THE LINK LAYER - SWITCHES

COMP 30650: NETWORKS AND INTERNET SYSTEMS

Dr. Gavin McArdle
Email: gavin.mcardle@ucd.ie
Office: A1.09 Computer Science

RECAP

Sharing of Resources

- TDM
- FDM
  - Hybrids

Wireless Coordination

- Carrier Sense
- Collision Avoidance
  - Binary Exponential Back off

Modern Switched Ethernet

- Hubs v Switches

TODAY'S PLAN

How Switches Work

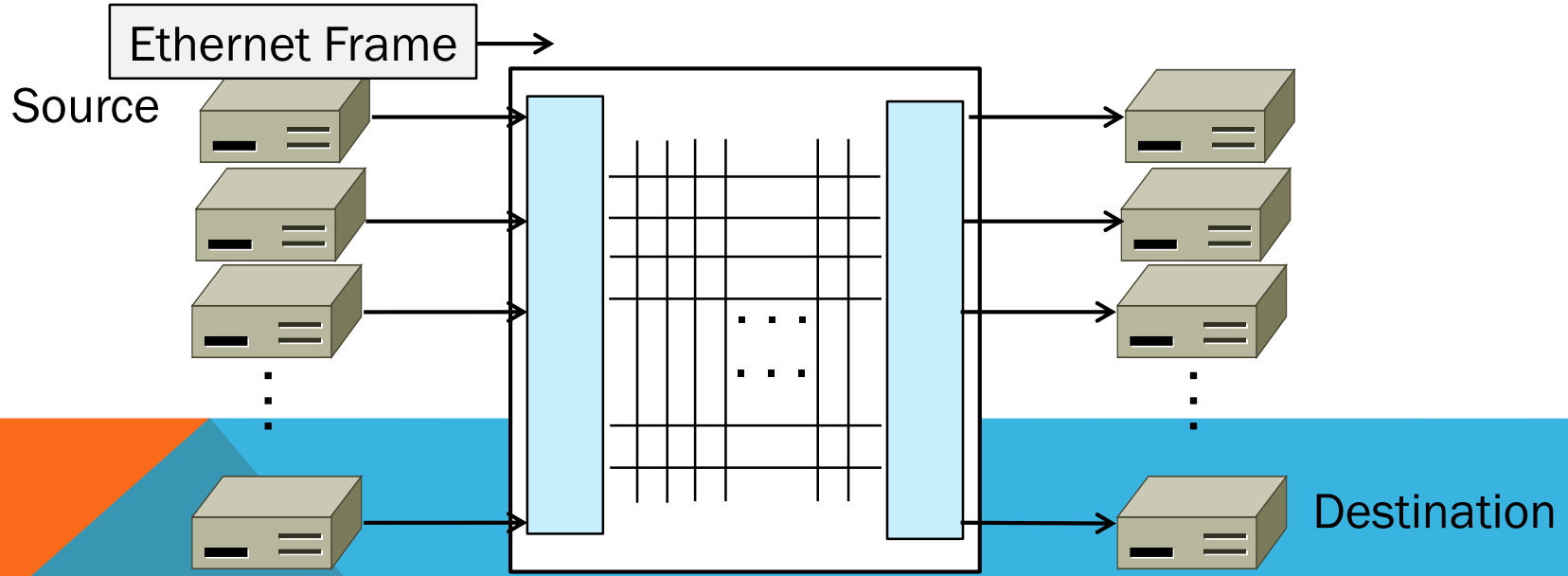
- Backwards Learning
- Spanning Tree Coordination

The Bigger Picture

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!



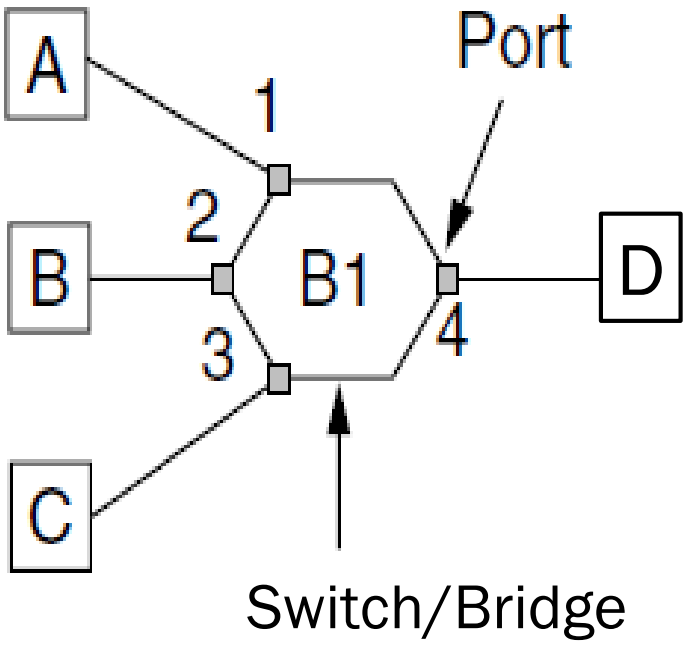
BACKWARD LEARNING

A Switch uses a port/address table which it creates.

1. To fill the table, it looks at the source address of input frames to construct the table mapping
2. To forward, it sends to the port (if known), or else broadcasts to all ports

BACKWARD LEARNING

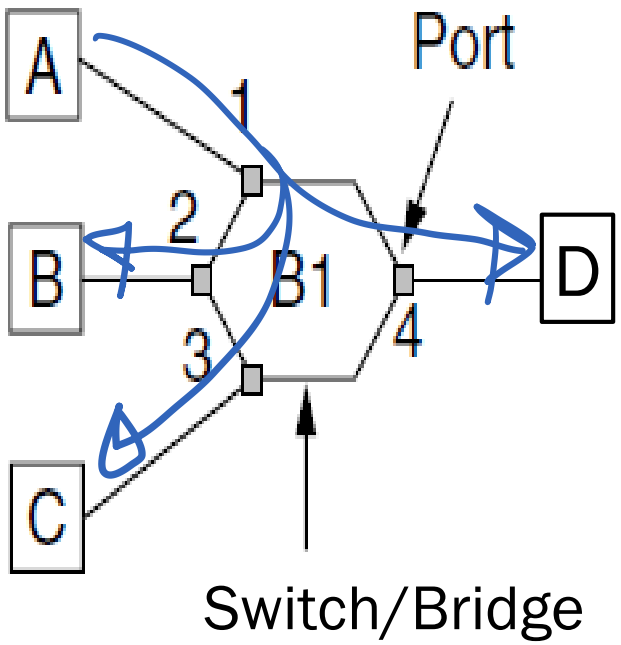
1: A sends to D



Switch B1	
Address	**Port**
A	
B	
C	
D	

BACKWARD LEARNING

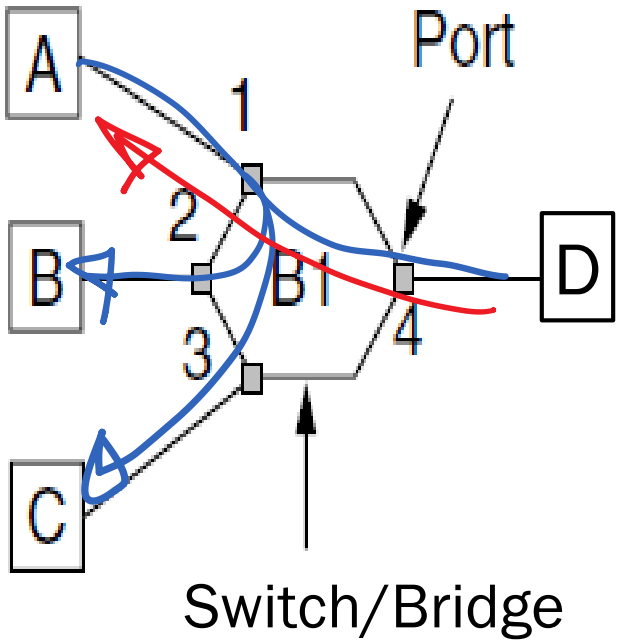
2: D sends to A



Switch B1	
Address	**Port**
A	1
B	
C	
D	

BACKWARD LEARNING

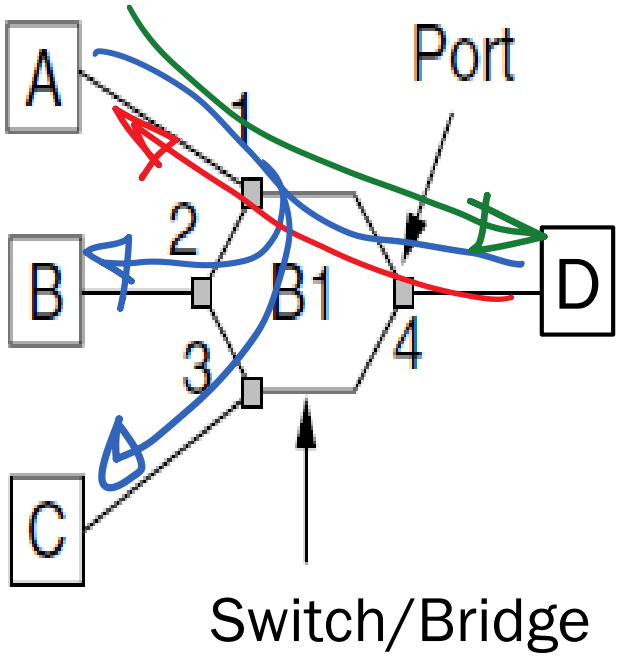
3: A sends to D



Switch B1	
Address	**Port**
A	1
B	
C	
D	4

BACKWARD LEARNING

3: A sends to D

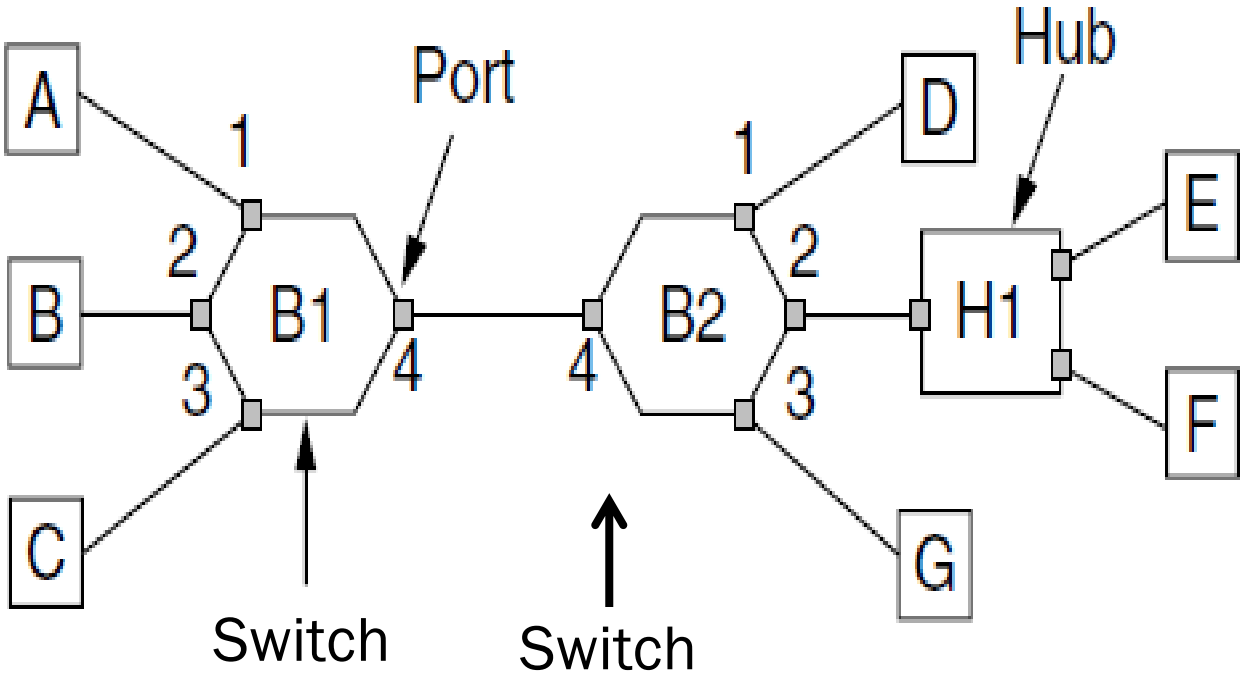


Switch B1	
Address	**Port**
A	1
B	
C	
D	4

LEARNING WITH MULTIPLE SWITCHES

Works the same way with multiple switches and a mix of hubs *assuming no loops*,

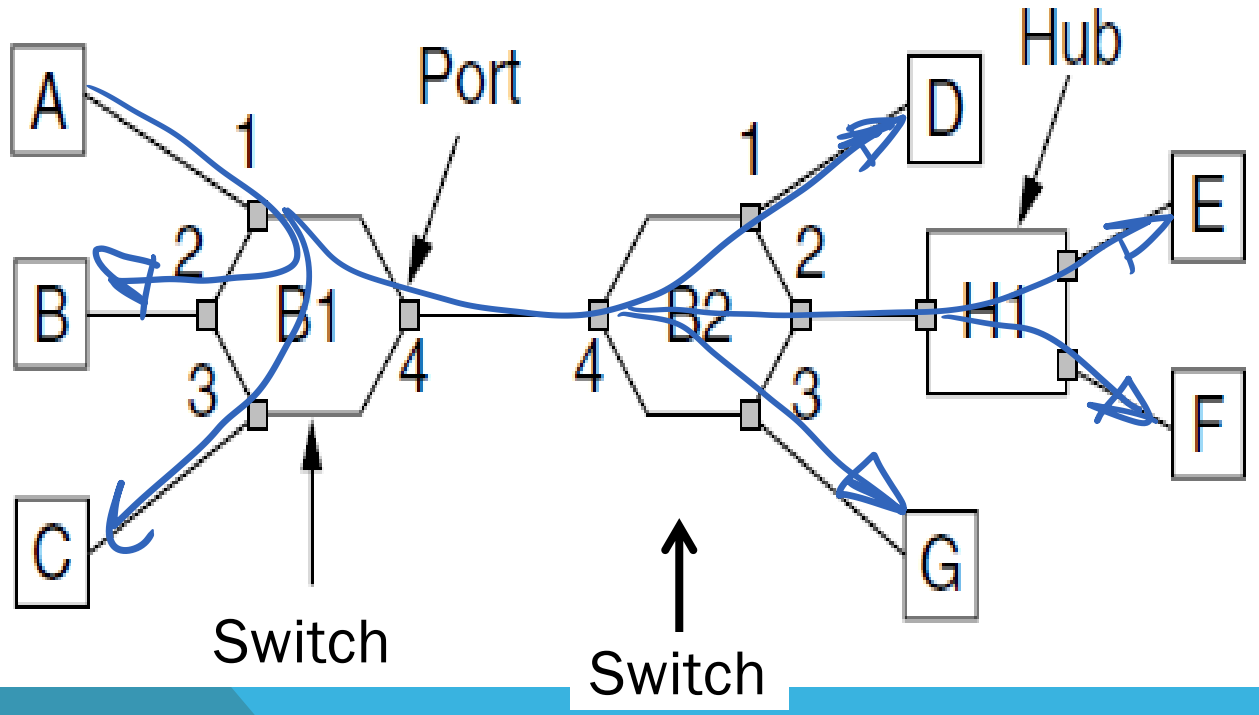
A sends to D then D sends to A



LEARNING WITH MULTIPLE SWITCHES

Works the same way with multiple switches and a mix of hubs *assuming no loops*,

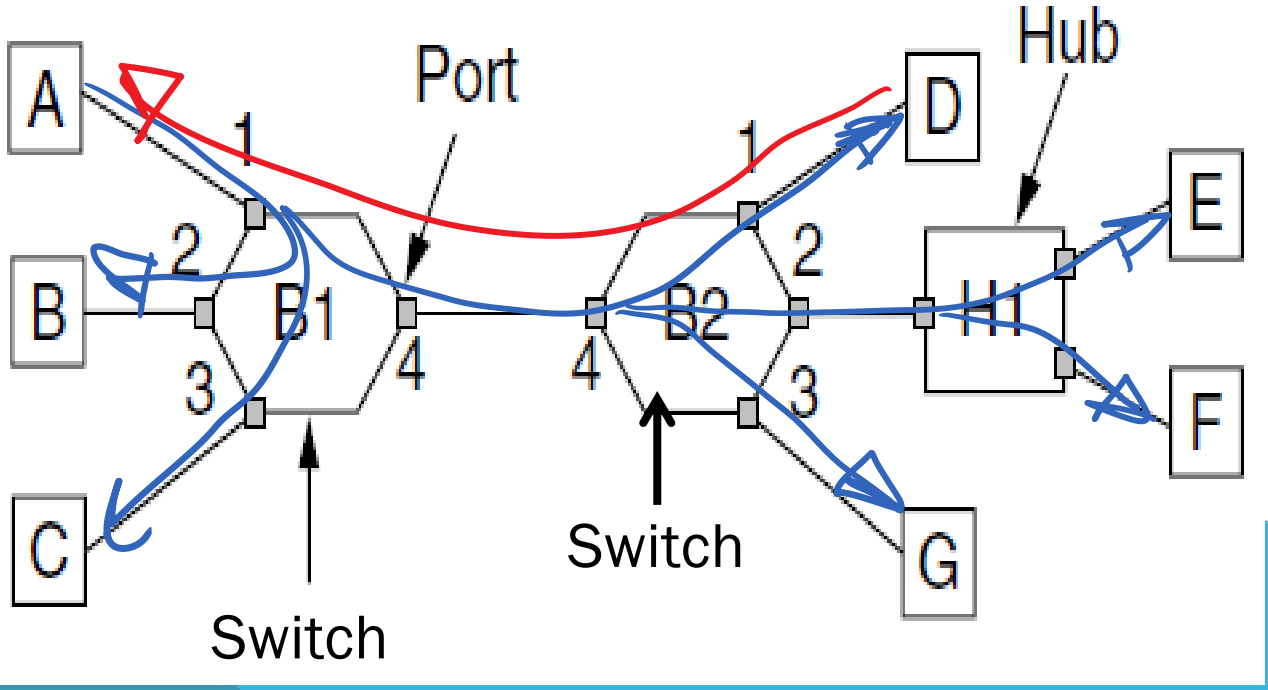
A sends to D then D sends to A



LEARNING WITH MULTIPLE SWITCHES (3)

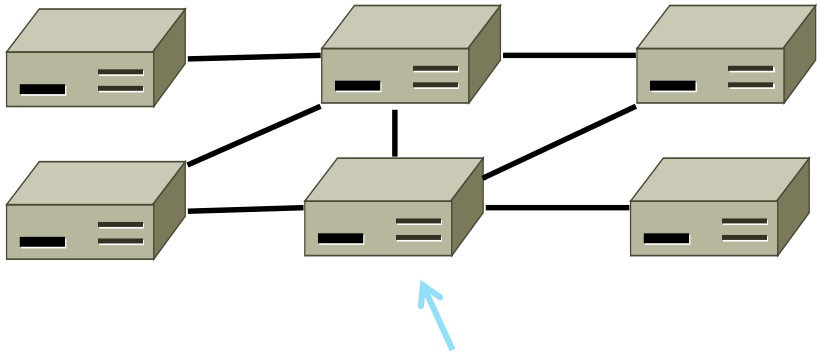
Works the same way with multiple switches and a mix of hubs *assuming no loops*,

A sends to D then D sends to A



TOPIC

How can we connect switches in any topology so they just work



Loops – yet another complication to solve!

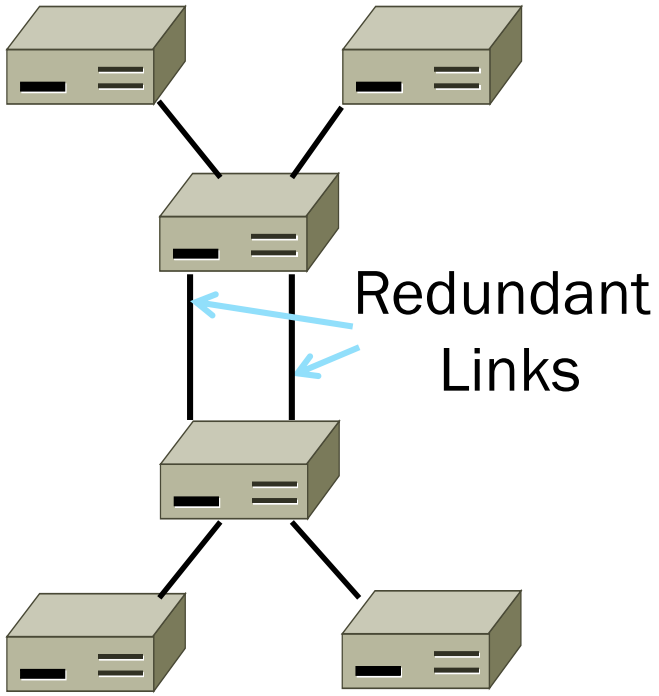
PROBLEM – FORWARDING LOOPS

May have a loop in the topology

- Redundancy in case of failures
- Or a simple mistake

Want LAN switches to "just work"

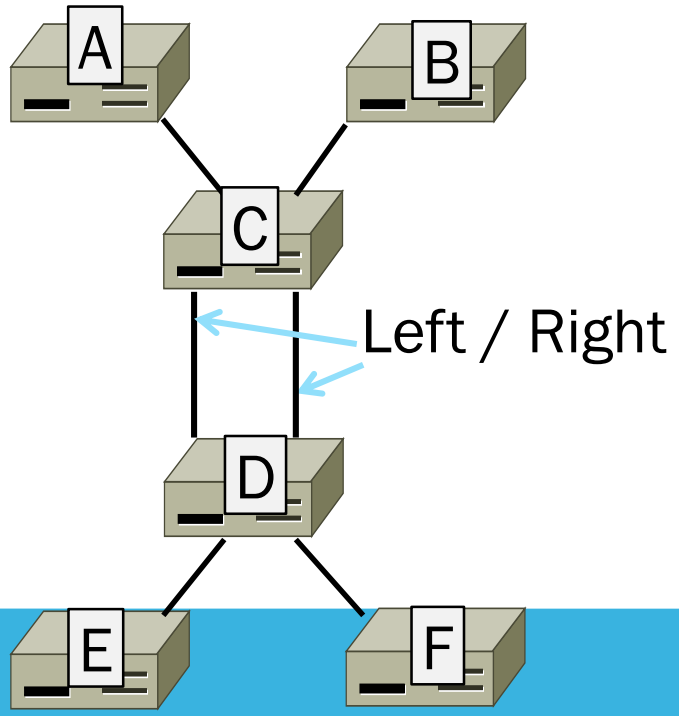
- Plug-and-play, no changes to hosts
- But loops cause a problem …



FORWARDING LOOPS

Suppose the network is started

A sends to F. What happens?

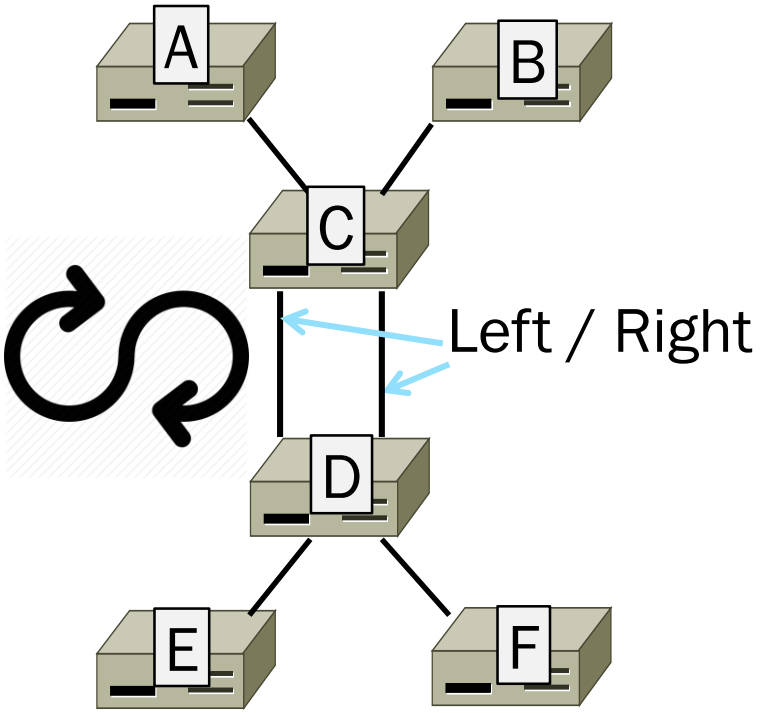


FORWARDING LOOPS

Suppose the network is started

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 → …



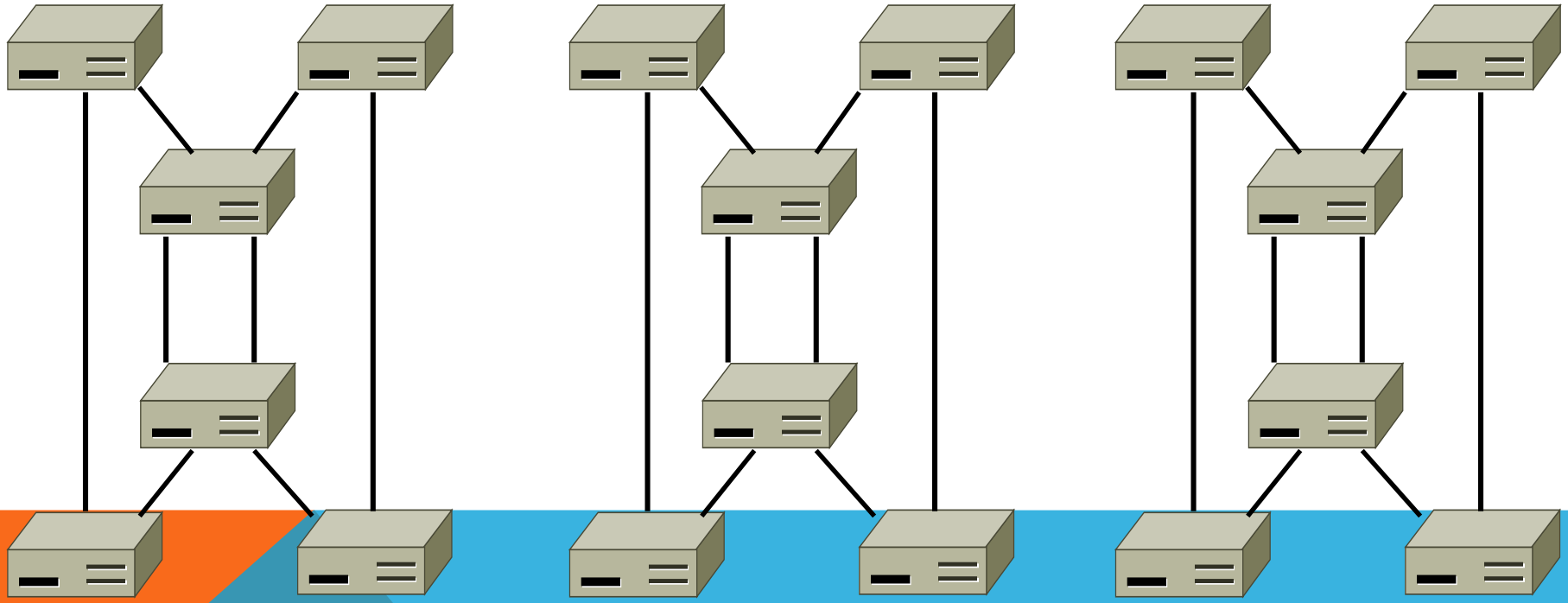
SPANNING TREE SOLUTION

Switches collectively find a spanning tree for the topology

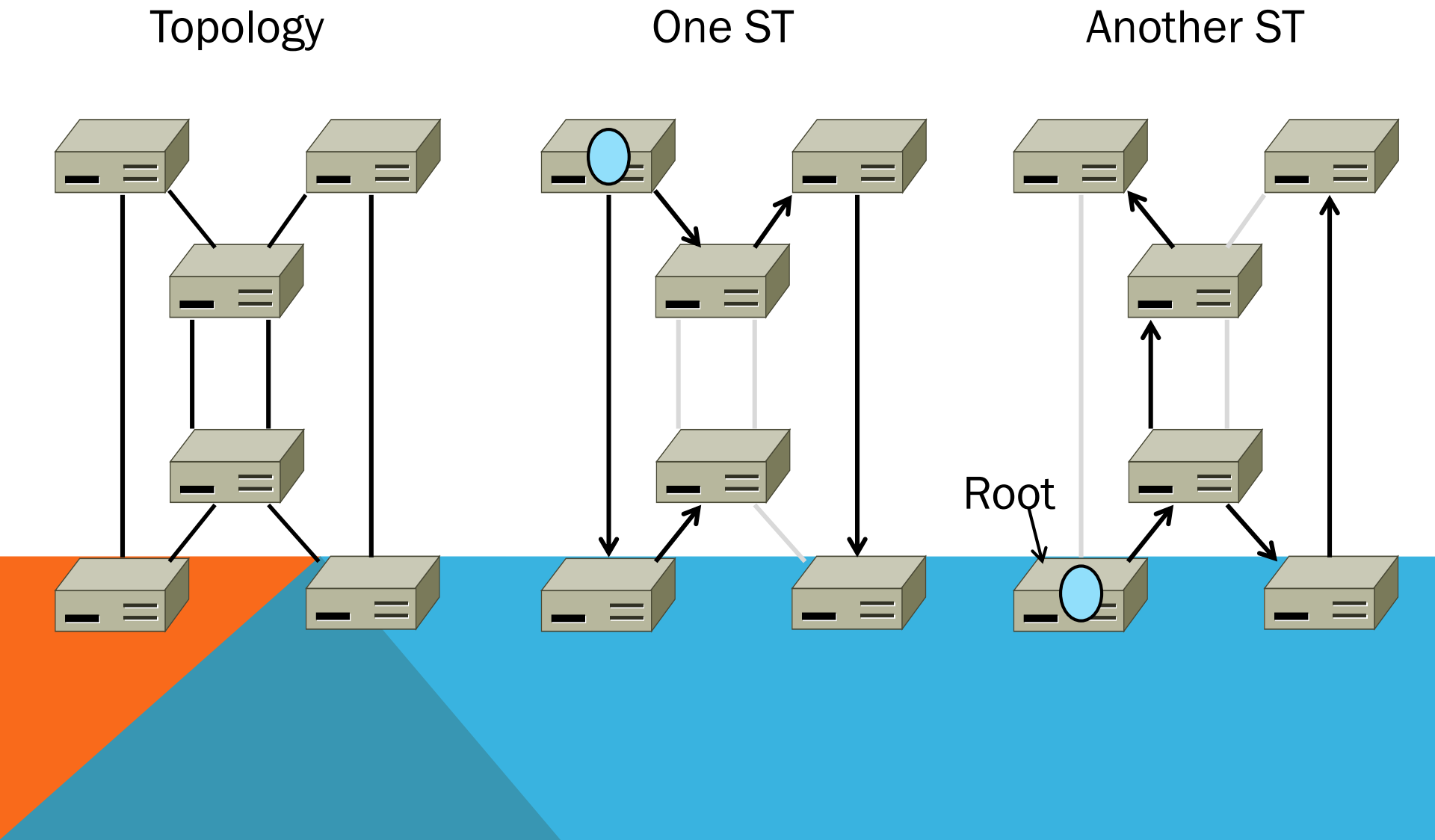
- A subset of links that is a tree (no loops) and reaches all switches
- The switches forward as normal on the spanning tree
- Broadcasts will go up to the root of the tree and down all the branches

SPANNING TREE

Topology



SPANNING TREE



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, …

SPANNING TREE ALGORITHM

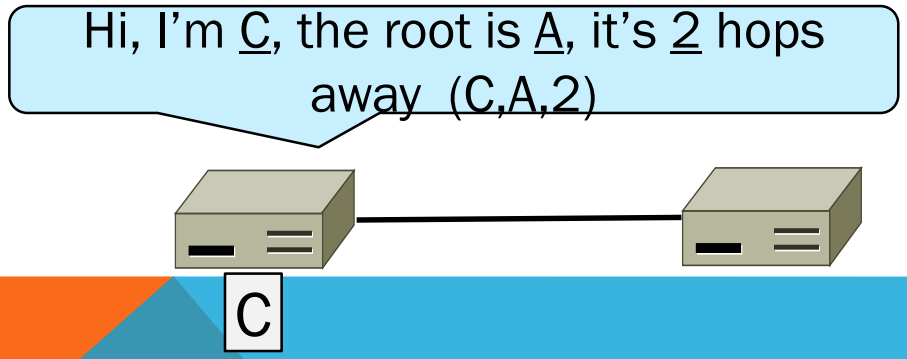
Outline:

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 the root, and distance (in hops) to root
- Switches favor ports with shorter distances to lowest root
  - Uses lowest address as a tie for distances



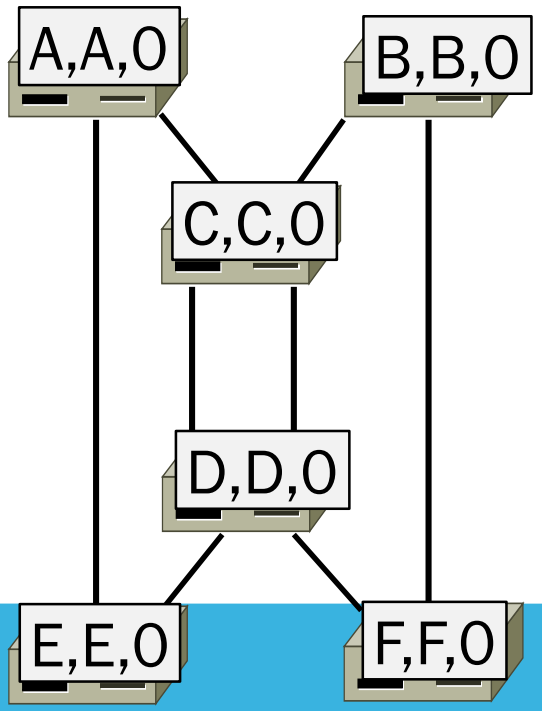
SPANNING TREE EXAMPLE

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 it is (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)



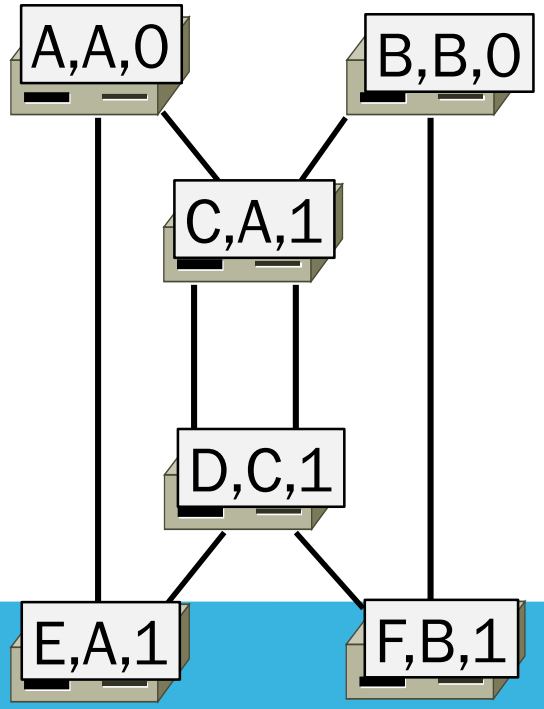
SPANNING TREE EXAMPLE

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)



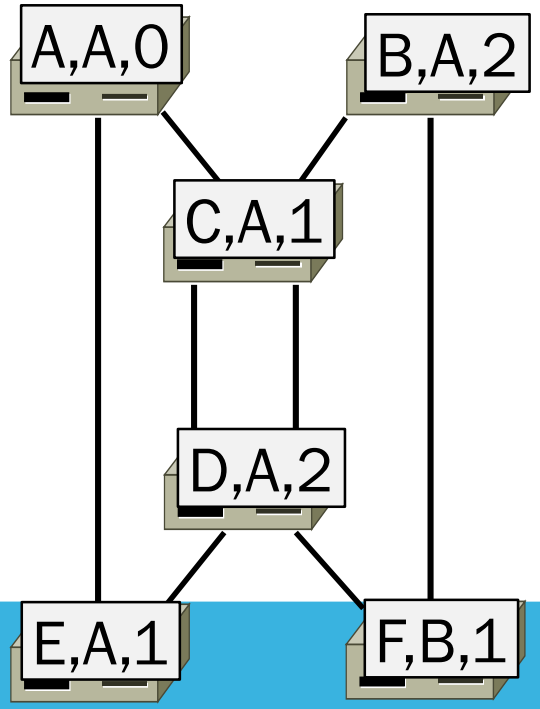
SPANNING TREE EXAMPLE

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



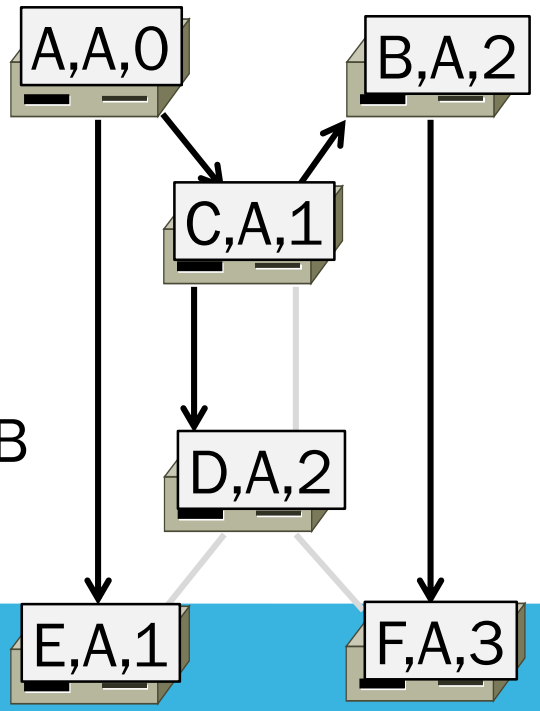
SPANNING TREE EXAMPLE

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

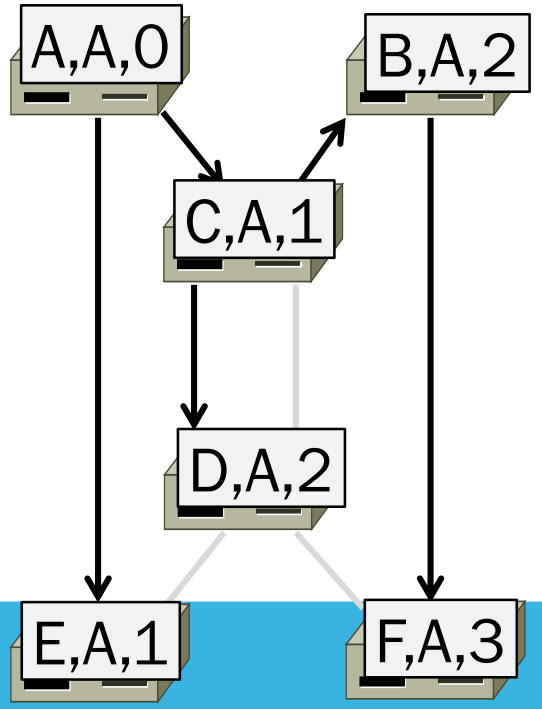


SPANNING TREE EXAMPLE

Forwarding proceeds as usual on the ST

Initially D sends to F:

And F sends back to D:



SPANNING TREE EXAMPLE

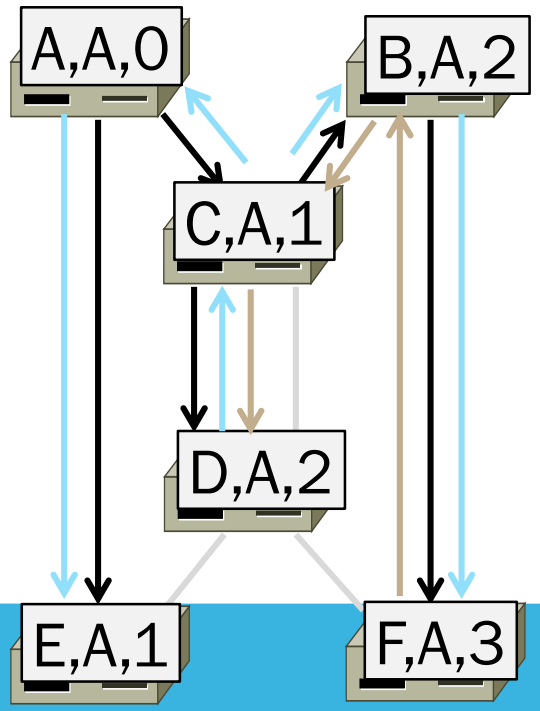
Forwarding proceeds as usual on the ST

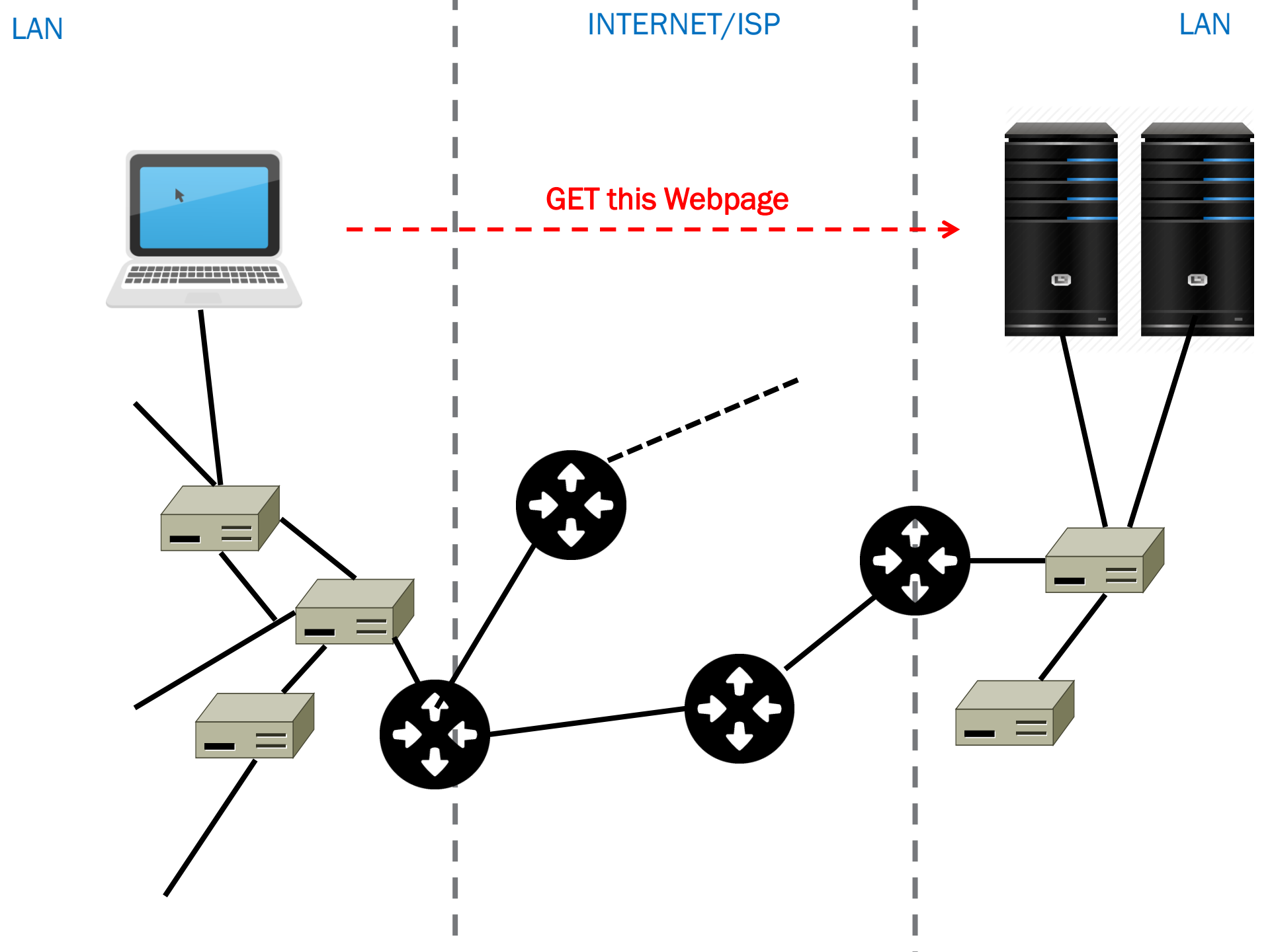
Initially D sends to F:

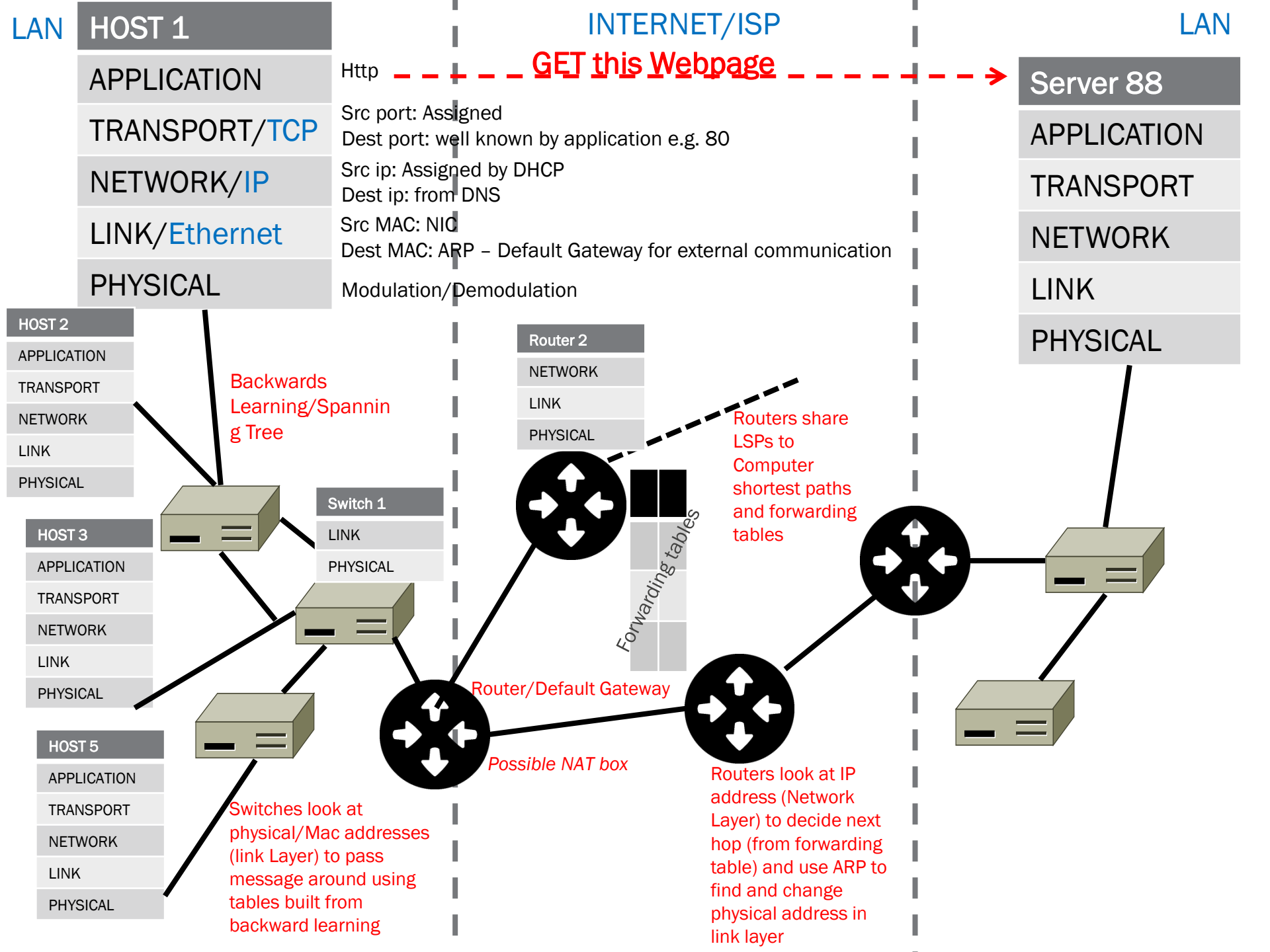
- D → C-left
- C → A, B
- A → E
- B → F

And F sends back to D:

- F → B
- B → C
- C → D




LAN
INTERNET/ISP
LAN
GET this Webpage


LAN
HOST 1
APPLICATION
TRANSPORT/TCP
NETWORK/IP
LINK/Ethernet
PHYSICAL
INTERNET/ISP
LAN
Http
GET this Webpage
Server 88
APPLICATION
TRANSPORT
NETWORK
LINK
PHYSICAL
Src port: Assigned
Dest port: well known by application e.g. 80
Src ip: Assigned by DHCP
Dest ip: from DNS
Src MAC: NIC
Dest MAC: ARP – Default Gateway for external communication
Modulation/Demodulation
HOST 2
APPLICATION
TRANSPORT
NETWORK
LINK
PHYSICAL
Backwards Learning/Spanning Tree
Router 2
NETWORK
LINK
PHYSICAL
Routers share LSPs to Computer shortest paths and forwarding tables
Switch 1
LINK
PHYSICAL
HOST 3
APPLICATION
TRANSPORT
NETWORK
LINK
PHYSICAL
Forwarding tables
Router/Default Gateway
HOST 5
APPLICATION
TRANSPORT
NETWORK
LINK
PHYSICAL
Possible NAT box
Switches look at physical/Mac addresses (link Layer) to pass message around using tables built from backward learning
Routers look at IP address (Network Layer) to decide next hop (from forwarding table) and use ARP to find and change physical address in link layer

THE LINK LAYER RECAP

- Builds on the link layer
  - Switches send frames over a link.
  - Error checking
  - Access control

THE NETWORK LAYER

- Network Layer
  - Routers send packets over one or more networks
  - Routing
  - Quality of Service
  - **Scale** to large networks
  - Support diverse technologies

RECAP

Switched Ethernet

- Hubs
- Switches
- Backwards Learning
  - Lookup table
- Spanning Trees
  - Circular reference problem

WHY DO WE NEED A NETWORK LAYER?

Shortcomings of switches

1. **Don't scale to large networks**
   - Blow up of routing table, broadcast messages!
2. **Don't work across more than one link layer technology**
   - Hosts on Ethernet + 3G + 802.11
3. **Don't give much traffic control**
   - Want to plan routes / bandwidth

NETWORK LAYER APPROACH

Scaling:

- Hierarchy, in the form of prefixes

Heterogeneity:

- IP for internetworking

Bandwidth Control:

- Lowest-cost routing
- Later QOS (Quality of Service)

NETWORK LAYER TOPICS

Network service

- Datagrams (packets), virtual circuits

IP (Internet Protocol)

- Internetworking
- Forwarding (Longest Matching Prefix)
- Helpers: *ARP* and DHCP
- Fragmentation and MTU discovery
- Errors: ICMP

NAT, a “middlebox”

Routing algorithms

ROUTING VS. FORWARDING

Routing is the process of deciding in which direction to send traffic

- Network wide (global) and expensive

Forwarding is the process of sending a packet on its way

- Node process (local) and fast