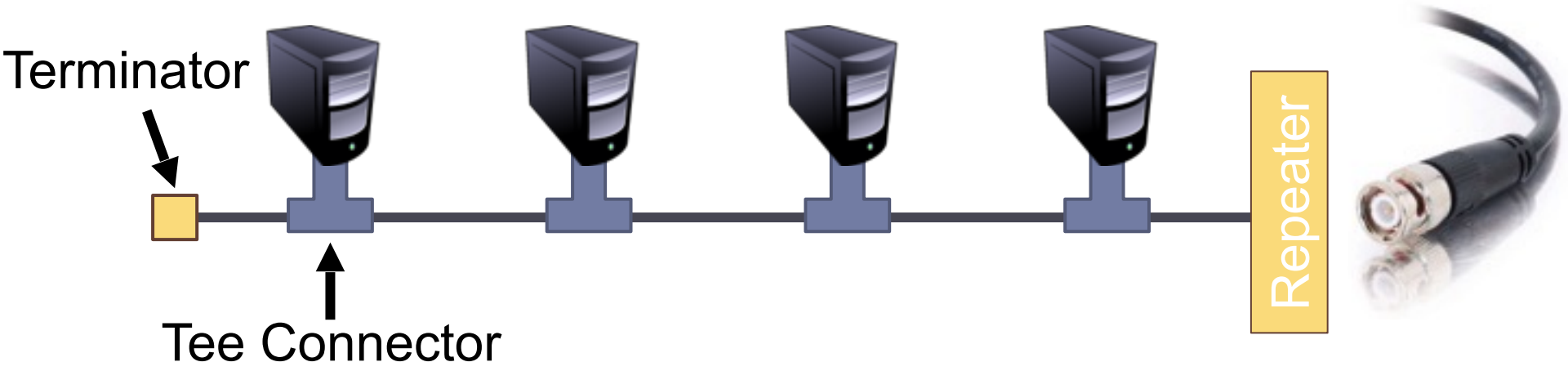
Just Above the Data Link Layer



- Bridging
  - How do we connect LANs?
- Function:
  - Route packets between LANs
- Key challenges:
  - Plug-and-play, self configuration
  - How to resolve loops

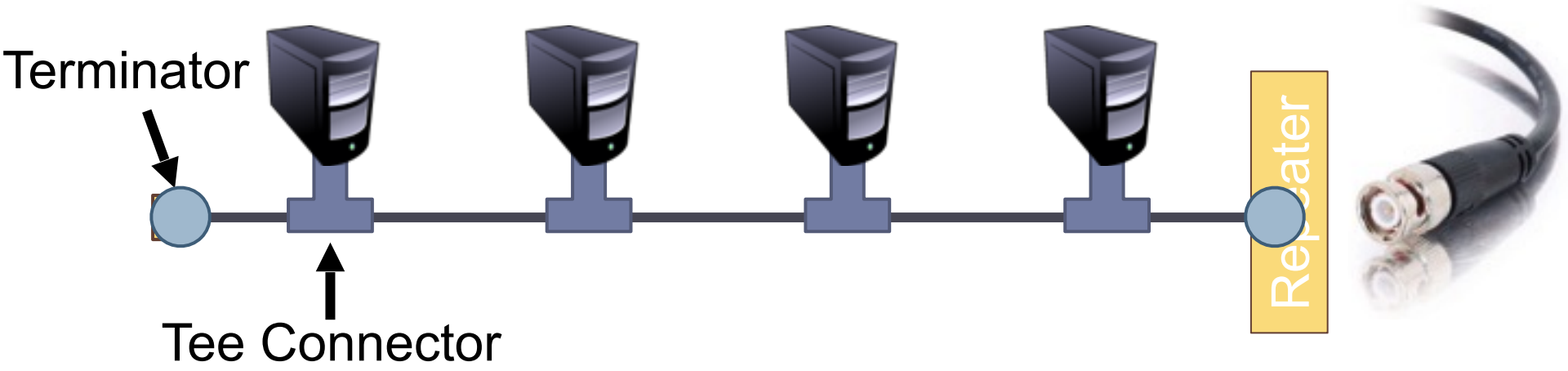
Recap

- Originally, Ethernet was a broadcast technology



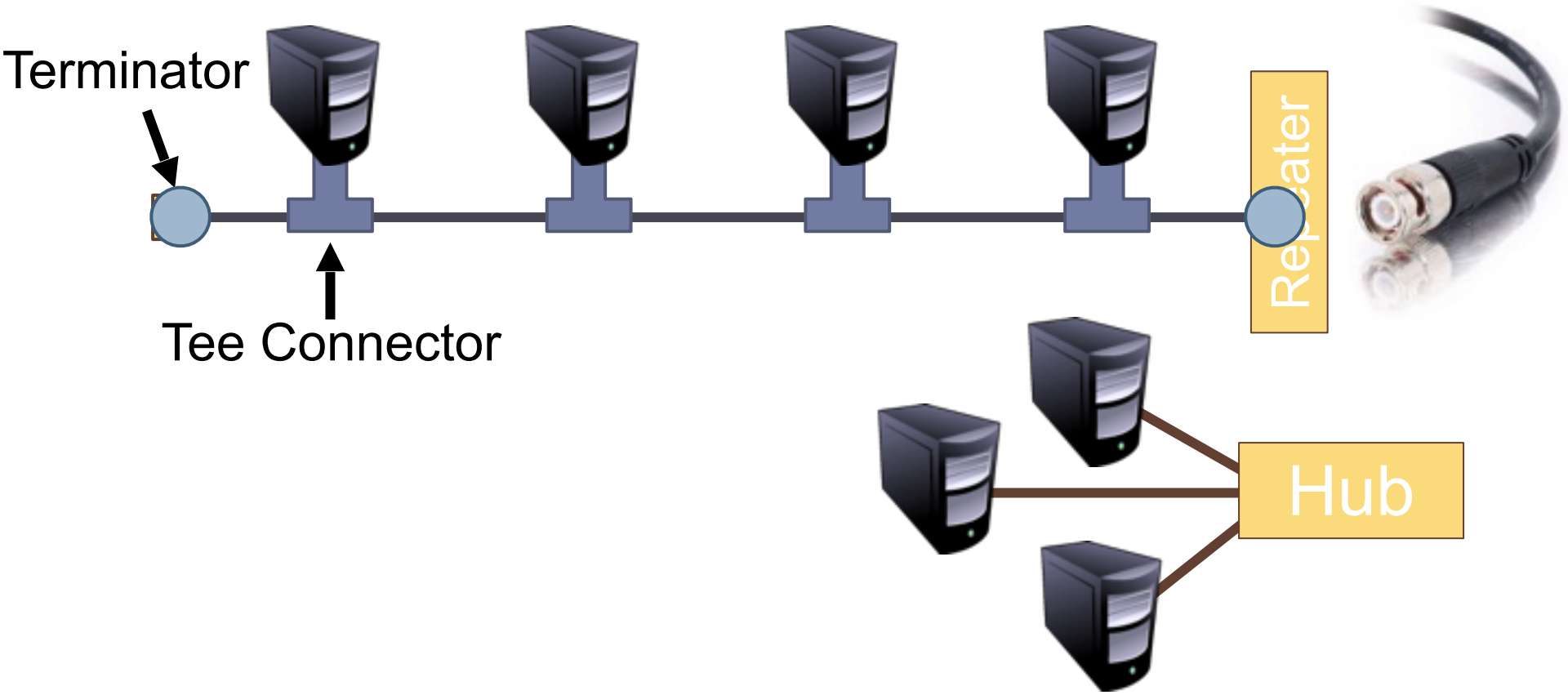
Recap

- Originally, Ethernet was a broadcast technology



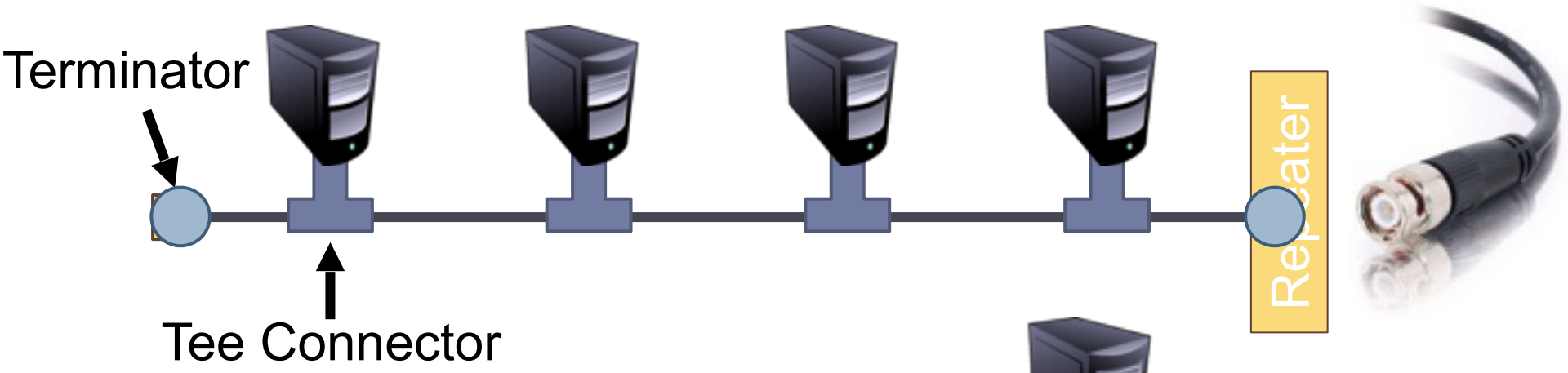
Recap

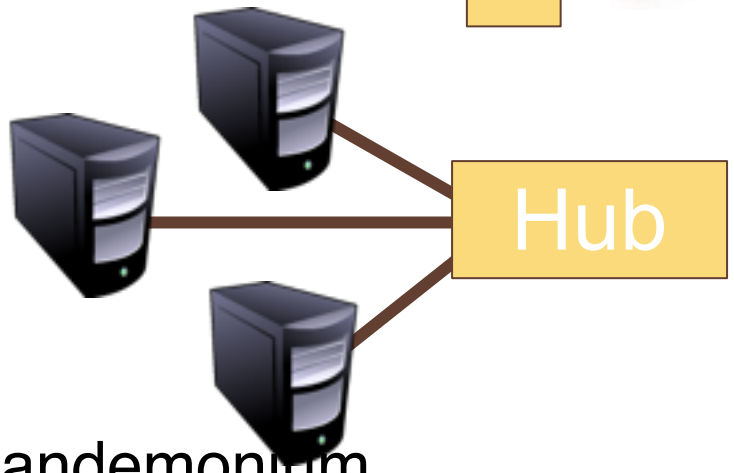
- Originally, Ethernet was a broadcast technology



Recap

- Originally, Ethernet was a broadcast technology

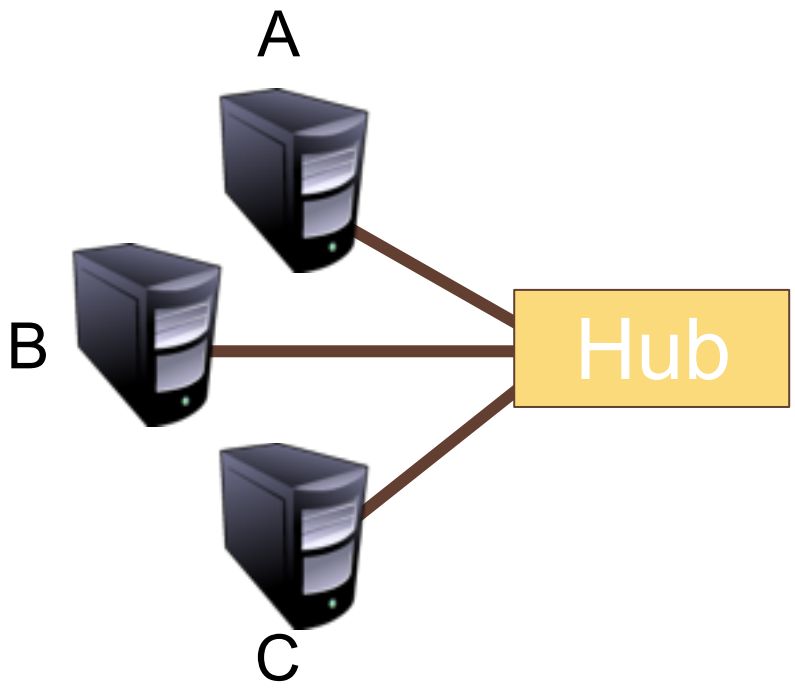




- Pros: Simplicity
  - Hardware is stupid and cheap
- Cons: No scalability
  - More hosts = more collisions = pandemonium

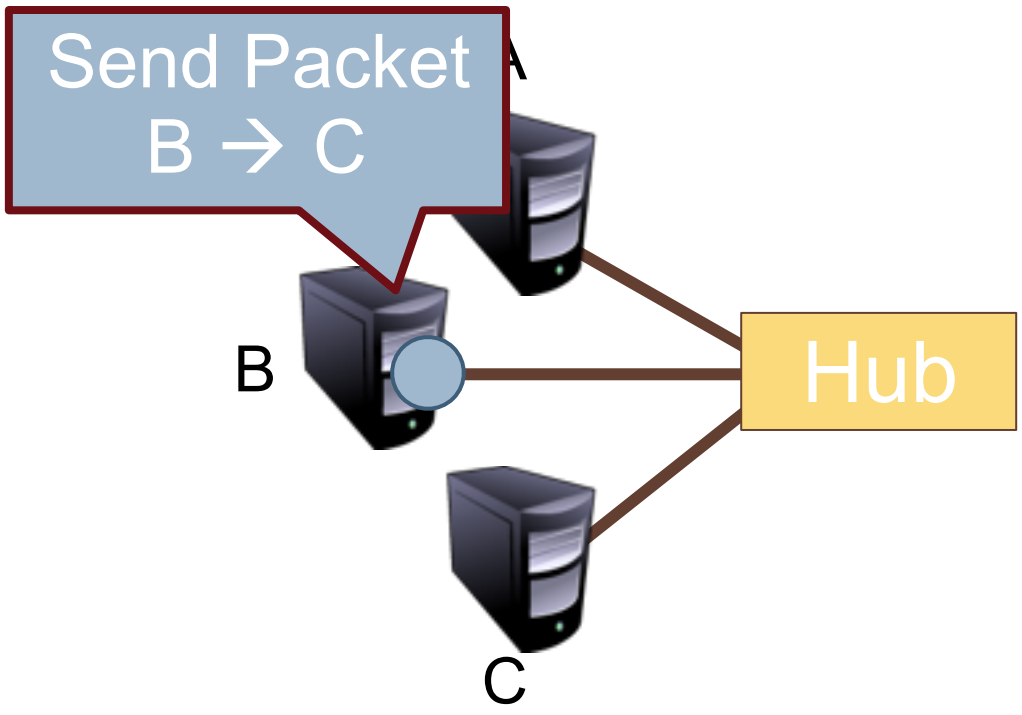
The Case for Bridging

- Need a device that can bridge different LANs
  - Only forward packets to intended recipients
  - No broadcast!



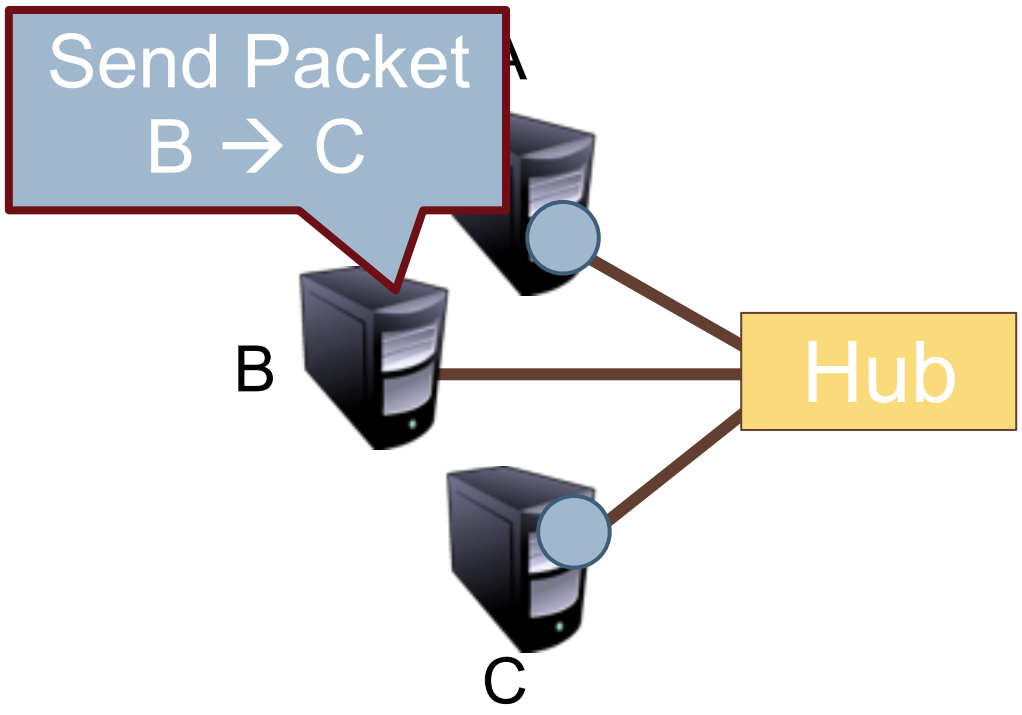
The Case for Bridging

- Need a device that can bridge different LANs
  - Only forward packets to intended recipients
  - No broadcast!



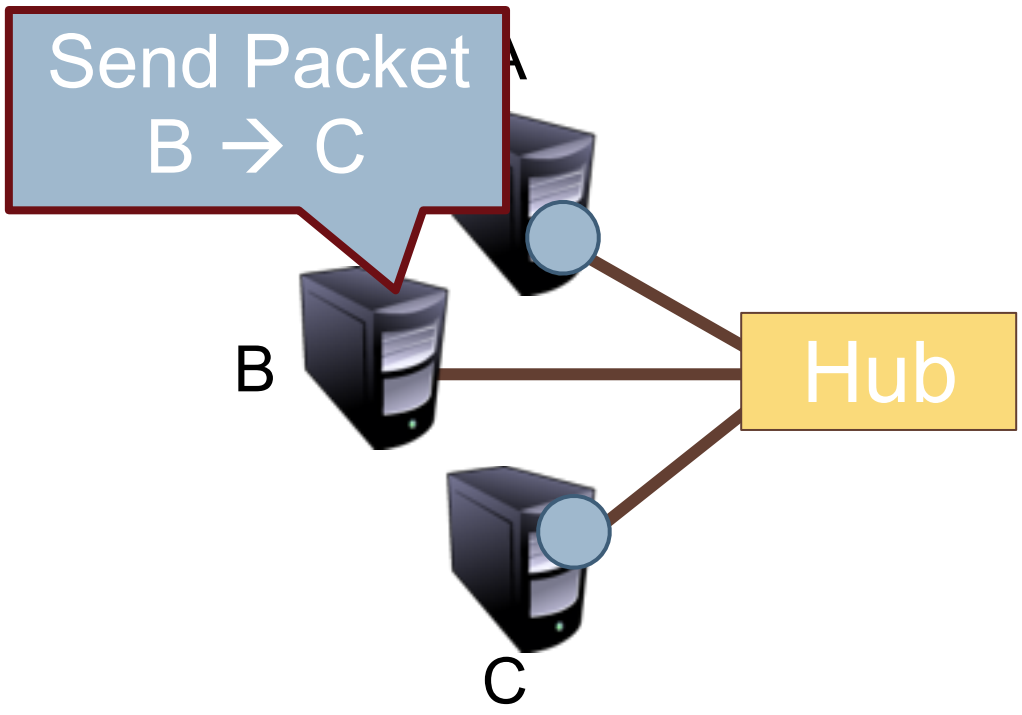
The Case for Bridging

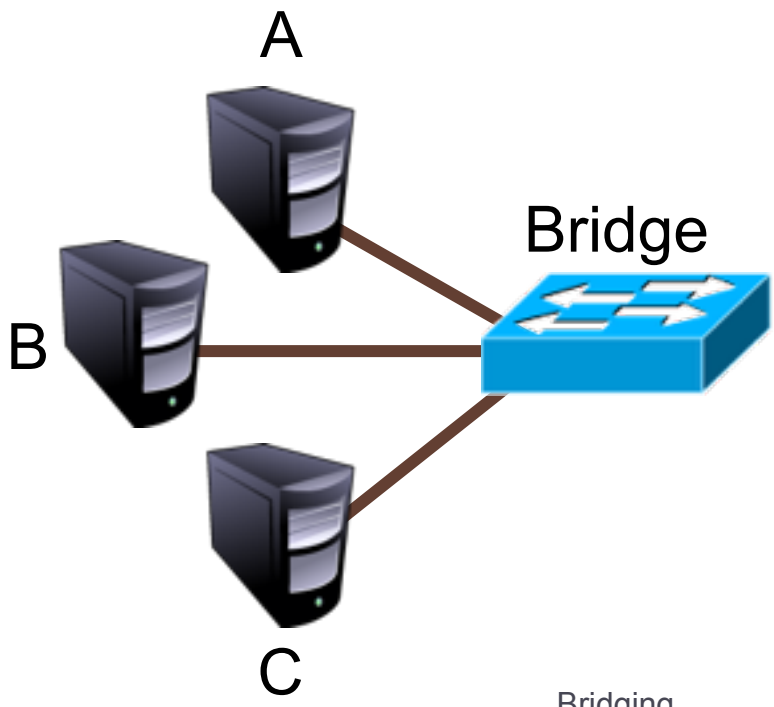
- Need a device that can bridge different LANs
  - Only forward packets to intended recipients
  - No broadcast!



The Case for Bridging

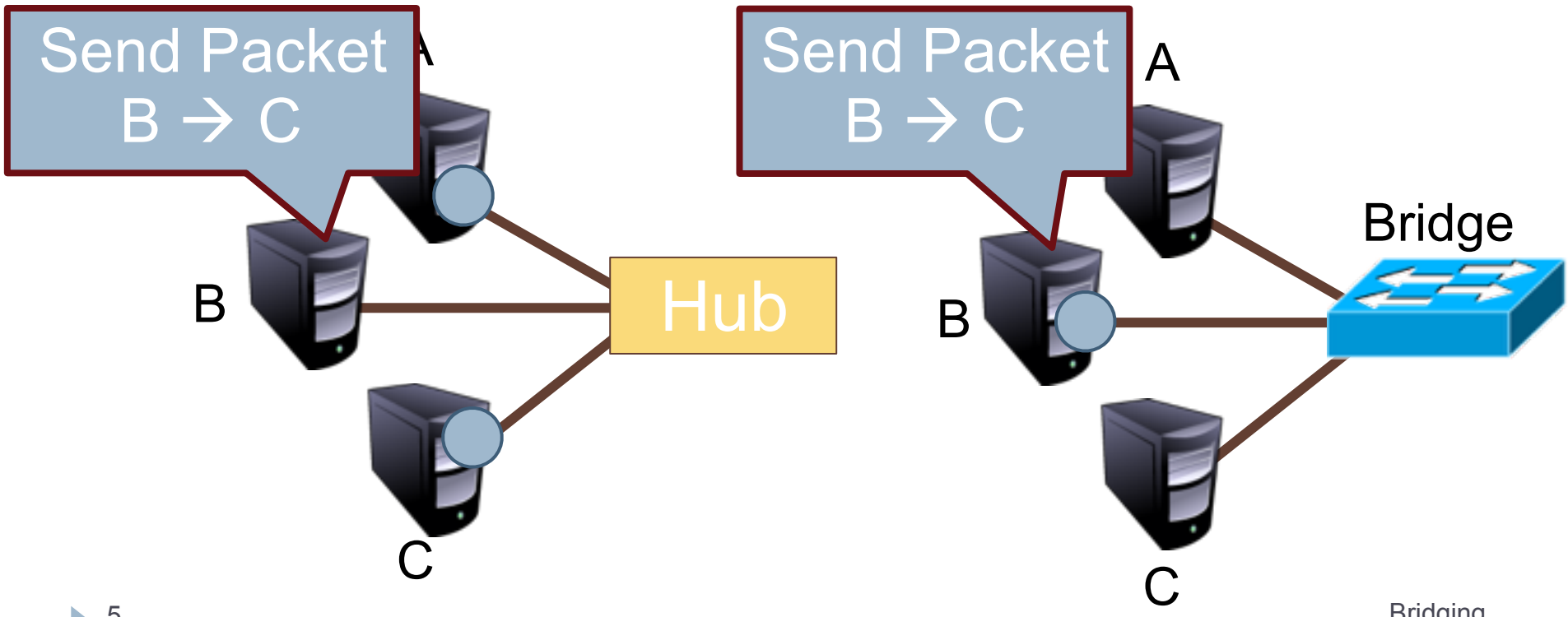
- Need a device that can bridge different LANs
  - Only forward packets to intended recipients
  - No broadcast!





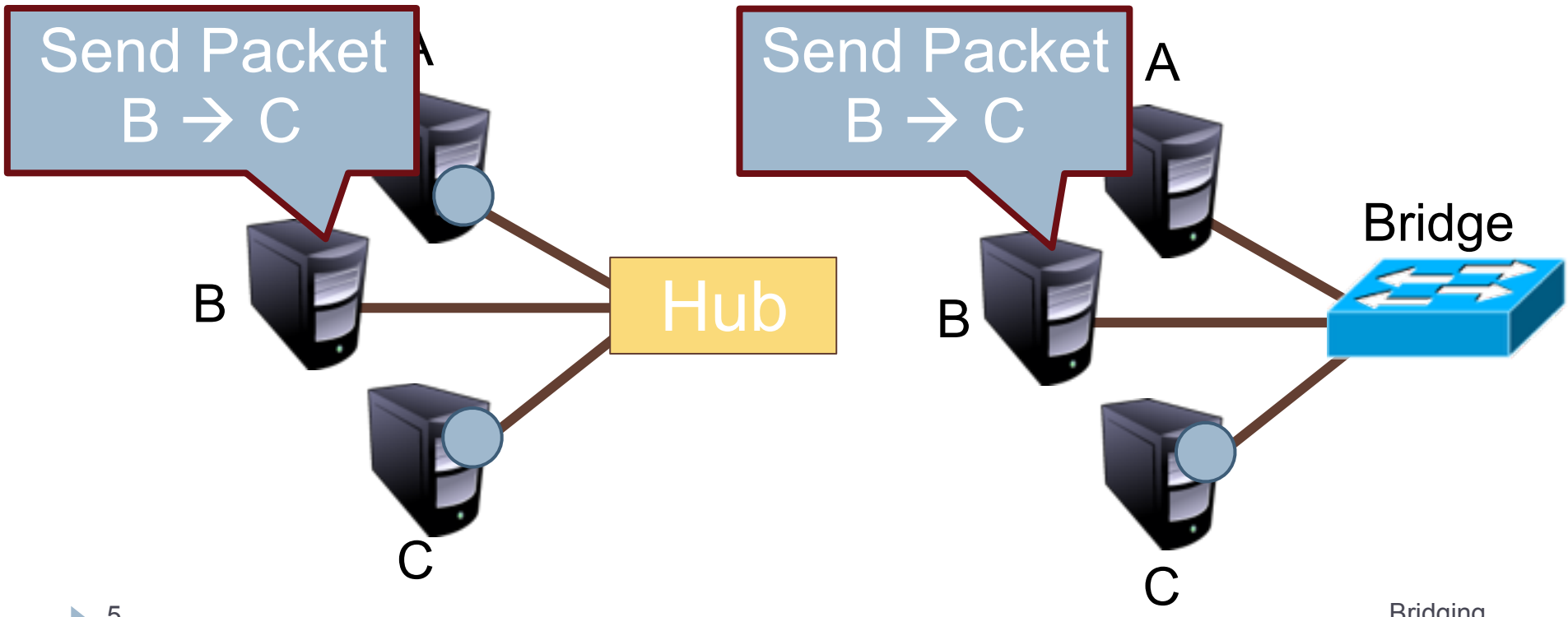
The Case for Bridging

- Need a device that can bridge different LANs
  - Only forward packets to intended recipients
  - No broadcast!

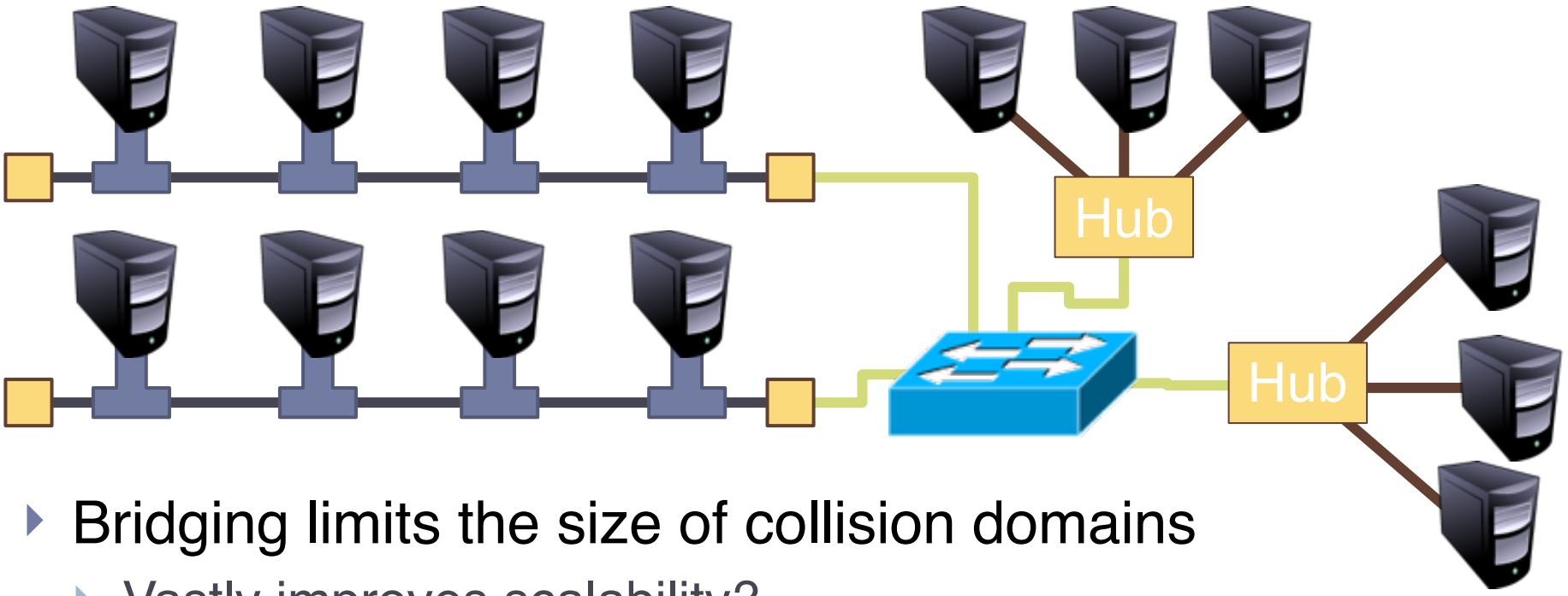


The Case for Bridging

- Need a device that can bridge different LANs
  - Only forward packets to intended recipients
  - No broadcast!

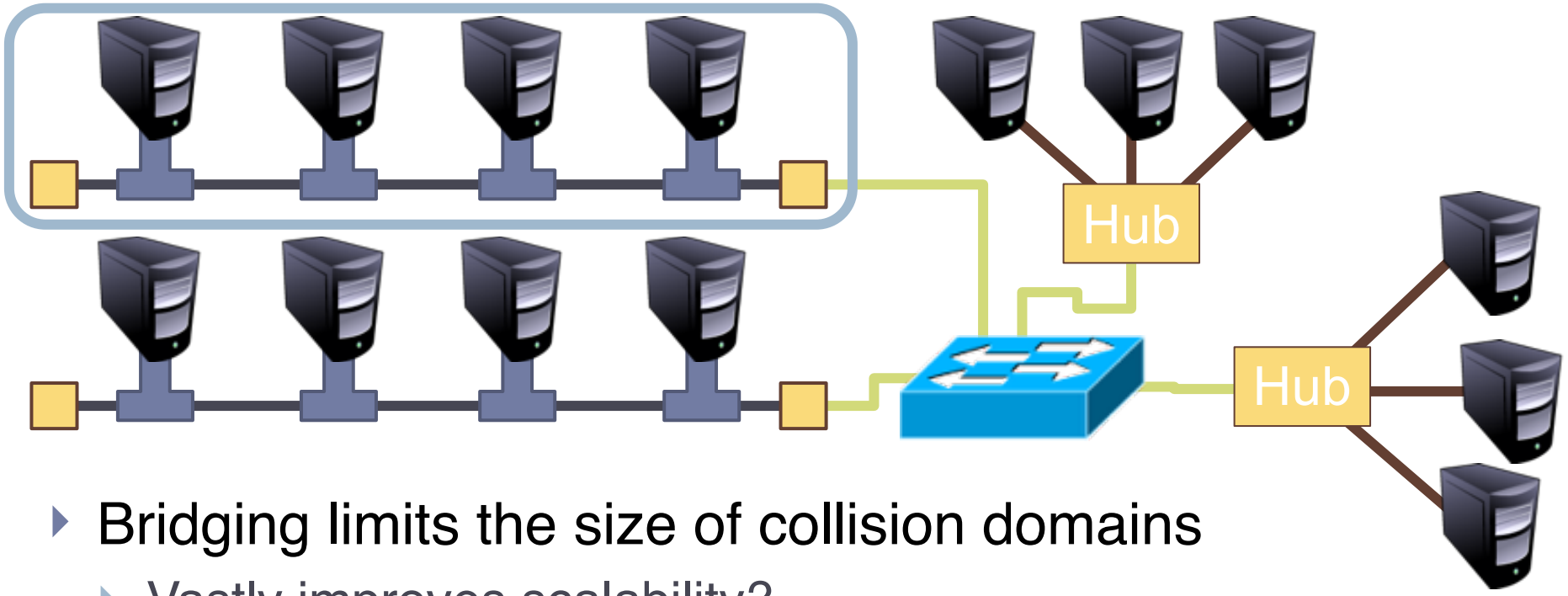


Bridging the LANs



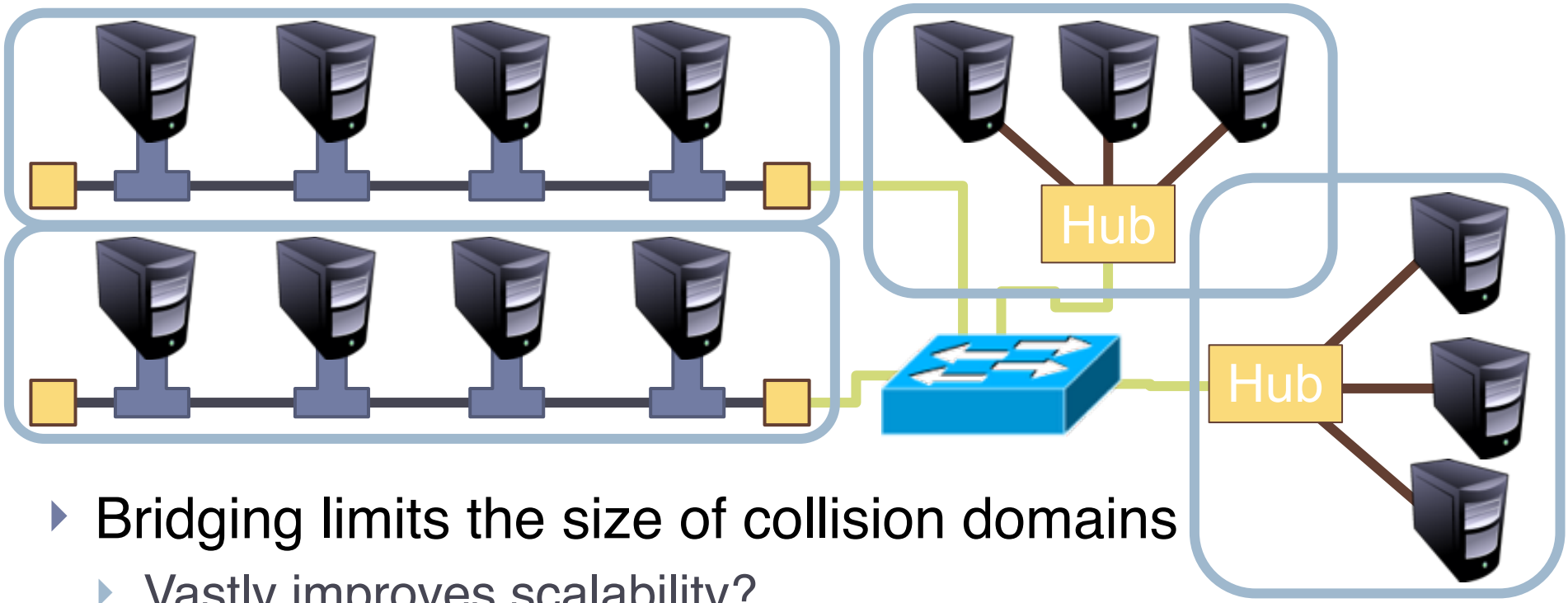
- Bridging limits the size of collision domains
  - Vastly improves scalability?
- Tradeoff: bridges are more complex than hubs
  - Physical layer device vs. data link layer device
  - Need memory buffers, packet processing hardware, routing tables

Bridging the LANs



- Bridging limits the size of collision domains
  - Vastly improves scalability?
- Tradeoff: bridges are more complex than hubs
  - Physical layer device vs. data link layer device
  - Need memory buffers, packet processing hardware, routing tables

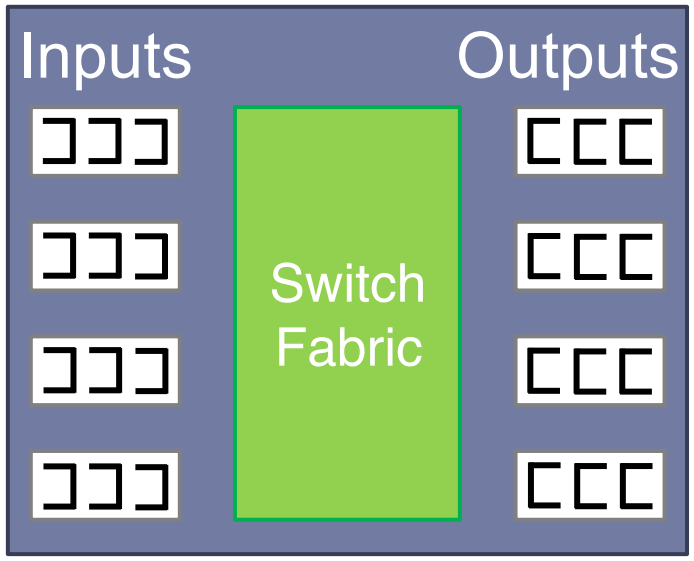
Bridging the LANs



- Bridging limits the size of collision domains
  - Vastly improves scalability?
- Tradeoff: bridges are more complex than hubs
  - Physical layer device vs. data link layer device
  - Need memory buffers, packet processing hardware, routing tables

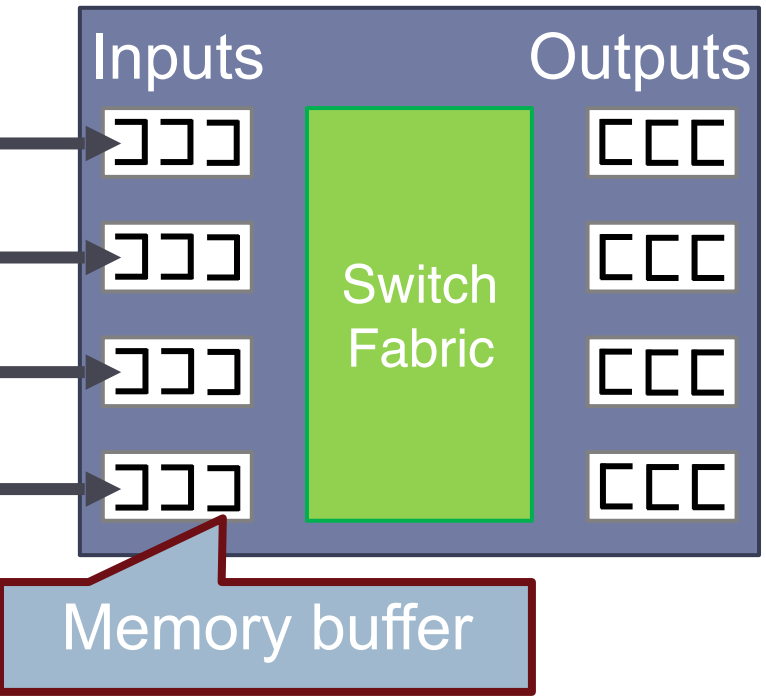
Bridge Internals

- Bridges have memory buffers to queue packets
- Bridge is intelligent, only forwards packets to the correct output
- Bridges are high performance, full N x line rate is possible



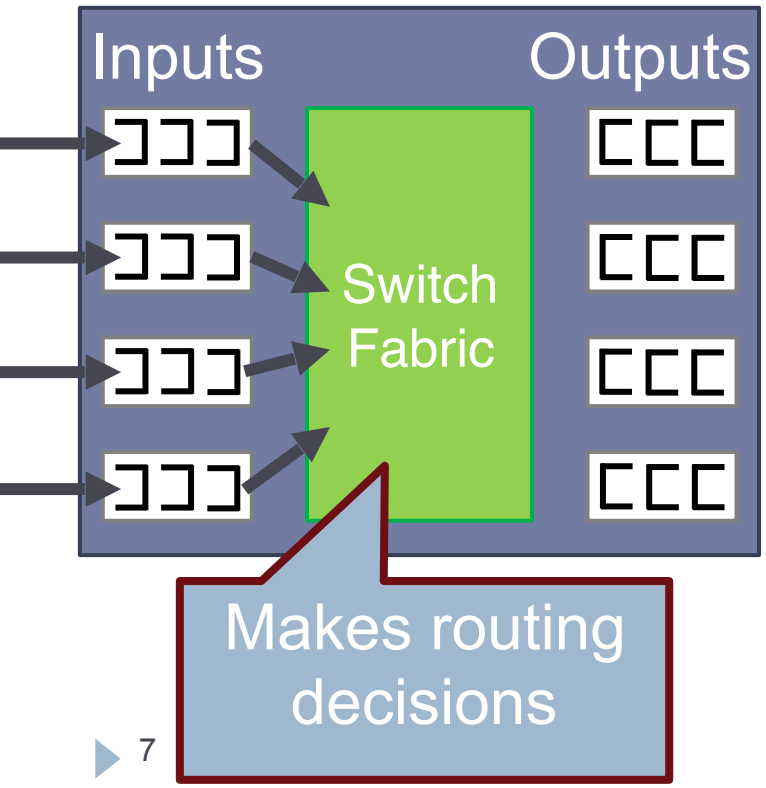
Bridge Internals

- Bridges have memory buffers to queue packets
- Bridge is intelligent, only forwards packets to the correct output
- Bridges are high performance, full N x line rate is possible



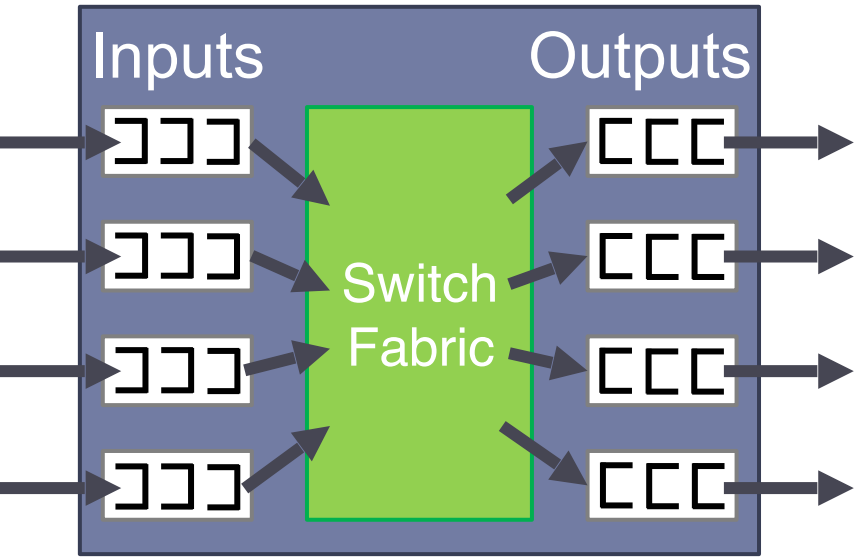
Bridge Internals

- Bridges have memory buffers to queue packets
- Bridge is intelligent, only forwards packets to the correct output
- Bridges are high performance, full N x line rate is possible



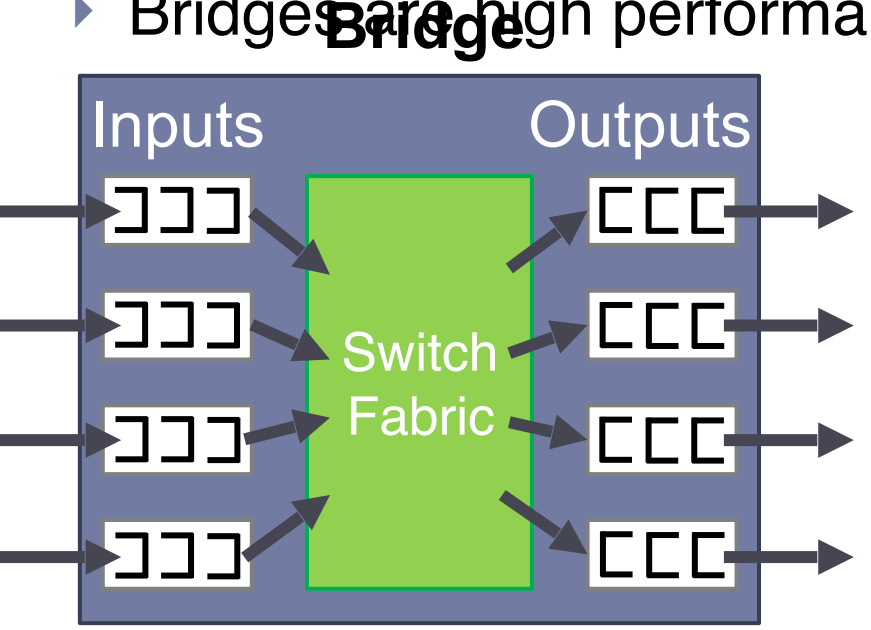
Bridge Internals

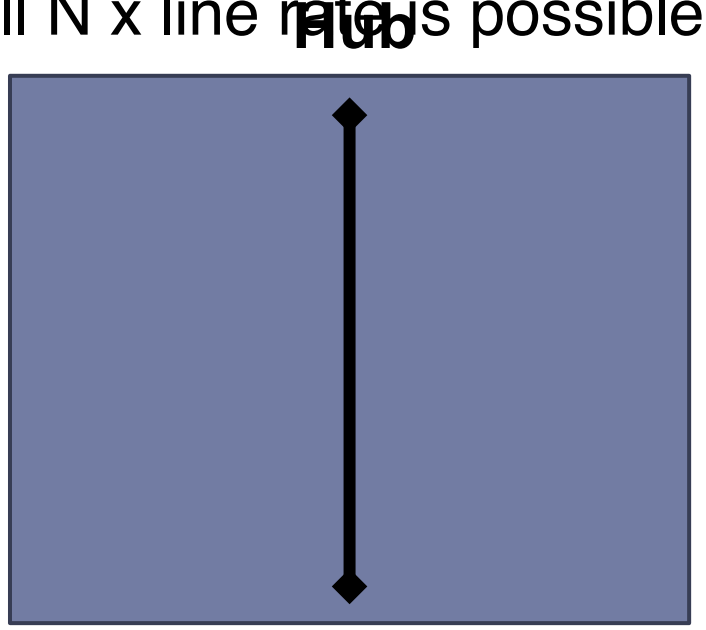
- Bridges have memory buffers to queue packets
- Bridge is intelligent, only forwards packets to the correct output
- Bridges are high performance, full N x line rate is possible



Bridge Internals

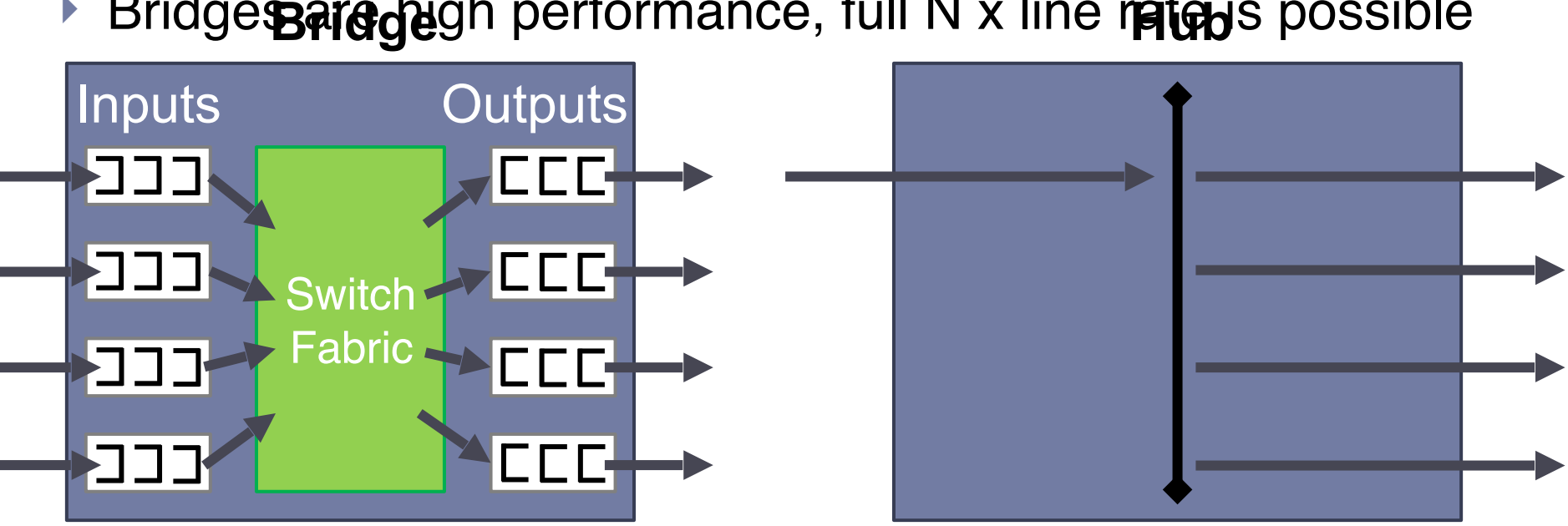
- Bridges have memory buffers to queue packets
- Bridge is intelligent, only forwards packets to the correct output
- Bridges are high performance, full N x line rate is possible





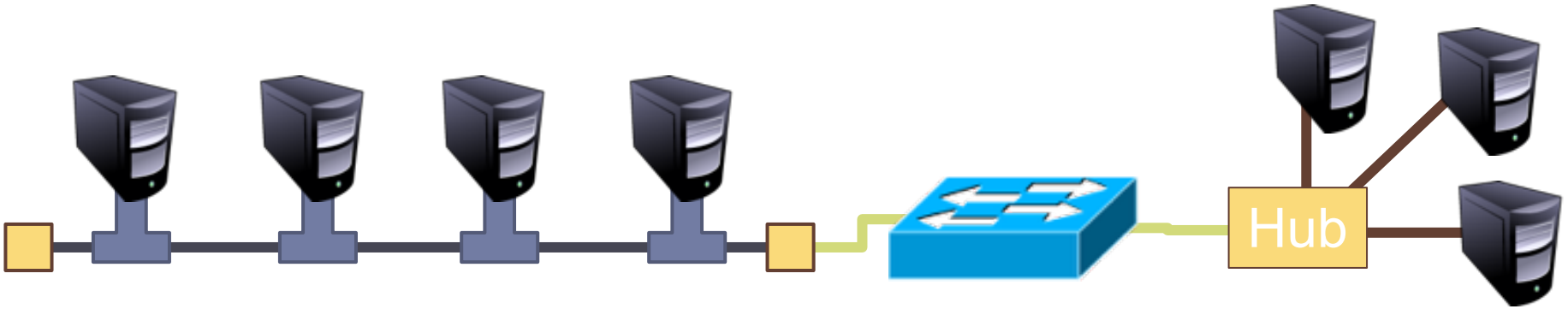
Bridge Internals

- Bridges have memory buffers to queue packets
- Bridge is intelligent, only forwards packets to the correct output
- Bridges are high performance, full N x line rate is possible



Bridges

- Original form of Ethernet switch
- Connect multiple IEEE 802 LANs at layer 2
- Goals
  - Reduce the collision domain
  - Complete transparency
    - "Plug-and-play," self-configuring
    - No hardware of software changes on hosts/hubs
    - Should not impact existing LAN operations



Next

Forwarding of frames

Learning of (MAC) Addresses

Spanning Tree Algorithm (to handle loops)

Frame Forwarding Tables

- Each bridge maintains a forwarding table

MAC Address	Port	Age
00:00:00:00:00:AA	1	1 minute
00:00:00:00:00:BB	2	7 minutes
00:00:00:00:00:CC	3	2 seconds

Frame Forwarding Tables

- Each bridge maintains a forwarding table

MAC Address	Port	Age
00:00:00:00:00:AA	1	1 minute
00:00:00:00:00:BB	2	7 minutes
00:00:00:00:00:CC	3	2 seconds

Frame Forwarding Tables

- Each bridge maintains a forwarding table

MAC Address	Port	Age
00:00:00:00:00:AA	1	1 minute
00:00:00:00:00:BB	2	7 minutes
00:00:00:00:00:CC	3	2 seconds

Frame Forwarding Tables

- Each bridge maintains a forwarding table

MAC Address	Port	Age
00:00:00:00:00:AA	1	1 minute
00:00:00:00:00:BB	2	7 minutes
00:00:00:00:00:CC	3	2 seconds
00:00:00:00:00:DD	1	3 minutes

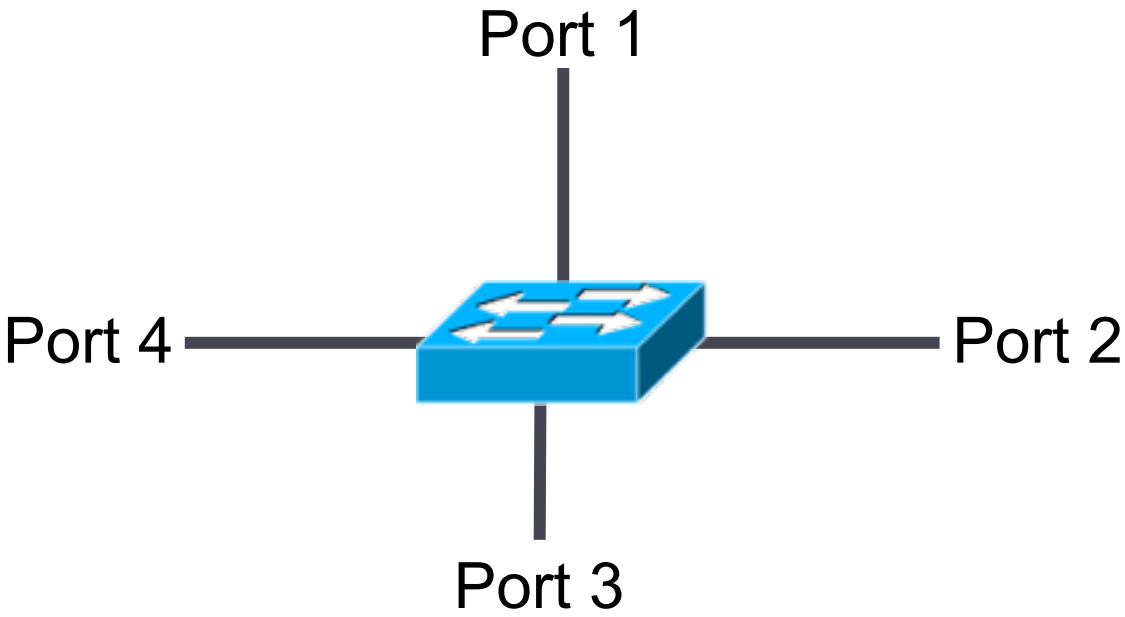
Frame Forwarding Tables

- Each bridge maintains a forwarding table

MAC Address	Port	Age
00:00:00:00:00:AA	1	1 minute
00:00:00:00:00:BB	2	7 minutes
00:00:00:00:00:CC	3	2 seconds
00:00:00:00:00:DD	1	3 minutes

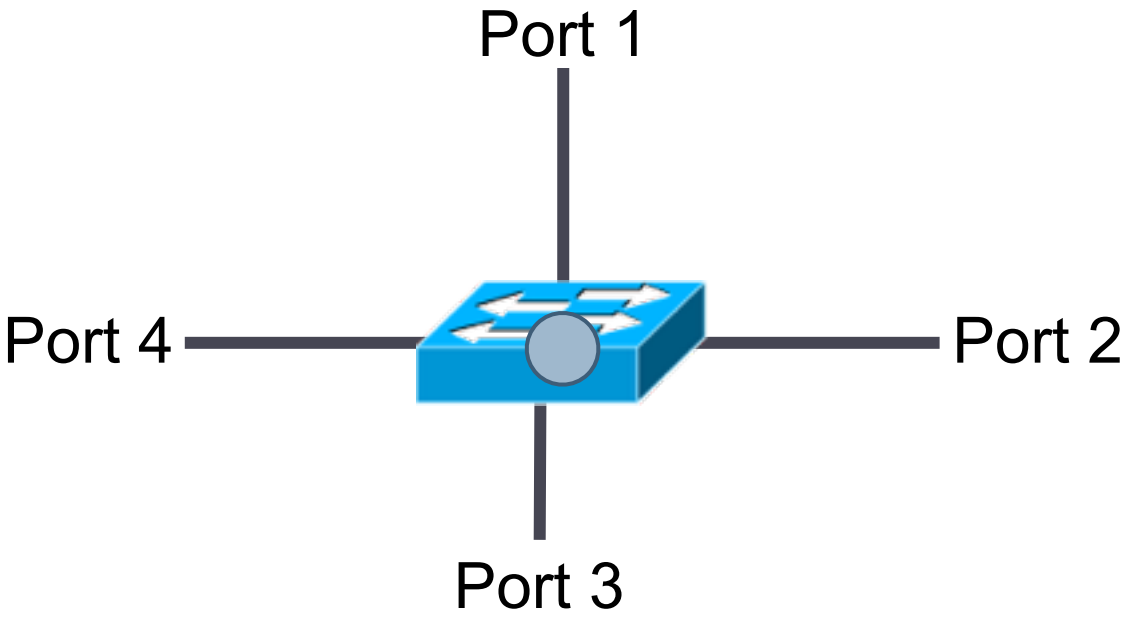
Frame Forwarding in Action

- Assume a frame arrives on port 1
- If the destination MAC address is in the forwarding table, send the frame on the correct output port
- If the destination MAC isn't in the forwarding table, broadcast the frame on all ports except 1



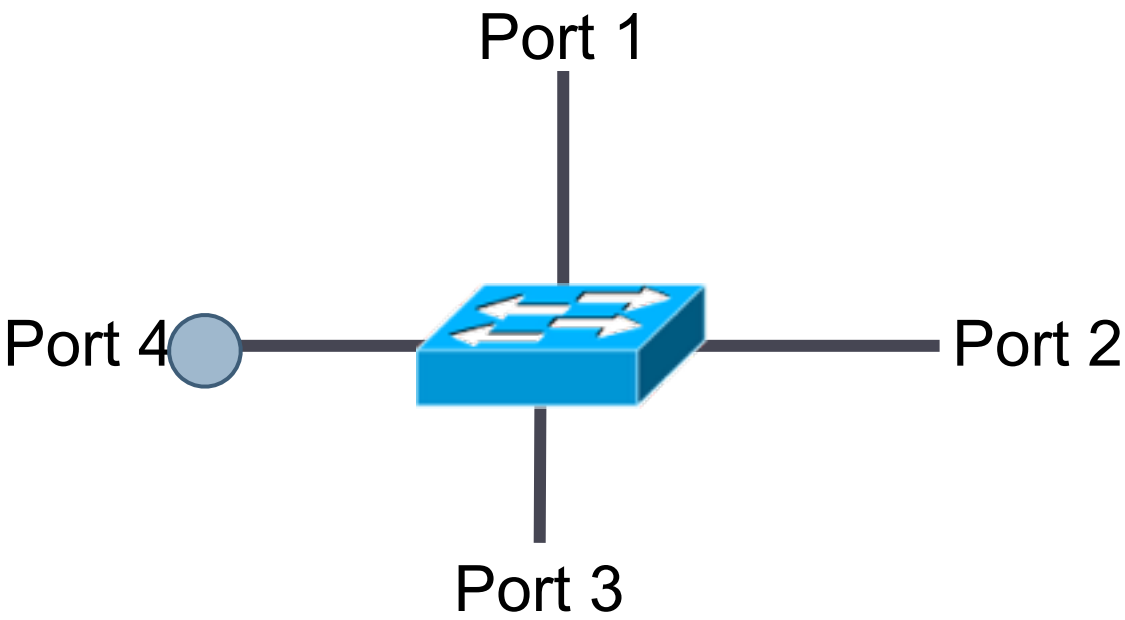
Frame Forwarding in Action

- Assume a frame arrives on port 1
- If the destination MAC address is in the forwarding table, send the frame on the correct output port
- If the destination MAC isn't in the forwarding table, broadcast the frame on all ports except 1



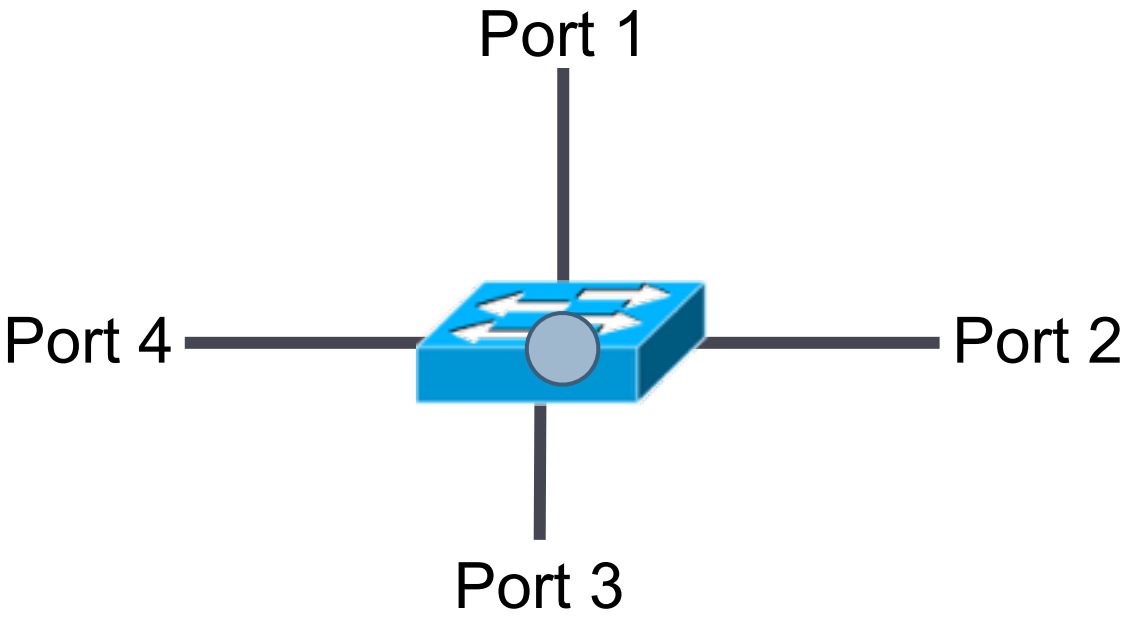
Frame Forwarding in Action

- Assume a frame arrives on port 1
- If the destination MAC address is in the forwarding table, send the frame on the correct output port
- If the destination MAC isn't in the forwarding table, broadcast the frame on all ports except 1



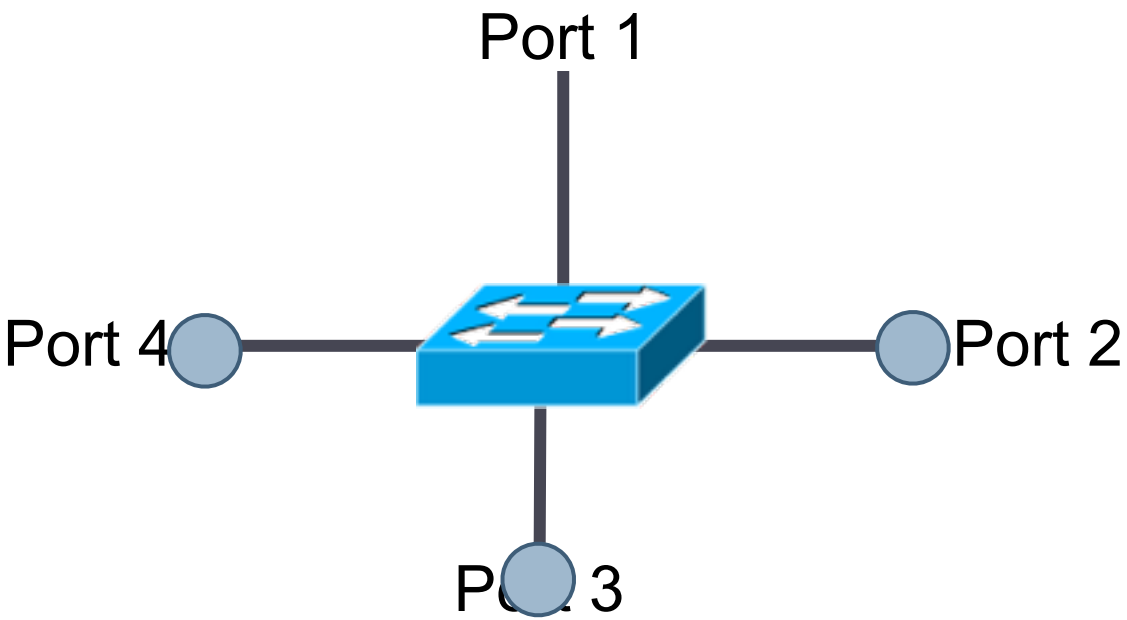
Frame Forwarding in Action

- Assume a frame arrives on port 1
- If the destination MAC address is in the forwarding table, send the frame on the correct output port
- If the destination MAC isn't in the forwarding table, broadcast the frame on all ports except 1



Frame Forwarding in Action

- Assume a frame arrives on port 1
- If the destination MAC address is in the forwarding table, send the frame on the correct output port
- If the destination MAC isn't in the forwarding table, broadcast the frame on all ports except 1

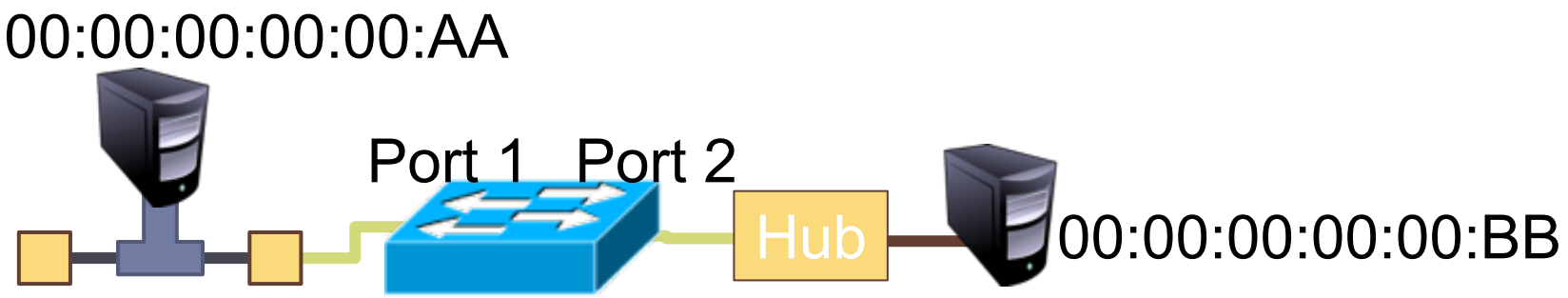


Learning Addresses

- Manual configuration is possible, but…
  - Time consuming
  - Error prone
  - Not flexible (add/ remove hosts)

Instead, learn addresses using a simple heuristic

- Look at the source of frames that arrive on each port

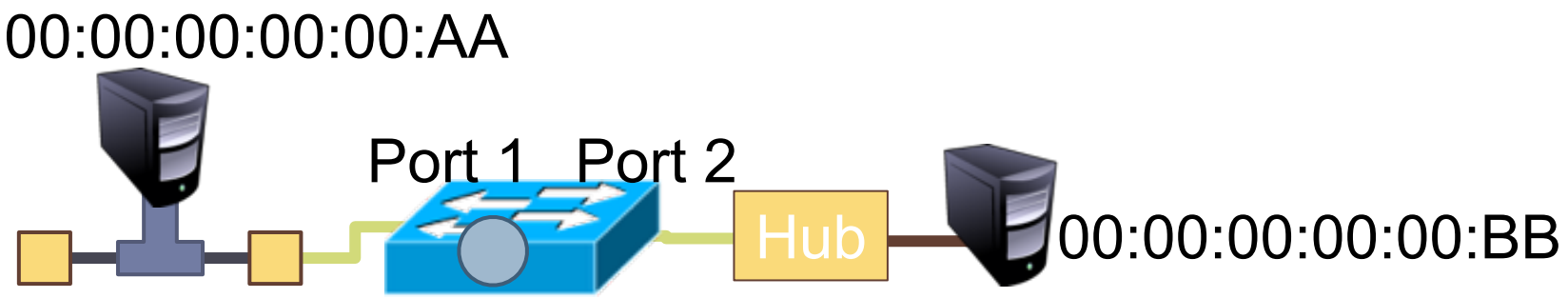


Learning Addresses

- Manual configuration is possible, but…
  - Time consuming
  - Error prone
  - Not flexible (add/ remove hosts)

Instead, learn addresses using a simple heuristic

- Look at the source of frames that arrive on each port

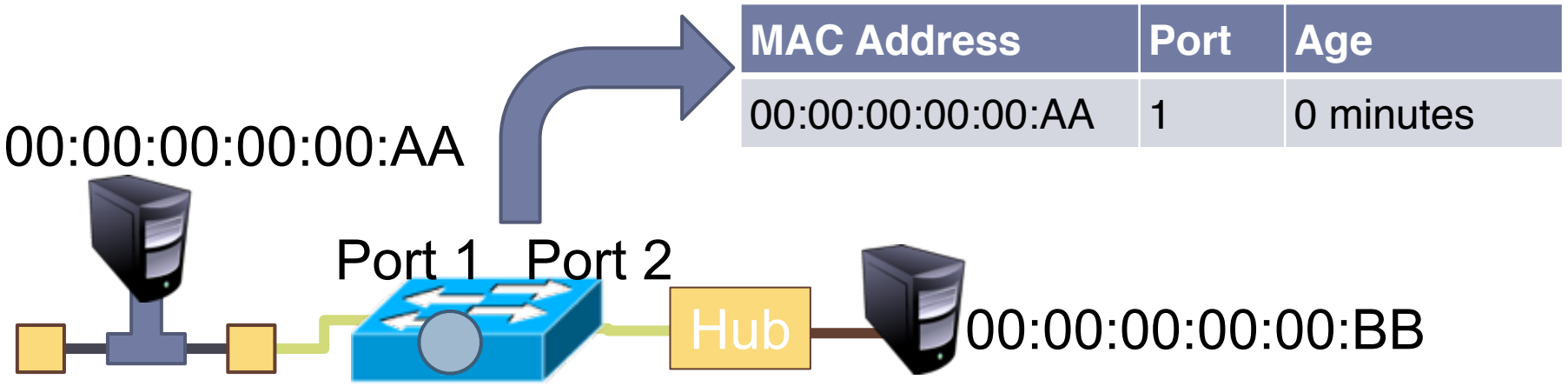


Learning Addresses

- Manual configuration is possible, but…
  - Time consuming
  - Error prone
  - Not flexible (add/ remove hosts)

Instead, learn addresses using a simple heuristic

- Look at the source of frames that arrive on each port



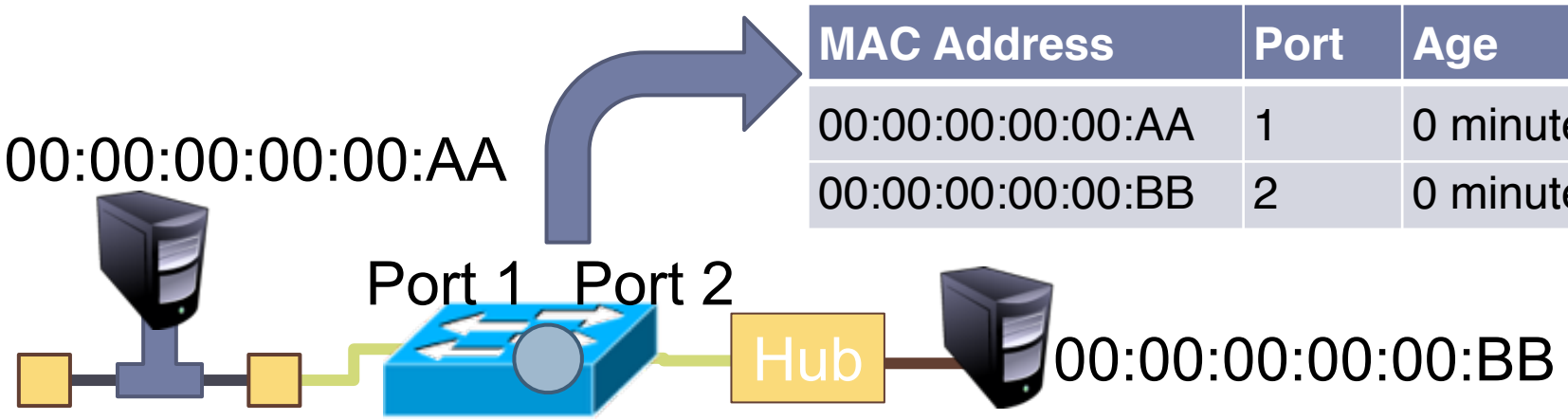
MAC Address	Port	Age
00:00:00:00:00:AA	1	0 minutes

Learning Addresses

- Manual configuration is possible, but…
  - Time consuming
  - Error prone
  - Not flexible (add/ remove hosts)

Instead, learn addresses using a simple heuristic

- Look at the source of frames that arrive on each port



MAC Address	Port	Age
00:00:00:00:00:AA	1	0 minutes
00:00:00:00:00:BB	2	0 minutes

Learning Addresses

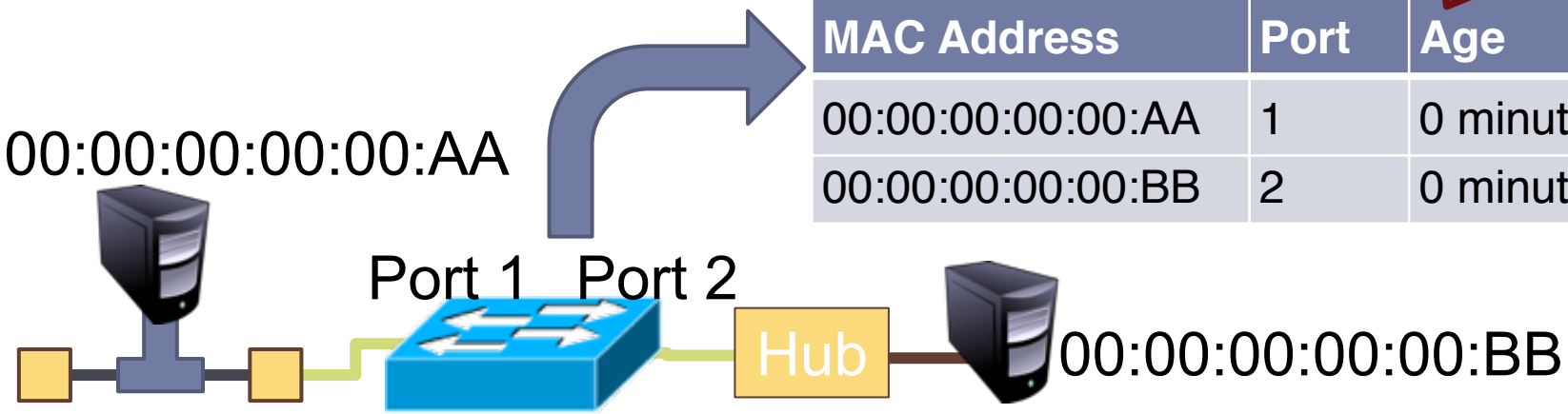
- Manual configuration is possible, but…
  - Time consuming
  - Error prone
  - Not flexible (add/ remove hosts)

Instead, learn addresses using a simple heuristic

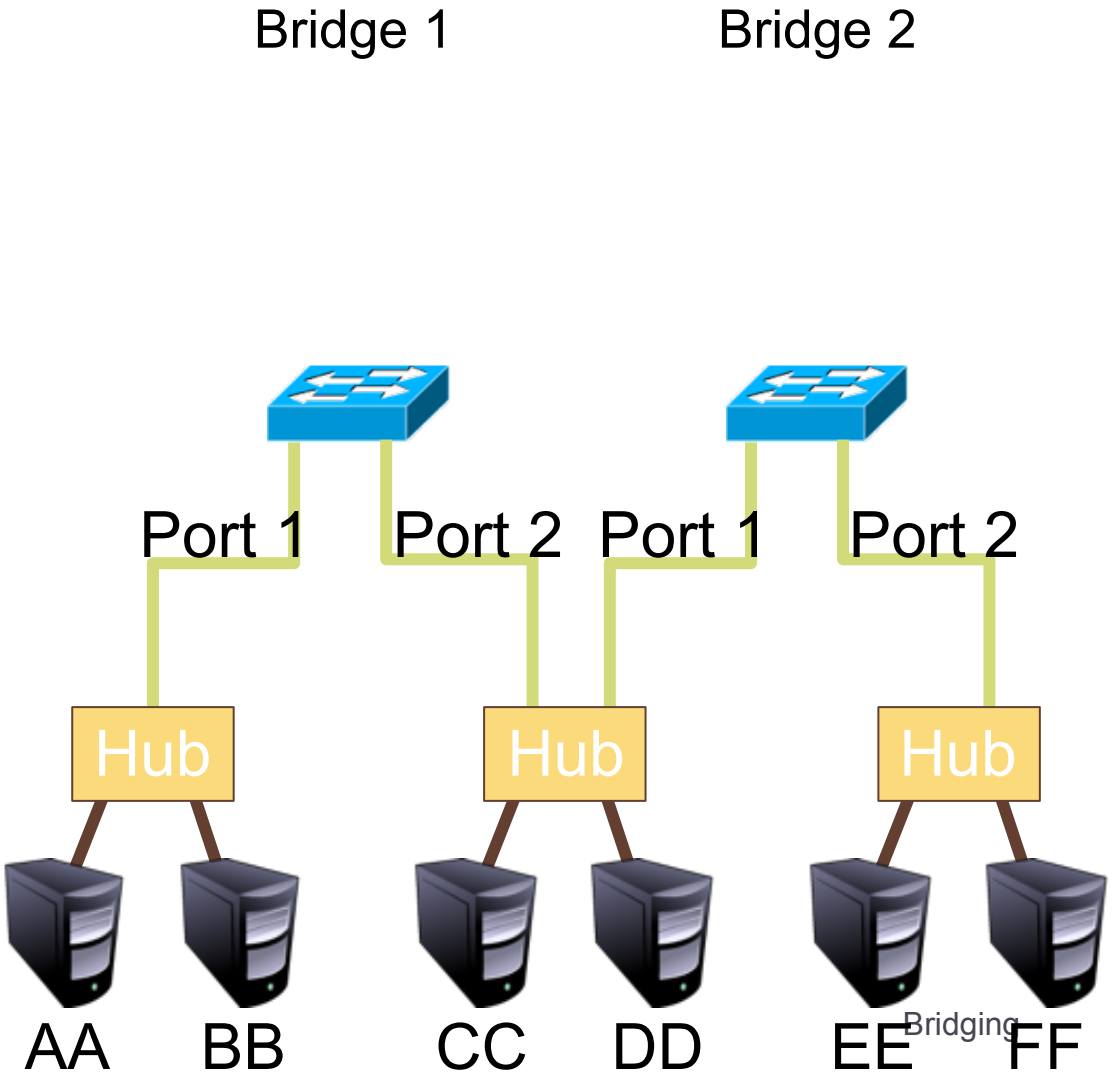
- Look at the source of frames that arrive on each port

Delete old entries after a timeout

MAC Address	Port	Age
00:00:00:00:00:AA	1	0 minutes
00:00:00:00:00:BB	2	0 minutes

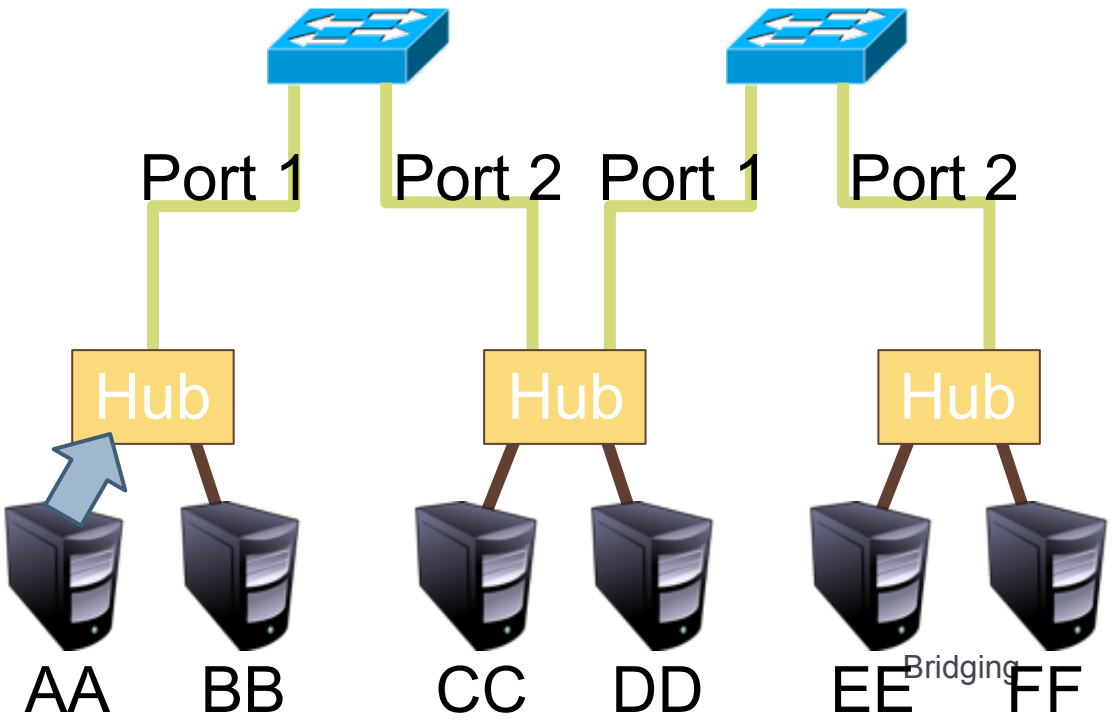


Complicated Learning Example



Complicated Learning Example

- <Src=AA, Dest=FF>



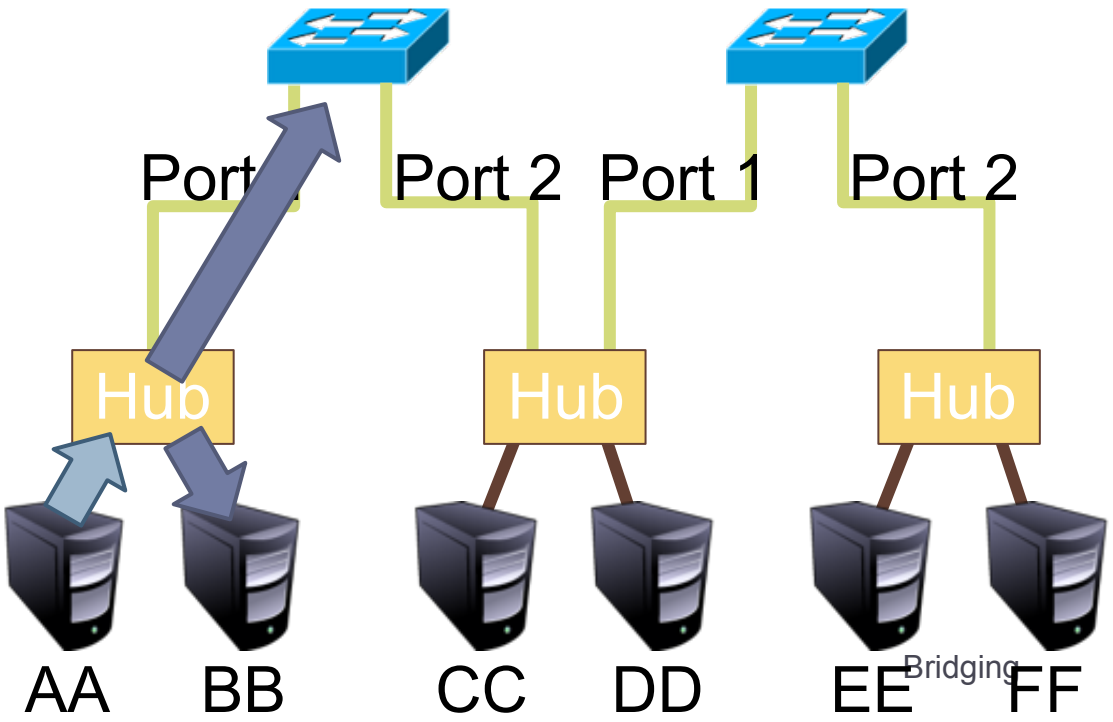
Complicated Learning Example

- <Src=AA, Dest=FF>

Bridge 1

AA	1

Bridge 2



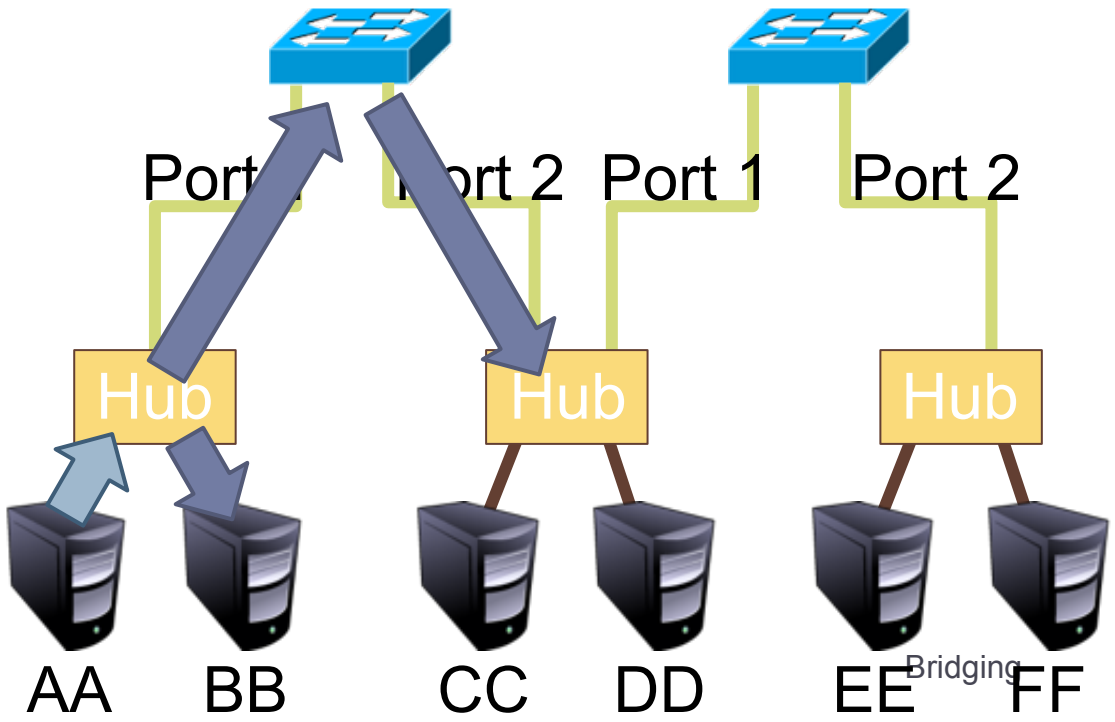
Complicated Learning Example

- <Src=AA, Dest=FF>

Bridge 1

AA	1

Bridge 2



Complicated Learning Example

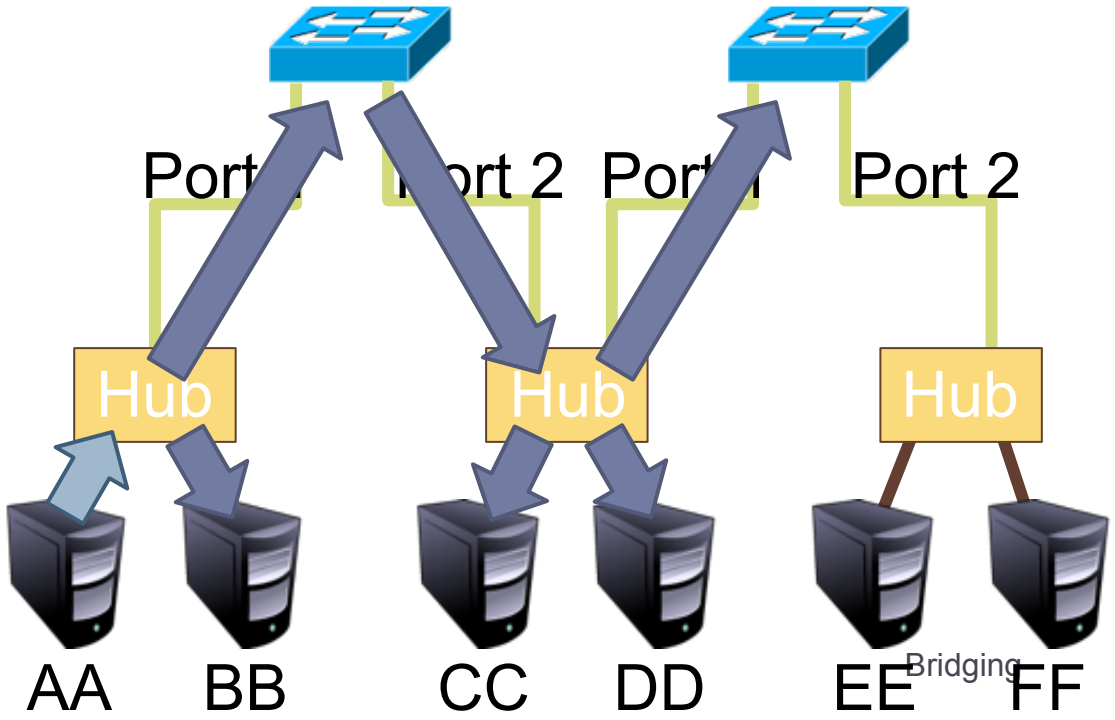
- <Src=AA, Dest=FF>

Bridge 1

AA	1

Bridge 2

AA	1



Complicated Learning Example

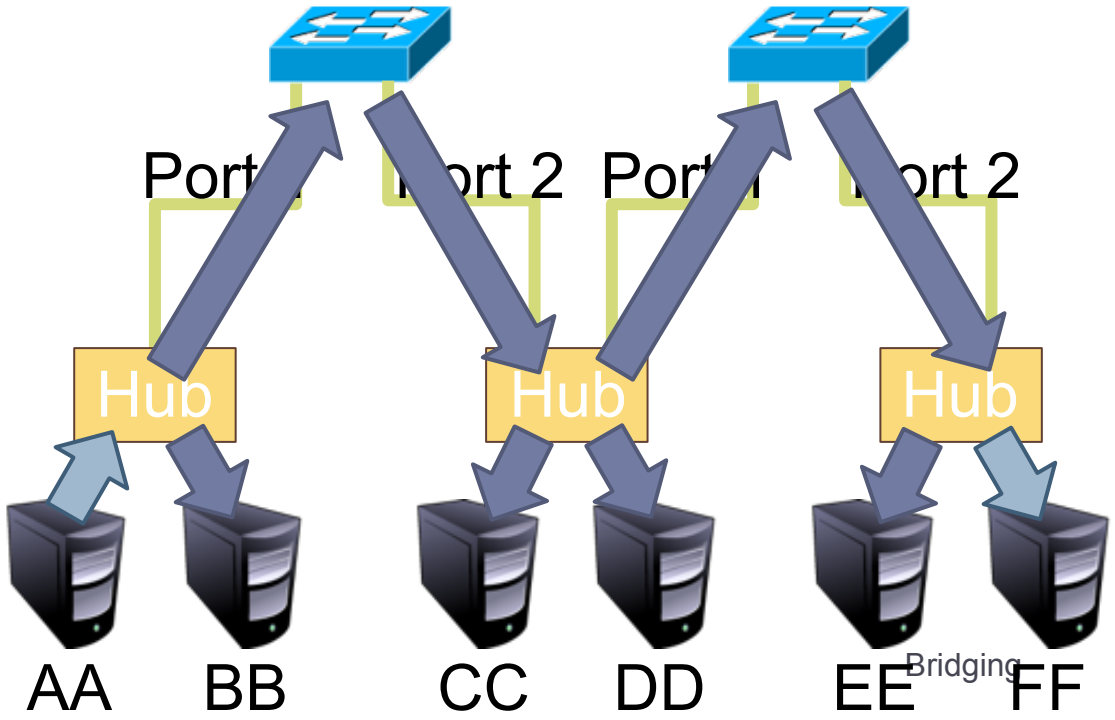
- <Src=AA, Dest=FF>

Bridge 1

AA	1

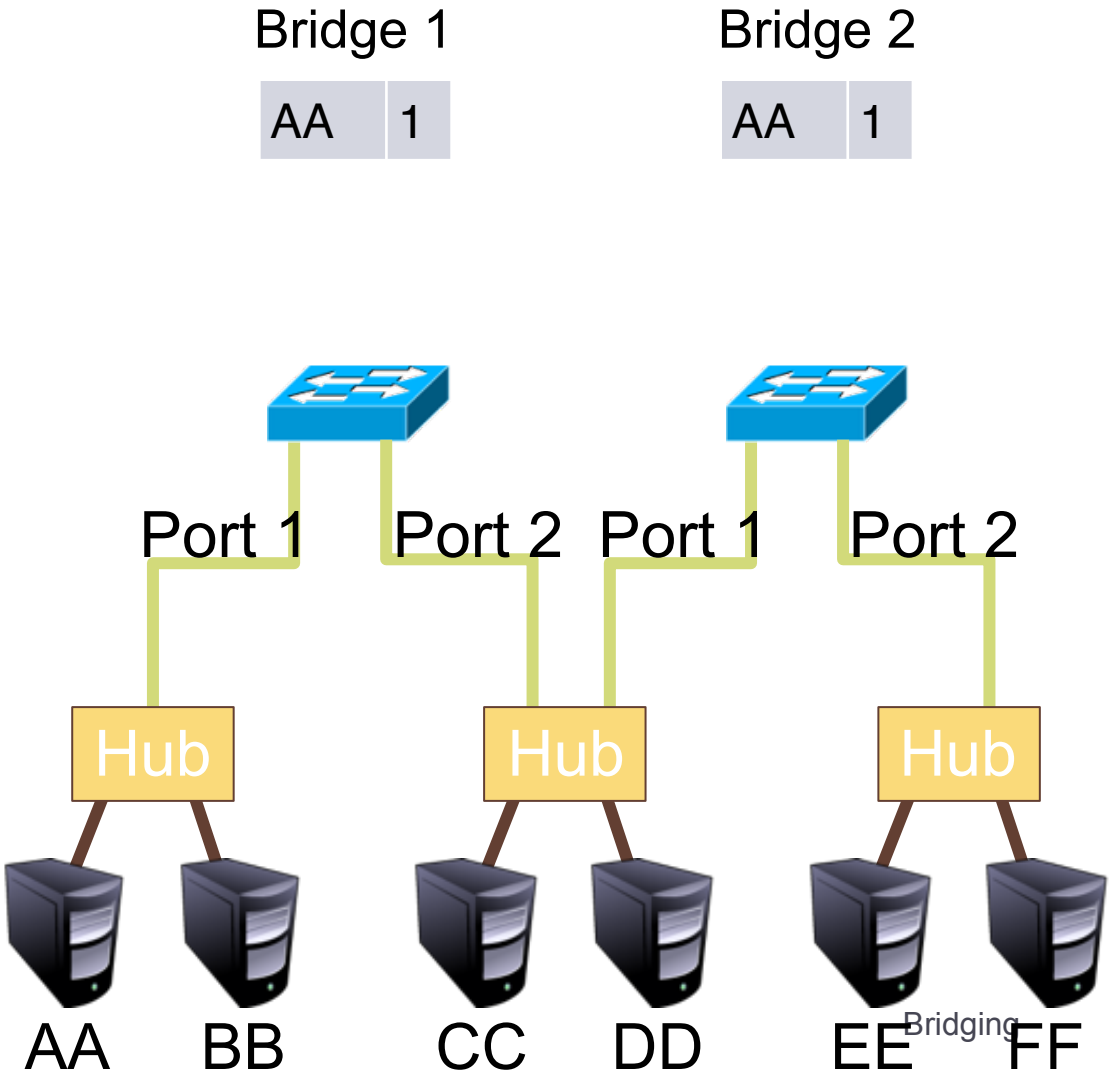
Bridge 2

AA	1



Complicated Learning Example

- <Src=AA, Dest=FF>



Complicated Learning Example

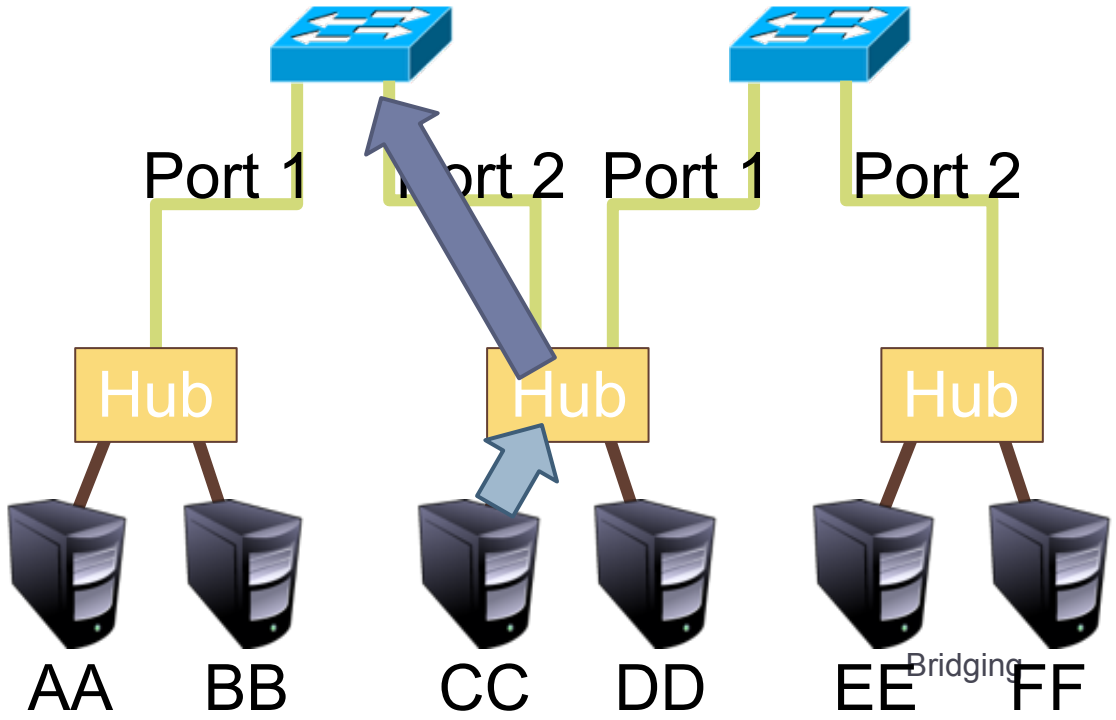
- <Src=AA, Dest=FF>
- <Src=CC, Dest=AA>

Bridge 1

AA	1
CC	2

Bridge 2

AA	1
CC	1



Complicated Learning Example

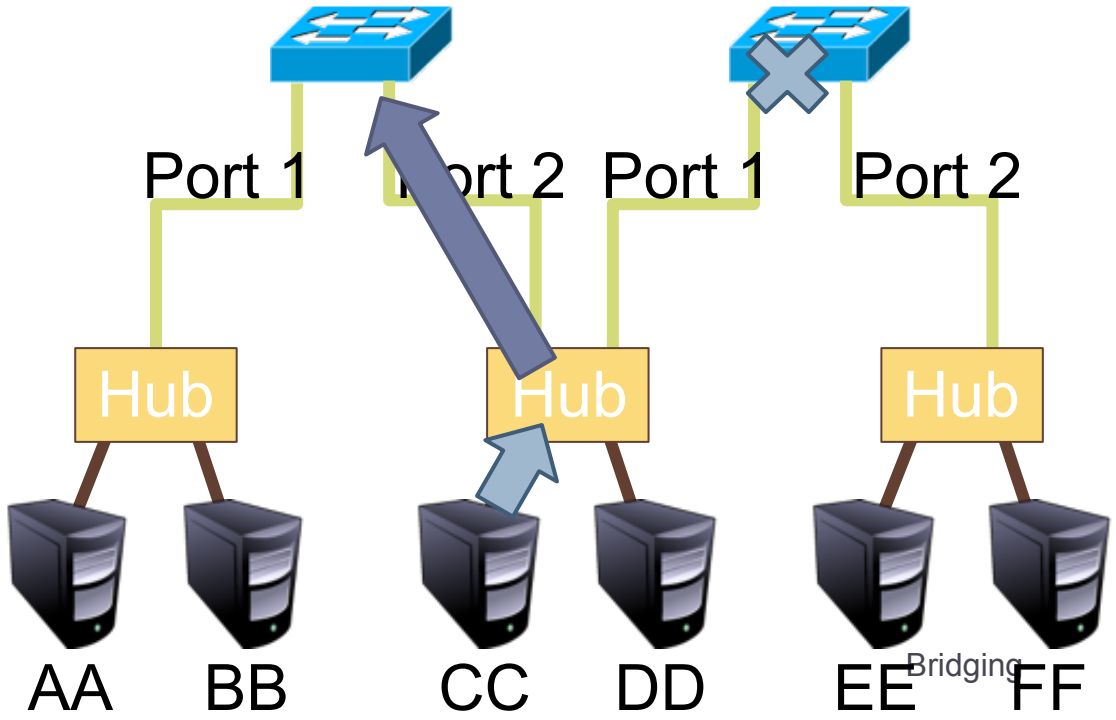
- <Src=AA, Dest=FF>
- <Src=CC, Dest=AA>

Bridge 1

AA	1
CC	2

Bridge 2

AA	1
CC	1



Complicated Learning Example

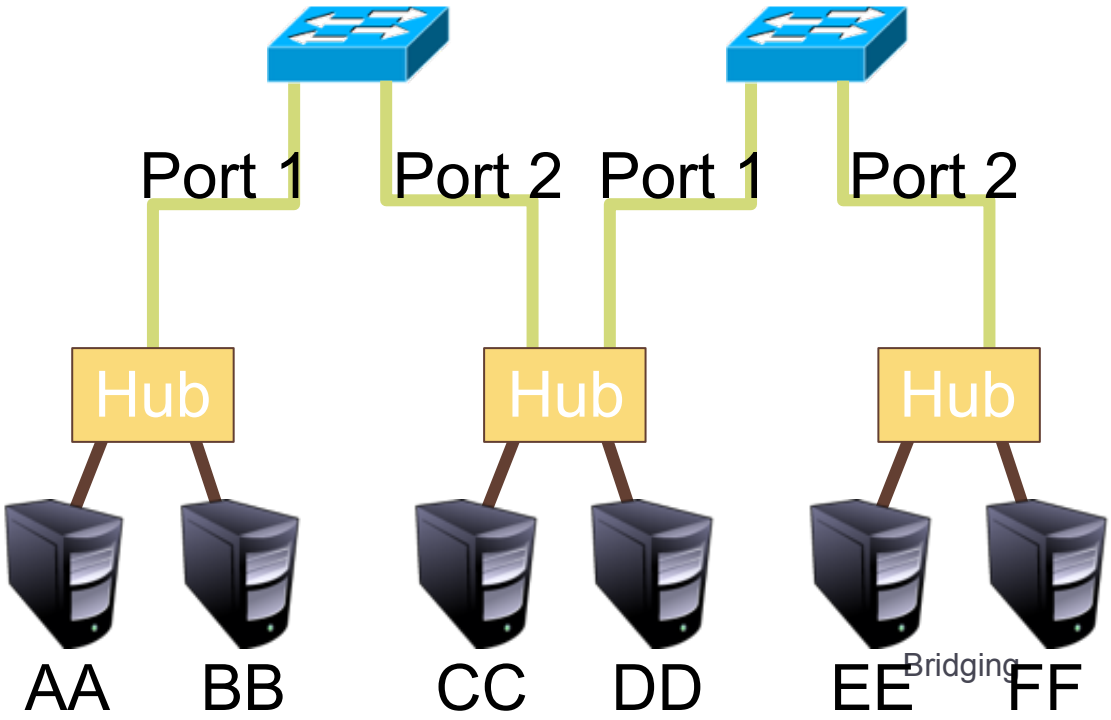
- <Src=AA, Dest=FF>
- <Src=CC, Dest=AA>

Bridge 1

AA	1
CC	2

Bridge 2

AA	1
CC	1



Complicated Learning Example

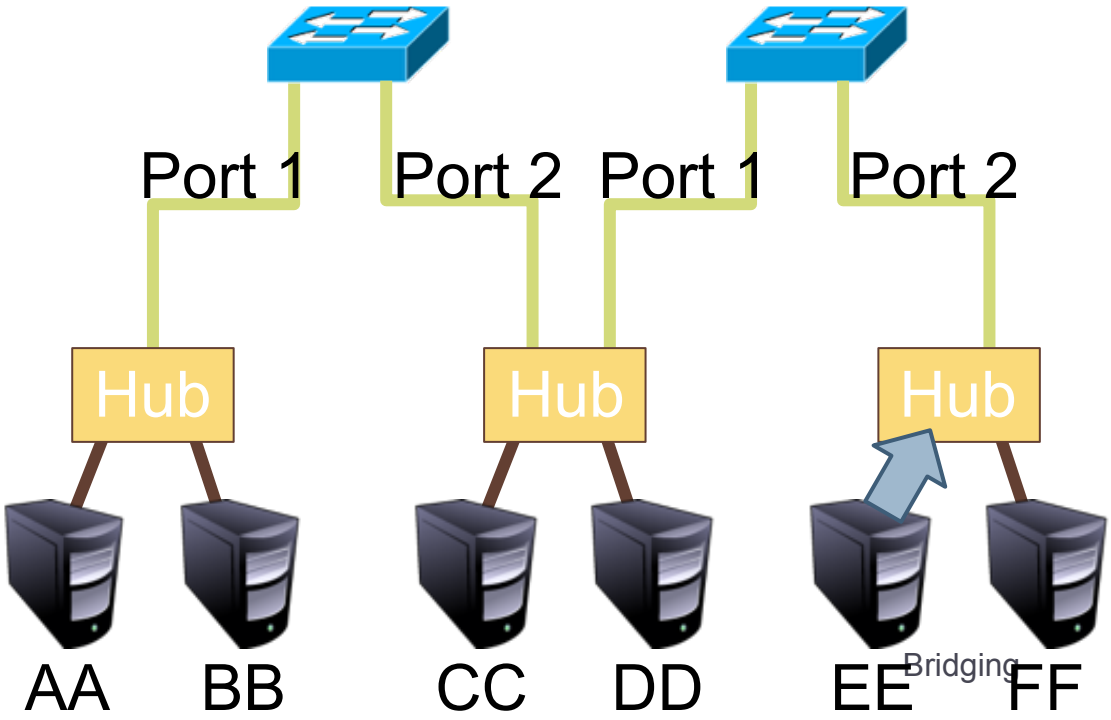
- <Src=AA, Dest=FF>
- <Src=CC, Dest=AA>
- <Src=EE, Dest=CC>

Bridge 1

AA	1
CC	2

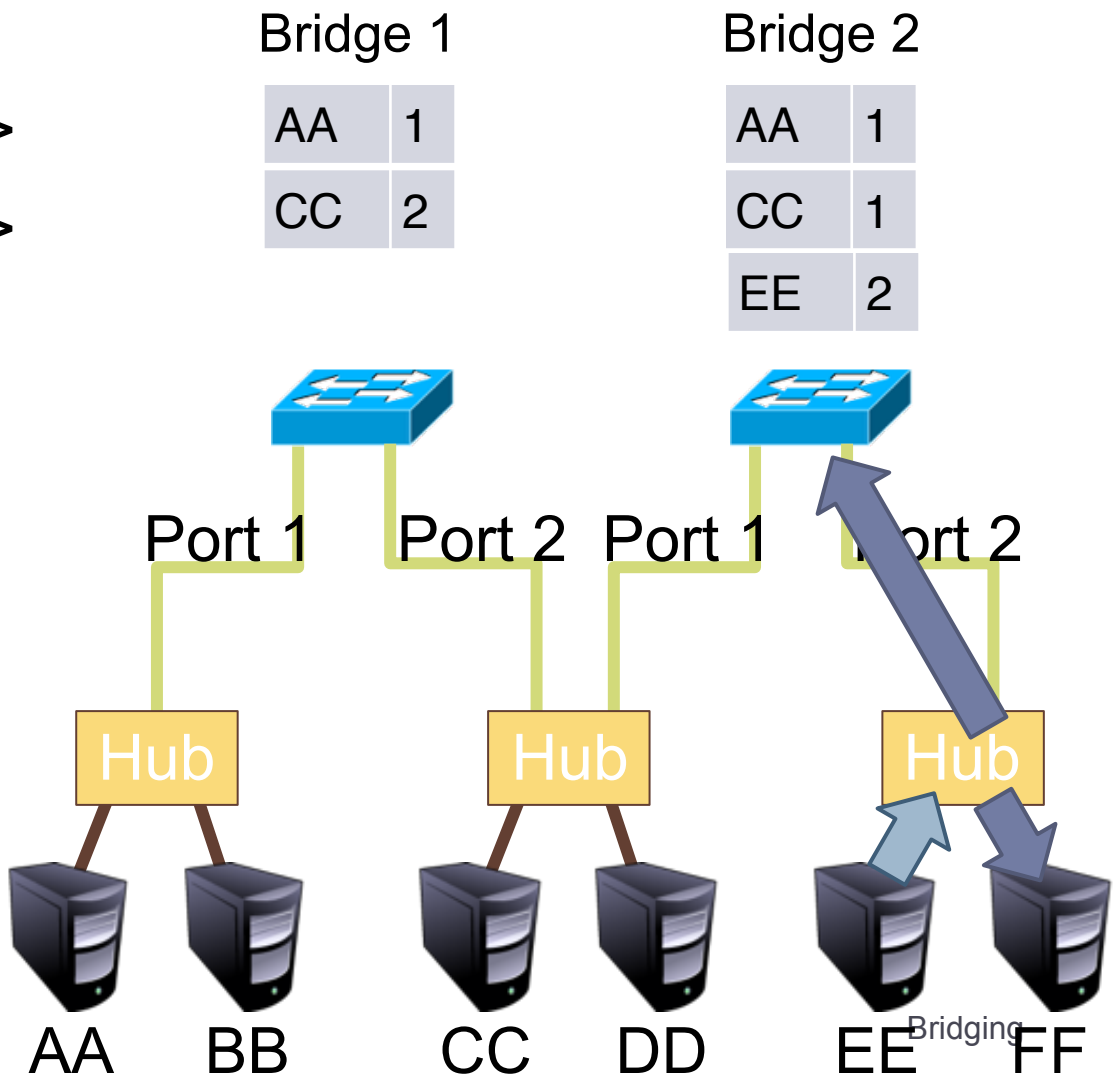
Bridge 2

AA	1
CC	1



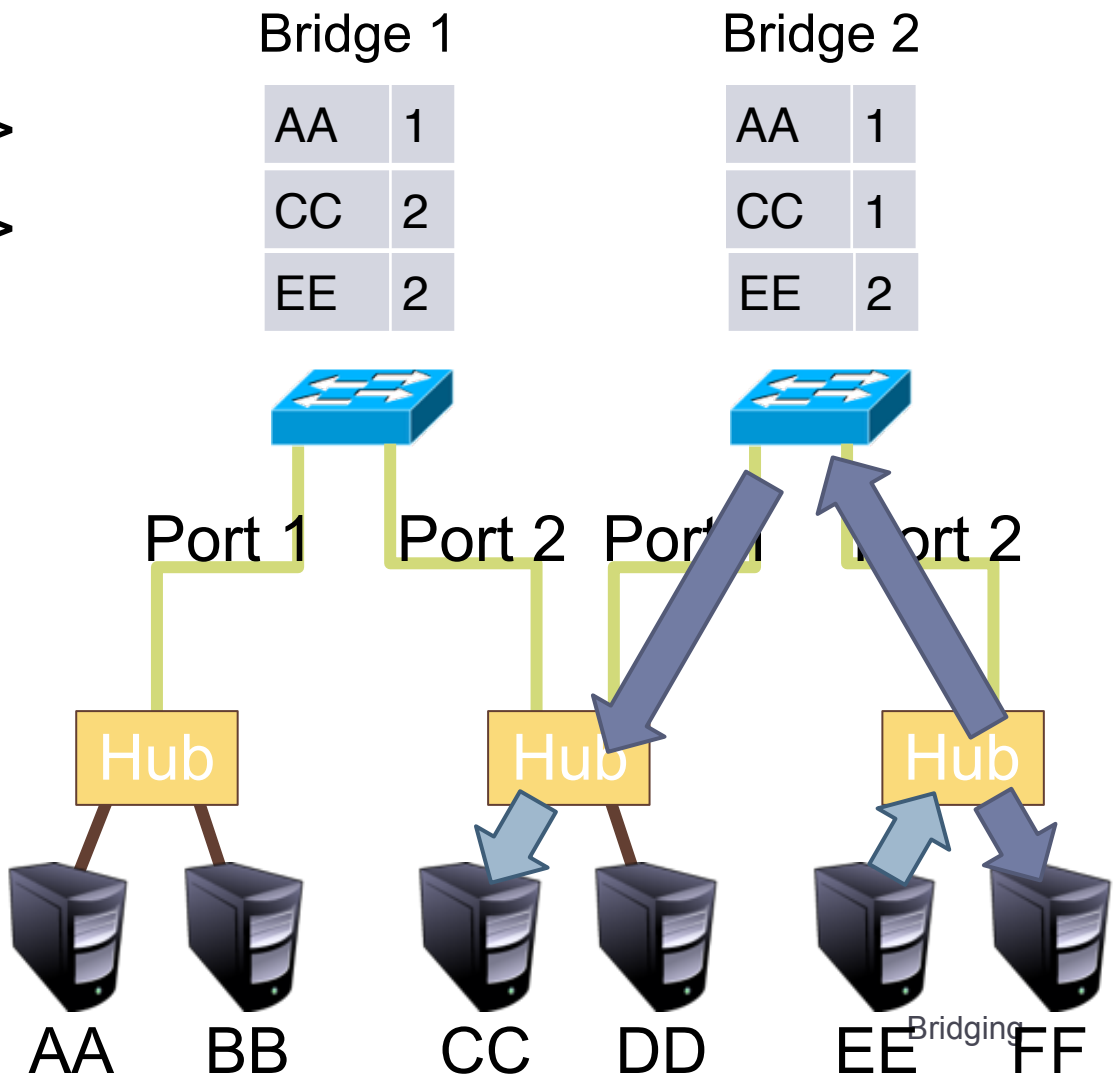
Complicated Learning Example

- <Src=AA, Dest=FF>
- <Src=CC, Dest=AA>
- <Src=EE, Dest=CC>



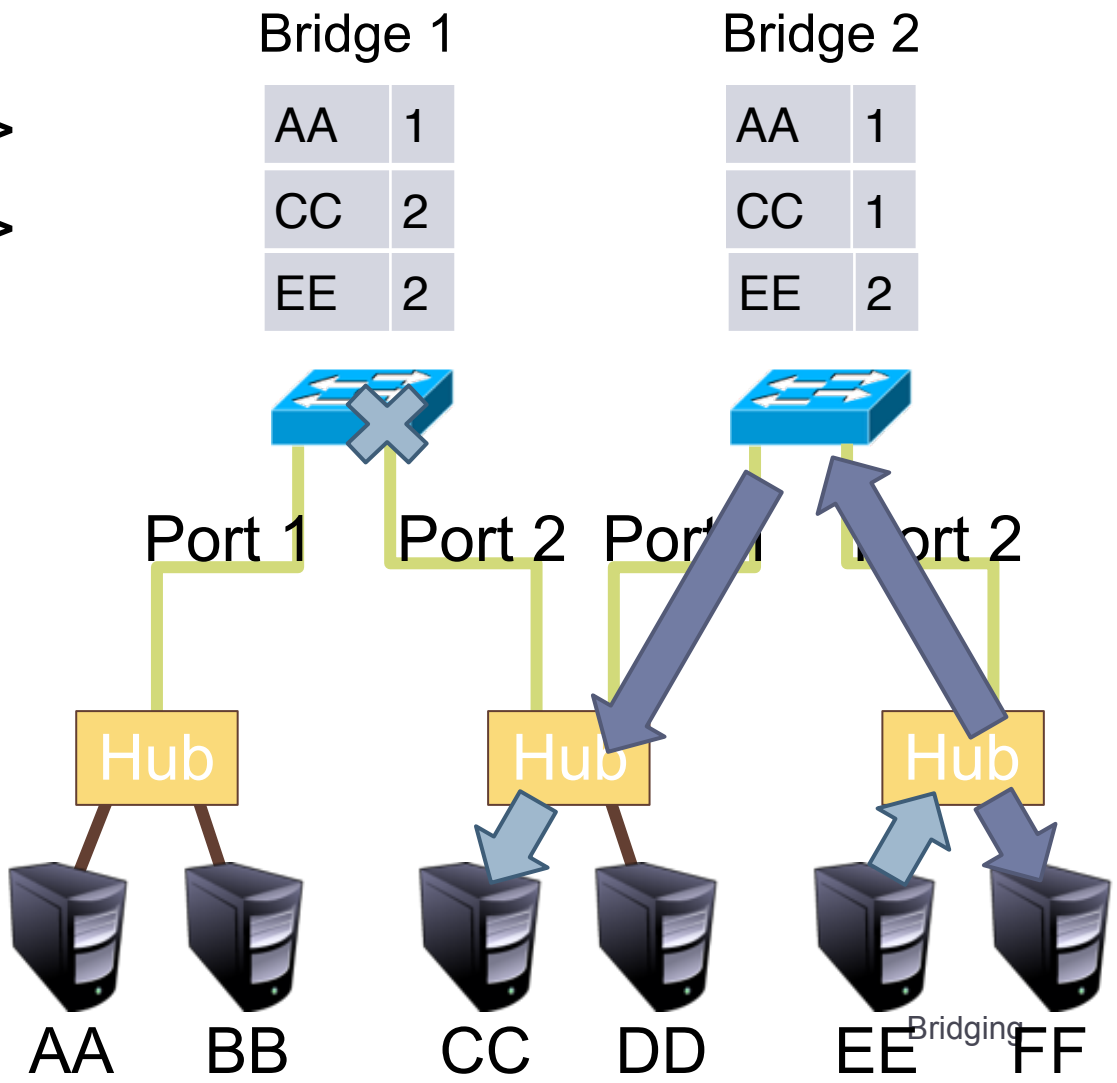
Complicated Learning Example

- <Src=AA, Dest=FF>
- <Src=CC, Dest=AA>
- <Src=EE, Dest=CC>



Complicated Learning Example

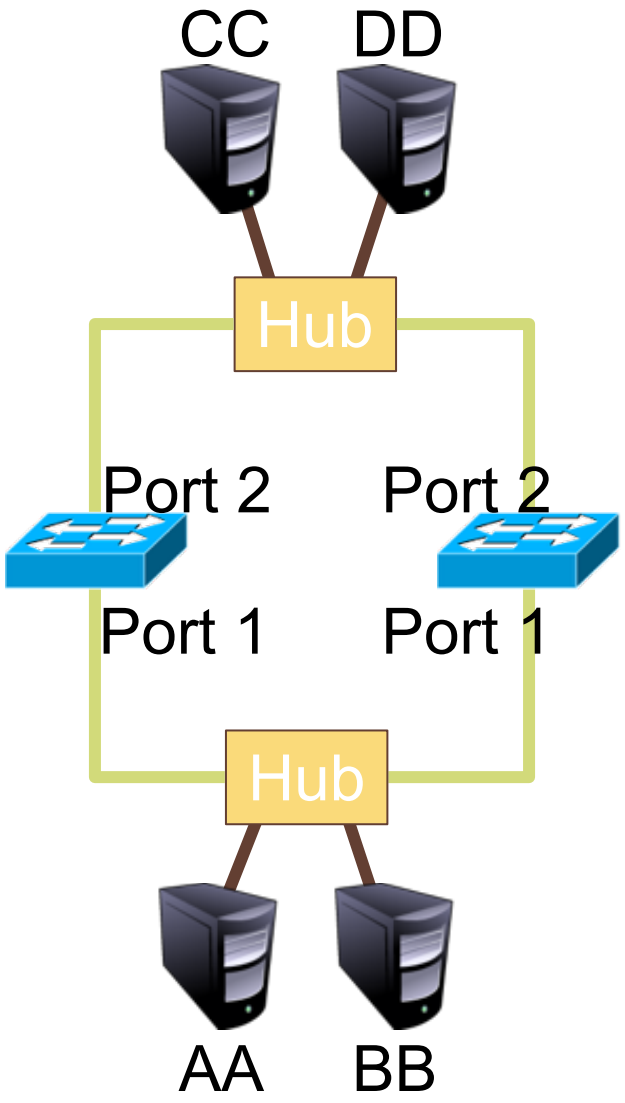
- <Src=AA, Dest=FF>
- <Src=CC, Dest=AA>
- <Src=EE, Dest=CC>



3: Spanning tree.

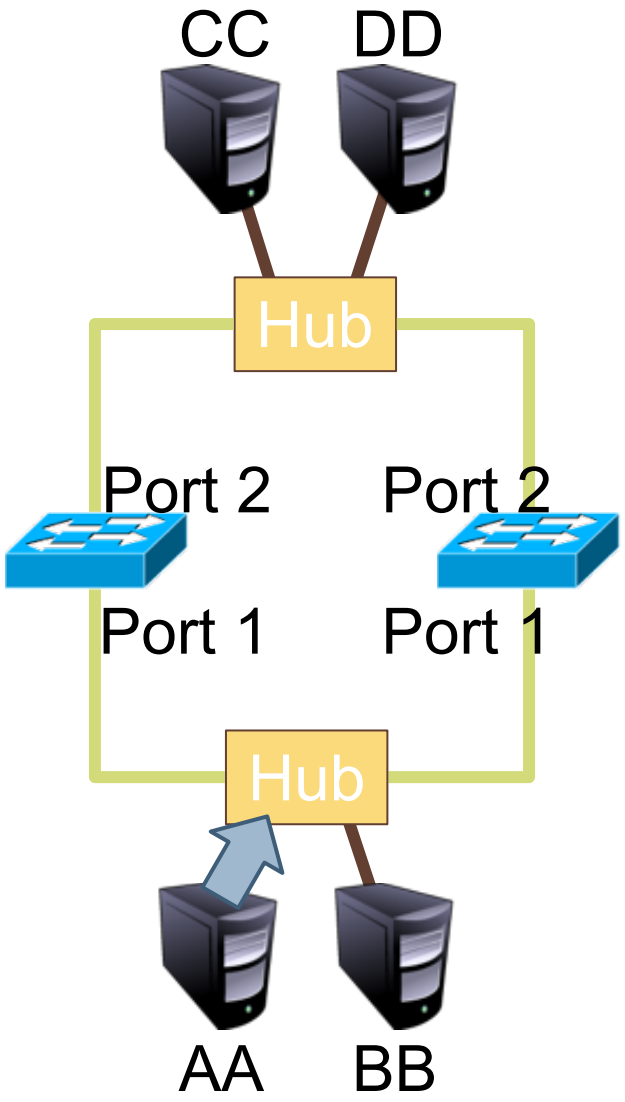
The Danger of Loops

- <Src=AA, Dest=DD>
- This continues to infinity
  - How do we stop this?
- Remove loops from the topology
  - Without physically unplugging cables
- 802.1 uses an algorithm to build and maintain a spanning tree for routing



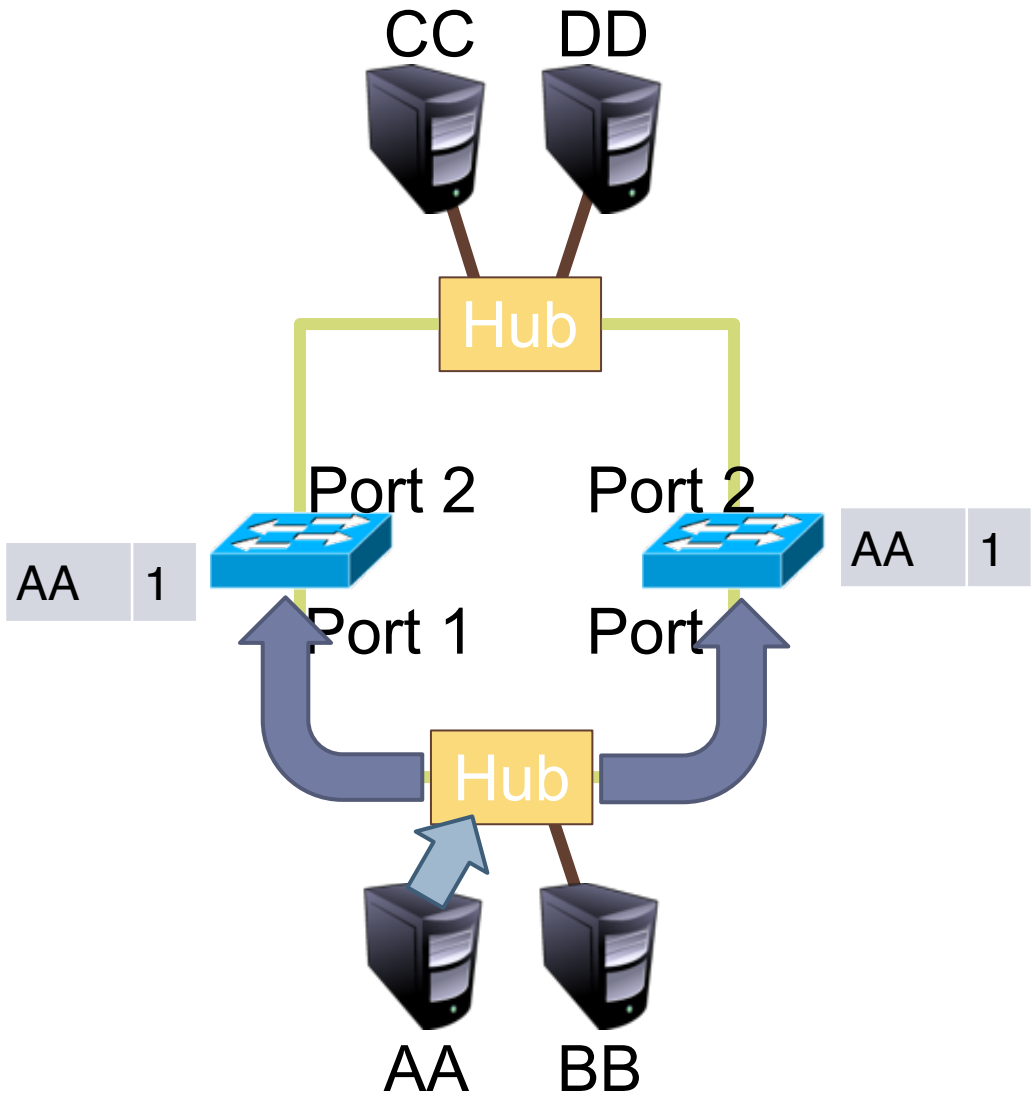
The Danger of Loops

- <Src=AA, Dest=DD>
- This continues to infinity
  - How do we stop this?
- Remove loops from the topology
  - Without physically unplugging cables
- 802.1 uses an algorithm to build and maintain a spanning tree for routing



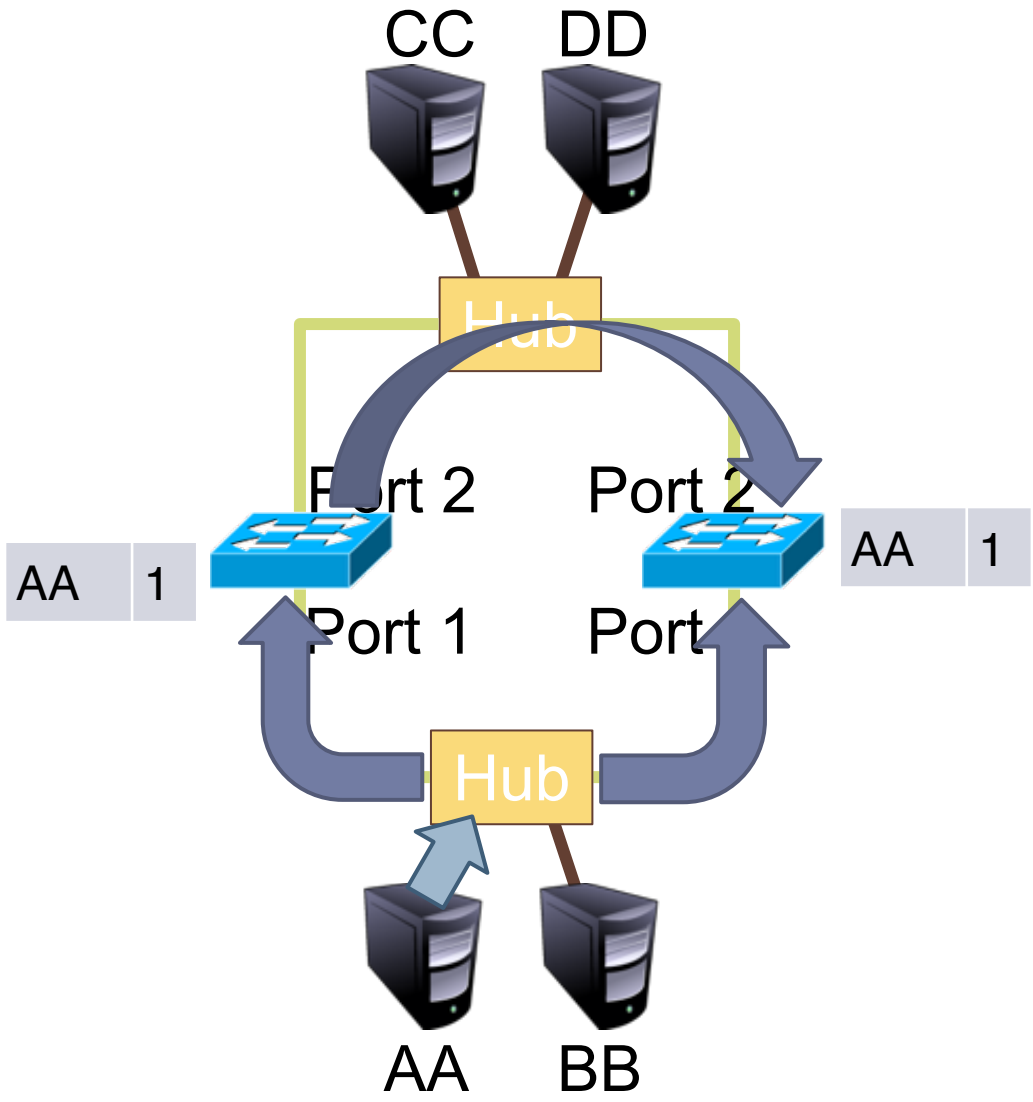
The Danger of Loops

- <Src=AA, Dest=DD>
- This continues to infinity
  - How do we stop this?
- Remove loops from the topology
  - Without physically unplugging cables
- 802.1 uses an algorithm to build and maintain a spanning tree for routing



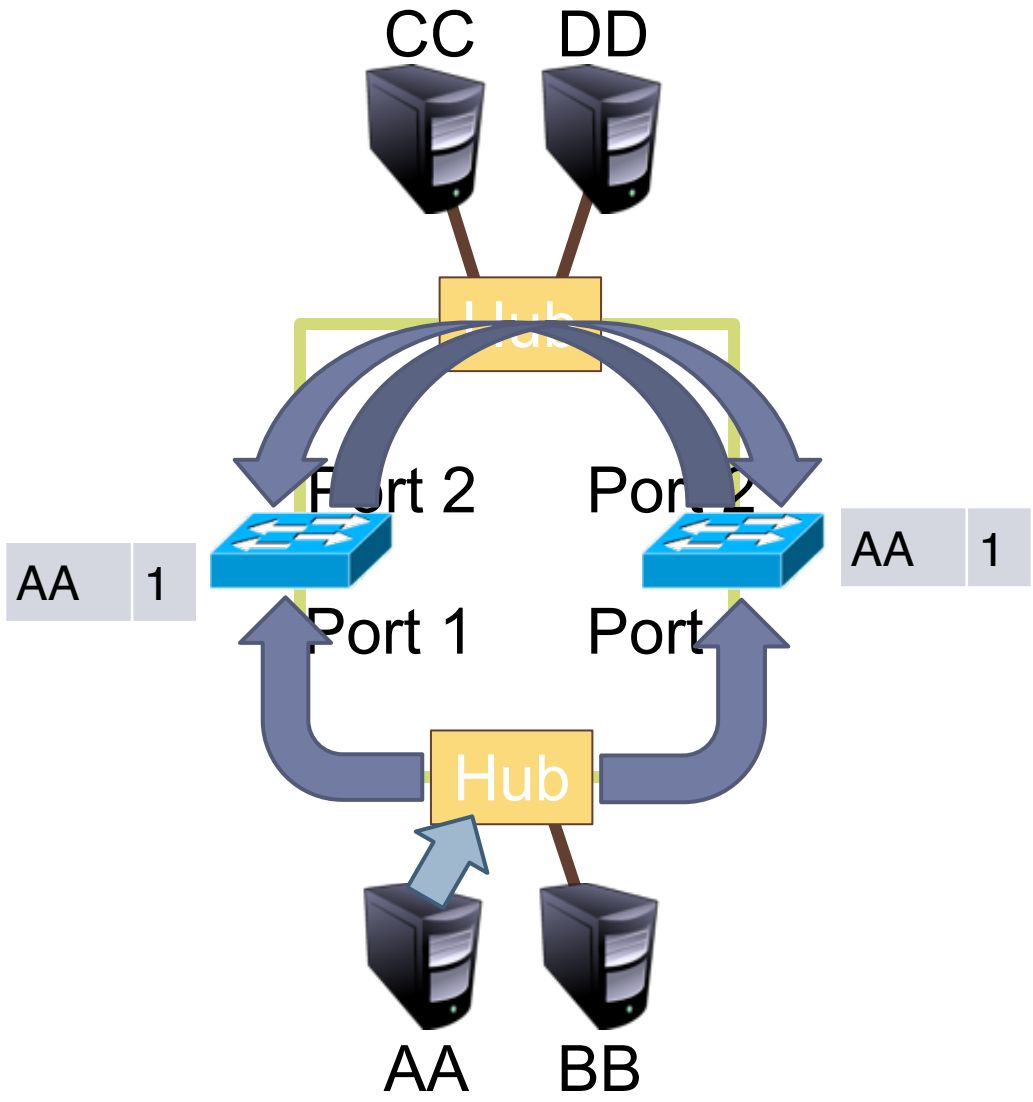
The Danger of Loops

- <Src=AA, Dest=DD>
- This continues to infinity
  - How do we stop this?
- Remove loops from the topology
  - Without physically unplugging cables
- 802.1 uses an algorithm to build and maintain a spanning tree for routing



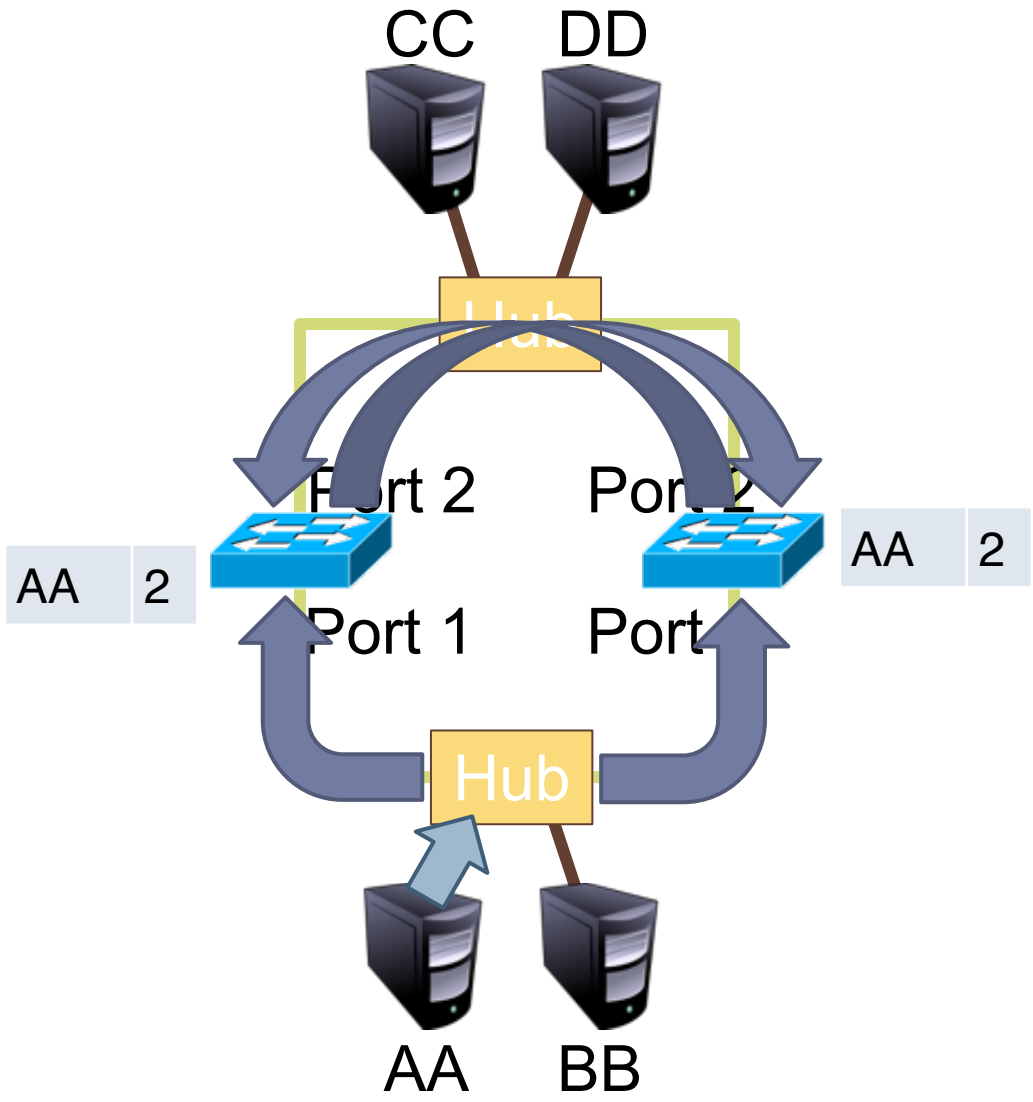
The Danger of Loops

- <Src=AA, Dest=DD>
- This continues to infinity
  - How do we stop this?
- Remove loops from the topology
  - Without physically unplugging cables
- 802.1 uses an algorithm to build and maintain a spanning tree for routing



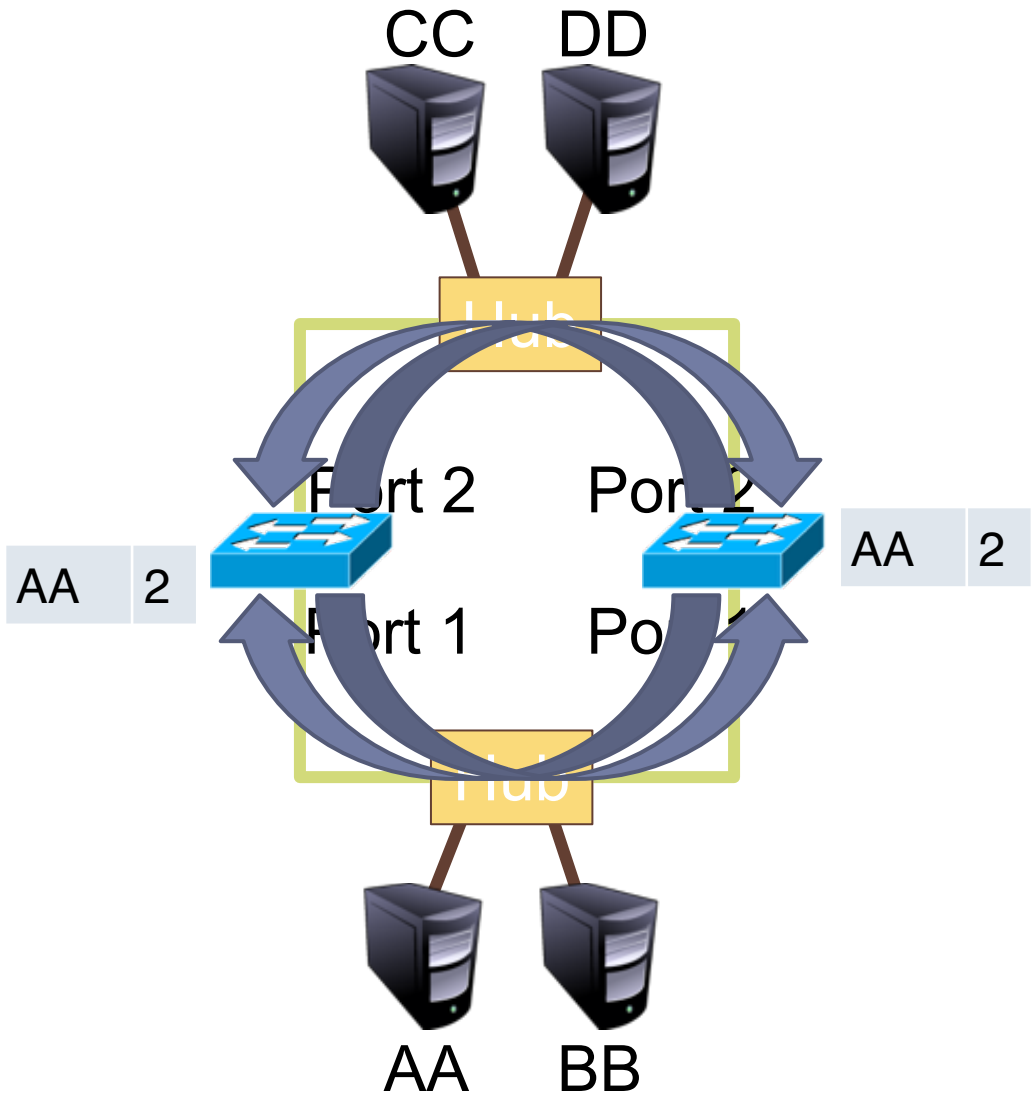
The Danger of Loops

- <Src=AA, Dest=DD>
- This continues to infinity
  - How do we stop this?
- Remove loops from the topology
  - Without physically unplugging cables
- 802.1 uses an algorithm to build and maintain a spanning tree for routing



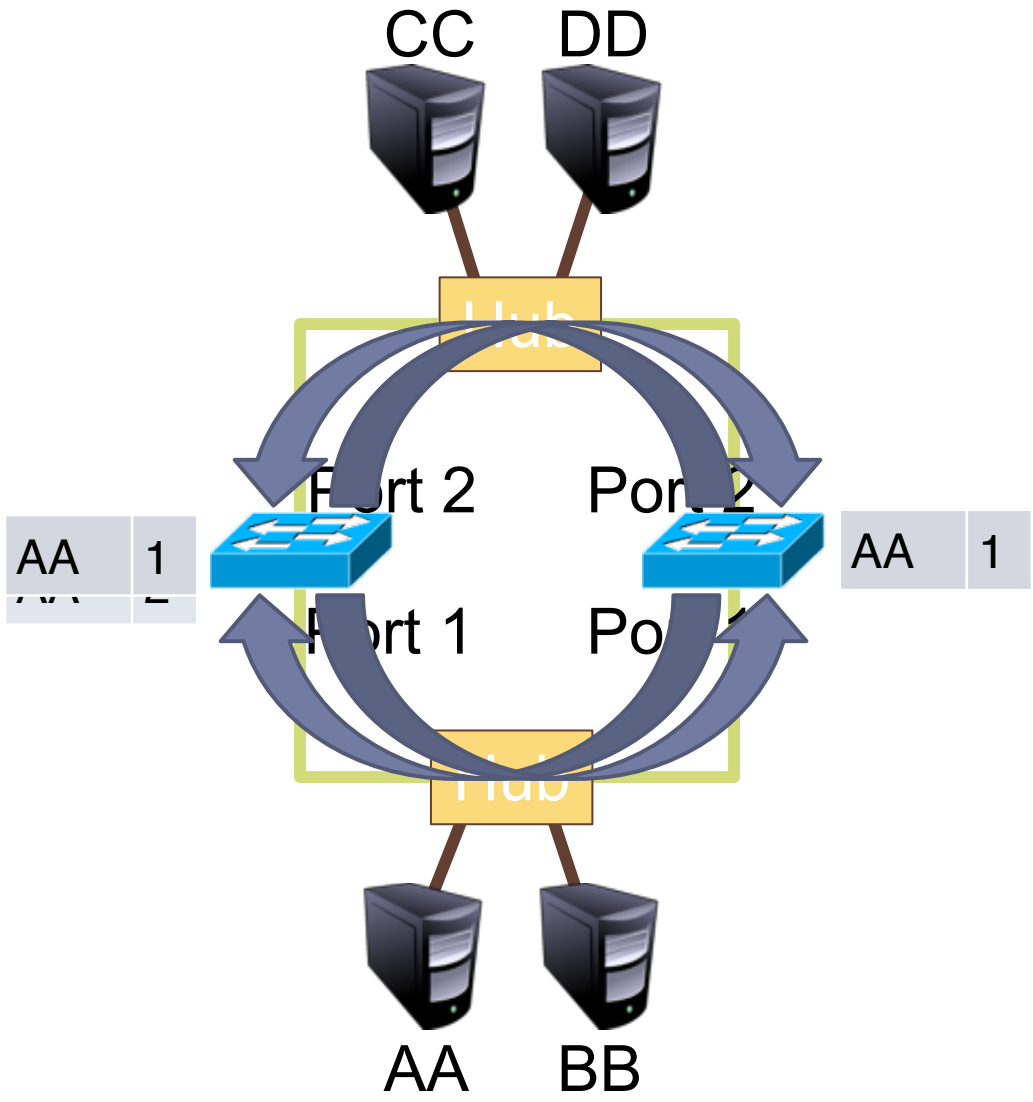
The Danger of Loops

- <Src=AA, Dest=DD>
- This continues to infinity
  - How do we stop this?
- Remove loops from the topology
  - Without physically unplugging cables
- 802.1 uses an algorithm to build and maintain a spanning tree for routing



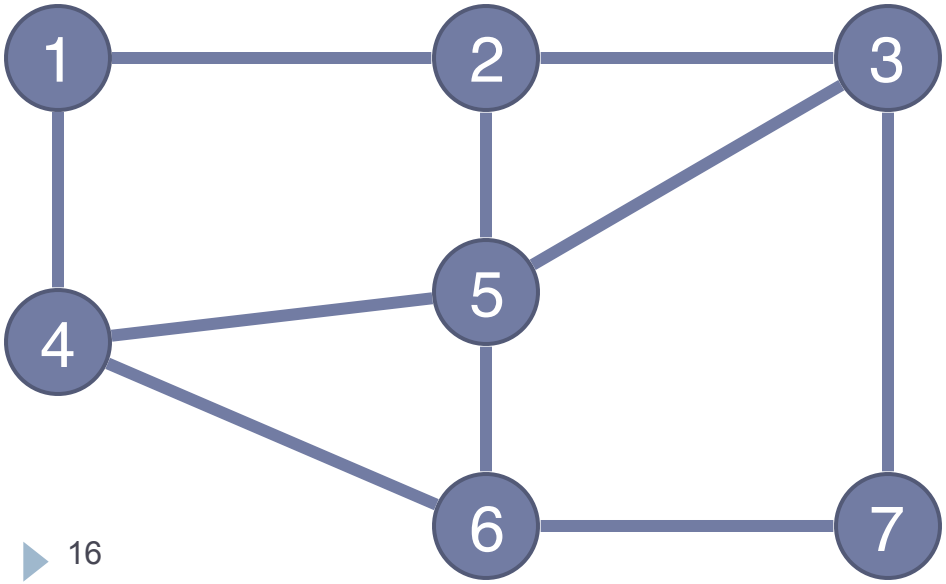
The Danger of Loops

- <Src=AA, Dest=DD>
- This continues to infinity
  - How do we stop this?
- Remove loops from the topology
  - Without physically unplugging cables
- 802.1 uses an algorithm to build and maintain a spanning tree for routing



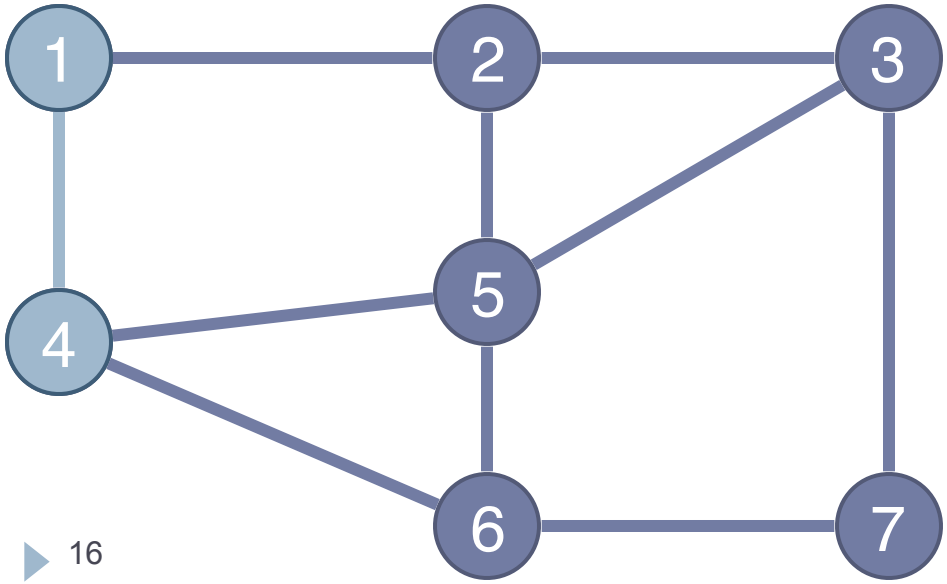
Spanning Tree Definition

- A subset of edges in a graph that:
  - Span all nodes
  - Do not create any cycles
- This structure is a tree



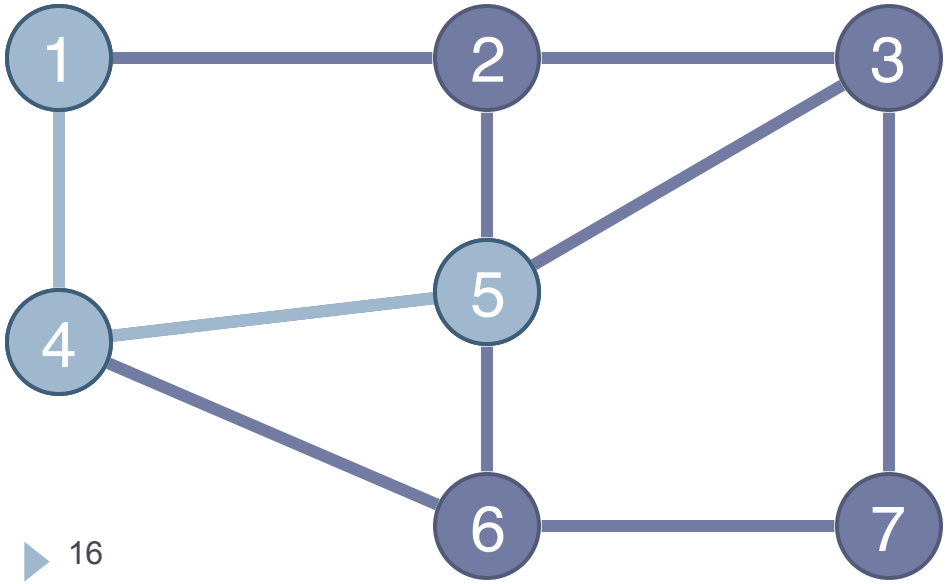
Spanning Tree Definition

- A subset of edges in a graph that:
  - Span all nodes
  - Do not create any cycles
- This structure is a tree



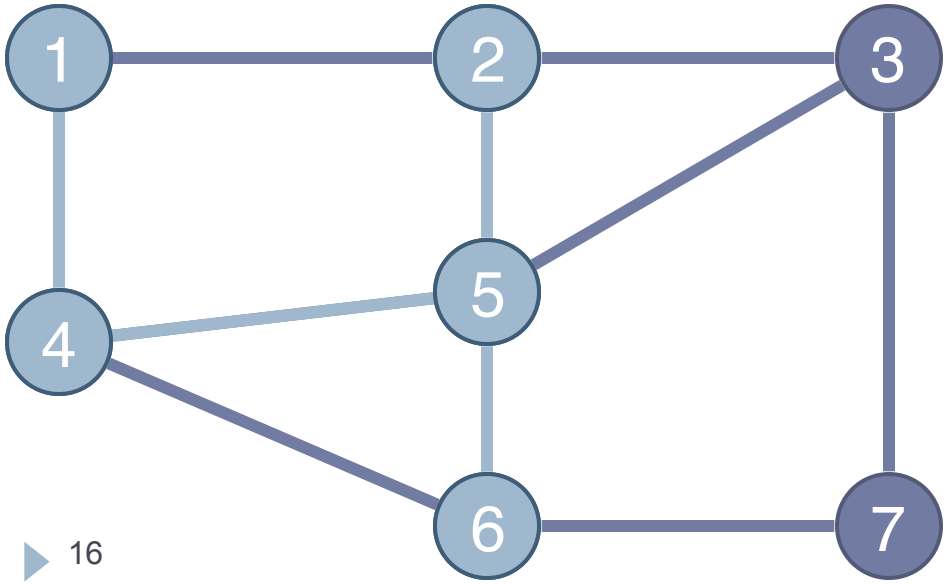
Spanning Tree Definition

- A subset of edges in a graph that:
  - Span all nodes
  - Do not create any cycles
- This structure is a tree



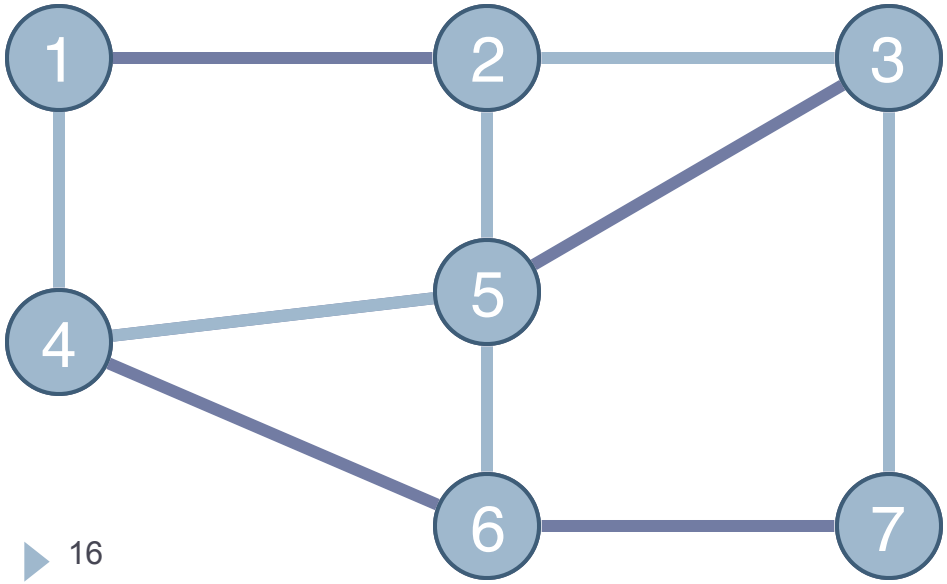
Spanning Tree Definition

- A subset of edges in a graph that:
  - Span all nodes
  - Do not create any cycles
- This structure is a tree



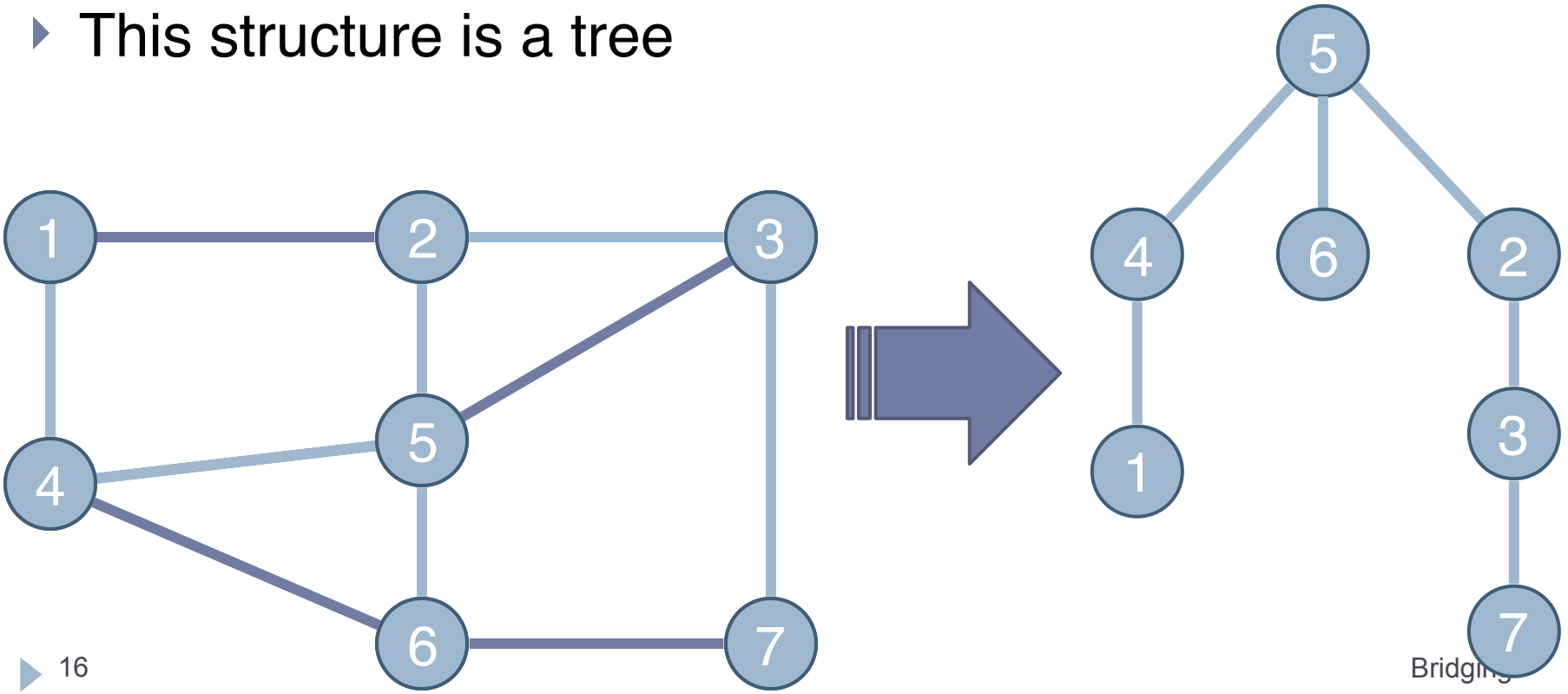
Spanning Tree Definition

- A subset of edges in a graph that:
  - Span all nodes
  - Do not create any cycles
- This structure is a tree



Spanning Tree Definition

- A subset of edges in a graph that:
  - Span all nodes
  - Do not create any cycles
- This structure is a tree



Spanning Tree Poem

Algorhyme

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 packet 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.

Radia Perlman

802.1 Spanning Tree Approach

1. Elect a bridge to be the root of the tree
2. Every bridge finds shortest path to the root
3. Union of these paths becomes the spanning tree

- Bridges exchange Configuration Bridge Protocol Data Units (BPDUs) to build the tree
  - Used to elect the root bridge
  - Calculate shortest paths
  - Locate the next hop closest to the root, and its port
  - Select ports to be included in the spanning trees

Definitions

- Bridge ID (BID) = <Random Number>
- Root Bridge: bridge with the lowest BID in the tree
- Path Cost: cost (in hops) from a transmitting bridge to the root
- Each port on a bridge has a unique Port ID
- Root Port: port that forwards to the root on each bridge
- Designated Bridge: the bridge on a LAN that provides the minimal cost path to the root
  - The designated bridge on each LAN is unique

Determining the Root

- Initially, all hosts assume they are the root
- Bridges broadcast BPDUs:

- Based on received BPDUs, each switch chooses:
  - A new root (smallest known Root ID)
  - A new root port (what interface goes towards the root)
  - A new designated bridge (who is the next hop to root)

Comparing BPDUs

if R1 < R2: use BPDU1

else if R1 == R2 and Cost1 < Cost2: use BPDU1

else if R1 == R2 and Cost1 == Cost 2 and B1 < B2:

use BPDU1

else: use BPDU2

Comparing BPDUs

if R1 < R2: use BPDU1

else if R1 == R2 and Cost1 < Cost2: use BPDU1

else if R1 == R2 and Cost1 == Cost 2 and B1 < B2:
use BPDU1

else: use BPDU2

Comparing BPDUs

if R1 < R2: use BPDU1

else if R1 == R2 and Cost1 < Cost2: use BPDU1

else if R1 == R2 and Cost1 == Cost 2 and B1 < B2:
use BPDU1

else: use BPDU2

Comparing BPDUs

if R1 < R2: use BPDU1

else if R1 == R2 and Cost1 < Cost2: use BPDU1

else if R1 == R2 and Cost1 == Cost 2 and B1 < B2:

use BPDU1

else: use BPDU2

Comparing BPDUs

if R1 < R2: use BPDU1

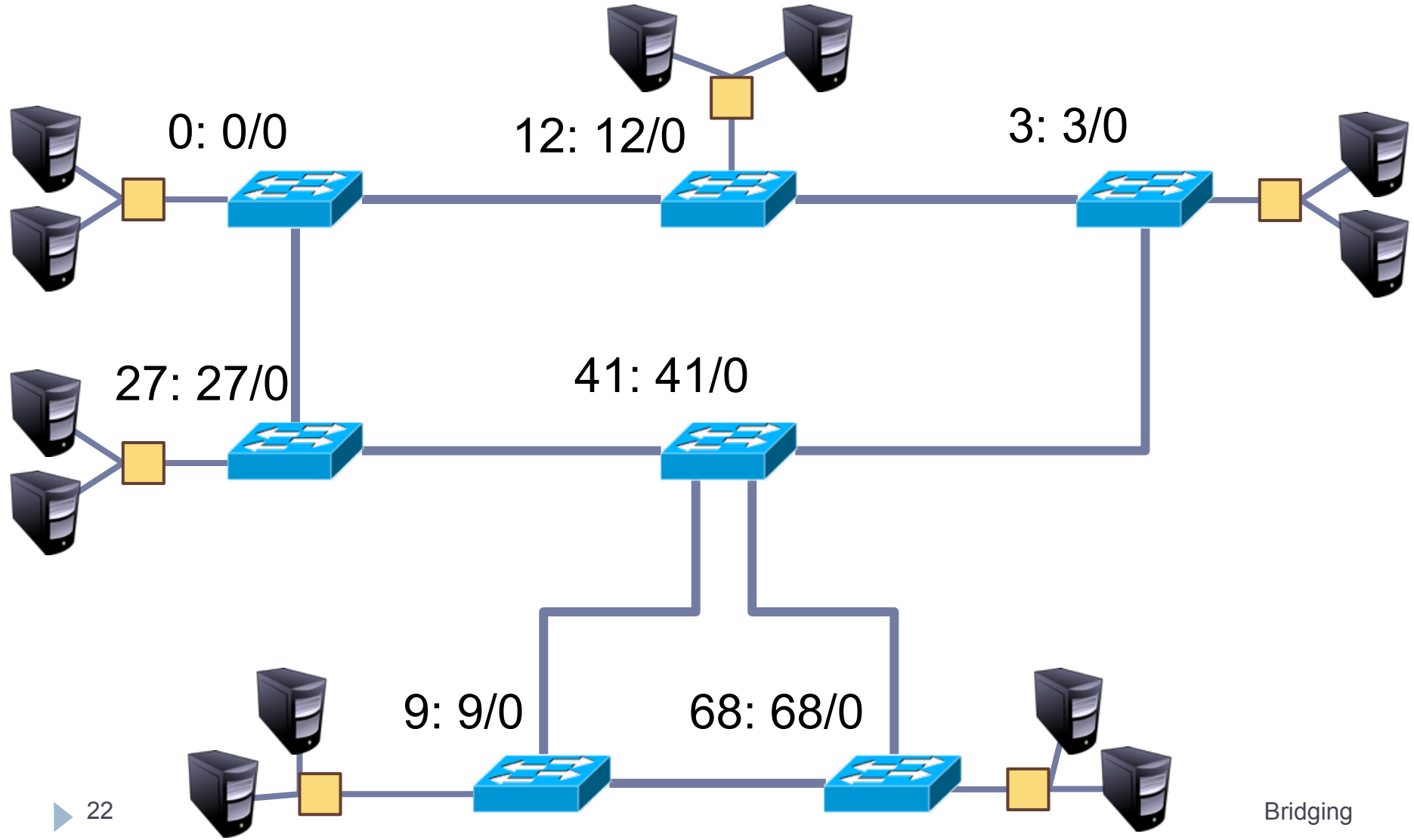
else if R1 == R2 and Cost1 < Cost2: use BPDU1

else if R1 == R2 and Cost1 == Cost 2 and B1 < B2:

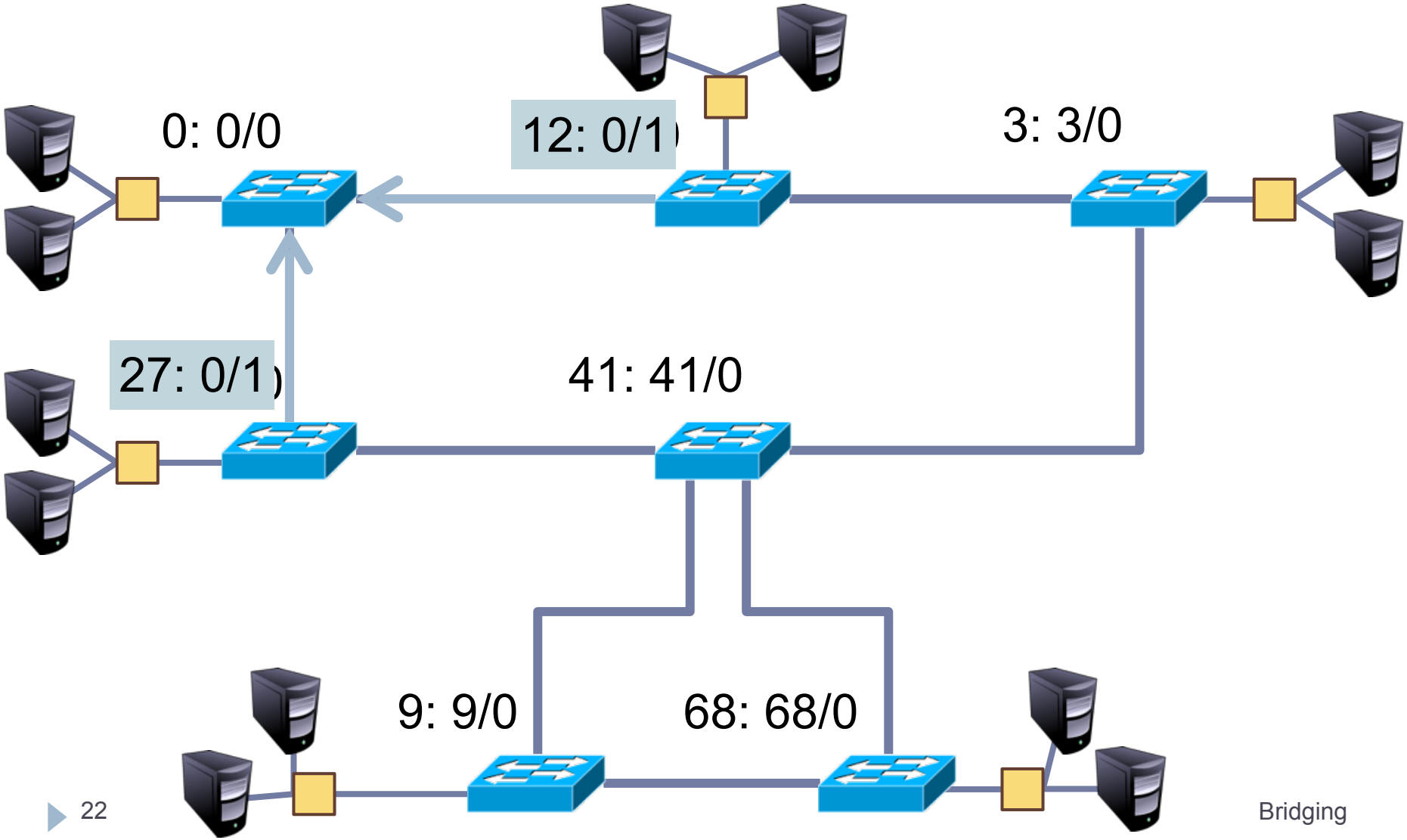
use BPDU1

else: use BPDU2

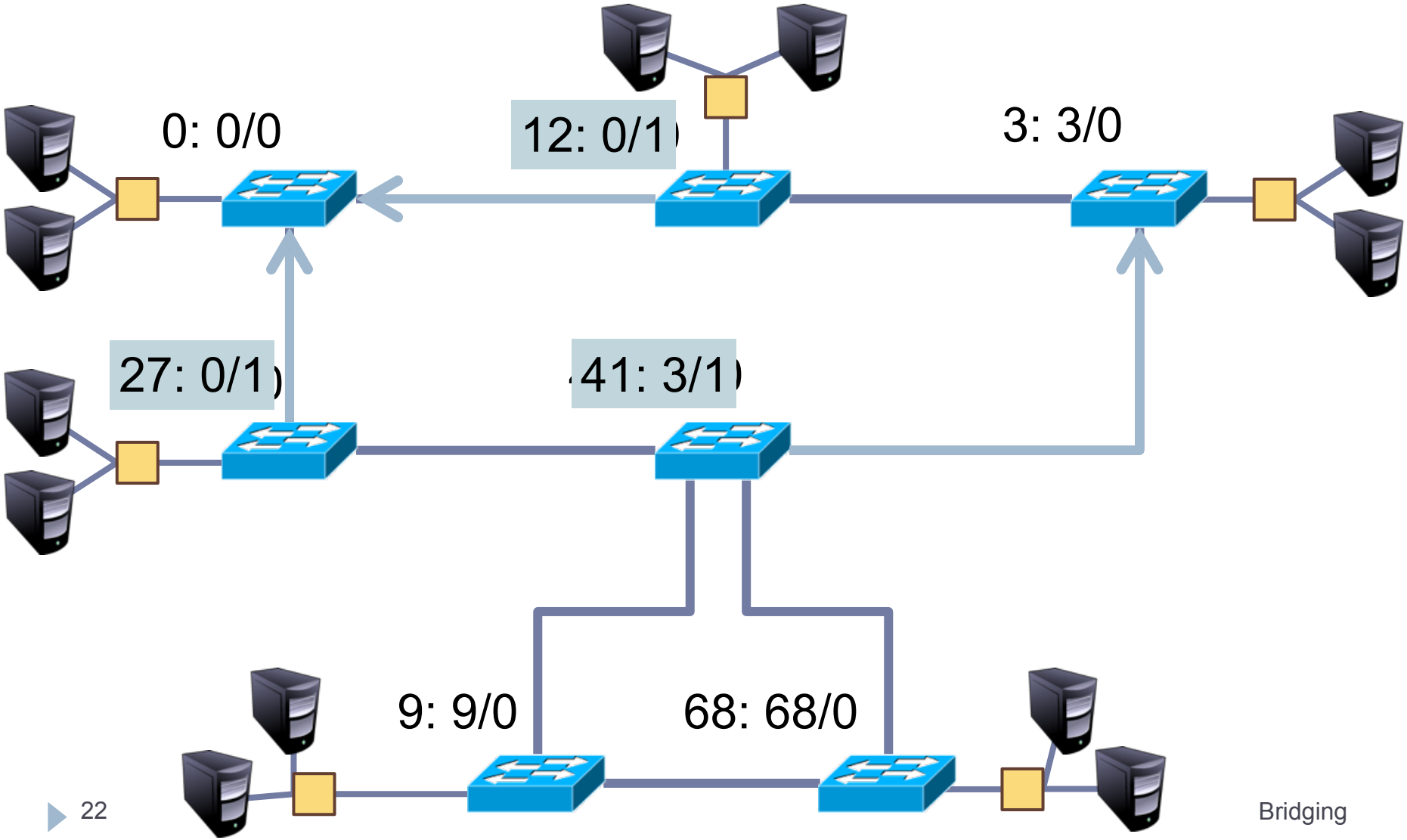
Spanning Tree Construction



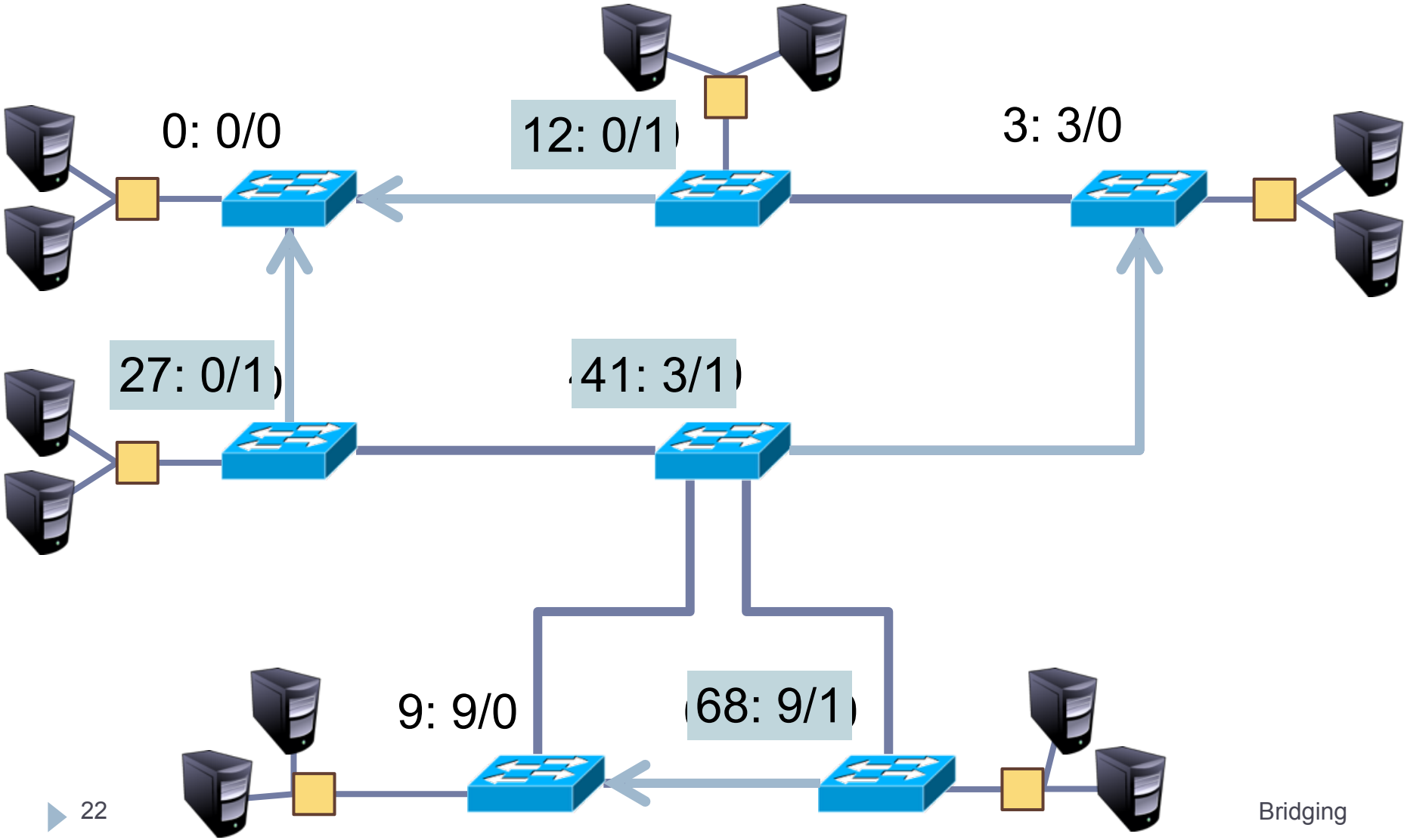
Spanning Tree Construction



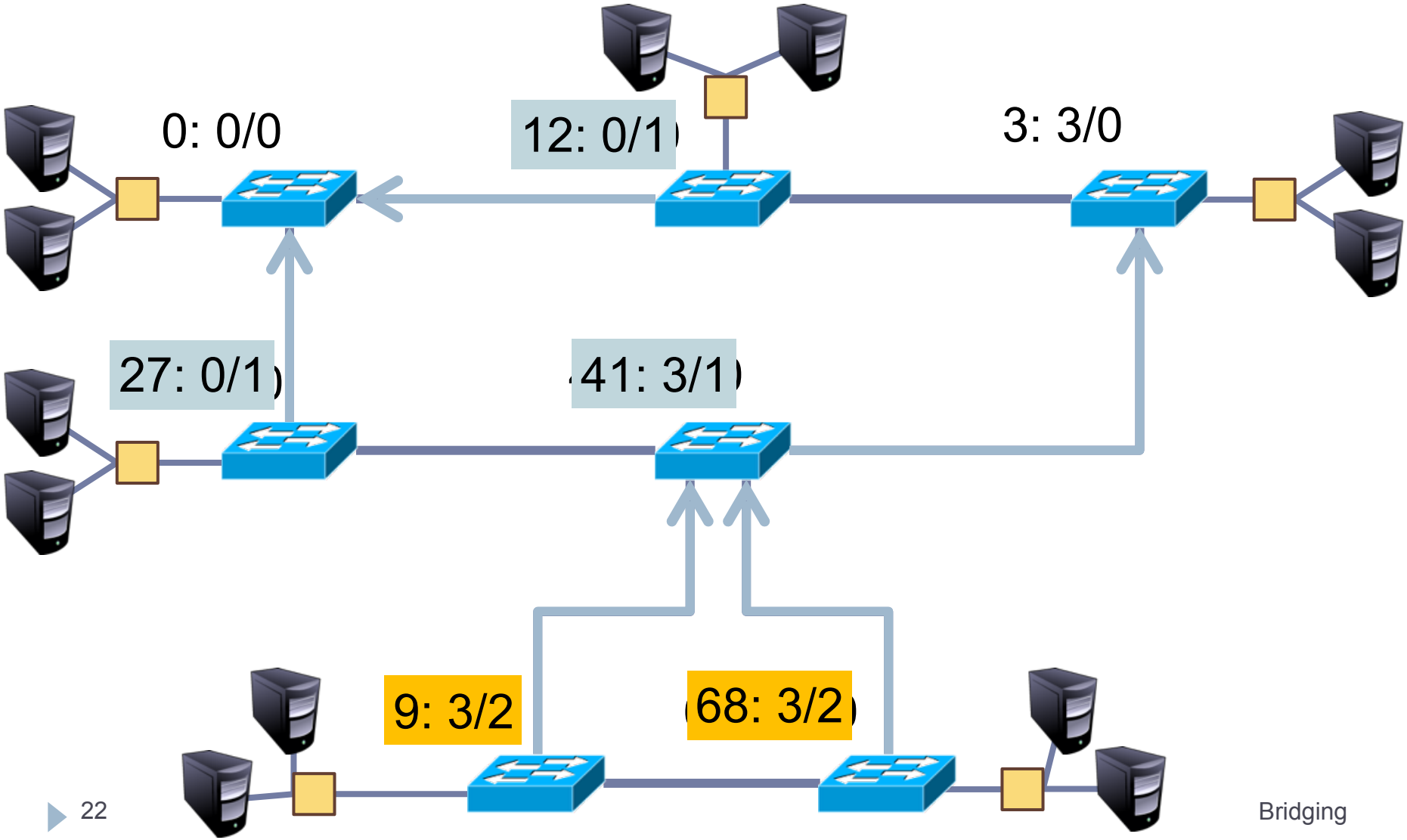
Spanning Tree Construction



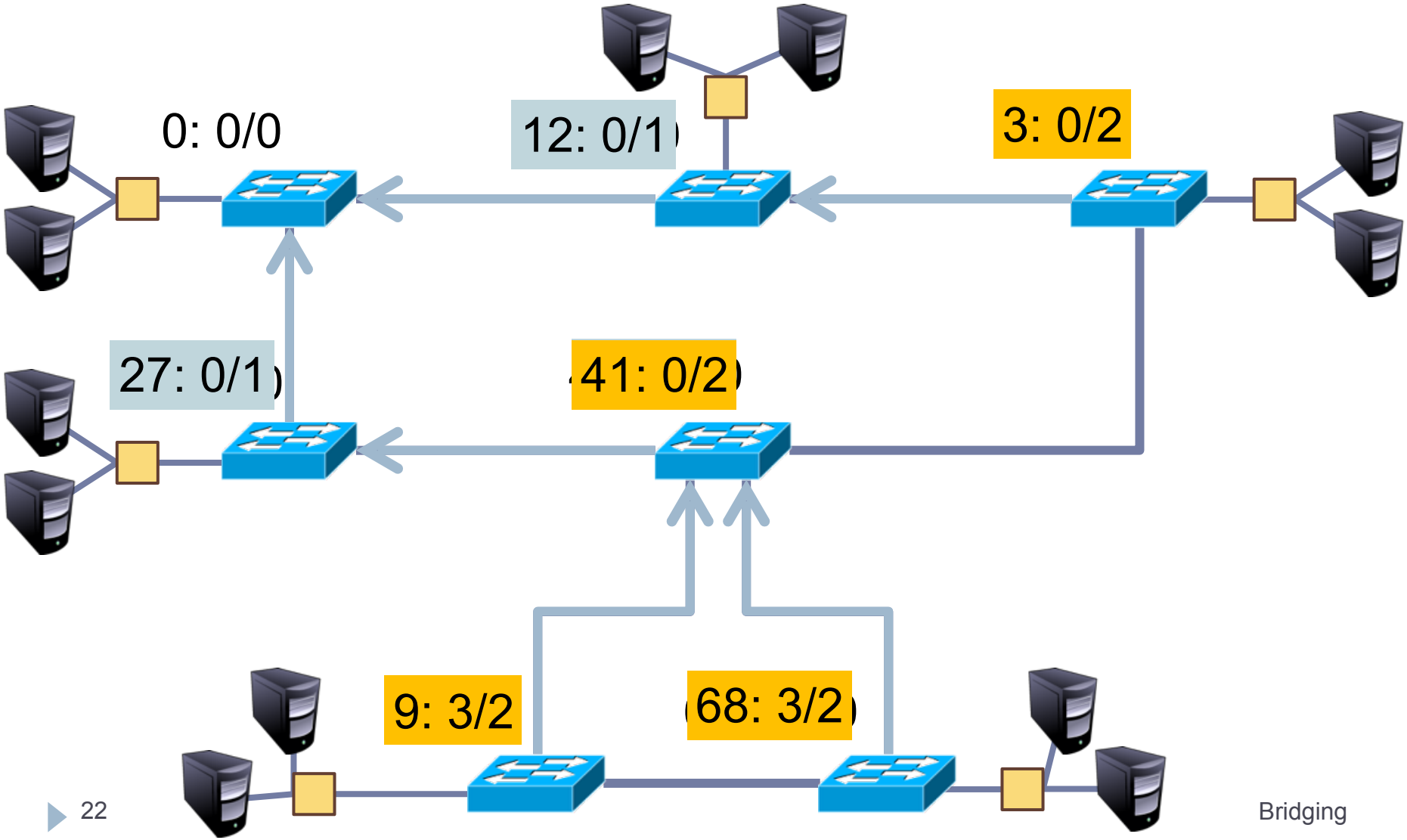
Spanning Tree Construction



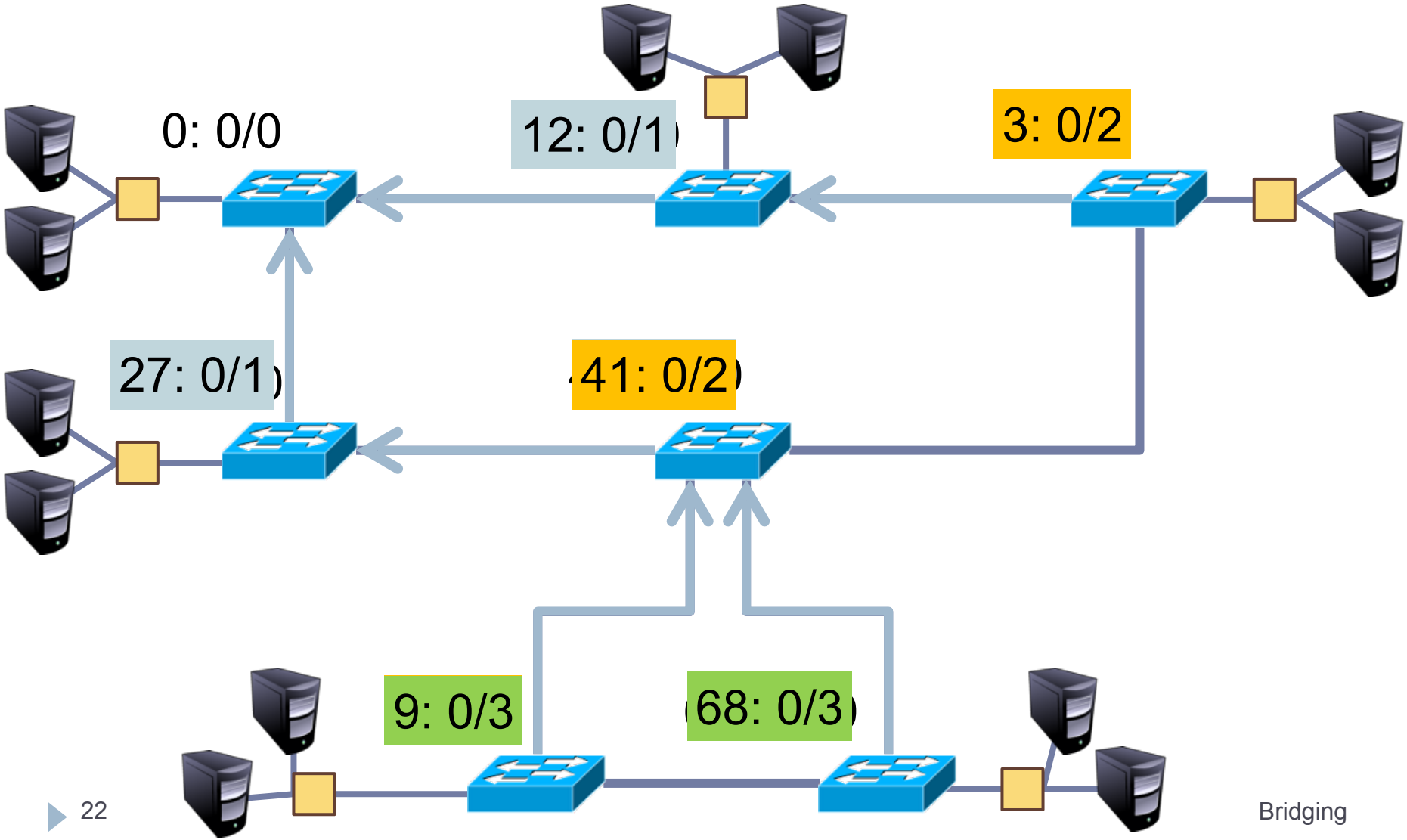
Spanning Tree Construction



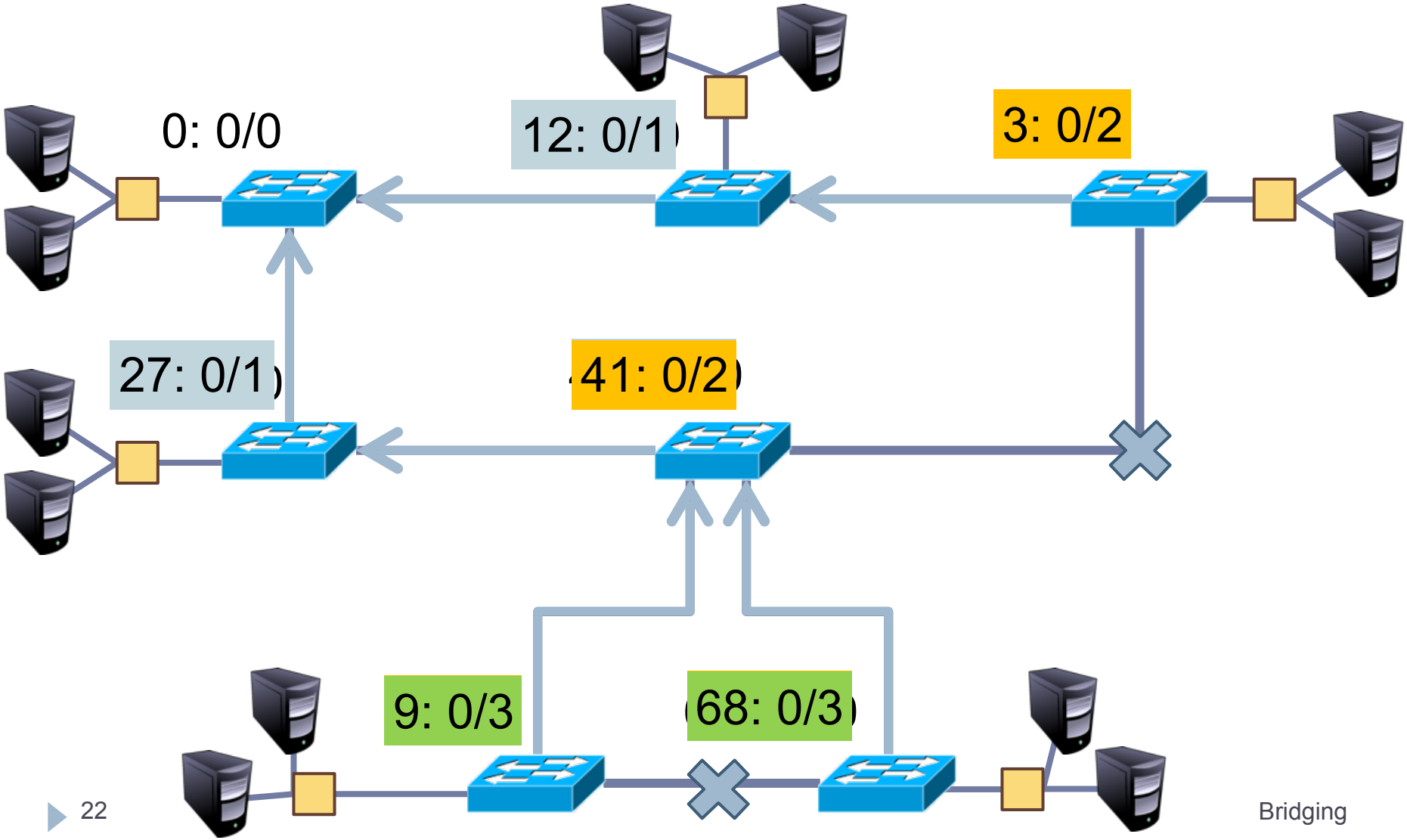
Spanning Tree Construction



Spanning Tree Construction



Spanning Tree Construction



2: Switches

Bridges vs. Switches

- Bridges make it possible to increase LAN capacity
  - Reduces the amount of broadcast packets
  - No loops
- Switch is a special case of a bridge
  - Each port is connected to a single host
    - Either a client machine
    - Or another switch
  - Links are full duplex
  - Simplified hardware: no need for CSMA/CD!
  - Can have different speeds on each port

Switching the Internet

- Capabilities of switches:
  - Network-wide routing based on MAC addresses
  - Learn routes to new hosts automatically
  - Resolve loops

Switching the Internet

- Capabilities of switches:
  - Network-wide routing based on MAC addresses
  - Learn routes to new hosts automatically
  - Resolve loops
- Could the whole Internet be one switching domain?

Switching the Internet

- Capabilities of switches:
  - Network-wide routing based on MAC addresses
  - Learn routes to new hosts automatically
  - Resolve loops
- Could the whole Internet be one switching domain?

NO

Limitations of MAC Routing

- Inefficient
  - Flooding packets to locate unknown hosts

Limitations of MAC Routing

- Inefficient
  - Flooding packets to locate unknown hosts
- Poor Performance
  - Spanning tree does not balance load
  - Hot spots

Limitations of MAC Routing

- Inefficient
  - Flooding packets to locate unknown hosts
- Poor Performance
  - Spanning tree does not balance load
  - Hot spots
- Extremely Poor Scalability
  - Every switch needs every MAC address on the Internet in its routing table!

Limitations of MAC Routing

- Inefficient
  - Flooding packets to locate unknown hosts
- Poor Performance
  - Spanning tree does not balance load
  - Hot spots
- Extremely Poor Scalability
  - Every switch needs every MAC address on the Internet in its routing table!
- IP addresses these problems (next week…)

Summary

- Bridges connect multiple LANs at layer 2
- Routing is based on MAC addresses
- MAC addresses are learned automatically
- Spanning tree is used to avoid loops