Last Lecture: Data Link Layer

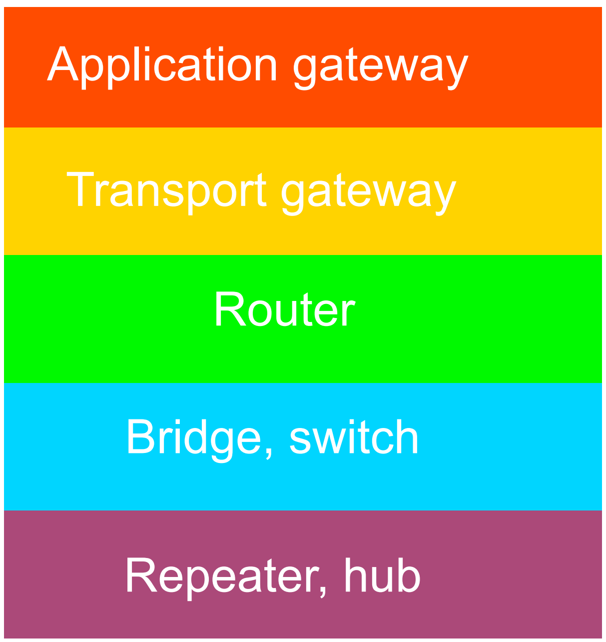
1. *Design goals and issues*
2. *(More on) Error Control and Detection*
3. *Multiple Access Control (MAC)*
4. *Ethernet, LAN Addresses and ARP* ✔
5. *Hubs, Bridges, Switches*
6. *Wireless LANs*
7. WLAN Security
8. Mobile Networking

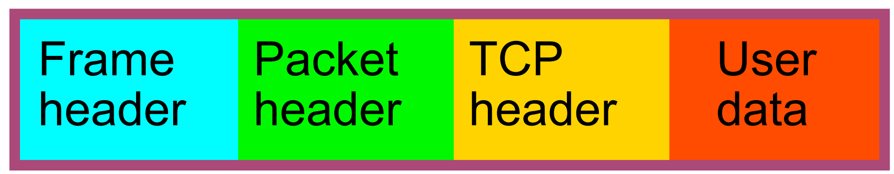
This Lecture: Data Link Layer

1. *Design goals and issues*
2. *(More on) Error Control and Detection*
3. *Multiple Access Control (MAC)*
4. *Ethernet, LAN Addresses and ARP*
5. *Hubs, Bridges, Switches* ✔
   - Credits: some slides from Jennifer Rexford @ Princeton
6. *Wireless LANs*
7. WLAN Security
8. Mobile Networking

Shuttling Data at Different Layers

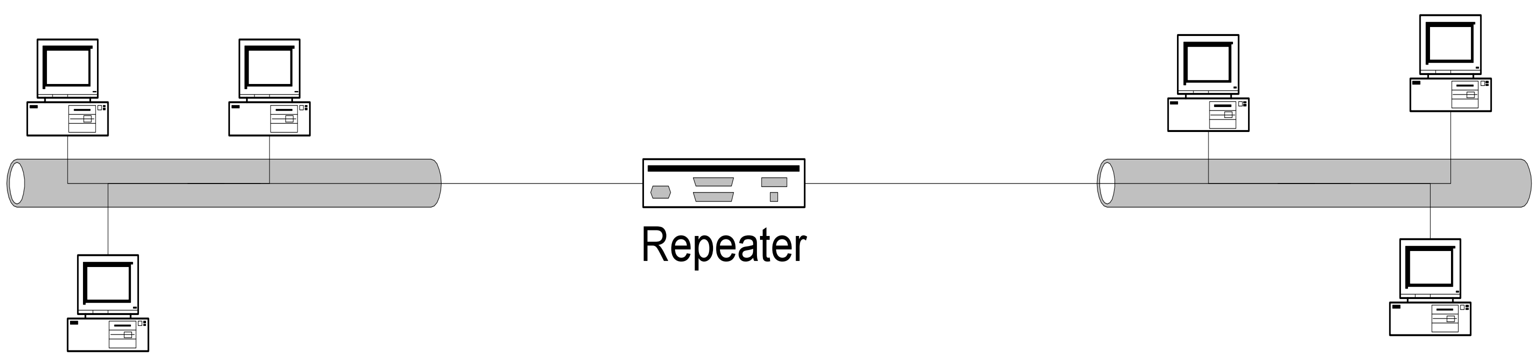
Layer	Unit type	Device
Network	Packets	Routers
Datalink	Frames	Switches, Bridges
Physical	Electrical signals	Repeaters, Hubs





Physical Layer: Repeaters

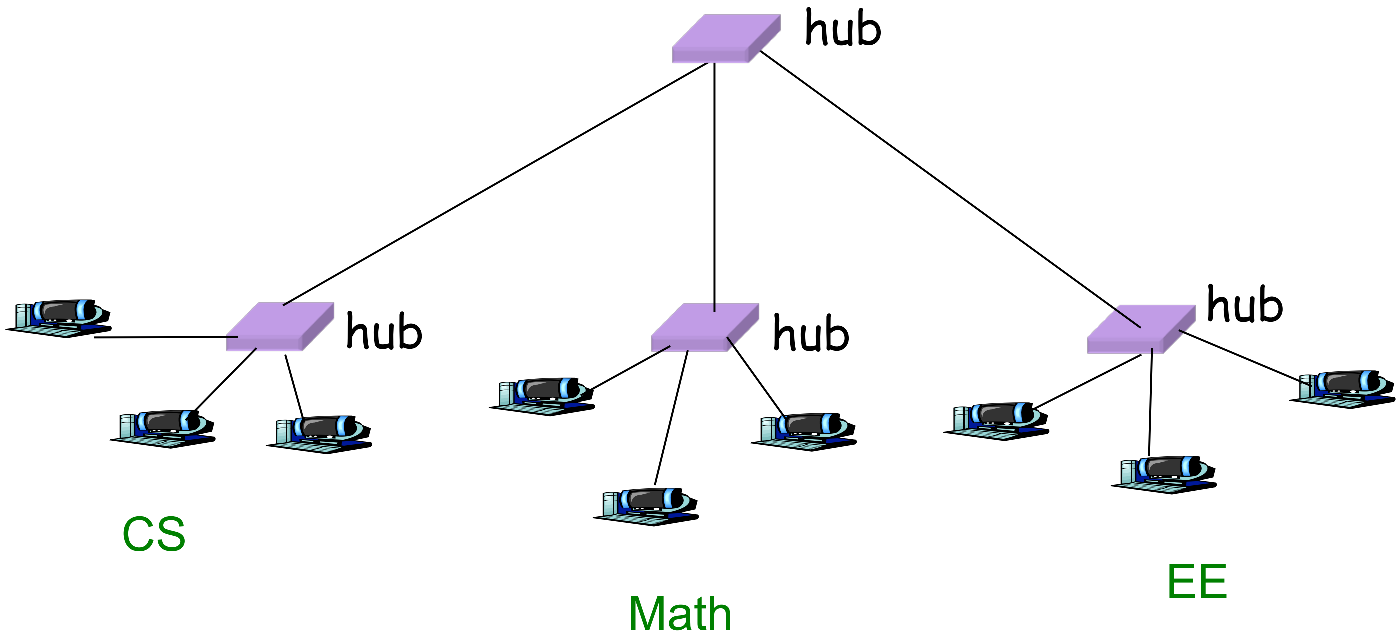
- Distance limitation in local-area networks
  - Electrical signal becomes weaker as it travels
  - Imposes a limit on the length of a LAN
- *Repeaters* join LANs together
  - Analog electronic device
  - Continuously monitors electrical signals on each LAN
  - *Transmits an amplified copy*



Magnum 200X Two-Port Repeater

Physical Layer: Hubs

- Joins multiple input lines electrically
  - Designed to hold multiple line cards
  - *Do not necessarily amplify the signal*
- Very similar to repeaters
  - Also operates at the physical layer



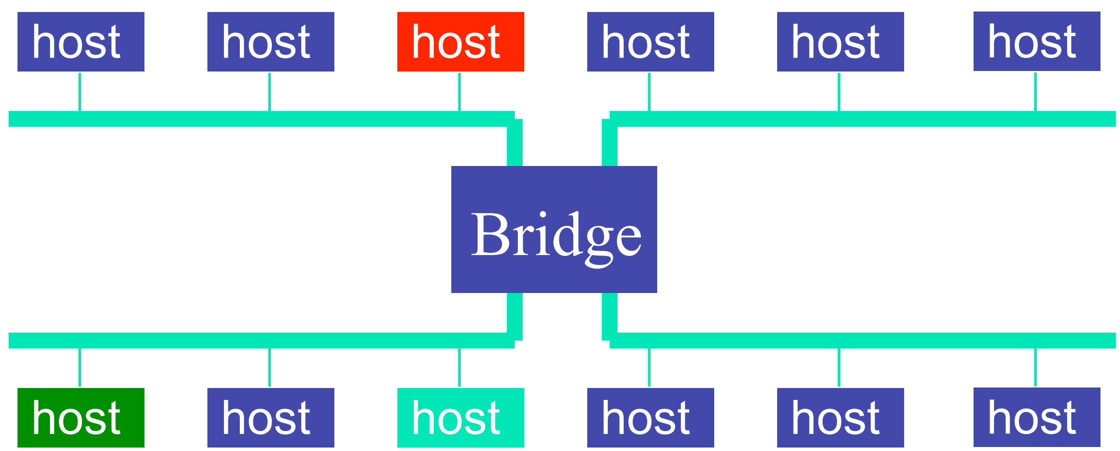
Magnum 3000 Series Stackable Hubs

Limitations of Repeaters and Hubs

- *One large shared link, thus throughput limited*
  - Each bit is sent everywhere
  - E.g., three departments each get 10 Mbps independently
  - … and then connect via a hub and must share 10 Mbps
- *Cannot support multiple LAN technologies*
  - Does not buffer or interpret frames
  - So, can't interconnect between different rates or formats
  - E.g., 10 Mbps Ethernet and 100 Mbps Ethernet
- *Limitations on maximum nodes and distances*
  - Shared medium imposes length limits ($2\tau R$)
  - E.g., cannot go beyond *2500* meters on Ethernet

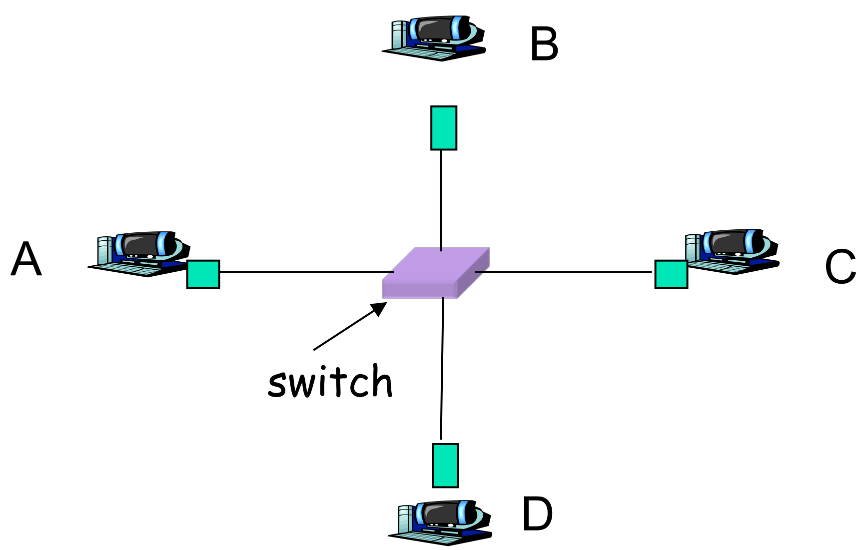
Link Layer: Bridges

- Connects two or more LANs at the link layer
  - Extracts destination address from the frame
  - Looks up the destination in a table
  - Forwards the frame to the appropriate LAN segment
- Each segment can carry its own traffic



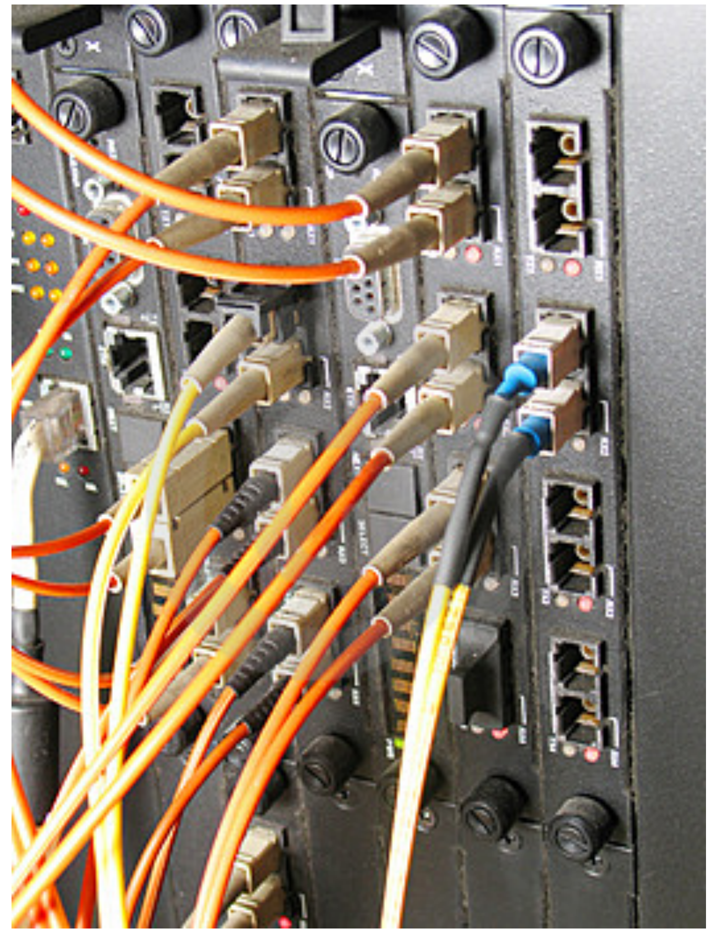
Link Layer: Switches

- Typically connects individual computers
  - A switch is essentially the same as a bridge
  - … though typically used to connect hosts, not LANs
- Like bridges, support concurrent communications
  - Host A can talk to C, while B talks to D



Netgear PE102 Ethernet/PNA Bridge

Some Modern Switches



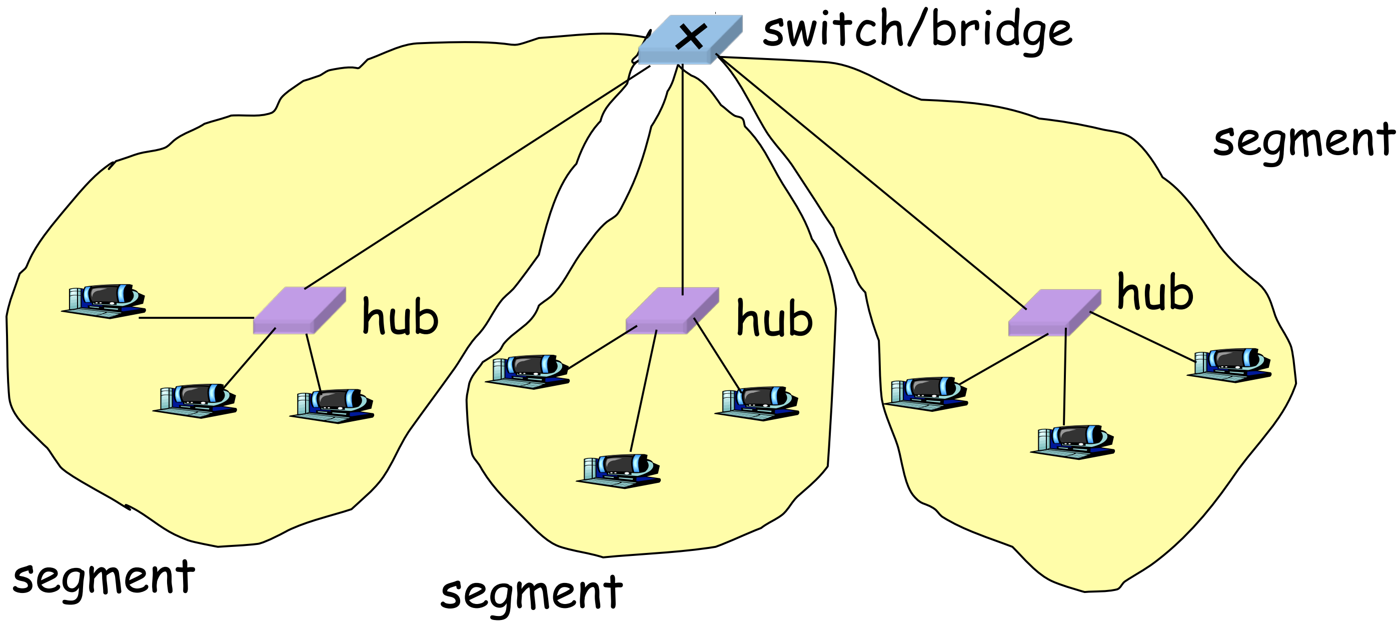
Cisco Nexus 7000 Network Switch
(15 Tbps total capacity)

Dedicated Access and Full Duplex

- *Dedicated access*
  - Host has direct connection to the switch
  - … rather than a shared LAN connection
- *Full duplex*
  - Each connection can send in both directions
  - Host sending to switch, and host receiving from switch
  - E.g., in 10BaseT and 100Base T
- Completely supports concurrent transmissions
  - Each connection is a bidirectional point-to-point link

Bridges/Switches: Traffic Isolation

- Switch breaks subnet into LAN segments
- Switch filters packets
  - Frame only forwarded to the necessary segments
  - Segments can support separate (concurrent) transmissions



Advantages Over Hubs/Repeaters

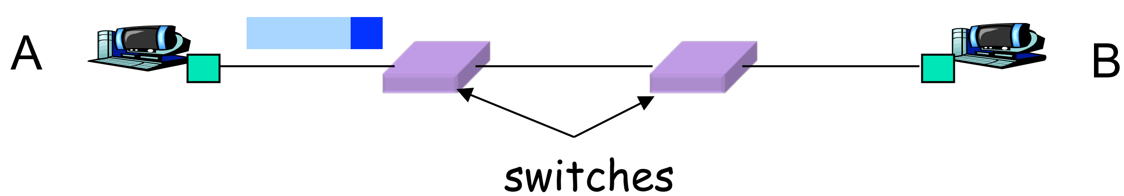
- Only forwards frames as needed, higher throughput
  - Filters frames to avoid unnecessary load on segments
  - Sends frames only to segments that need to see them
- Extends the geographic span of the network
  - Separate segments allow longer distances
- Improves security by limiting scope of frames
  - Hosts can "snoop" the traffic traversing their segment
  - … but not all the rest of the traffic
- Can join segments using different technologies

Disadvantages Over Hubs/Repeaters

- Delay in forwarding frames
  - Bridge/switch must receive and parse the frame
  - … and perform a look-up to decide where to forward
  - Storing and forwarding the packet introduces delay
  - Solution: *cut-through switching*
- Need to learn where to forward frames
  - Bridge/switch needs to construct a forwarding table
  - Ideally, without intervention from network administrators
  - Solution: *self-learning*
- Higher cost
  - More complicated devices that *cost more* money

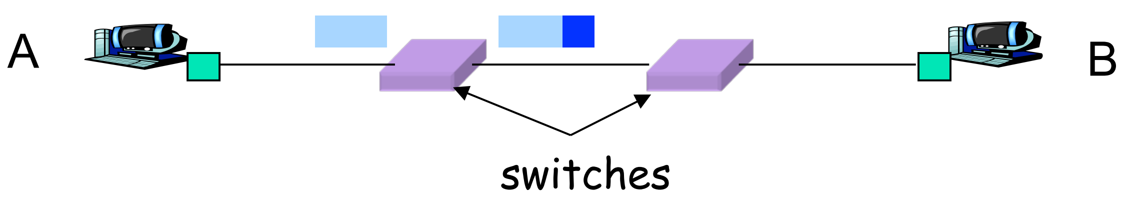
Motivation For Cut-Through Switching

- Buffering a frame takes time
  - Suppose L is the length of the frame
  - And R is the transmission rate of the links
  - Then, receiving the frame takes L/R time units
- Buffering delay can be a high fraction of total delay
  - Propagation delay is small over short distances
  - Making buffering delay a large fraction of total



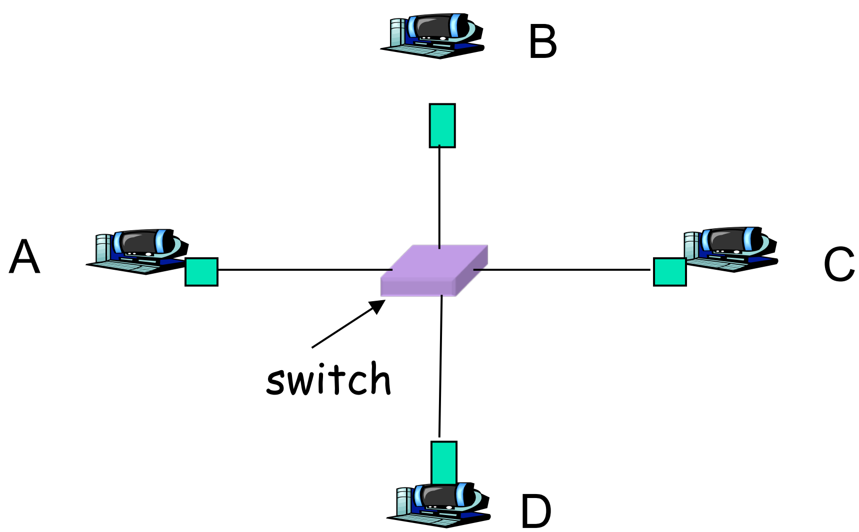
Cut-Through Switching

- Start transmitting as soon as possible
  - Inspect the frame header and do the look-up
  - If outgoing link is idle, start forwarding the frame
- Overlapping transmissions
  - Transmit the head of the packet via the outgoing link
  - … while still receiving the tail via the incoming link
  - Analogy: different folks crossing different intersections



Motivation For Self Learning

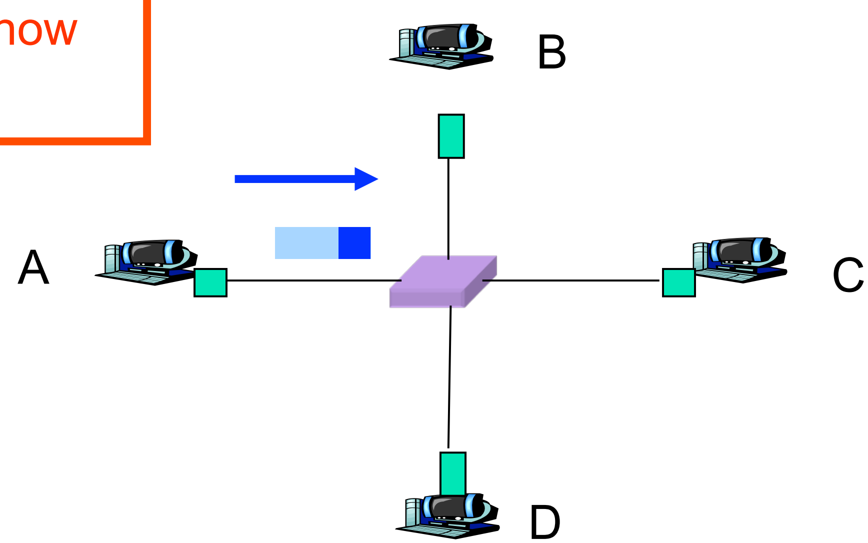
- Switches forward frames selectively
  - Forward frames only on segments that need them
- Switch table
  - Maps destination MAC address to outgoing interface
  - Goal: construct the switch table automatically



Self Learning: Building the Table

- When a frame arrives
  - Inspect the *source* MAC address
  - Associate the address with the *incoming* interface
  - Store the mapping in the switch table
  - Use a time-to-live field to eventually forget the mapping

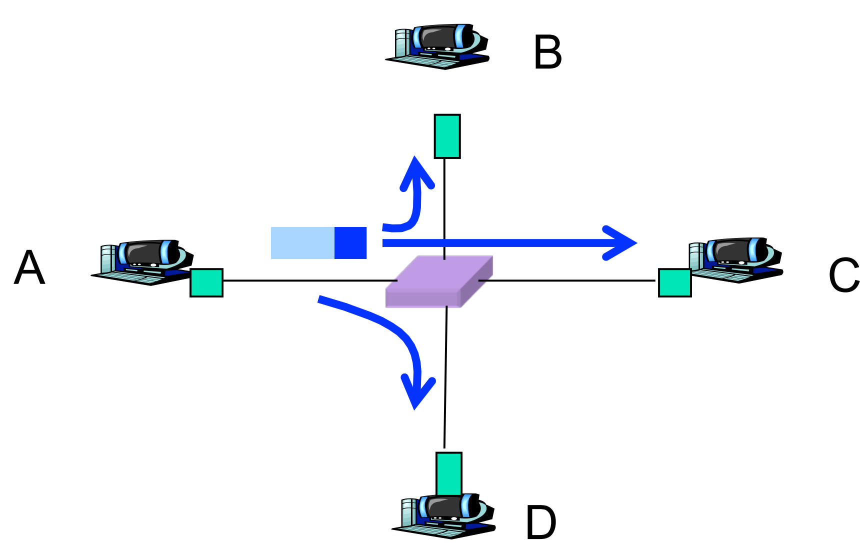
Switch learns how to reach A.



Self Learning: Handling Misses

- When frame arrives with unfamiliar destination
  - Forward the frame out all of the interfaces
  - … except for the one where the frame arrived
  - Hopefully, this case won't happen very often

When in doubt, shout!



Switch Filtering/Forwarding

When switch receives a frame:

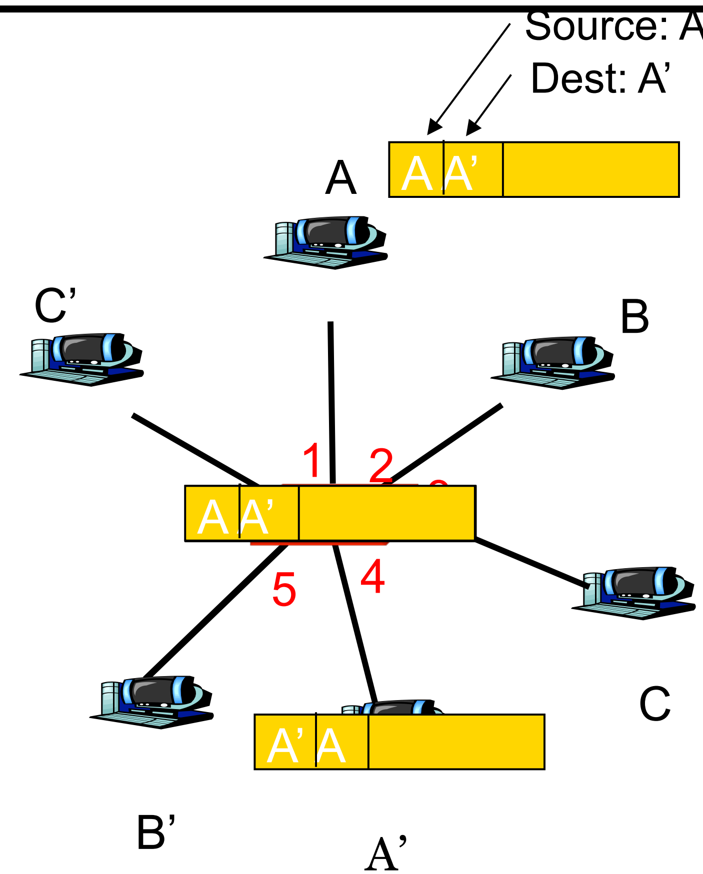
Index switch table using MAC destination address

if entry found for destination
  then{
   if dest on segment from which frame arrived
     then drop the frame
     else forward the frame on interface indicated
   }
  else flood

forward on all but the interface on which the frame arrived

Self-learning, forwarding: example

- Frame destination unknown: *flood*
- Frame destination known: *selective forward*

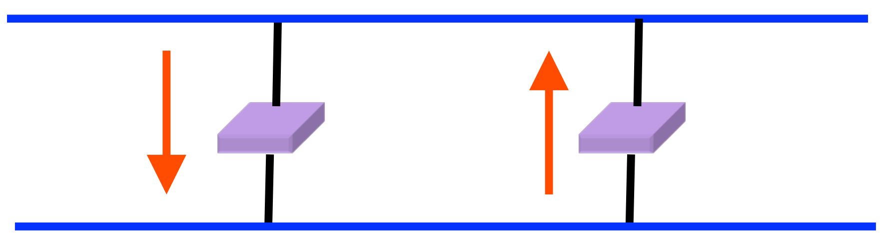


MAC addr	interface	TTL
A	1	60
A'	4	60

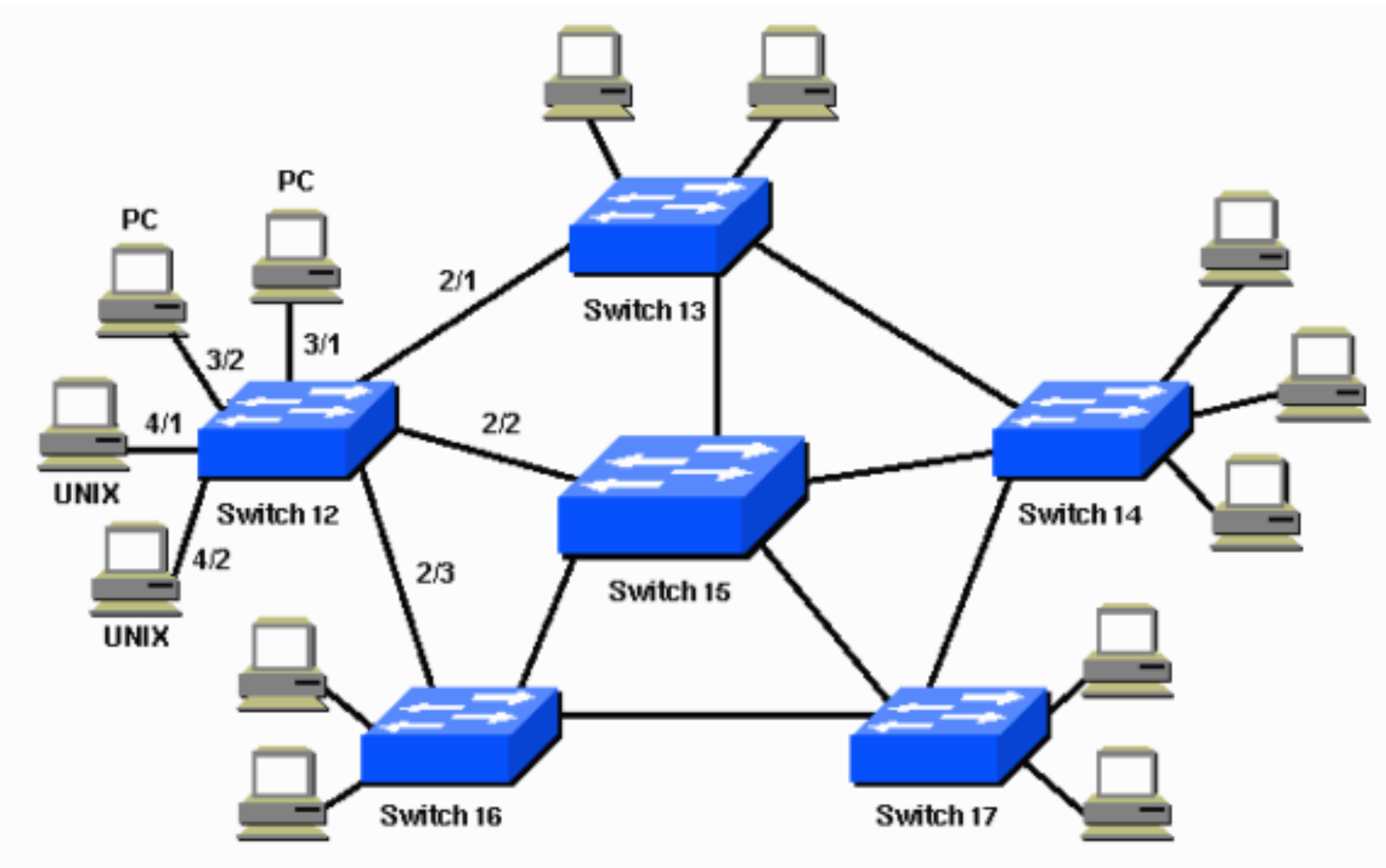
Switch table (initially empty)

Flooding Can Lead to Loops

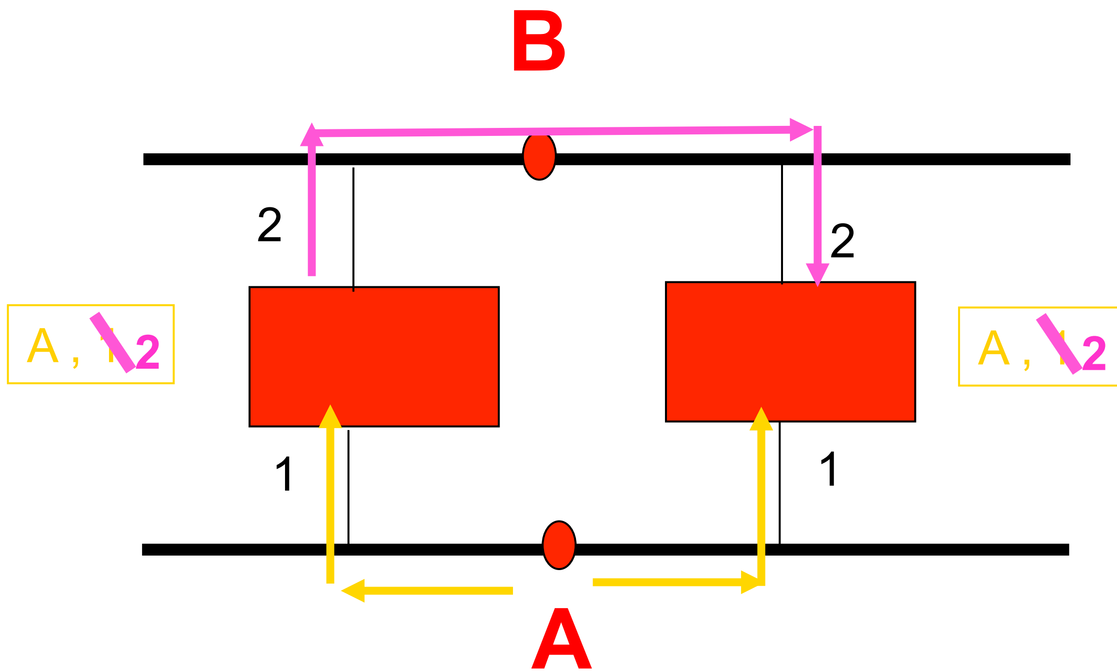
- Switches sometimes need to broadcast frames
  - Upon receiving a frame with an unfamiliar destination
  - Upon receiving a frame sent to the broadcast address
- Broadcasting is implemented by flooding
  - Transmitting frame out every interface
  - … except the one where the frame arrived
- Flooding can lead to forwarding loops
  - E.g., if the network contains a cycle of switches
  - Either accidentally, or by design for higher reliability



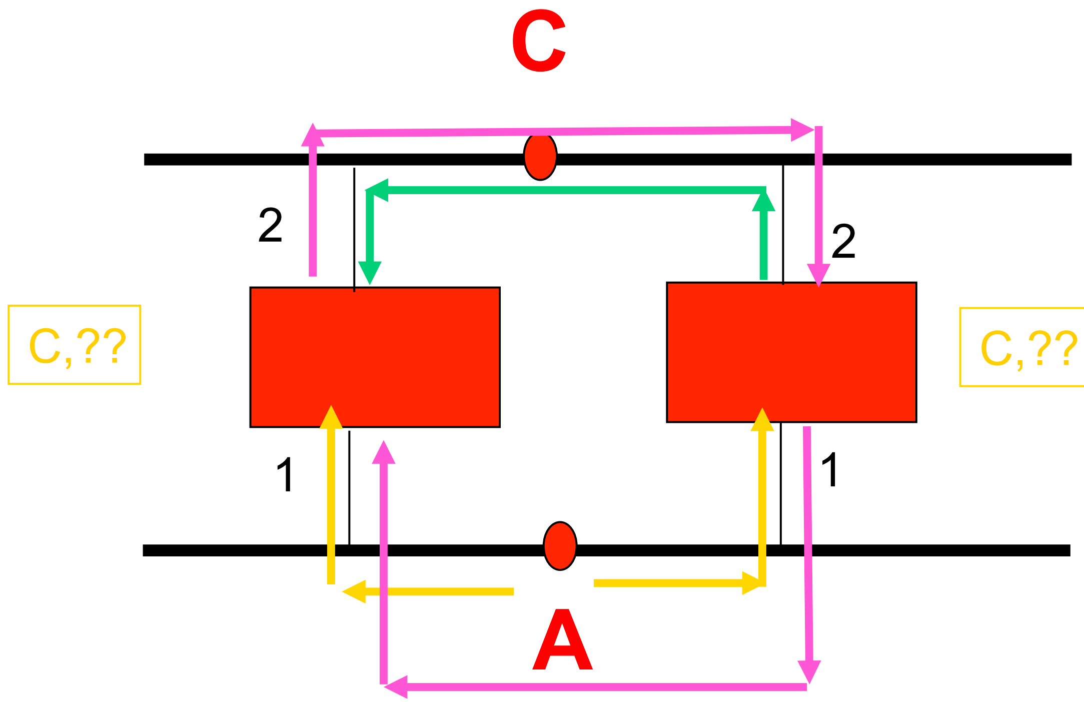
For Instance



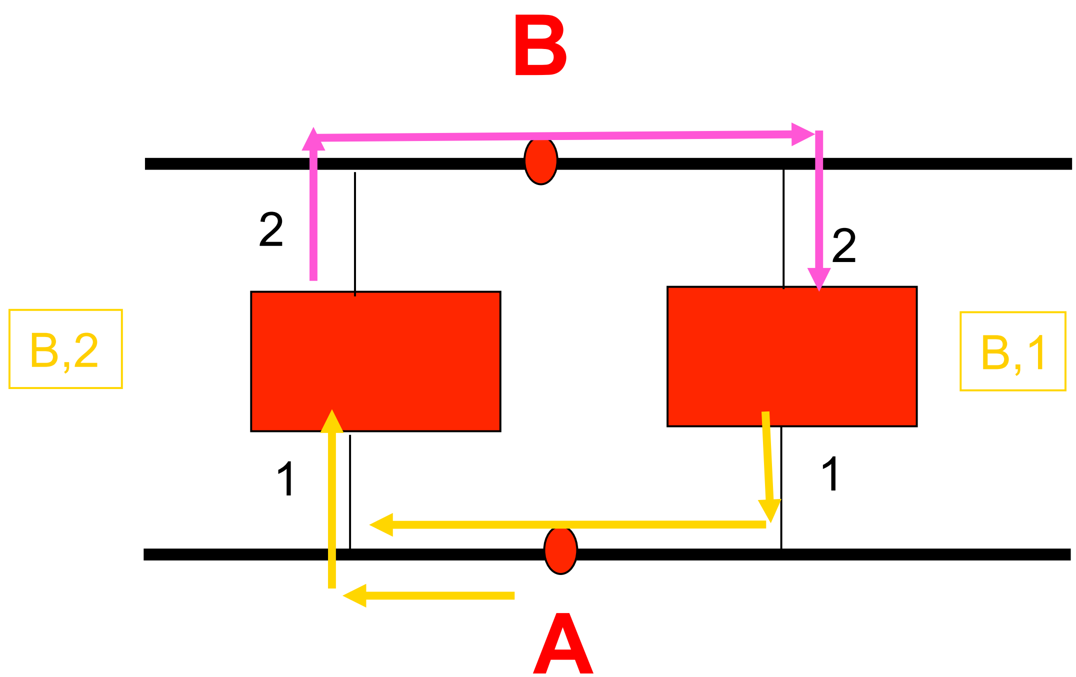
Loops ➔ Incorrect Learning



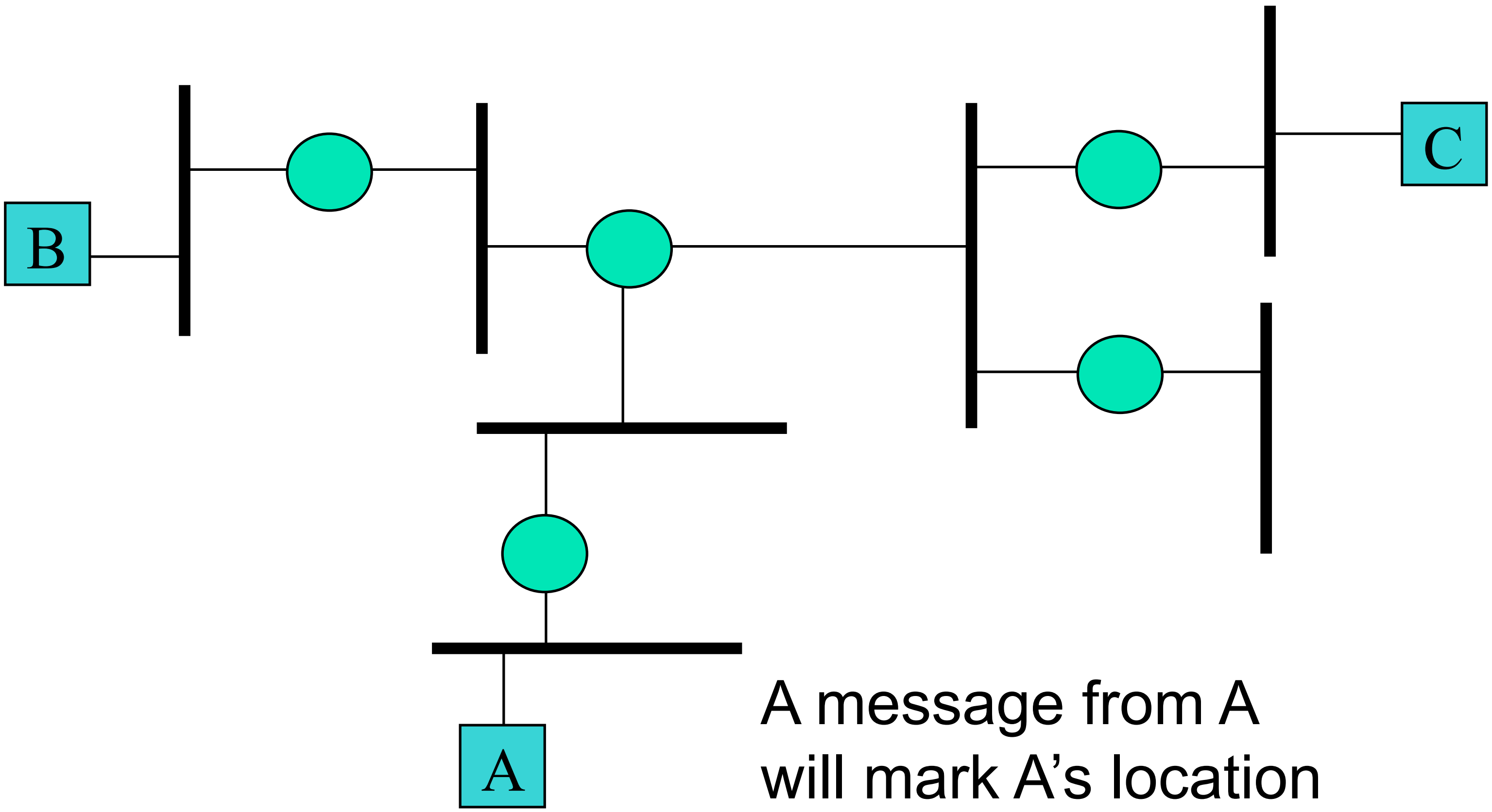
Loops ➔ Frame Looping



Loops: Frame looping

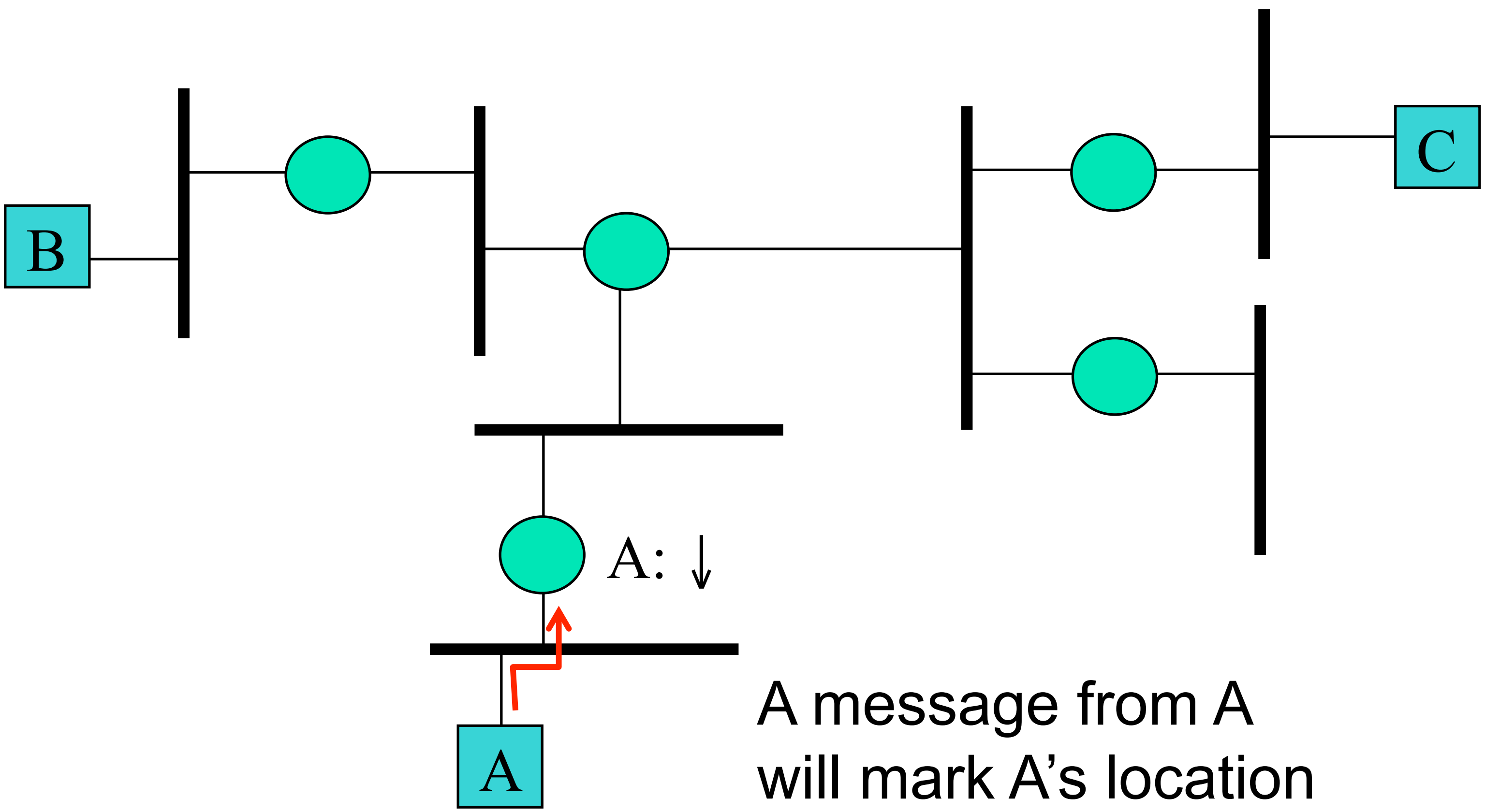


Loop-free: tree



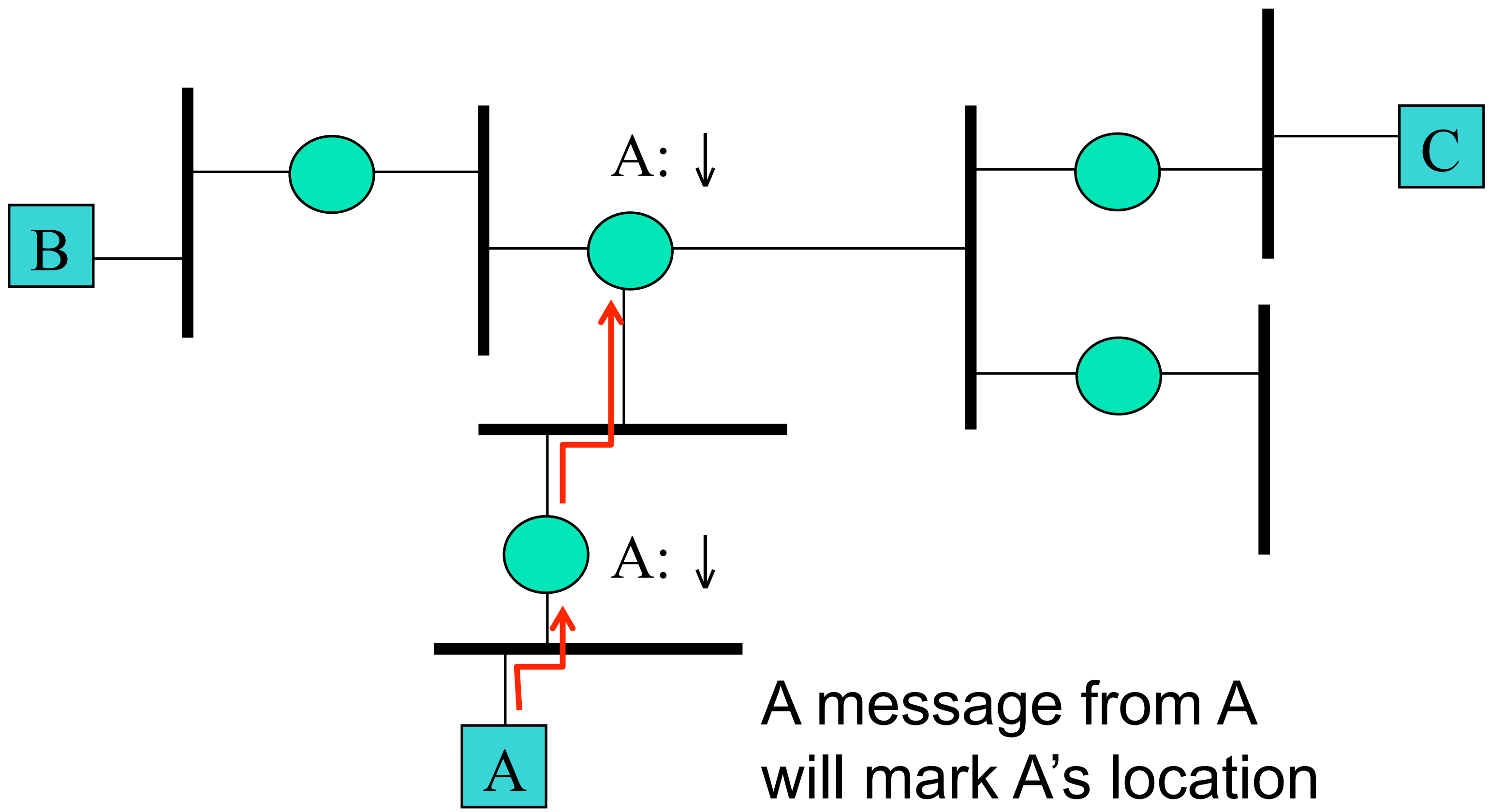
A message from A
will mark A's location

Loop-free: tree



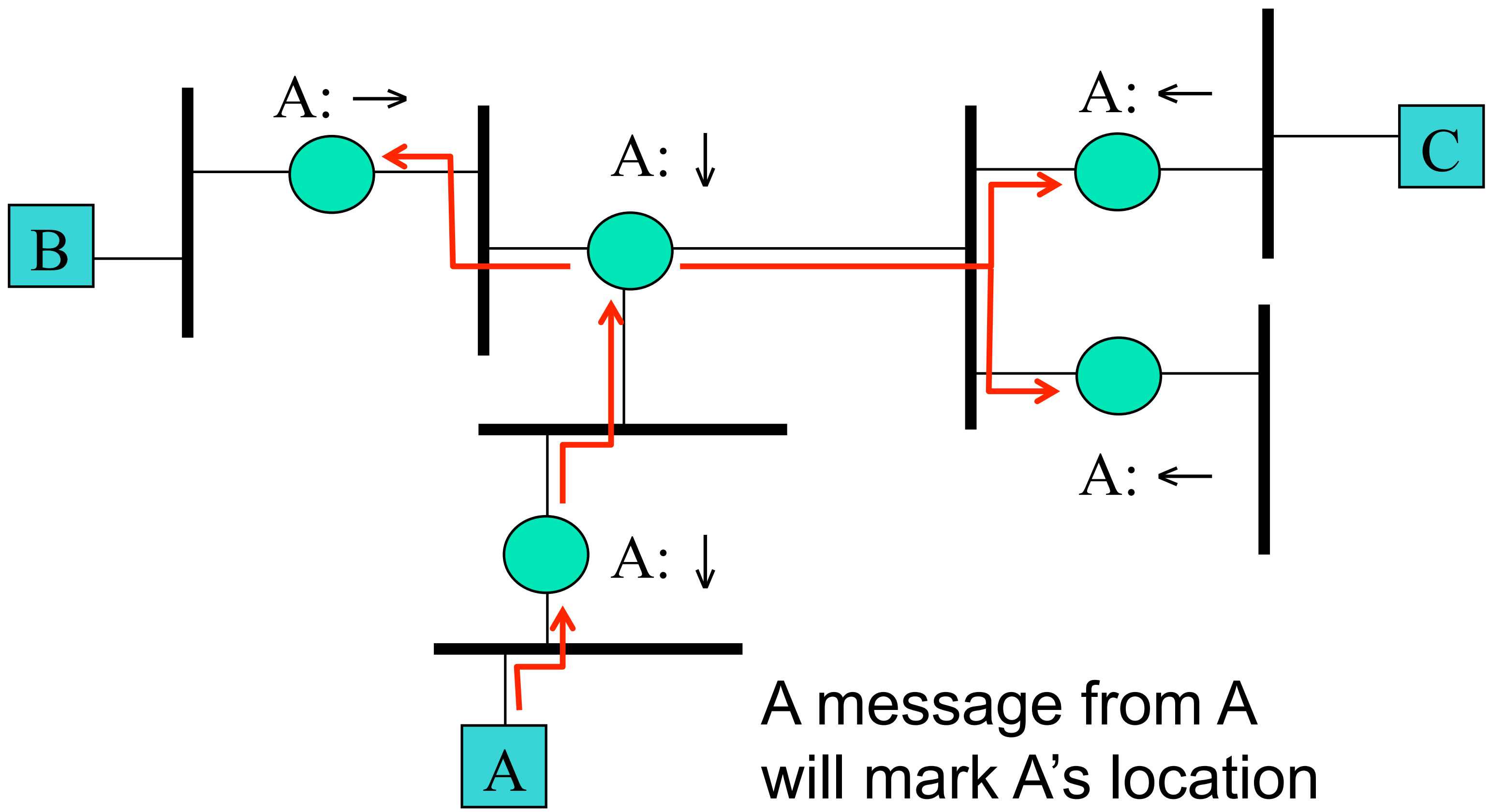
A message from A
will mark A's location

Loop-free: tree



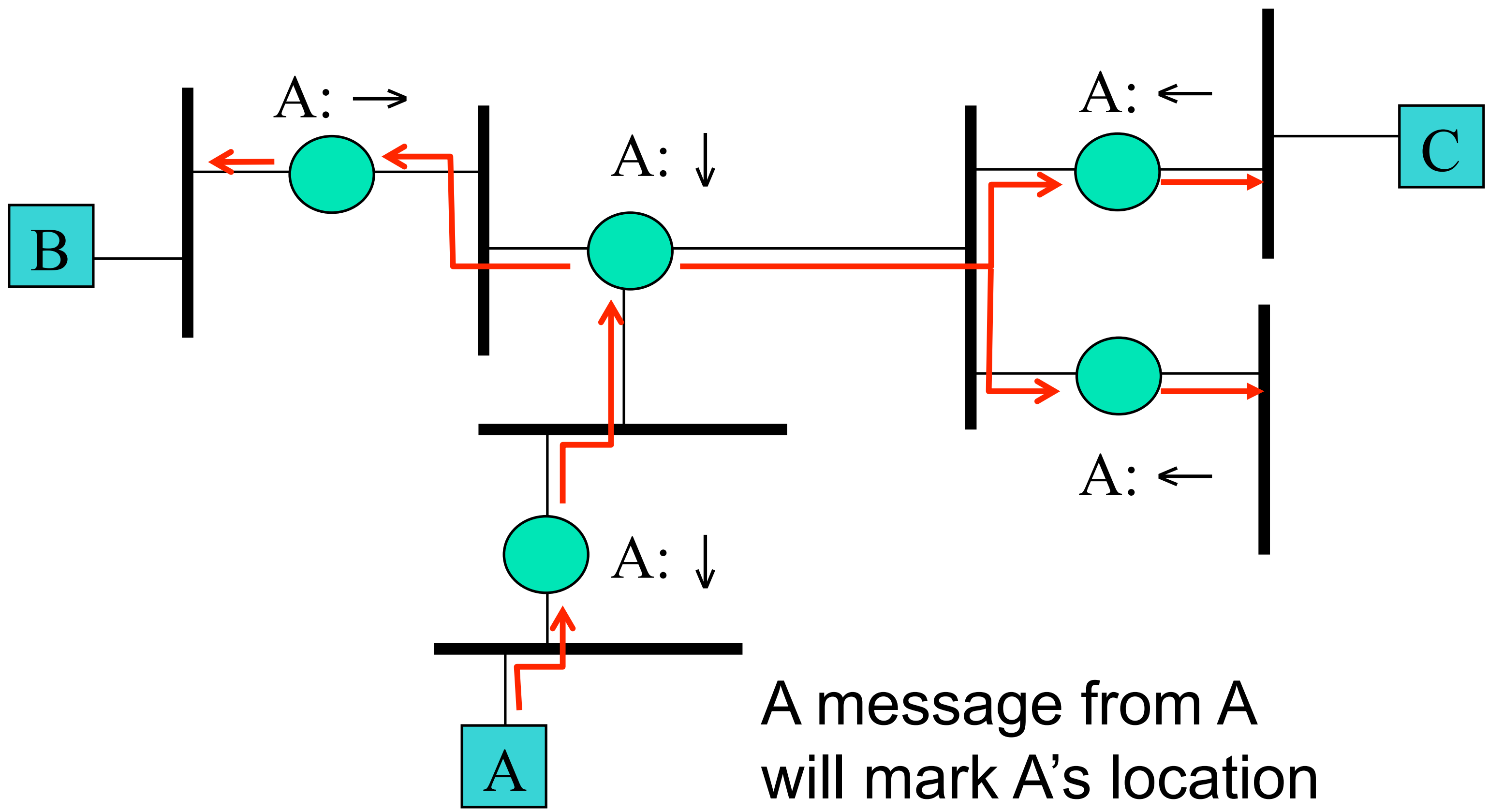
A message from A
will mark A's location

Loop-free: tree



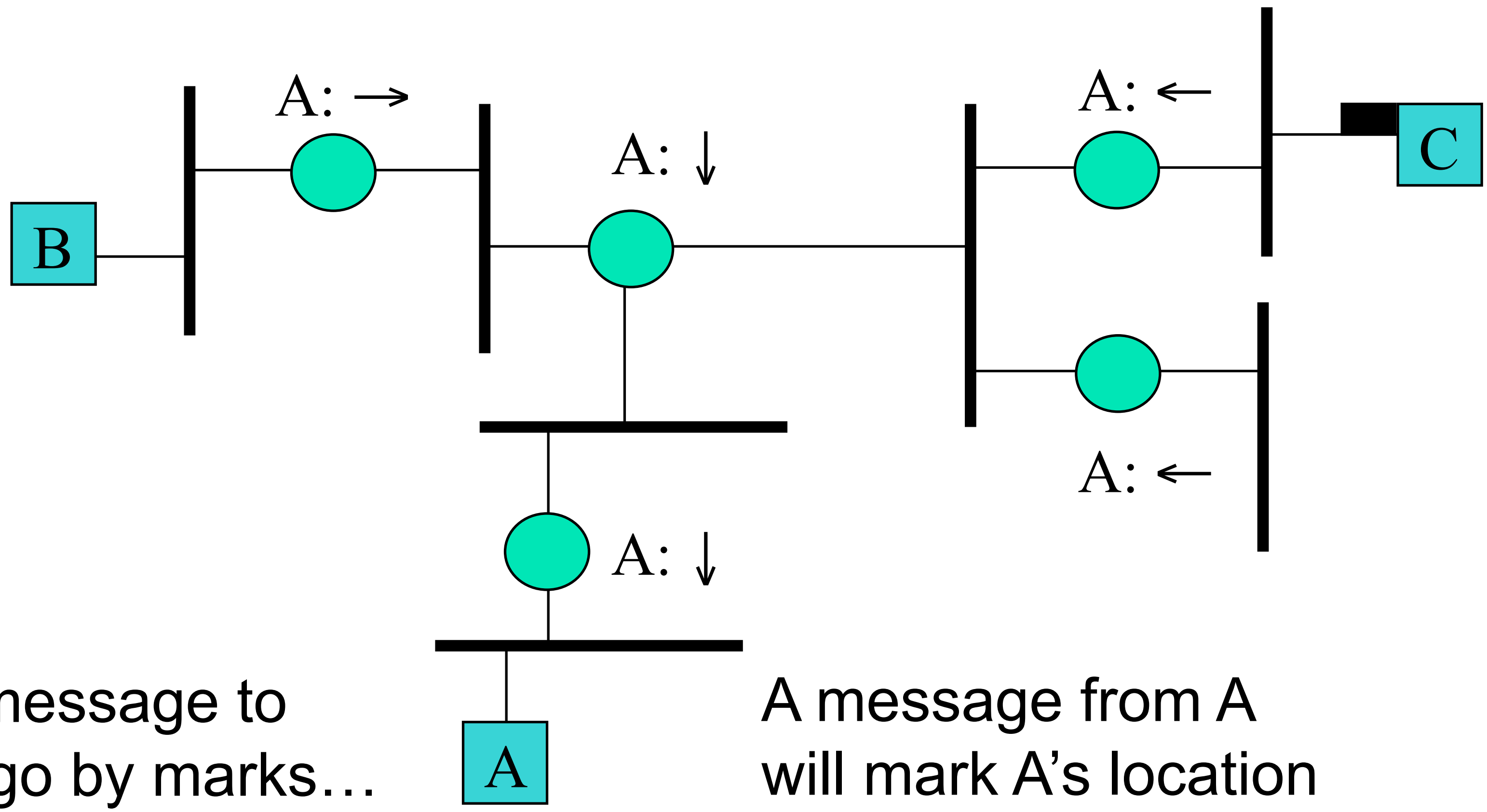
A message from A will mark A's location

Loop-free: tree



A message from A
will mark A's location

Loop-free: tree

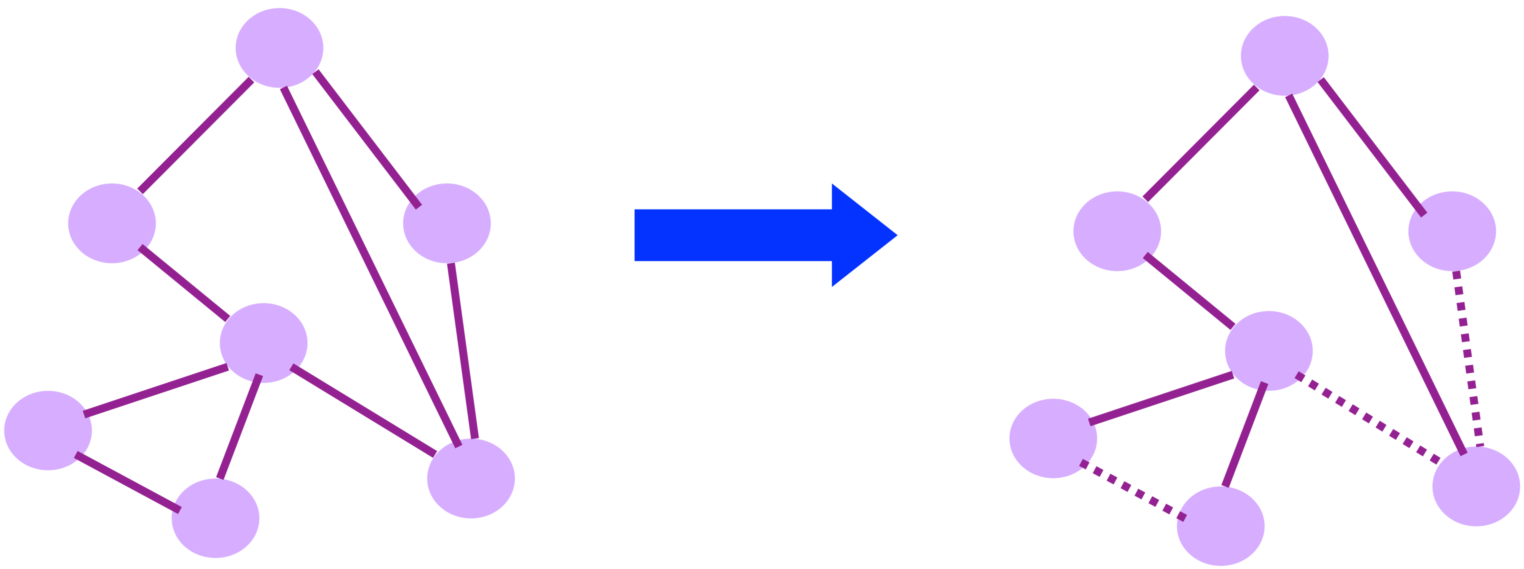


So a message to
A will go by marks…

A message from A
will mark A's location

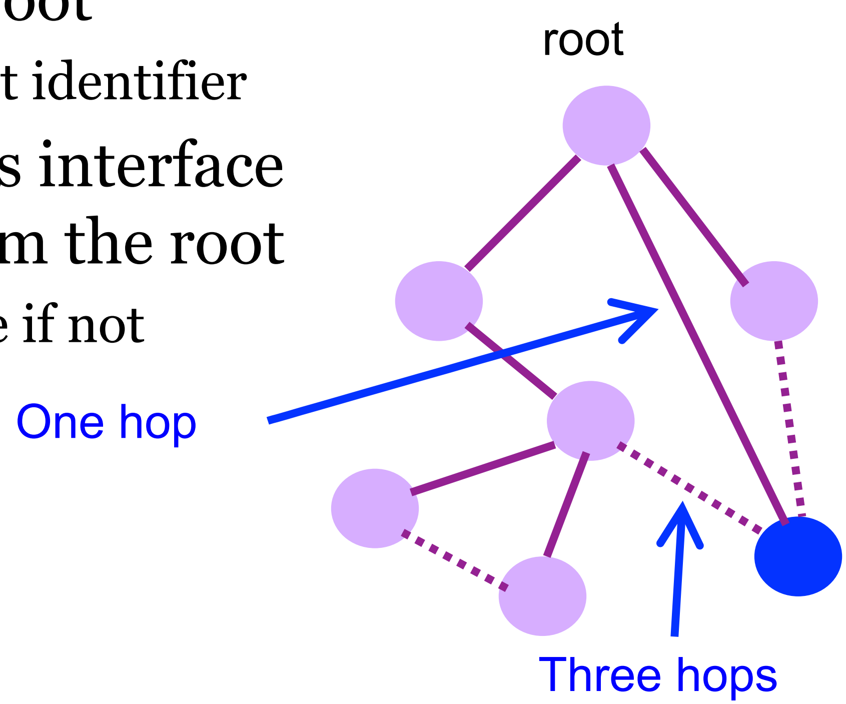
Solution: Spanning Trees

- Ensure the topology has no loops
  - Avoid using some of the links when flooding
  - … to avoid forming a loop
- *Spanning tree*
  - Sub-graph that covers all vertices but contains no cycles
  - Links not in the spanning tree do not forward frames



Constructing a Spanning Tree

- Need a distributed algorithm
  - Switches cooperate to build the spanning tree
  - … and adapt automatically when failures occur
- Key ingredients of the algorithm
  - Switches need to elect a “root”
    - The switch with the smallest identifier
  - Each switch identifies if its interface is on the shortest path from the root
    - And it exclude from the tree if not
  - Messages (Y, d, X)
    - From node X
    - Claiming Y is the root
    - And the distance is d

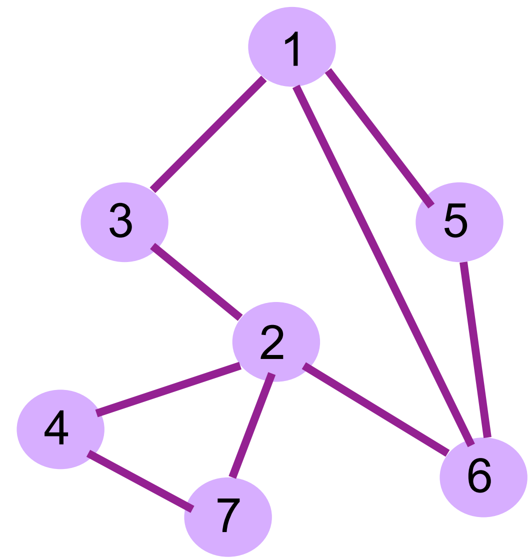


Steps in Spanning Tree Algorithm

- Initially, each switch thinks it is the root
  - Switch sends a message out every interface
  - … identifying itself as the root with distance 0
  - Example: switch X announces (X, 0, X)
- Switches update their view of the root
  - Upon receiving a message, check the root id
  - If the new id is smaller, start viewing that switch as root
- Switches compute their distance from the root
  - Add 1 to the distance received from a neighbor
  - Identify interfaces not on a shortest path to the root
  - … and exclude them from the spanning tree

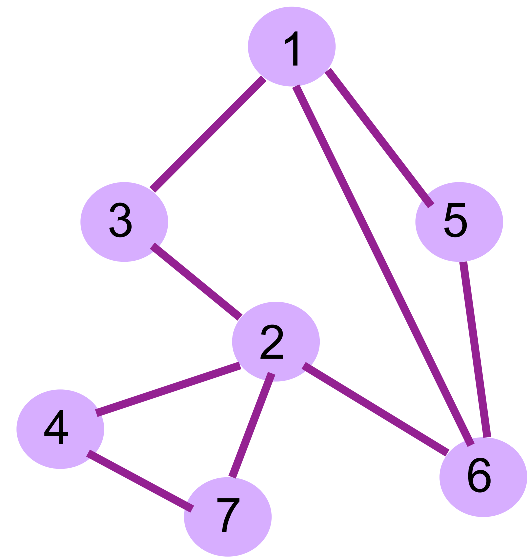
Example From Switch #4's Viewpoint

- Switch #4 thinks it is the root
  - Sends (4, 0, 4) message to 2 and 7
- Then, switch #4 hears from #2
  - Receives (2, 0, 2) message from 2
  - … and thinks that #2 is the root
  - And realizes it is just one hop away
- Then, switch #4 hears from #7
  - Receives (2, 1, 7) from 7
  - And realizes this is a longer path
  - So, prefers its own one-hop path
  - And removes 4-7 link from the tree



Example From Switch #4's Viewpoint

- Switch #2 hears about switch #1
  - Switch 2 hears (1, 1, 3) from 3
  - Switch 2 starts treating 1 as root
  - And sends (1, 2, 2) to neighbors
- Switch #4 hears from switch #2
  - Switch 4 starts treating 1 as root
  - And sends (1, 3, 4) to neighbors
- Switch #4 hears from switch #7
  - Switch 4 receives (1, 3, 7) from 7
  - And realizes this is a longer path
  - So, prefers its own three-hop path
  - And removes 4-7 Iink from the tree



Robust Spanning Tree Algorithm

- Algorithm must react to failures
  - Failure of the root node
    - Need to elect a new root, with the next lowest identifier
  - Failure of other switches and links
    - Need to recompute the spanning tree
- Root switch continues sending messages
  - Periodically reannouncing itself as the root (1, 0, 1)
  - Other switches continue forwarding messages
- Detecting failures through timeout (soft state!)
  - Switch waits to hear from others
  - Eventually times out and claims to be the root

Evolution Toward Virtual LANs

- In the olden days…
  - Thick cables snaked through cable ducts in buildings
  - Every computer they passed was plugged in
  - All people in adjacent offices were put on the same LAN
  - Independent of whether they belonged together or not
- More recently…
  - Hubs and switches changed all that
  - Every office connected to central wiring closets
  - Often multiple LANs (*k* hubs) connected by switches
  - Flexibility in mapping offices to different LANs

> Group users based on organizational structure, rather than the physical layout of the building.

Why Group by Organizational Structure?

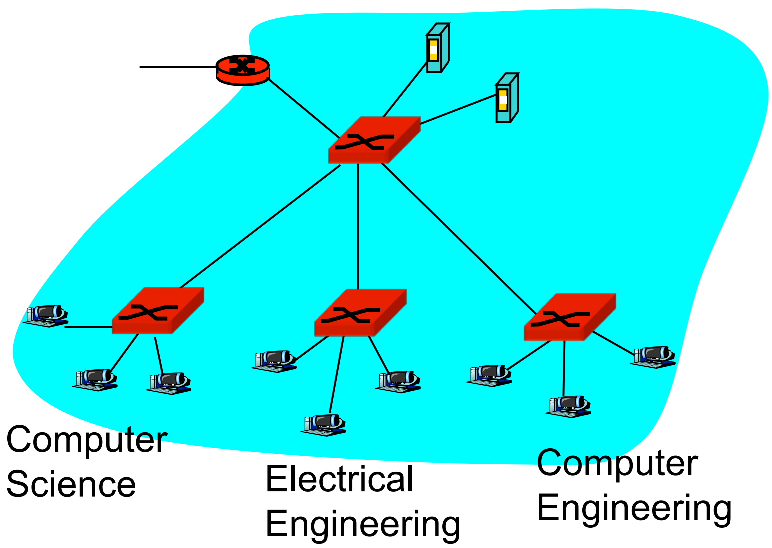
- Security
  - Ethernet is a shared media
  - Any interface card can be put into "promiscuous" mode
  - … and get a copy of all of the traffic (e.g., final exam)
  - So, isolating traffic on separate LANs improves security
- Load
  - Some LAN segments are more heavily used than others
  - E.g., researchers running experiments get out of hand
  - … can saturate their own segment and not the others
  - Plus, there may be natural locality of communication
  - E.g., traffic between people in the same research group

People Move, and Roles Change

- Organizational changes are frequent
  - E.g., faculty office becomes a grad-student office
  - E.g., graduate student becomes a faculty member
- Physical rewiring is a major pain
  - Requires unplugging the cable from one port
  - … and plugging it into another
  - … and hoping the cable is long enough to reach
  - … and hoping you don't make a mistake
- Would like to "rewire" the building in software
  - The resulting concept is a *Virtual LAN (VLAN)*

VLANs: motivation

What's wrong with this picture?

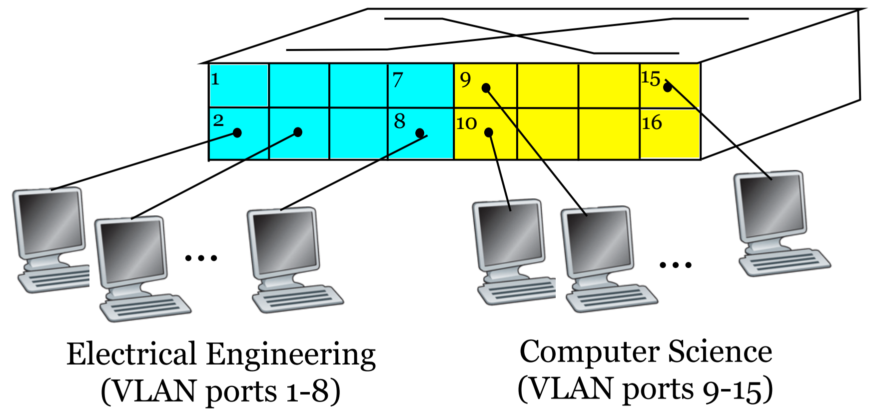


What happens if:

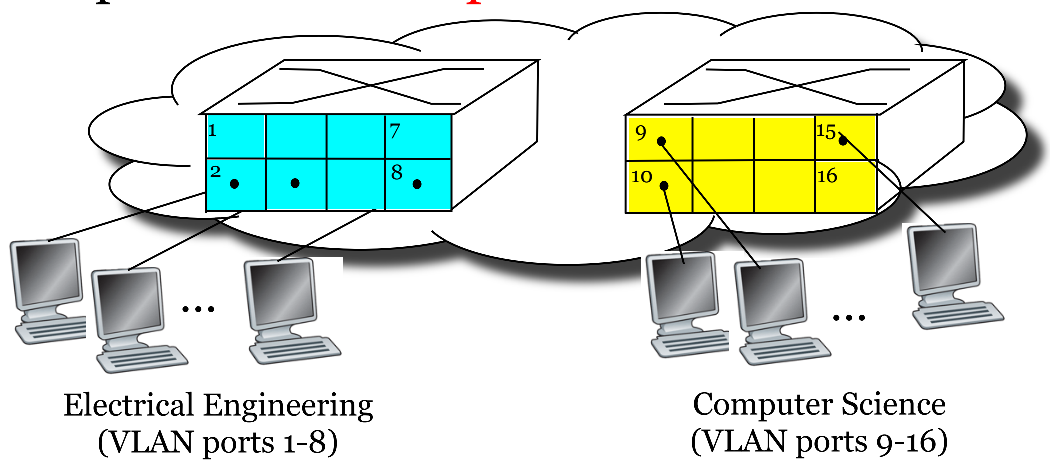
- CS user moves office to EE, but wants connect to CS switch?
- single broadcast domain:
  - all layer-2 broadcast traffic (ARP, DHCP) crosses entire LAN (security/privacy, efficiency issues)
- each lowest level switch has only few ports in use

VLANs

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



… operates as multiple virtual switches

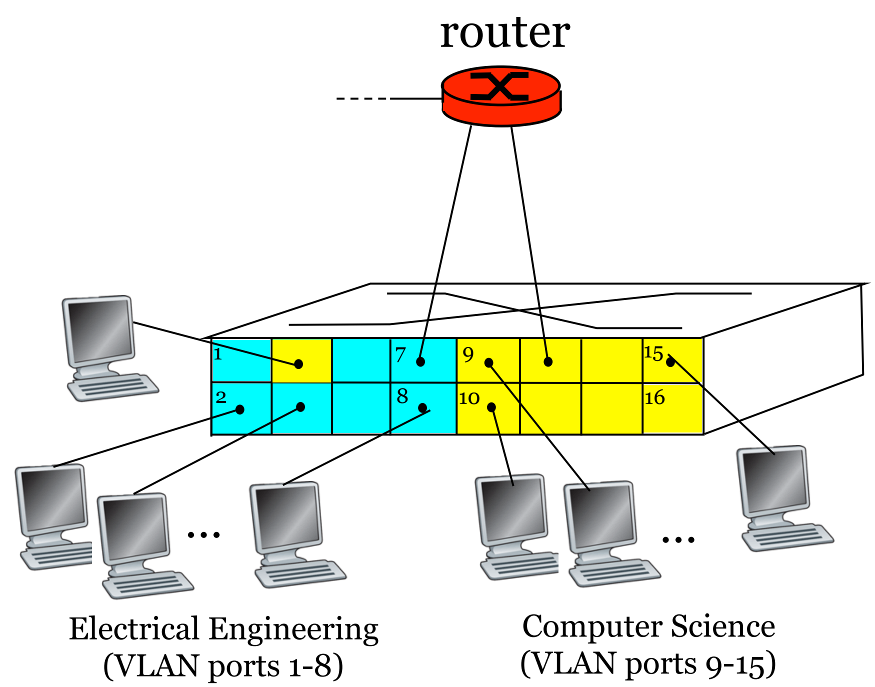


Virtual Local Area Network

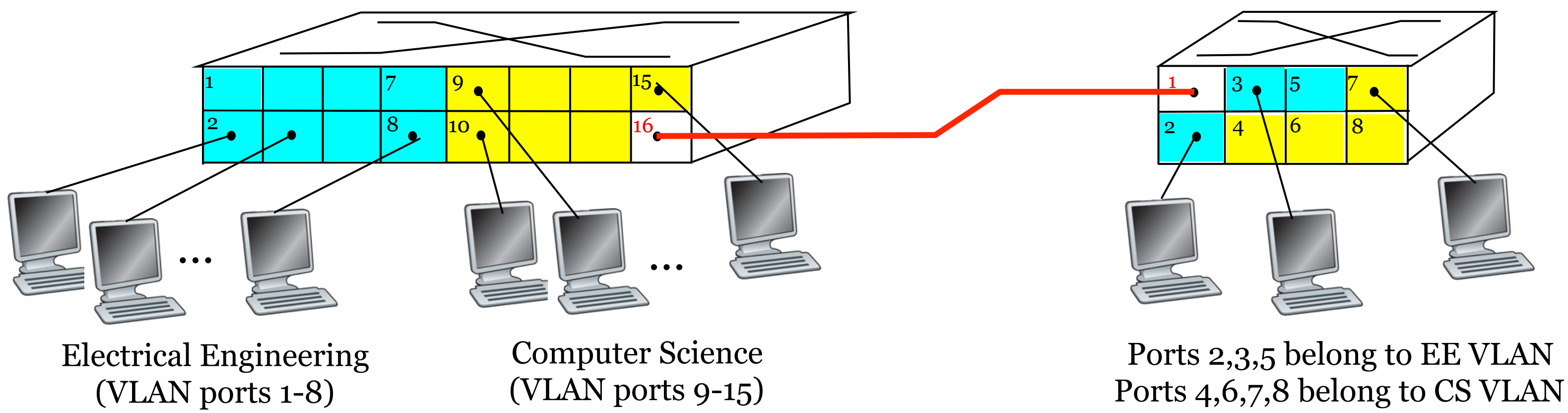
Switch(es) supporting VLAN capabilities can be configured to define multiple **virtual** LANS over single physical LAN infrastructure.

Port-based VLAN

- *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

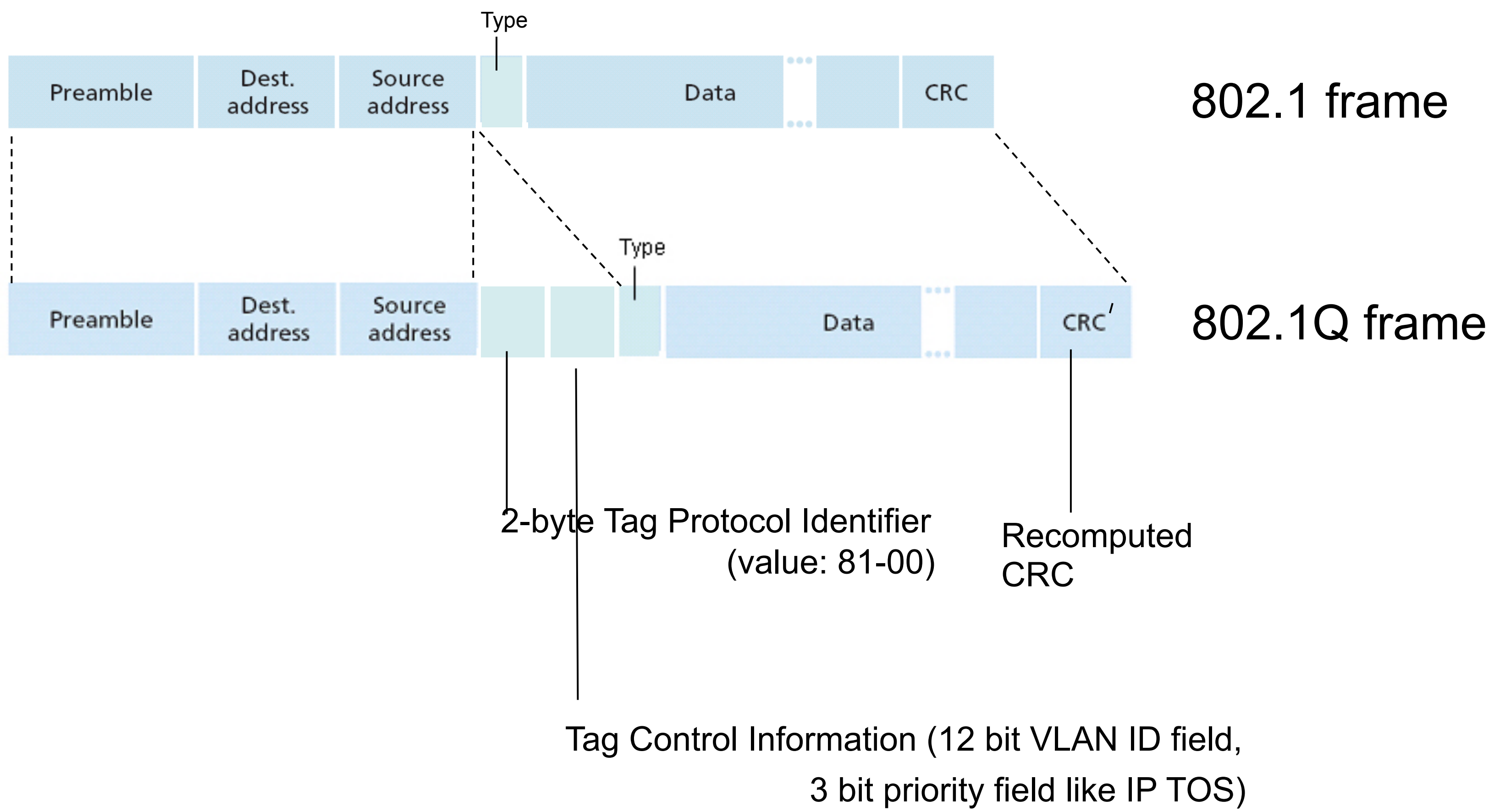


VLANS spanning multiple switches



- *trunk port:* carries frames between VLANS defined over multiple physical switches
  - frames forwarded within VLAN between switches can't be vanilla 802.1 frames (must carry VLAN ID info)
  - 802.1q protocol adds/removed additional header fields for frames forwarded between trunk ports

802.1Q VLAN Frame Format



Summary: Making VLANs Work

- Bridges/switches need configuration tables
  - Saying which VLANs are accessible via which interfaces
- Approaches to mapping to VLANs
  - Each interface has a VLAN "color" (i.e. number)
    - Only works if all hosts on same segment belong to same VLAN
  - Each MAC address has a VLAN color
    - Useful when hosts on same segment belong to different VLANs
    - Useful when hosts move from one physical location to another
- Changing the Ethernet header
  - Adding a field for a VLAN tag
  - Implemented on the bridges/switches
  - … but can still interoperate with old Ethernet cards

Moving From Switches to Routers

- Advantages of switches over routers
  - Plug-and-play
  - Fast filtering and forwarding of frames
  - No pronunciation ambiguity (e.g., “rooter” vs. “rowter”)
- Disadvantages of switches over routers
  - Topology is restricted to a spanning tree
  - Large networks require large ARP tables
  - *Broadcast storms* can cause the network to collapse

Comparing Hubs, Switches, Routers

	Hub/ Repeater	Bridge/ Switch	Router
Traffic isolation	no	yes	yes
Plug and Play	yes	yes	no
Efficient routing	no	no	yes
Cut through	yes	yes	no

Conclusion

- Shuttling data from one link to another
  - Bits, frames, packets, …
  - Repeaters/hubs, bridges/switches, routers, …

- Key ideas in switches
  - Cut-through switching
  - Self learning of the switch table
  - Spanning trees
  - Virtual LANs (VLANs)