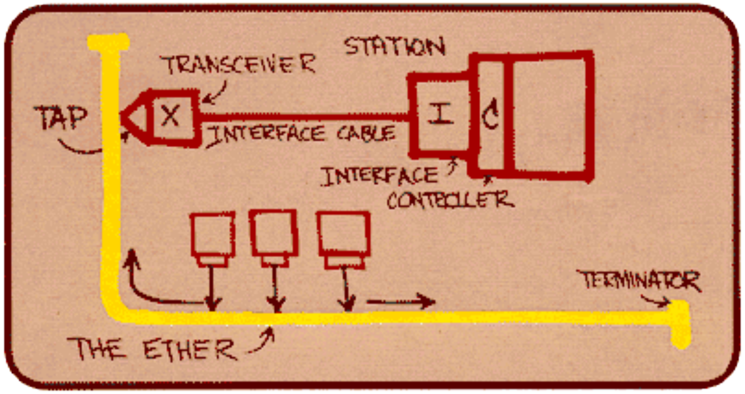


Local Area Networks: Ethernet, Switching

Sangtae Ha

CSCI 4273/5273 Network Systems

http://ngn.cs.colorado.edu/~sangtaeha/courses/csci4273/fall15/

Note: The slides are adapted from the materials from Prof. Richard Han at CU Boulder and Profs. Jennifer Rexford and Mike Freedman at Princeton University.

Announcements

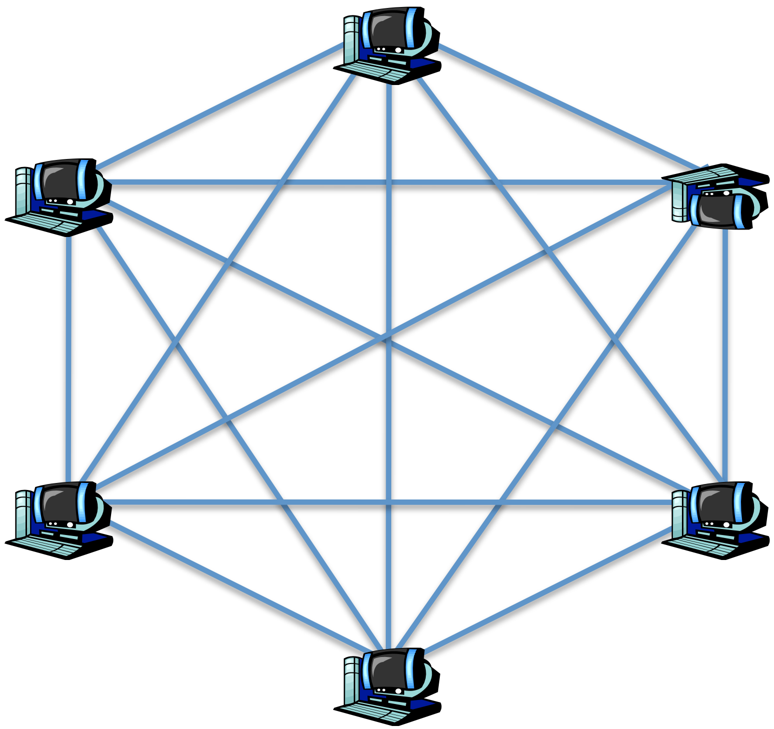
- PA #1 due is extended by one week (11:59pm, Sept 17)
- Lab #1 begins this Friday

Ethernet and Switching

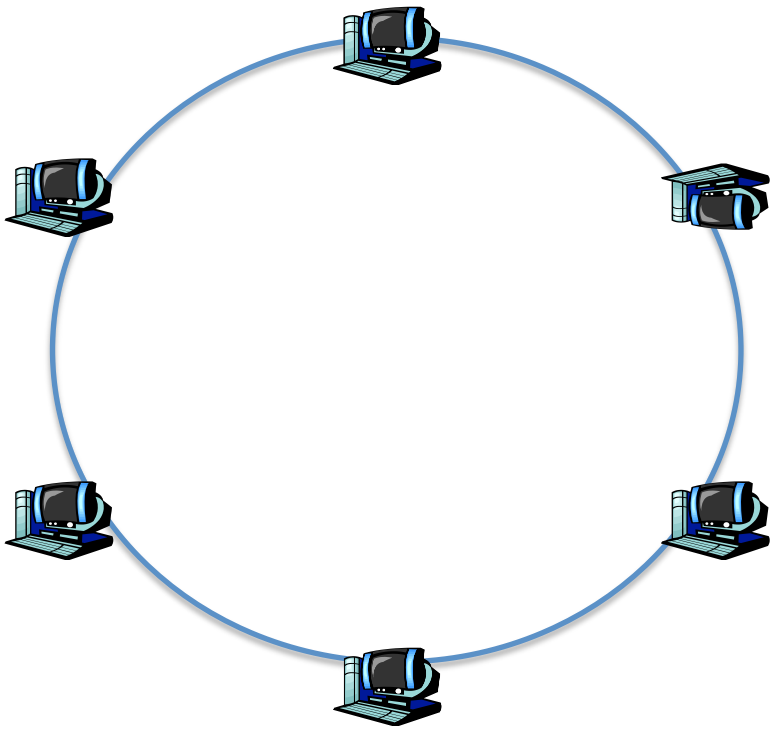
Goals of Today's Lecture

- Devices that shuttle data at different layers
  - Repeaters and hubs
  - Bridges and switches
  - Routers
- Switch protocols and mechanisms
  - Dedicated access and full-duplex transfers
  - Cut-through switching
  - Self learning of the switch table
  - Spanning trees
- Virtual LANs (VLANs)

Fully-connected links



Shared broadcast medium

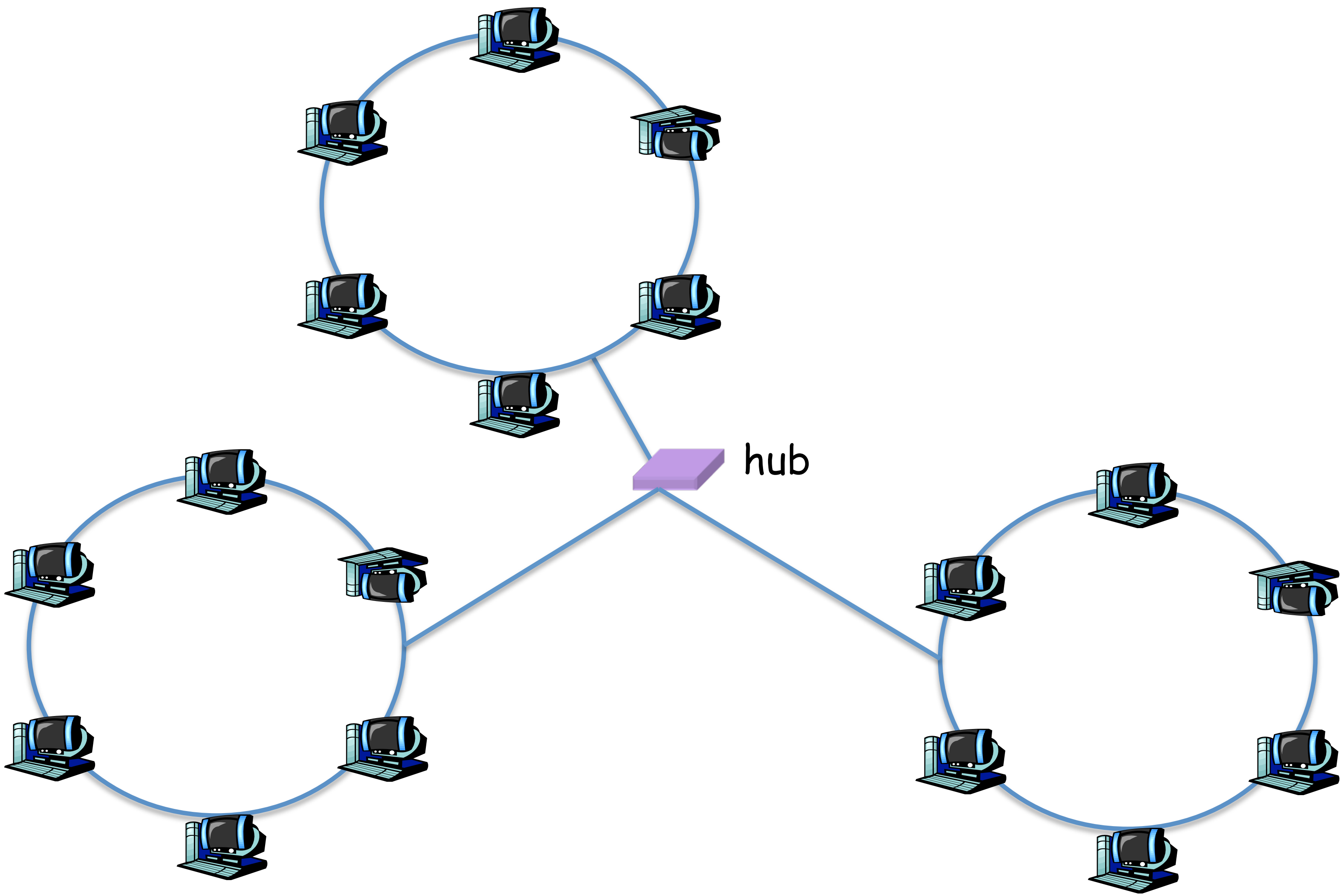


It's all about resource allocation

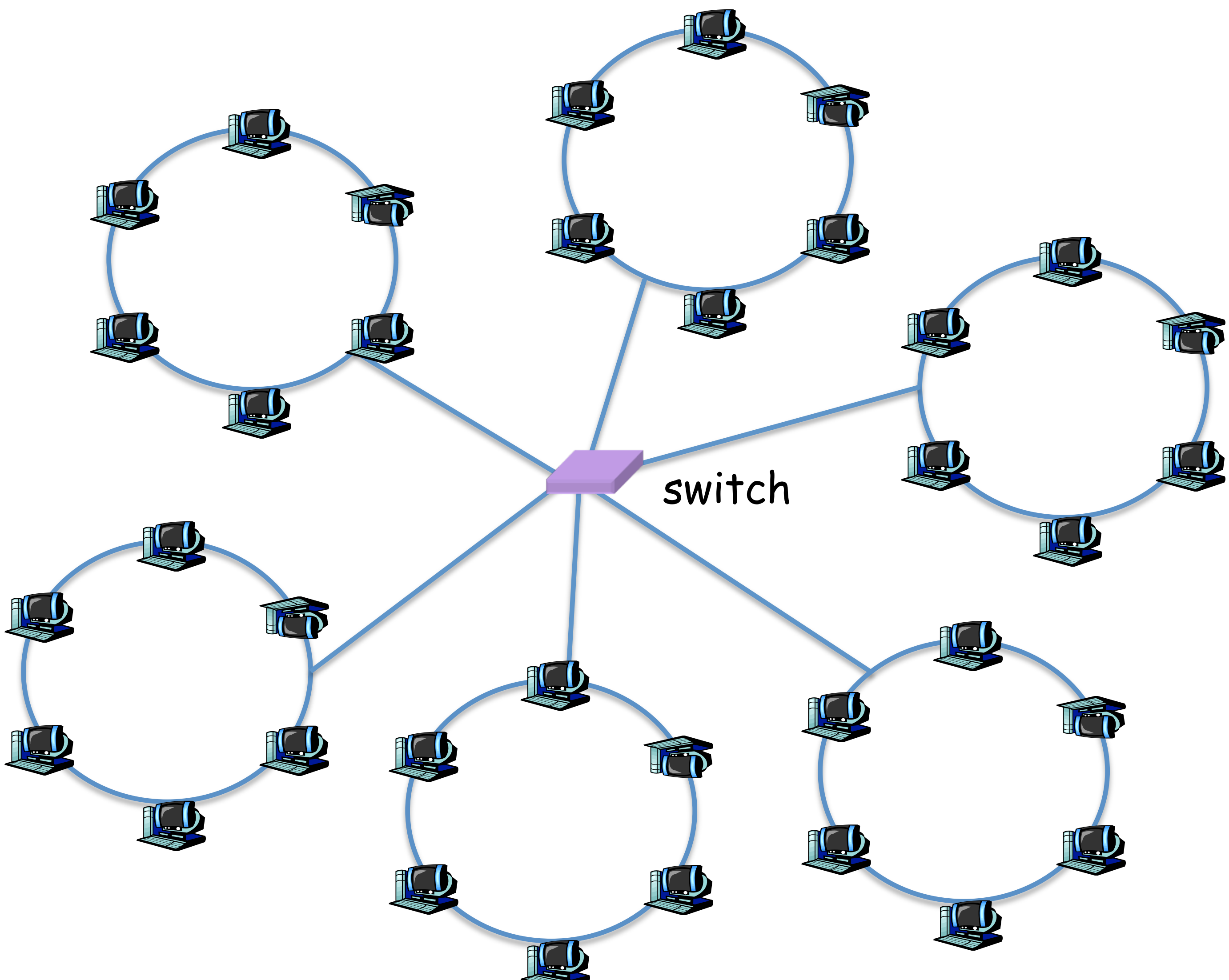
Three Ways to Share the Media

- Channel partitioning MAC protocols:
  - Share channel efficiently and fairly at high load
  - Inefficient at low load: unused go idle
- "Taking turns" protocols
  - Eliminates empty slots without causing collisions
  - Vulnerable to failures
- Random access MAC protocols
  - Efficient at low load: single node can fully utilize channel
  - High load: collision overhead

Hubs:
Joining broadcast mediums

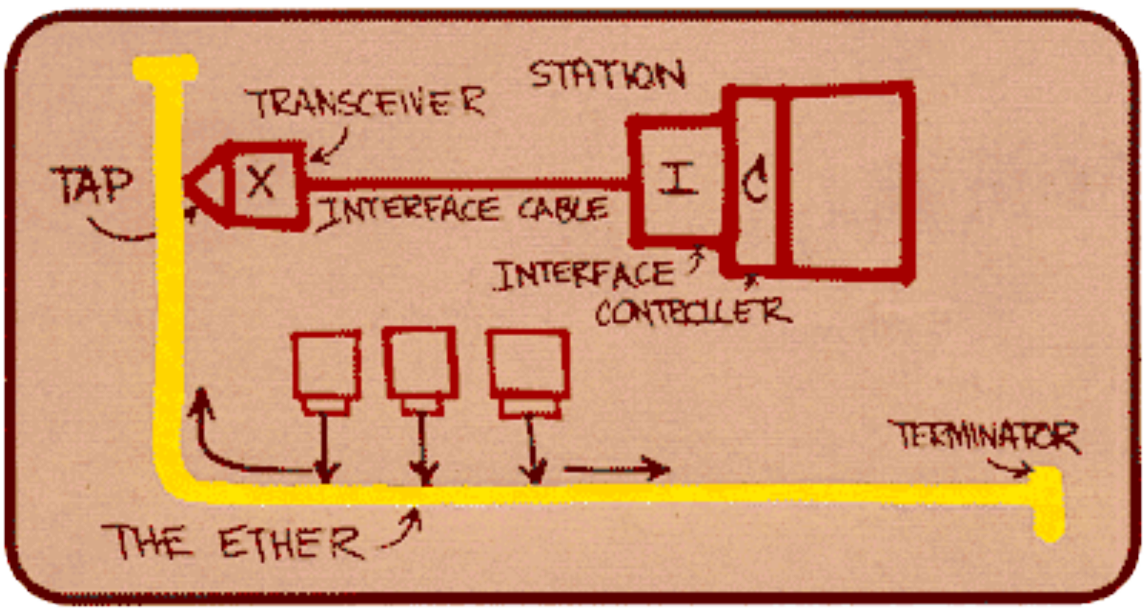


Bridges / Switches: Isolating broadcast mediums



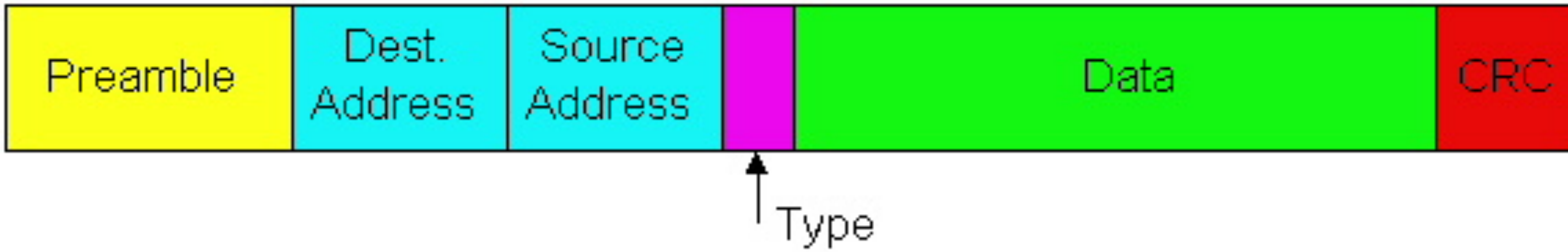
Ethernet

- Dominant wired LAN technology, first widely used
- Simpler, cheaper than token LANs and ATM
- Kept up with speed race: 10 Mbps – 10 Gbps



Metcalfe's Ethernet sketch

Ethernet Frame Structure

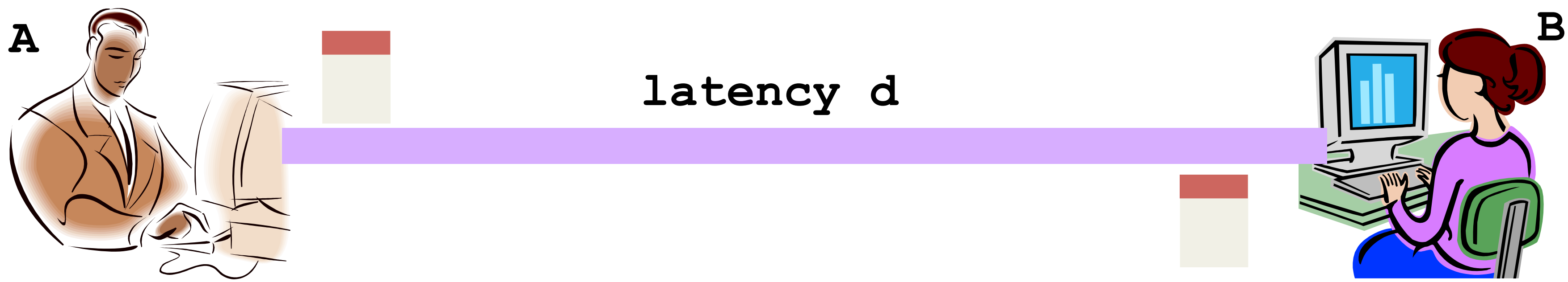


- Preamble: synchronization: $(10101010)^7$ 10101011
- Addresses: 6-byte source and dest MAC addresses
  – Adaptor passes frame to OS stack if destination matches adaptor or is broadcast address; otherwise, discard frame
- Type: higher-layer protocol (IP, AppleTalk, …)
- Error detection: CRC: cyclic redundancy check
- Best effort: Connectionless, unreliable

Ethernet Uses CSMA/CD

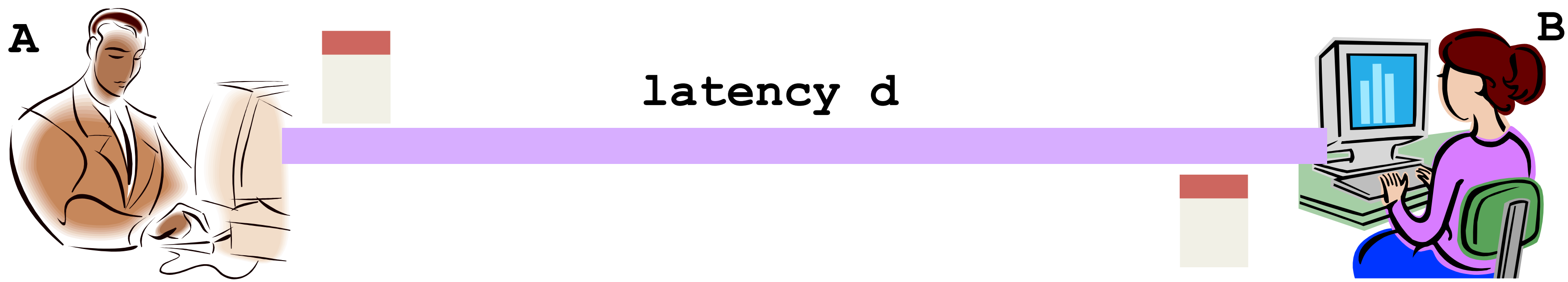
- Carrier Sense: wait for link to be idle before transmit
- Collision Detection: listen while transmitting
  - No collision: transmission complete
  - Collision: abort and send jam signal
- Random access: exponential back-off
  - After collision, wait a random time before retry
  - After m^{th} collision, choose K randomly from $\{0, \ldots, 2^m-1\}$
  - … and wait for K*64 byte times before retry

Limitations on Ethernet Length



- Latency depends on physical length of link
  – Time to propagate a packet from one end to the other
- Suppose A sends a packet at time t
  – And B sees an idle line just before time t+d, so transmits
- B detects a collision, and sends jamming signal
  – But A doesn't see collision till t+2d

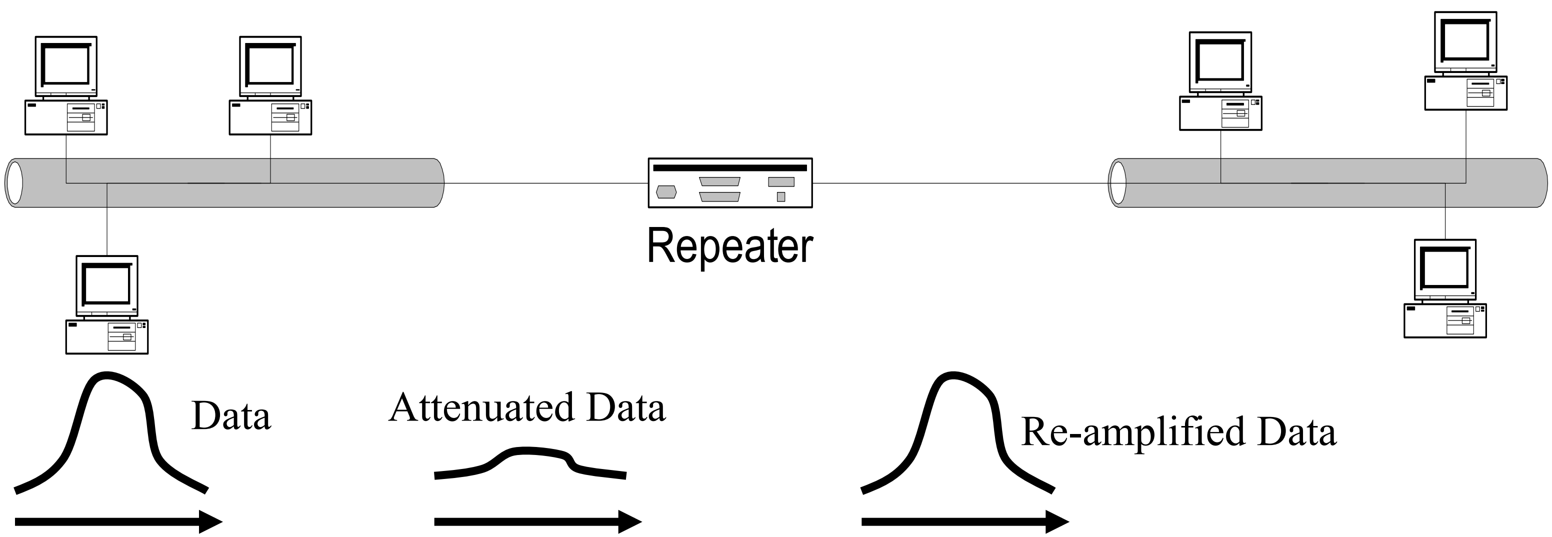
Limitations on Ethernet Length



- A needs to wait for time 2d to detect collision
  - So, A should keep transmitting during this period
  - … and keep an eye out for a possible collision
- Imposes restrictions on Ethernet
  - Max length of wire: 2500 meters
  - Min length of packet: 512 bits (64 bytes)

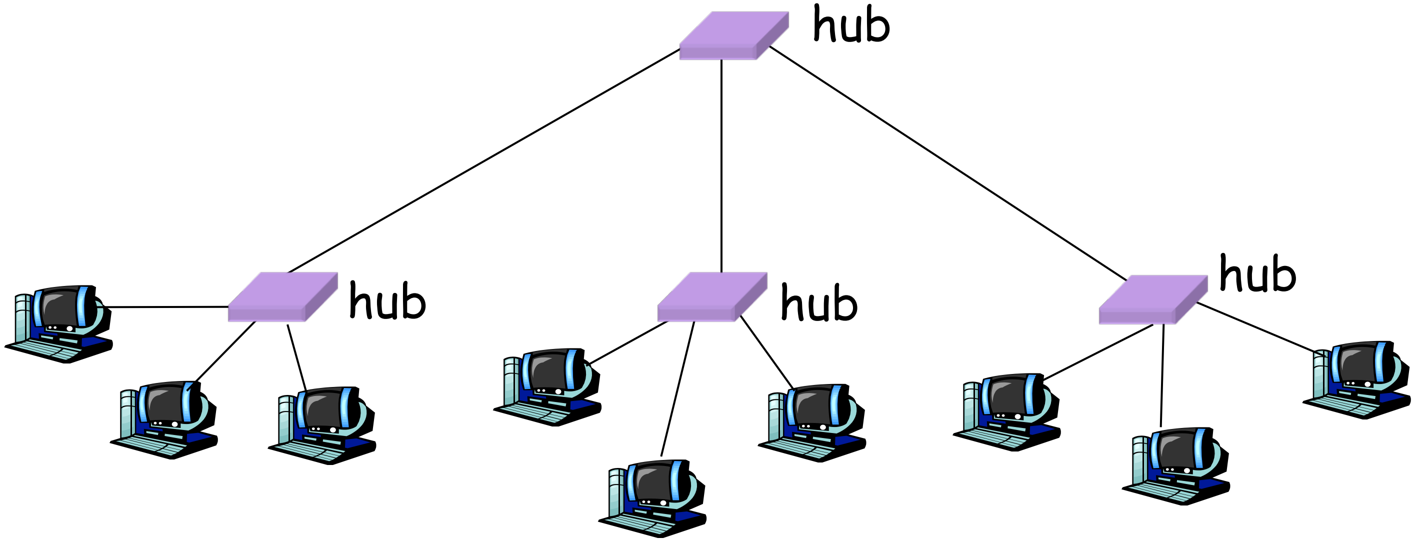
Physical Layer: Repeaters

- 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
  - Monitors signals on each LAN and transmits amplified copies



Physical Layer: Hubs

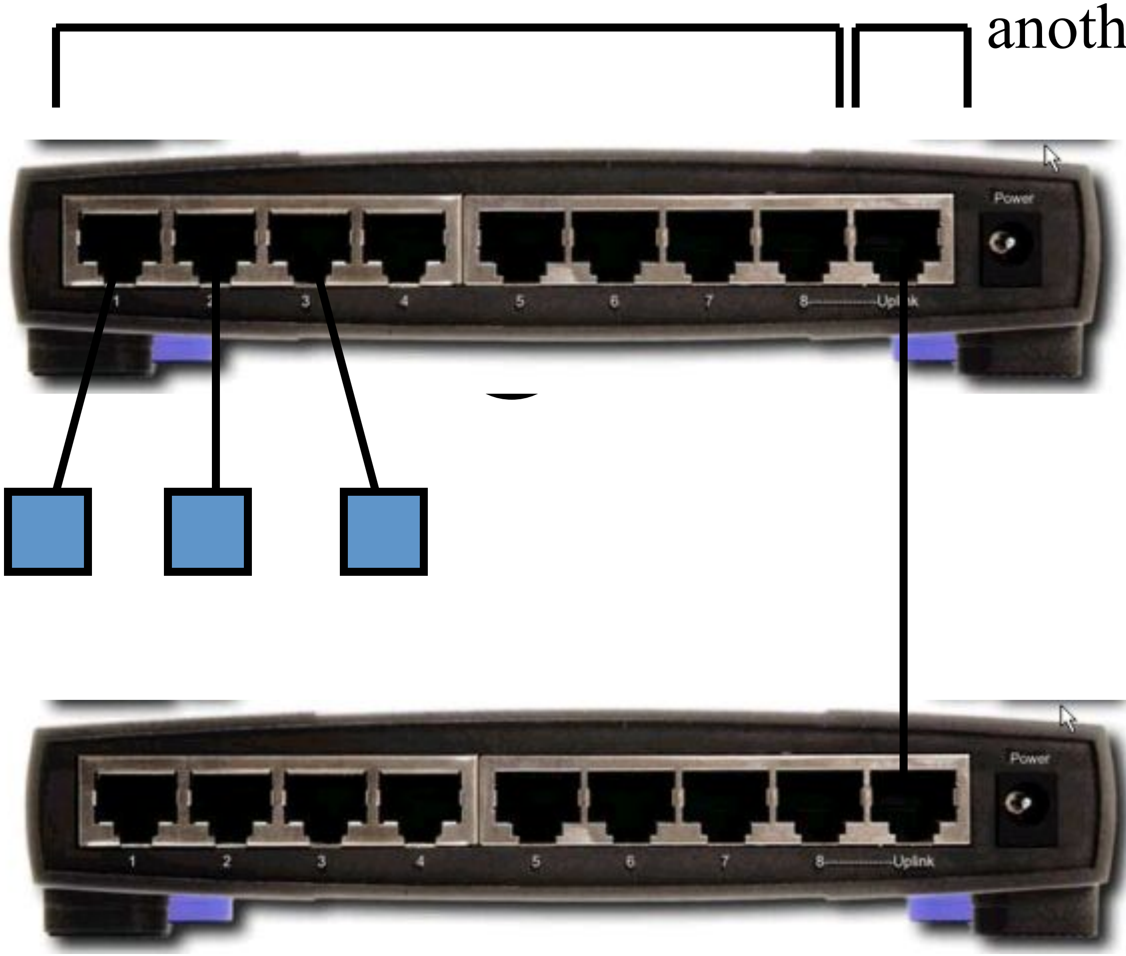
- Joins multiple input lines electrically
  - Designed to hold multiple line cards
  - Do not necessarily amplify the signal
  - Monitoring and fault isolation are supported
- Very similar to repeaters
  - Also operates at the physical layer



An Actual Ethernet Hub

8 ports: connect hosts here

Uplink: connect to another hub



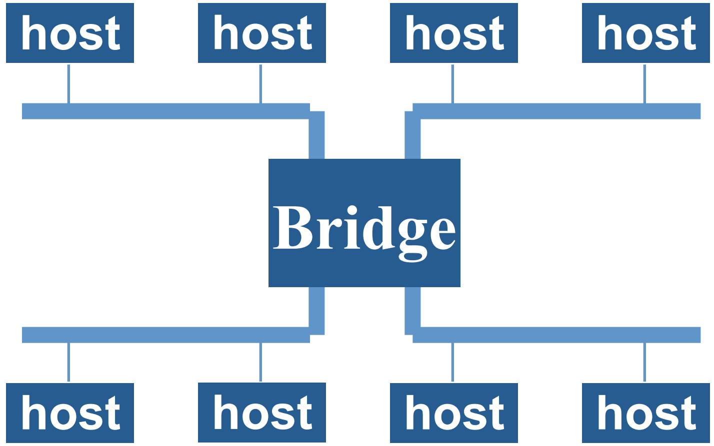
Limitations of Repeaters and Hubs

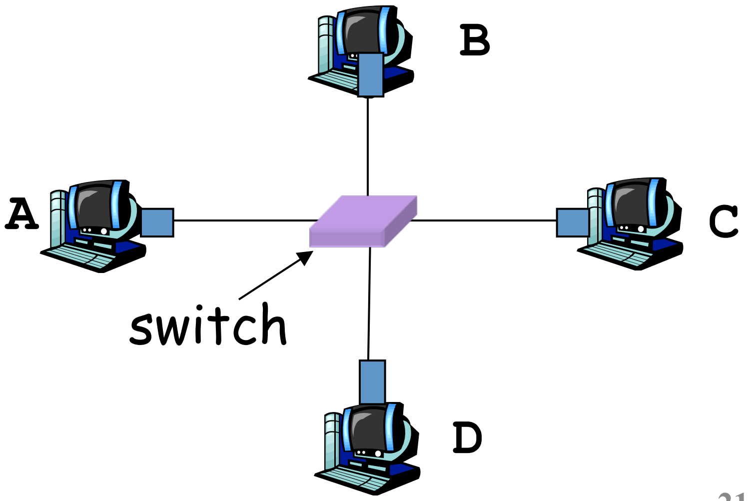
- One large shared link
  - Each bit sent everywhere, aggregate throughput limited
- Cannot support multiple LAN technologies
  - Does not buffer or interpret frames
  - So, can't interconnect different rates or formats
- Limitations on maximum nodes and distances

Switching for resource isolation

Link Layer: Bridges and Switches

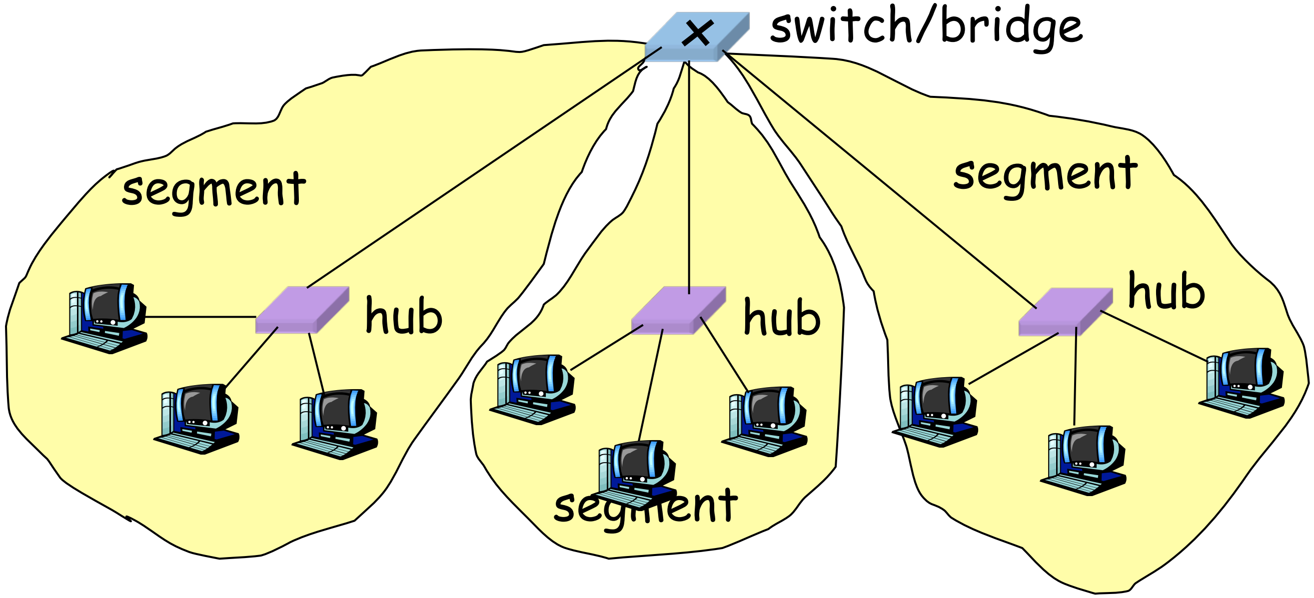
- Connects two or more LANs at the link layer
  - Extracts destination address from the frame
  - Looks up the destination in a table, forwards to appropriate
- Each segment can carry its own traffic
  - Concurrent traffic between LANs/host: A to B while D to C
- Bridge: connecting LANs; Switches: connecting hosts





Bridges/Switches: Traffic Isolation

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



High-density switching

48-port switch

rack

Facebook rack

- Each rack has 42 U (“pizza boxes”)
- Typically servers + 1-2 “top-of-rack” switch(es)

Advantages Over Hubs/Repeaters

- Only forwards frames as needed
  - E.g. to destination segments or for broadcast traffic
  - Reduces unnecessary traffic on segments
- Extends the geographic span of the network
  - Ethernet collisions (and distance limitations) only on segment
- Improves privacy by limiting scope of frames
  - Hosts can only “snoop” the traffic traversing *their* segment
- Can join segments using different technologies

Disadvantages Over Hubs/Repeaters

- Delay in forwarding frames
  - Bridge/switch must receive frame, parse, lookup, and send
  - Storing and forwarding the packet introduces delay
  - Sol'n: cut-through switching (start send after receive header)
- Need to learn where to forward frames
  - Forwarding table: destination MAC → outgoing interface
  - Needs to construct forwarding table, ideally w/o static config
  - Sol'n: self-learning
- Higher cost
  - More complicated devices that cost more money

Switches vs. Hubs

- Compared to hubs, Ethernet switches support
  (a) Larger geographic span
  (b) Similar span
  (c) Smaller span

- Compared to hubs, switches provides
  (a) Higher load on links
  (b) Less privacy
  (c) Heterogenous communication technologies

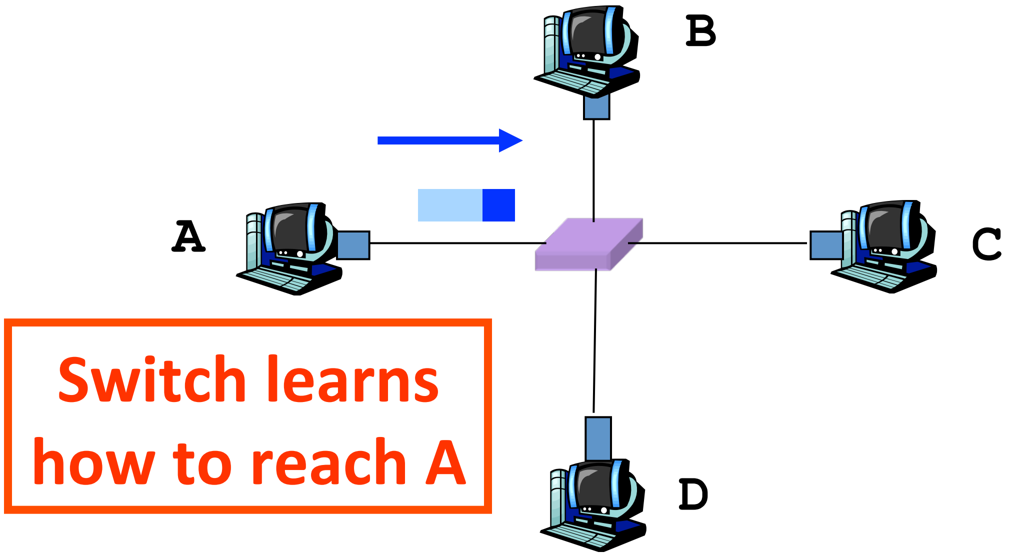
Switches vs. Hubs

- Compared to hubs, Ethernet switches support
  - (a) Larger geographic span
  - (b) Similar span
  - (c) Smaller span

- Compared to hubs, switches provides
  - (a) Higher load on links
  - (b) Less privacy
  - (c) Heterogenous communication technologies

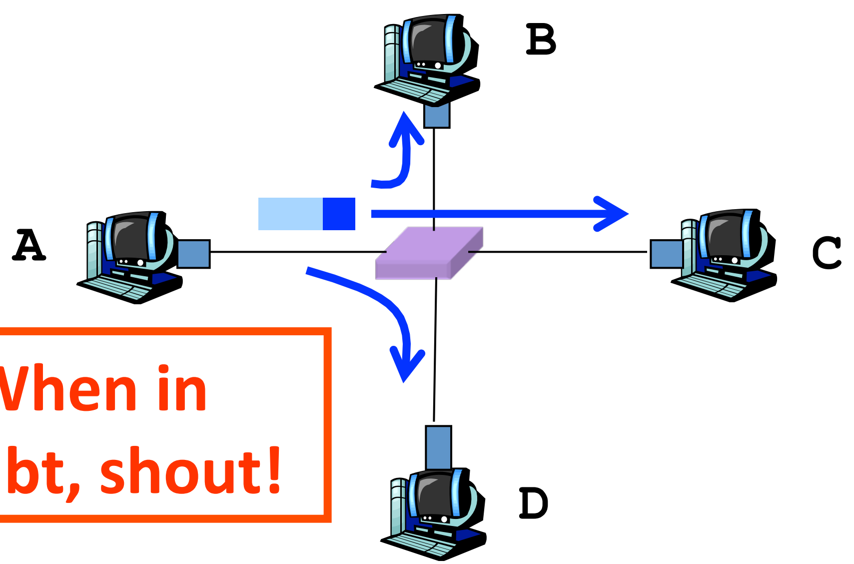
Self Learning: Building the Table

- When a frame arrives
  - Inspect *source* MAC address
  - Associate addr with *incoming* interface/port
  - Store mapping in forwarding table
  - Use TTL field to eventually forget mapping



Self Learning: Handling Misses

- When frame arrives with unfamiliar destination
  - Forward frame out all interfaces except source
  - Hopefully, won't happen very often



When in doubt, shout!

Switch Filtering/Forwarding

When switch receives a frame:

index switch table using MAC dest address

if (entry found for destination) then

    if (dest on segment from which frame arrived) then

        drop the frame

    else

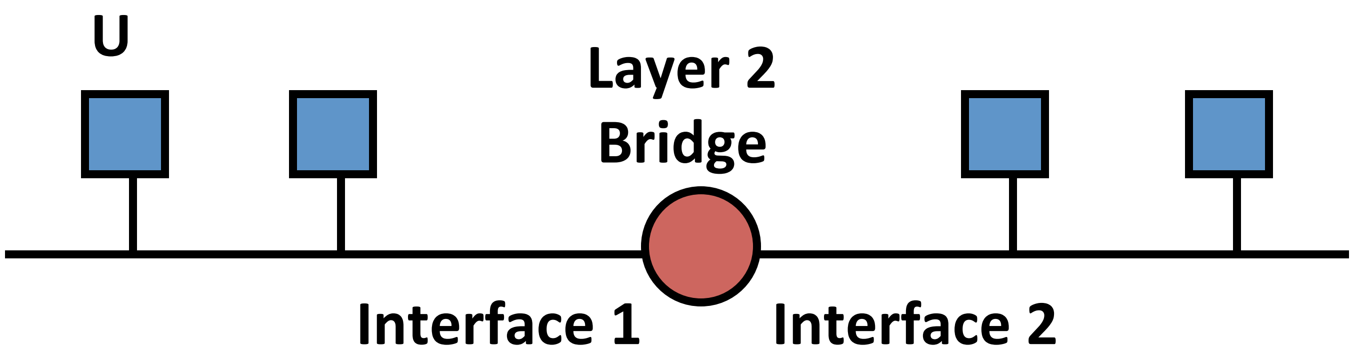
        forward the frame on interface indicated

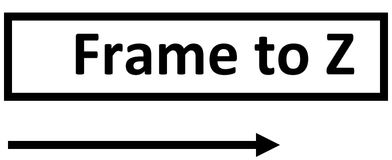
else flood

forward on all but the interface on which the frame arrived

Table for Forwarding and Filtering

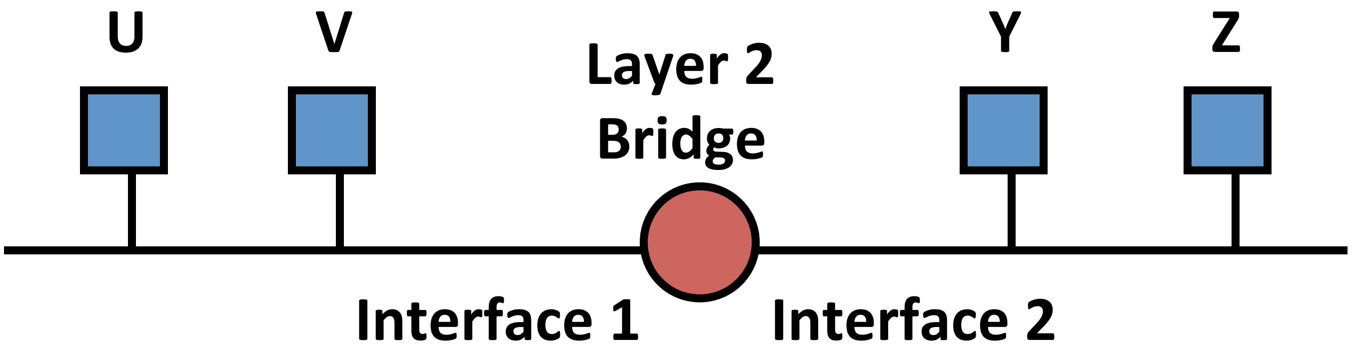
- Label the interfaces into/out of the bridge
- When a frame arrives, store the source address and the originating interface from which the frame came





Src Addr	Originating Interface	Time
U	1	--

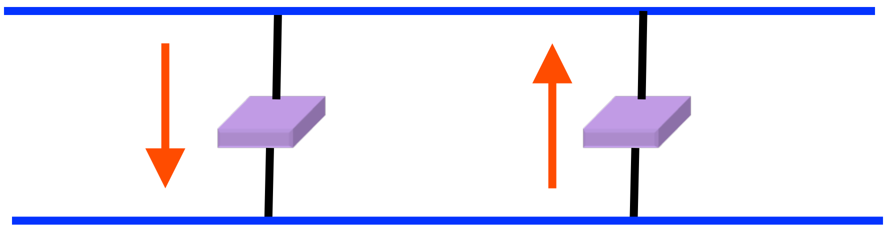
Frame Forwarding Rules



Event	Originating Interface 1	Originating Interface 2	Bridge Forwarding Action
Boot state	--	--	
U sends to V	U	--	Forward to all outgoing (e.g. 2)
V sends to U	U,V	--	none
Z sends to U	U,V	Z	Forward to 1
Z sends to Y	U,V	Z	Forward to all outgoing (e.g. 1)
Y sends to V	U,V	Y,Z	Forward to 1

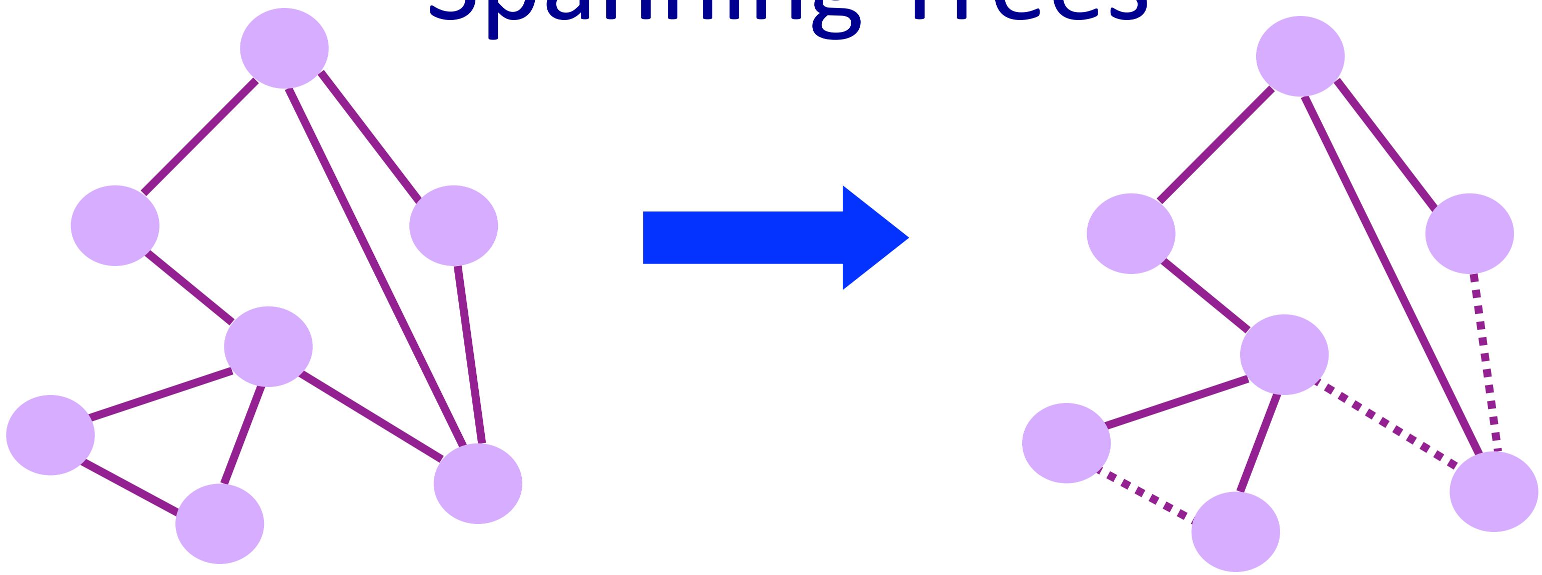
Flooding Can Lead to Loops

- E.g., if the network contains a cycle of switches
- Either accidentally or by design for higher reliability



- Solution: Spanning Tree
  - Ensure the topology has no loops
  - Avoid using some of the links when flooding
  - Spanning tree: Sub-graph that covers all vertices but contains no cycles

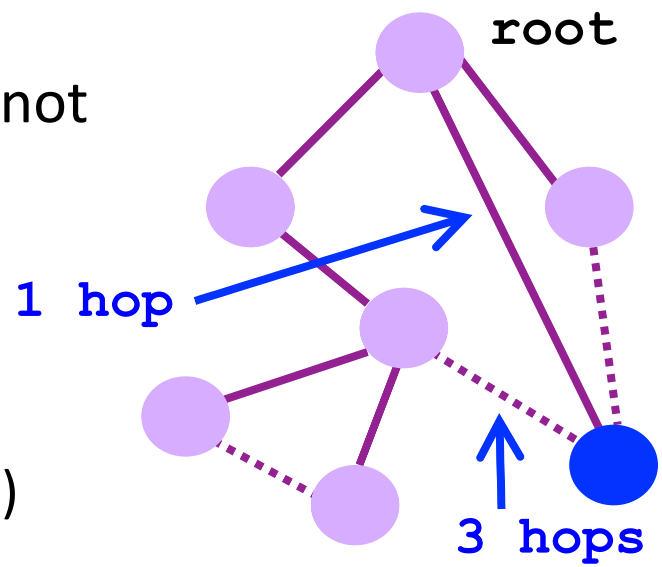
Spanning Trees



- Solution: Spanning Tree
  - Ensure the topology has no loops
  - Avoid using some of the links when flooding
  - Spanning tree: Sub-graph that covers all vertices but contains no cycles

Constructing a Spanning Tree

- Distributed algorithm
  - Switches cooperate to build, auto-adapt on failures
- Key ingredients of the algorithm
  - Switches elect a “root” (e.g. one w/ smallest ID)
  - Each determines if interface is on shortest path from root, excludes if not
  - Learned via messages from peers
    - (root Y, distance d, from X)
  - Reacts to root/switch/link failures
    - Path entries have TTL (i.e. soft state)
    - Root periodically reannounces

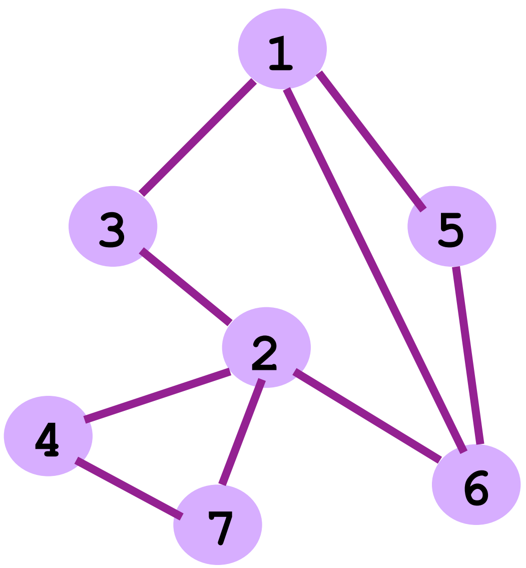


Steps in Spanning Tree Algorithm

- 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

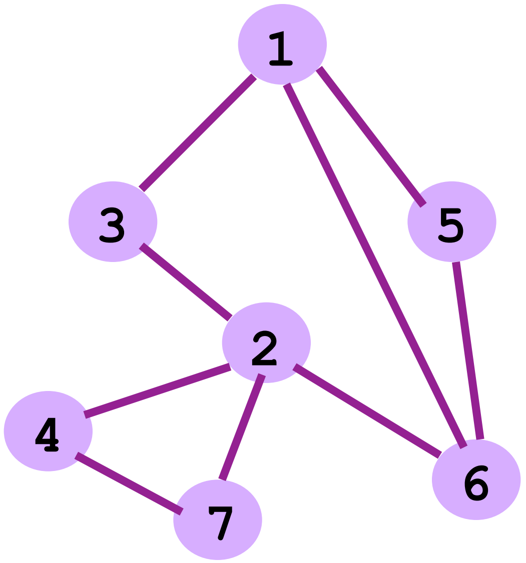
Example From Switch #4's Viewpoint

- 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

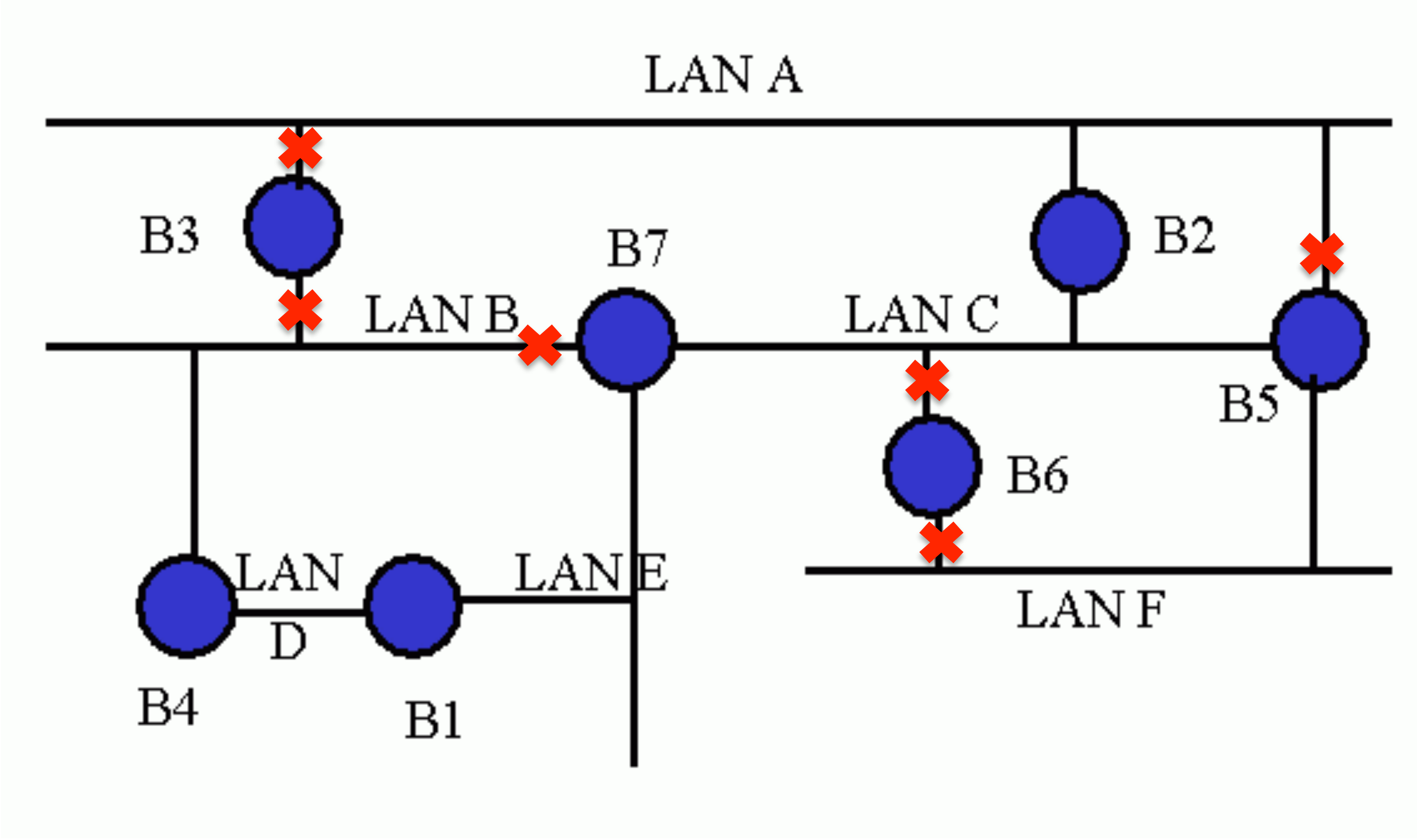


Example From Switch #4's Viewpoint

- 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 link from the tree



Spanning Tree Protocol Problem



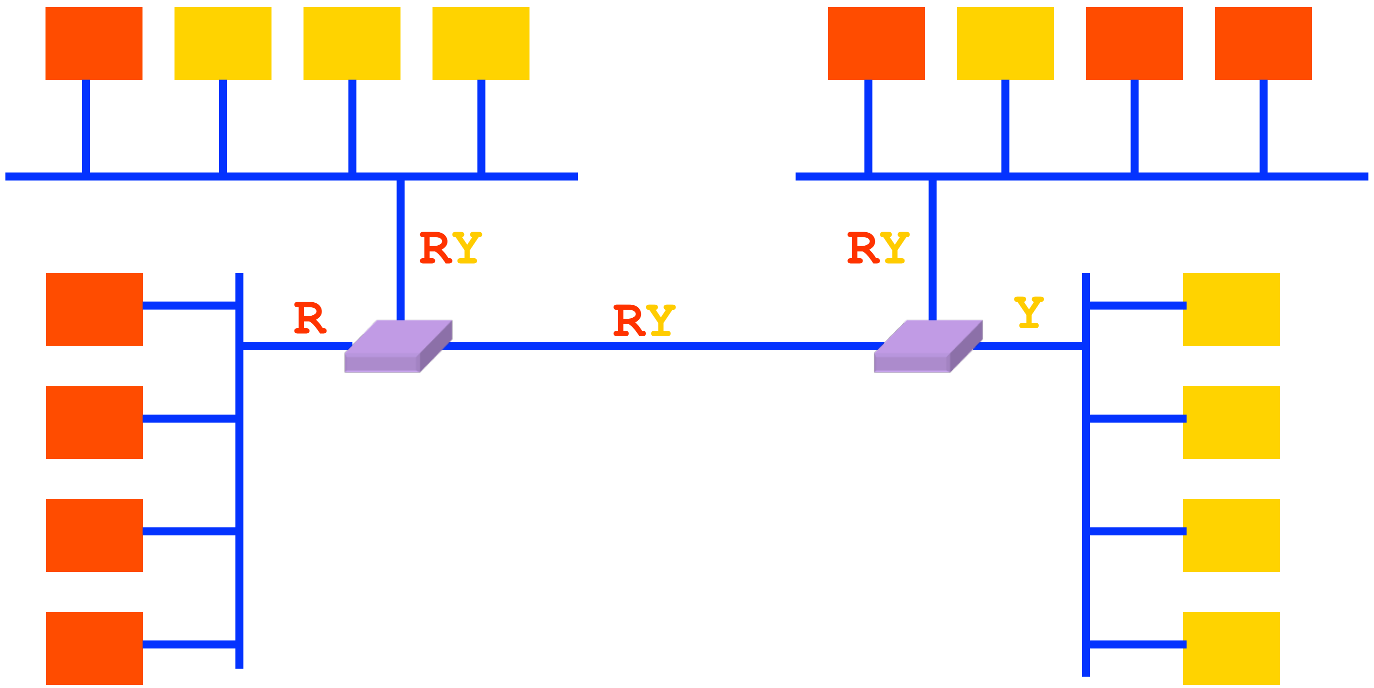
Evolution Toward Virtual LANs

- In the olden days…
  - Thick cables snaked through cable ducts in buildings
  - Every computer was plugged in
  - All people in adjacent offices were on same LAN
- More recently due to hubs and switches…
  - Every office connected to central wiring closets
  - Flexibility in mapping offices to different LANs
- Evolution to grouping users based on org structure, not physical layout of building

Why Group by Org Structure?

- Security
  - Ethernet is a shared media
  - Interfaces can be put in "promiscuous" mode to see all traffic
- Load
  - Some LAN segments are more heavily used than others
    - E.g., researchers can saturate own segment, but not others
  - May be natural locality of communication
    - E.g., traffic between people in the same research group
- But people move, organizations changes
  - Physical rewiring is a huge pain!

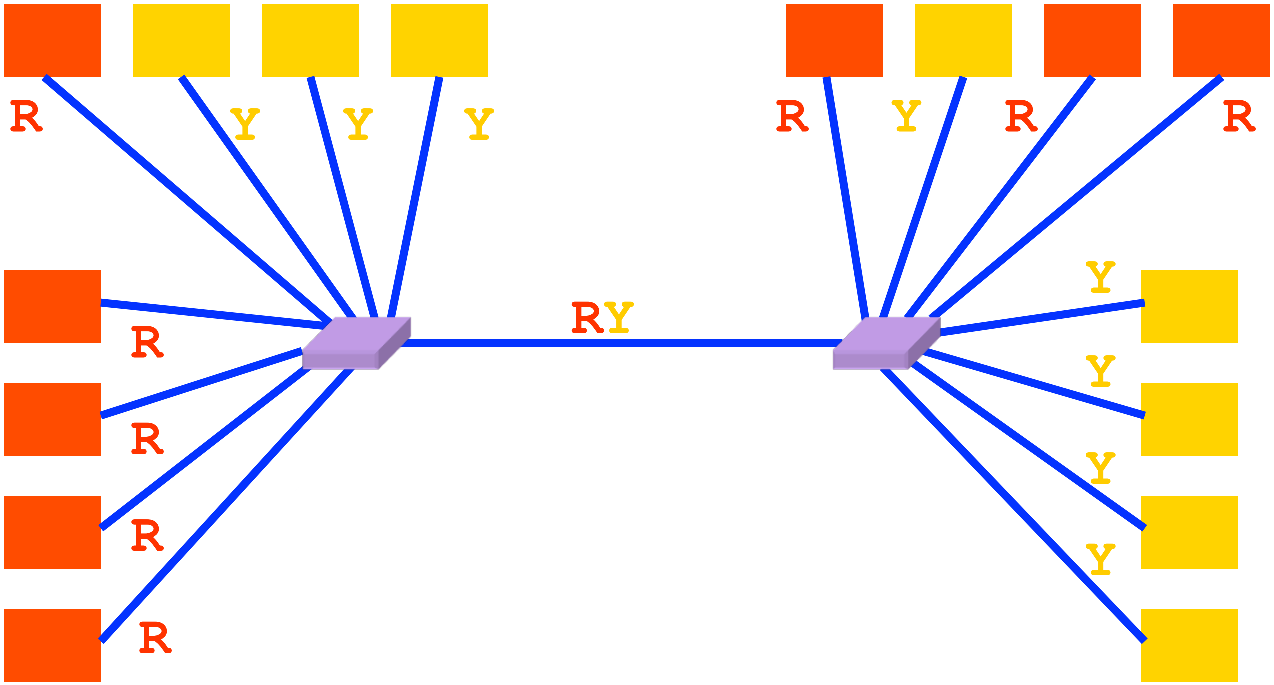
Virtual LANs



Red VLAN and Yellow VLAN

Switches forward traffic as needed

Virtual LANs



Red VLAN and Yellow VLAN

Switches forward traffic as needed

Making VLANs Work

- Switches need configuration tables
  - Saying which VLANs are accessible via which interfaces
- Approaches to mapping to VLANs
  - VLAN color per interface
    - Only if all hosts on segment belong to same VLAN
  - VLAN color per MAC address
- Changing the Ethernet header
  - Adding a field for a VLAN tag
  - VLAN tag added/removed by switches
    - Hosts unaware (backwards compat), cannot spoof (security)

Comparing Hubs, Switches, Routers

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