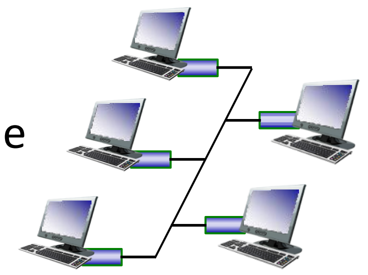
LAN Switches and Bridges

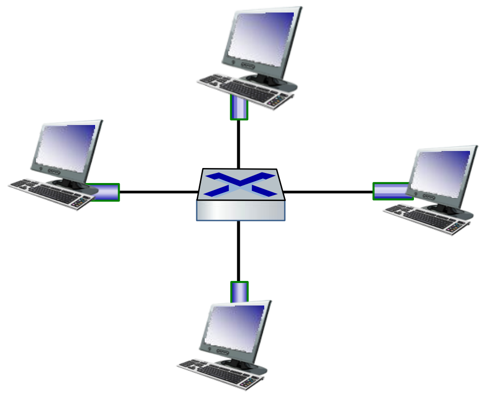
Chapter 3.2

Ethernet: Physical topology

- bus: popular through mid 90s
  - all nodes in same collision domain (can collide with each other)
- switched: prevails today
  - active link-layer 2 *switch* in center
  - each "spoke" runs a (separate) Ethernet protocol (nodes do not collide with each other)

bus: coaxial cable

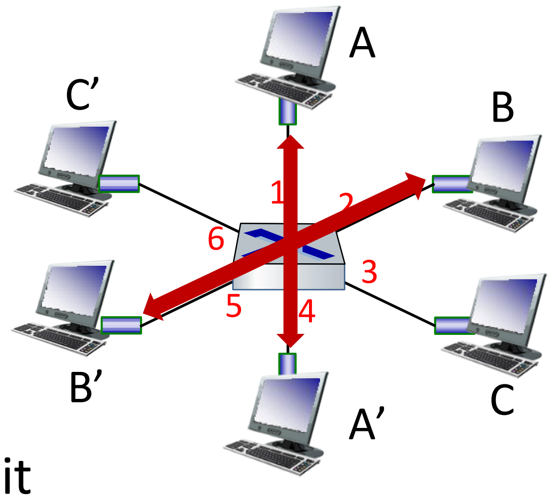




switched

Switch: Multiple simultaneous transmissions

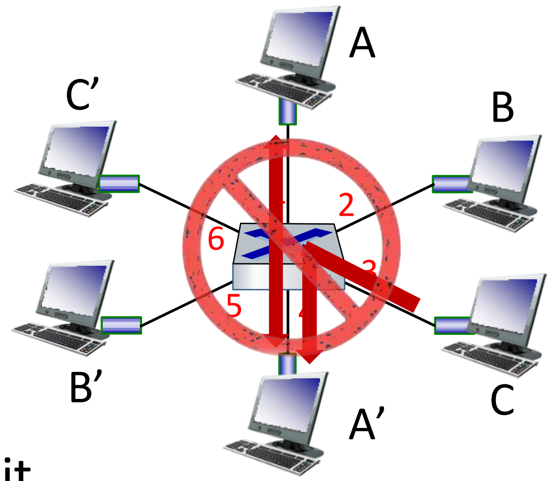
- hosts have dedicated, direct connection to switch
- switches buffer packets
- Ethernet protocol used on *each* incoming link, so:
  - no collisions; full duplex
  - each link is its own collision domain
- switching: A-to-A' and B-to-B' can transmit simultaneously, without collisions



switch with six interfaces (1,2,3,4,5,6)

Switch: Multiple simultaneous transmissions

- hosts have dedicated, direct connection to switch
- switches buffer packets
- Ethernet protocol used on *each* incoming link, so:
  - no collisions; full duplex
  - each link is its own collision domain
- switching: A-to-A' and B-to-B' can transmit simultaneously, without collisions
  - but A-to-A' and C to A' can *not* happen simultaneously



switch with six interfaces (1,2,3,4,5,6)

Ethernet switch

- Switch is a link-layer device: takes an *active* role
  - store, forward Ethernet frames
  - examine incoming frame's MAC address, *selectively* forward frame to one-or-more outgoing links when frame is to be forwarded on segment, uses CSMA/CD with binary backoff to access segment
- *Connectionless:* NO handshaking between sending and receiving NICs.
- *Unreliable:* Receiving NIC doesn't send ACKs or NAKs to sending NIC
  - Data in dropped frames recovered only if initial sender uses higher layer rdt (e.g.,TCP), otherwise dropped data lost.

Switch forwarding table

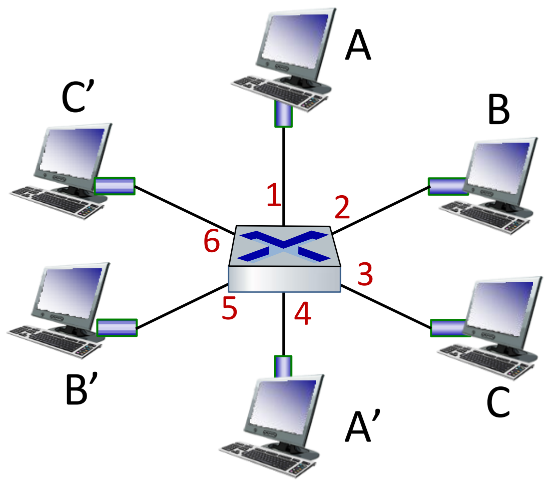
Q: how does switch know A’ reachable via interface 4, B’ reachable via interface 5?

A: each switch has a switch table, each entry:

- (MAC address of host, interface to reach host, time stamp) → E.g., Datagram switching
- looks like a routing table!

Q: how are entries created, maintained in switch table?

- something like a routing protocol?



Learning Switches

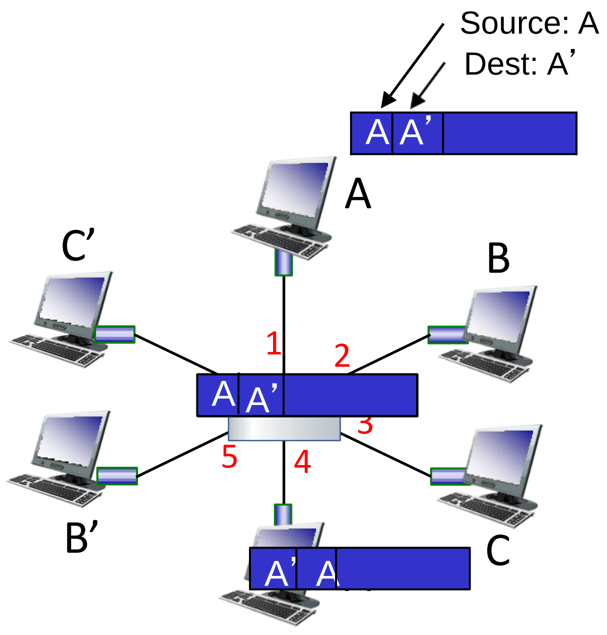
- Trivial algorithm
  - Switches simply accept LAN frames on their inputs and forward them out on all other outputs (i.e. all (other) LANs) → Hub.
  - Potentially heavy traffic and processing overhead
- Optimize by using address information
  - “Learn” which hosts live on which LAN
  - Maintain forwarding table
  - Only forward when necessary
  - Reduces switch workload

Details

- When a switch first boots, this table is empty; entries are added over time
- a timeout is associated with each entry, and the switch discards the entry after a specified period of time
- Should the switch receive a frame that is addressed to a host not currently in the table, it goes ahead and forwards the frame out on the other ports.

Switch Self-learning: Example

- frame destination, A', location unknown: flood
- destination A location known: selectively send on just one link

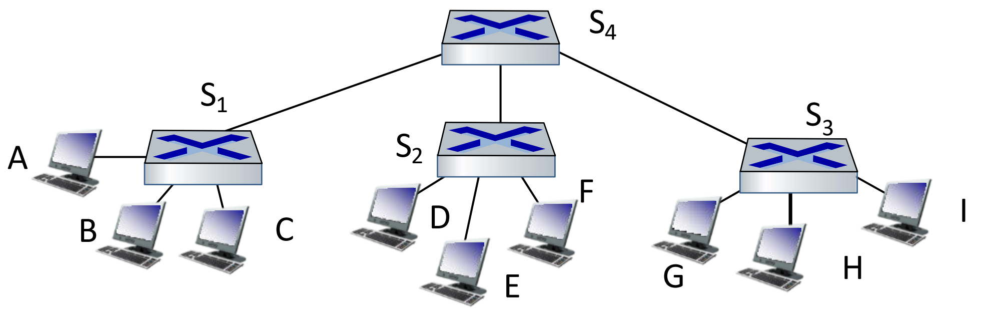


MAC addr	interface	TTL
A	*1*	*60*
A'	*4*	*60*

switch table (initially empty)

Interconnecting switches

self-learning switches can be connected together:



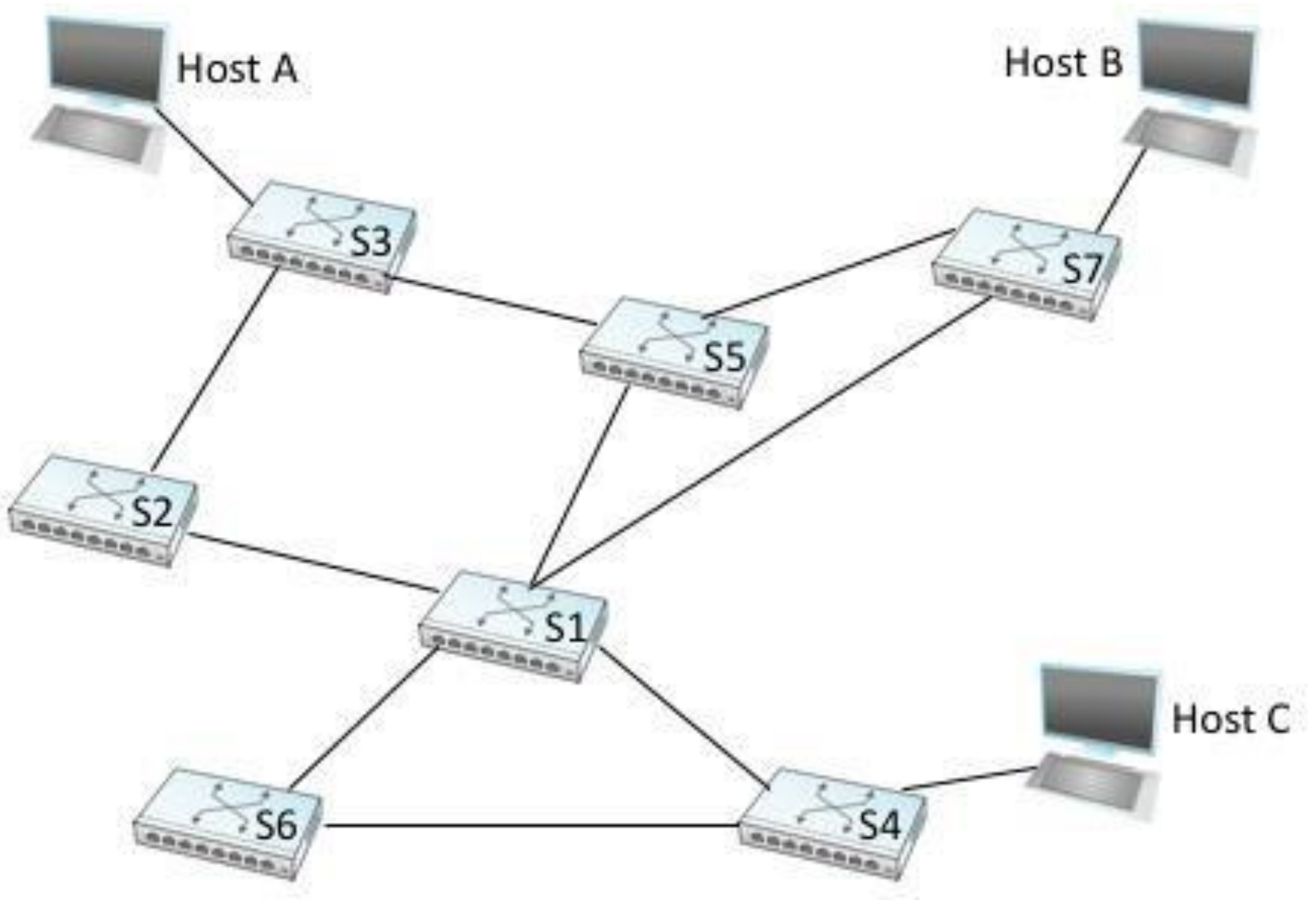
Q: sending from A to G - how does S_1 know to forward frame destined to G via S_4 and S_3?

- *A:* self learning! (works exactly the same as in single-switch case!)

Learning switches

- Problem
  - If there is a loop in the extended LAN, a packet could circulate forever
- Solution
  - Create a spanning tree to eliminate unnecessary edges
- Loops
  - Adds robustness
  - Complicates learning/forwarding

Example Extended LAN with LOOPS



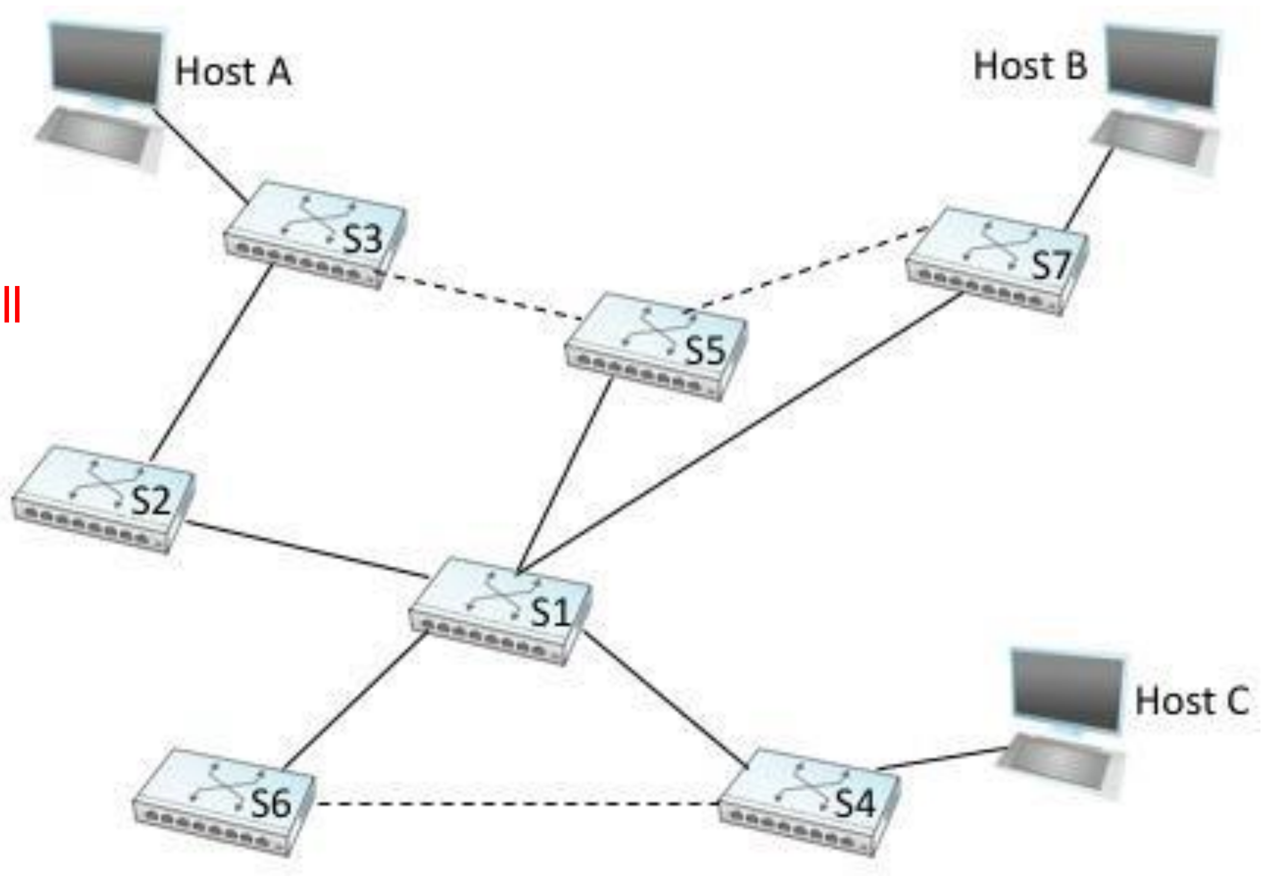
Spanning Tree Algorithm

- View extended LAN as a graph
  - Switches are graph nodes
  - Ports are edges connecting LAN’s to switches

- Objective
  - Obtain a spanning tree which connects all LAN’s
  - Subset of the actual network topology that has no loops and that reaches all the devices in the network.

Spanning Tree Algorithm

Select the ports over which they will forward frames



Challenge: A single view of the spanning tree among the switches.

Distributed Spanning Tree Algorithm

- Switches exchange configuration messages
    - (Y,d,X)
        - Y = root node
        - d = distance to root node
        - X = originating node
- Each switch records current best configuration message for each port
- Initially, each switch believes it is the root
- When a switch discovers it is not the root, stop generating messages
- Steady State
    - Root periodically send configuration messages
    - A timeout is used to restart the algorithm

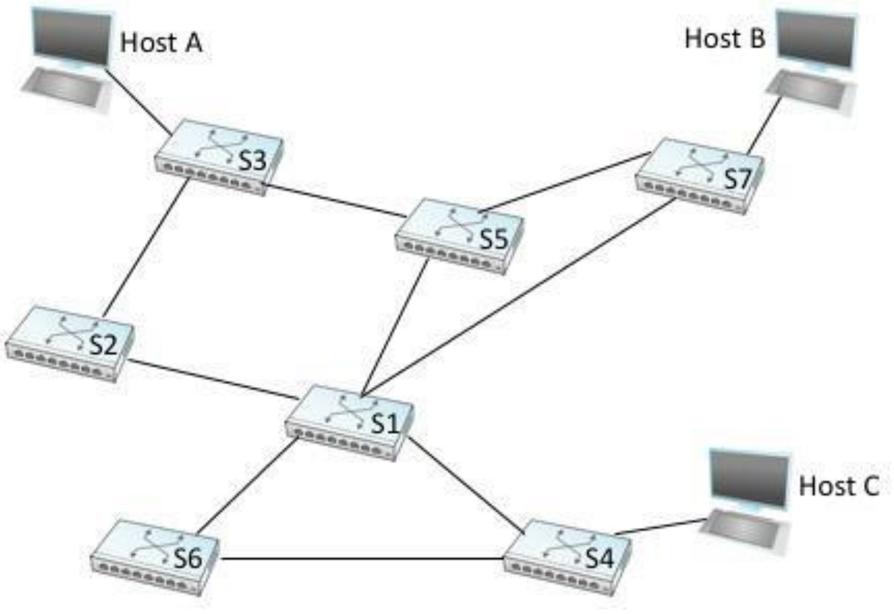
Defining a Spanning Tree

- Basic Rules
  - Bridge with the lowest ID is the **root**
  - The new configuration message is considered better than the currently recorded information if any of the following is true:
    - It identifies a root with a smaller ID.
    - It identifies a root with an equal ID but with a shorter distance.
    - The root ID and distance are equal, but the sending switch has a smaller ID

Spanning Tree Algorithm

- Example at switch S3

1. S3 receives (S2, 0, S2)
2. Since 2 < 3, S3 accepts S2 as root
3. S3 adds one to the distance advertised by S2 and sends (S2, 1, S3) to S5
4. Meanwhile S2 accepts S1 as root and sends (S1, 1, S2) to S3
5. S5 accepts S1 as root and sends (S1, 1, S5) to S3
6. S3 accepts S1 as root and it notes that both S2 and S5 are closer to the root than it is, but S2 has the smaller id, so it remains on S3's path to the root.

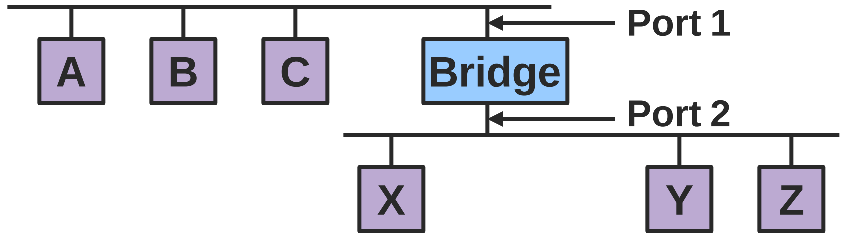


Bridges

- Bridges
  - “bridge” ethernet segments to build an extended LAN
    - Extend LAN concept (i.e.aka “LAN switches”)
  - Forward packets between LANs, i.e. “LAN switches”
    - Switched ethernet == extended LAN
  - Increase bandwidth
    - Achieving n times of bandwidth where n is # of ports
  - Limited scalability
    - to O(1,000) hosts
    - not to global networks – we need hierarchy
- Bridge outline
  - Learning bridges
  - Spanning tree algorithm

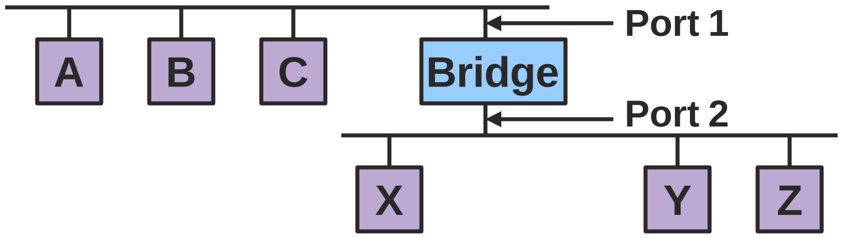
Example

- Whenever a frame from host A that is addressed to host B arrives on port 1, there is no need for the bridge to forward the frame out over port 2
- The question: how does a bridge come to learn on which port the various hosts reside?



Learning Bridges

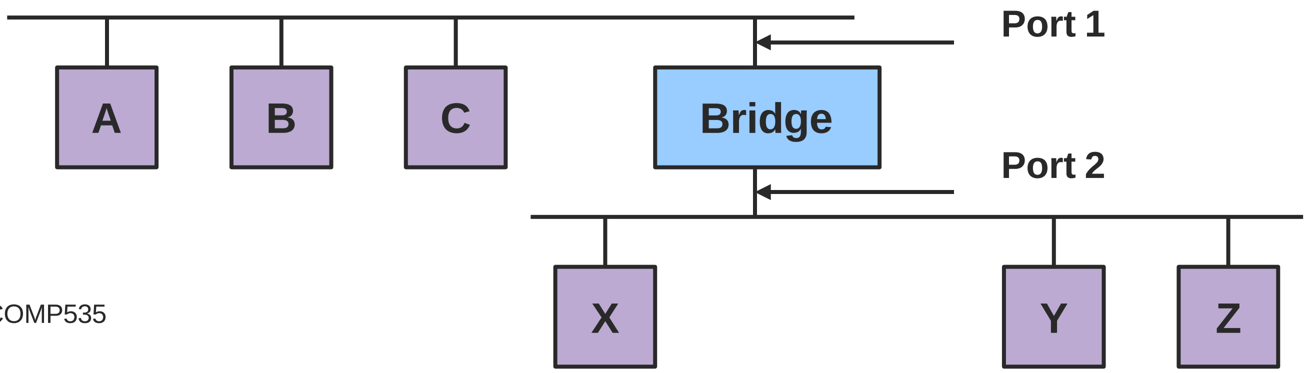
- Bridge learns table entries based on source address
  - When receive frame from A on port 1
    add A to list of hosts on port 1
  - Time out entries to allow movement of hosts
- Always forward broadcast frames



Host	Port
A	1
B	1
C	1
X	2
Y	2
Z	2

Learning Bridges

- Examples
  - Frame for A received on port 1
    - The bridge does nothing (on the same ethernet)
  - Frame for C received on port 2
    - The bridge forwards the frame to port 1
  - Frame for S received on port 2
    - The bridge forwards the frame to port 1



Bridges - Spanning Tree Algorithm

Example at bridge B3

B3 receives (B2, 0, B2)

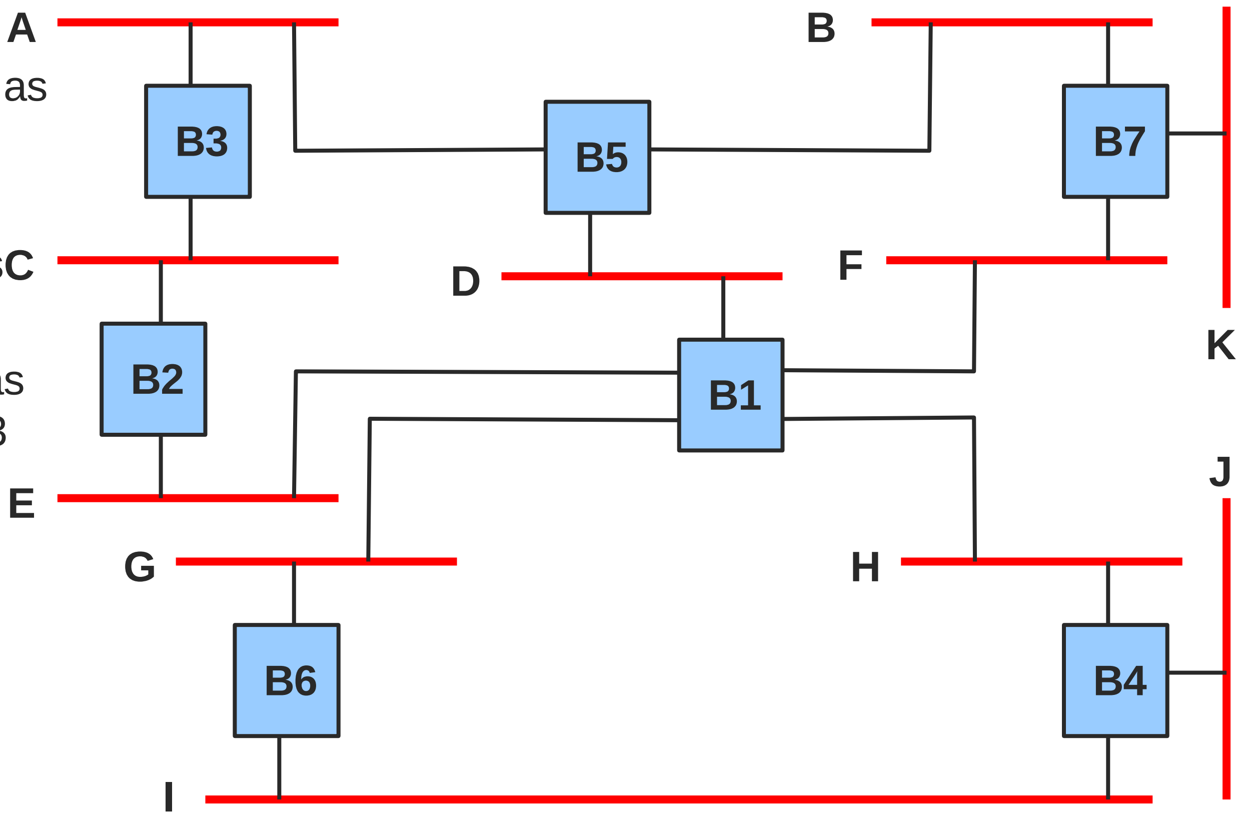
Since 2 < 3, B3 accepts B2 as root

B3 adds one to the distance advertised by B2 and sends (B2, 1, B3) to B5

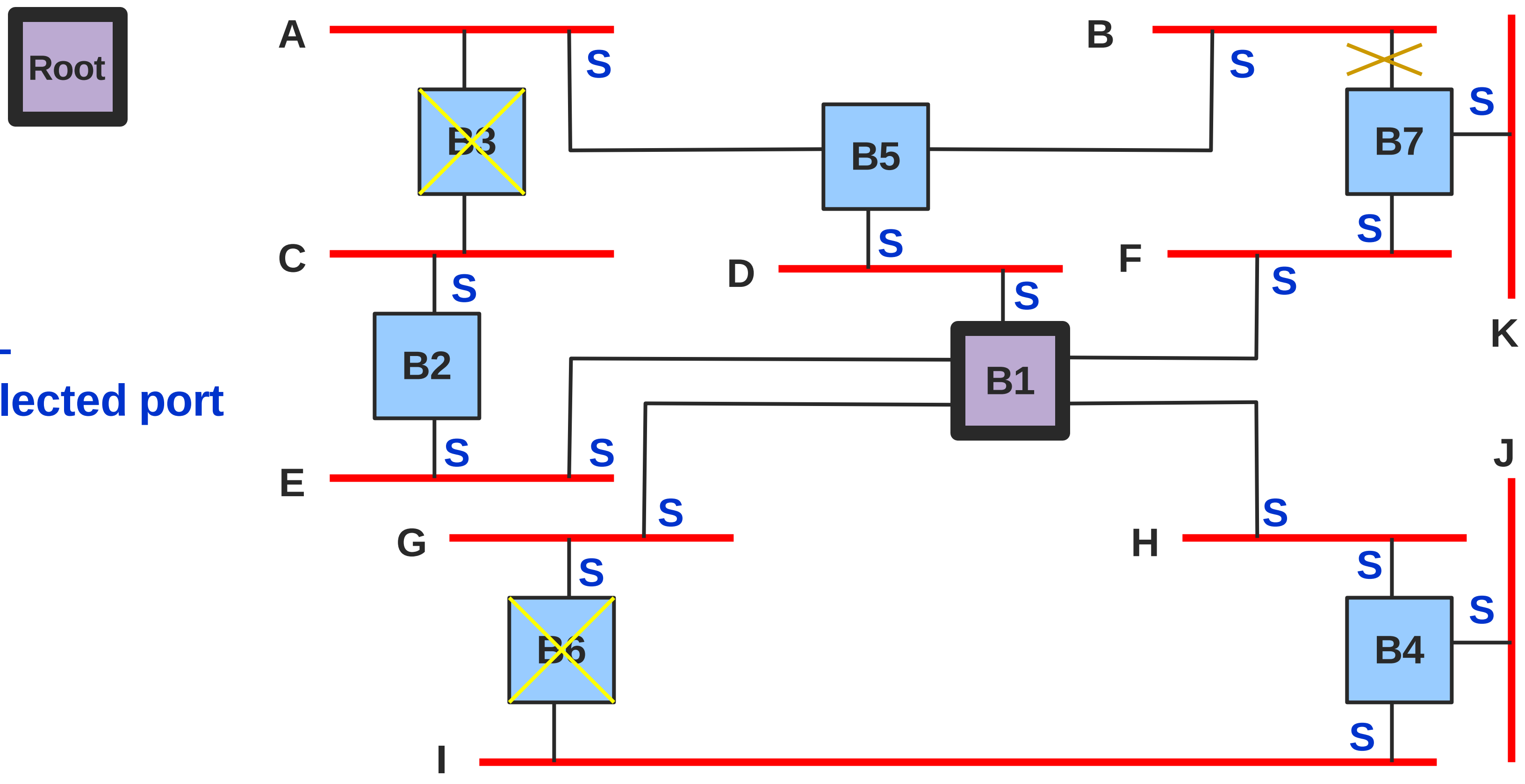
Meanwhile B2 accepts B1 as root and sends (B1, 1, B2) to B3

B5 accepts B1 as root and sends (B1, 1, B5) to B3

B3 accepts B1 as root and stops forwarding to both interfaces (because B2, B5 are closer to root)



Bridges - Spanning Tree Algorithm



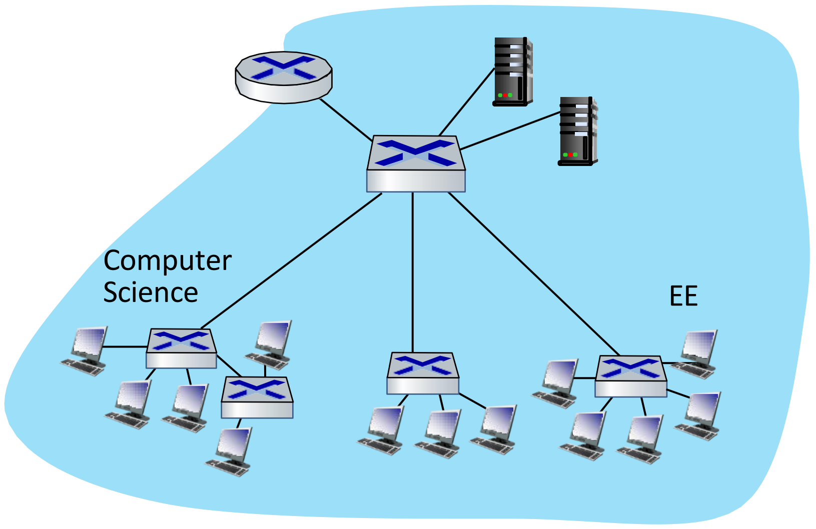
S – Selected port

Virtual LANs (VLANs): Motivation

Q: what happens as LAN sizes scale?

single broadcast domain:

- *scaling:* all layer-2 broadcast traffic (e.g., unknown MAC) must cross entire LAN
- Efficiency, security, privacy issues

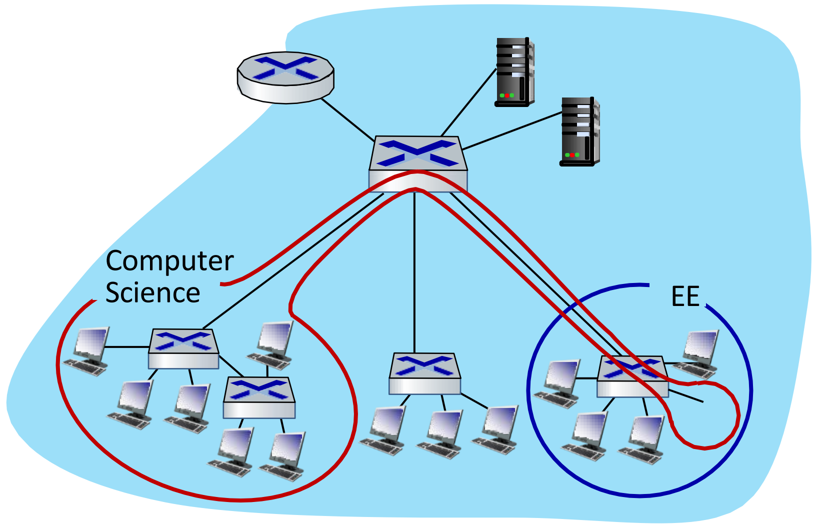


Virtual LANs (VLANs): Motivation

Q: what happens as LAN sizes scale, users change point of attachment?

Single broadcast domain:

- *scaling:* all layer-2 broadcast traffic (unknown MAC) must cross entire LAN
- efficiency, security, privacy, efficiency issues



Administrative issues:

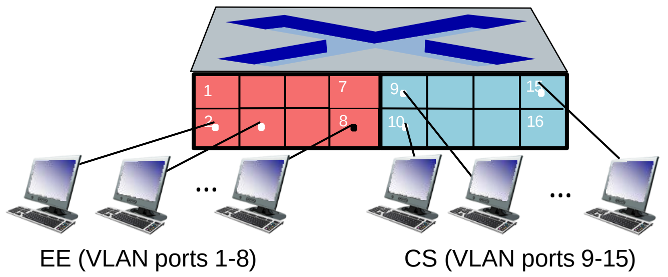
- CS user moves office to EE - *physically* attached to EE switch, but wants to remain *logically* attached to CS switch

Port-based VLANs

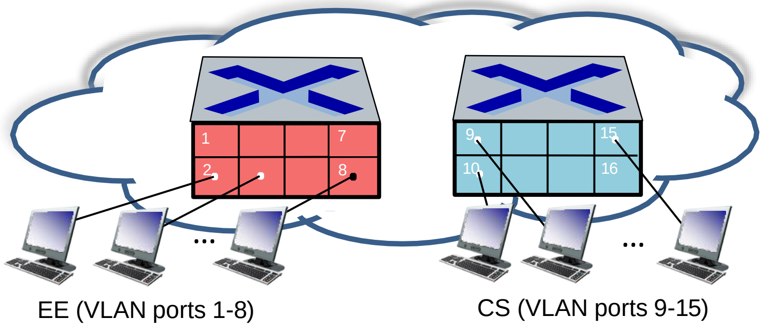
Virtual Local Area Network (VLAN)

switch(es) supporting VLAN capabilities can be configured to define multiple *virtual* LANS over single physical LAN infrastructure.

port-based VLAN: switch ports grouped (by switch management software) so that *single* physical switch ……

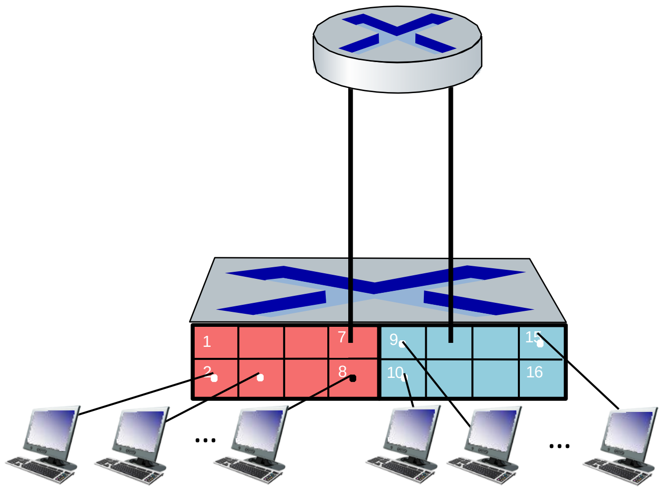


… operates as multiple virtual switches



Port-based VLANs

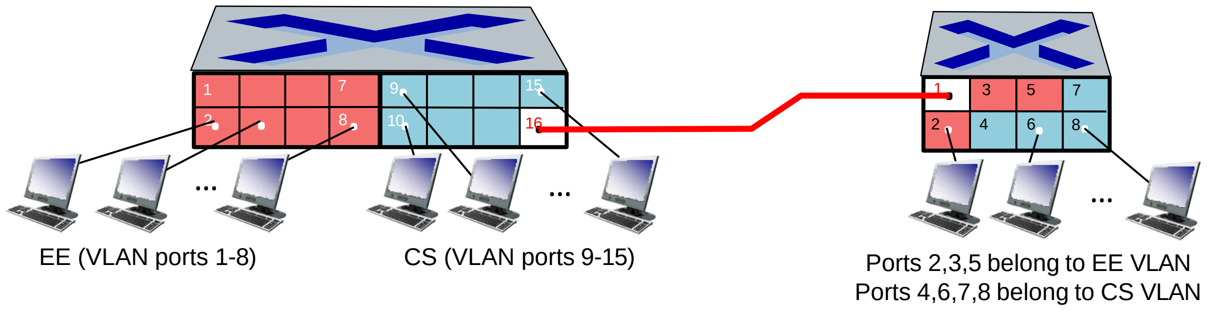
- Traffic isolation: frames to/from ports 1-8 can *only* reach ports 1-8
  - can also define VLAN based on MAC addresses of endpoints, rather than switch port
- Dynamic membership: ports can be dynamically assigned among VLANs
- Forwarding between VLANS: done via routing (just as with separate switches)
  - in practice vendors sell combined switches plus routers



EE (VLAN ports 1-8)

CS (VLAN ports 9-15)

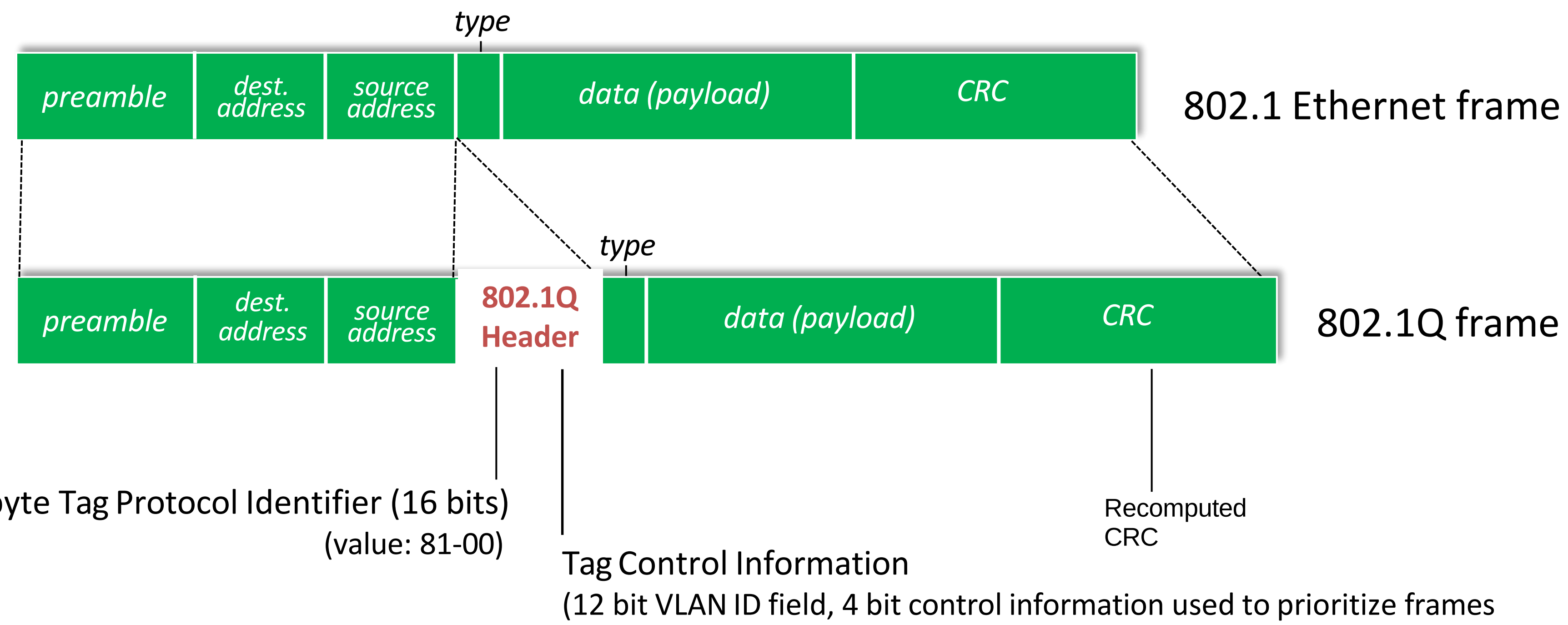
VLANS spanning multiple switches


1
7
2
8
9
15
10
16
…
EE (VLAN ports 1-8)
CS (VLAN ports 9-15)
1
3
5
7
2
4
6
8
…
Ports 2,3,5 belong to EE VLAN
Ports 4,6,7,8 belong to CS VLAN

Trunk port: carries frames between VLANS defined over multiple physical switches

- frames forwarded within VLAN between switches must carry VLAN ID info
- 802.1q protocol to support VLANs..

802.1Q VLAN frame format



Q: How many possible VLANs can be mapped onto one physical LAN?