Georgia Tech College of Computing

CS 3251- Computer Networks 1:

Link Layer(2)

Professor Patrick Traynor

10/10/13

Lecture 16

Announcements

- Midterm is just over *one week* from today
  - This is the last lecture of NEW material before the midterm.
  - Next class will be a midterm review: format, sample questions, etc.
- Look for email for demo setup
- Project 3 Posted
- Drop date: October 11th (tomorrow)
  - If you are doing VERY poorly, this is your last chance to drop the class.

Last Time

- What is EDC? How does it work?
- Why do we use different EDC techniques at the link layer than are used at higher layers?
- Why does slotted ALOHA have higher efficiency than ALOHA?
- What is CSMA? How is it different than CSMA/CD?

Link Layer

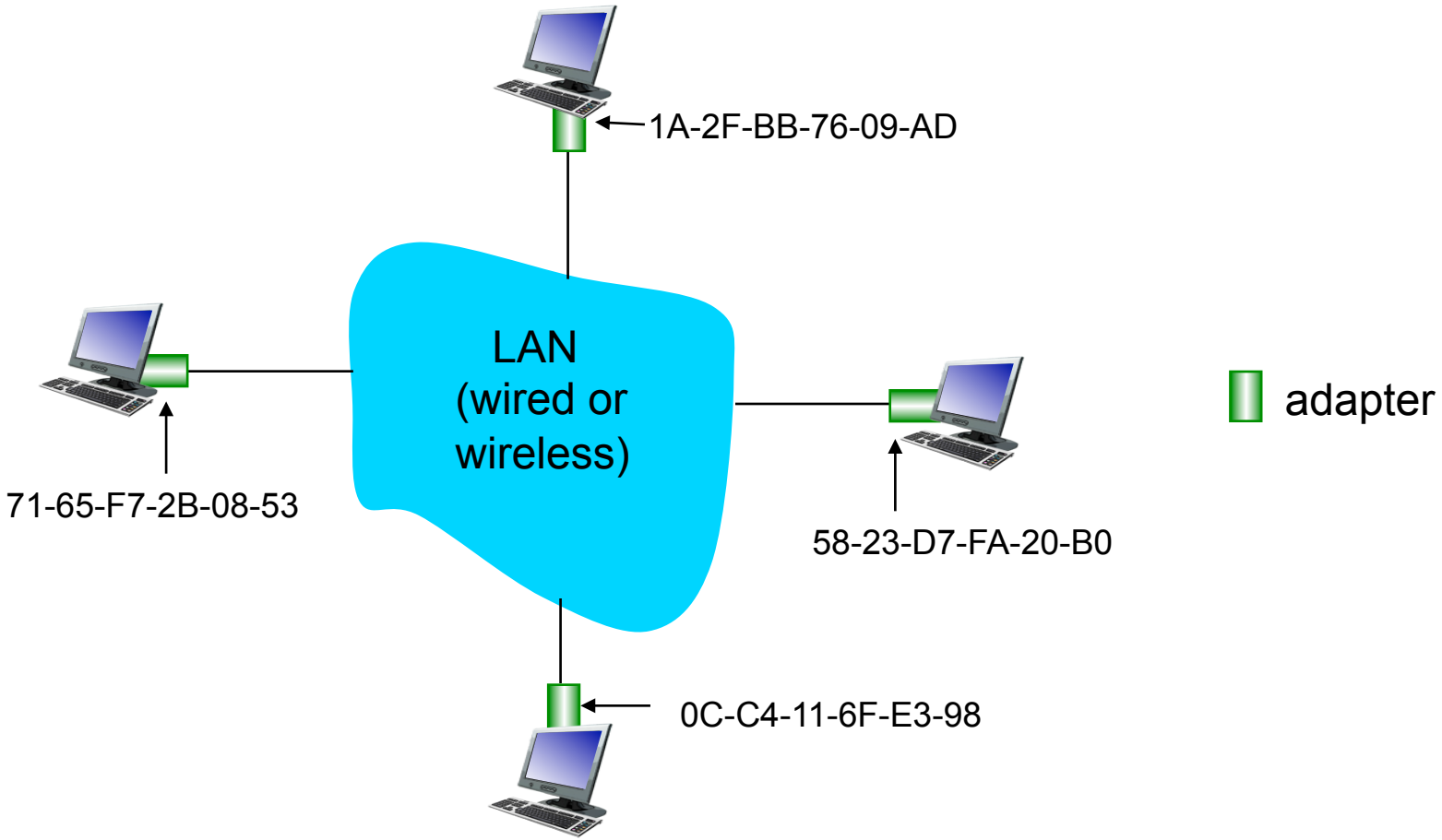
- 5.1 Introduction and services
- 5.2 Error detection and correction
- 5.3 Multiple access protocols
- 5.4 LANs
  - addressing, ARP
  - Ethernet
  - switches
  - VLANS
- 5.5 link virtualization: MPLS
- 5.6 data center networking
- 5.7 a day in the life of a web request

MAC Addresses and ARP

- 32-bit IP address:
  - *network-layer* address
  - used to get datagram to destination IP subnet
- MAC (or LAN or physical or Ethernet) address:
  - used to get frame from one interface to another physically-connected interface (same network)
  - 48 bit MAC address (for most LANs) burned in the adapter ROM
    - e.g.: 1A-2F-BB-76-09-AD; 00:1F:5B:38:FC:04

LAN Addresses and ARP

Each adapter on LAN has unique LAN address



LAN Address (more)

- MAC address allocation administered by IEEE
- manufacturer buys portion of MAC address space (to assure uniqueness)
- Analogy:

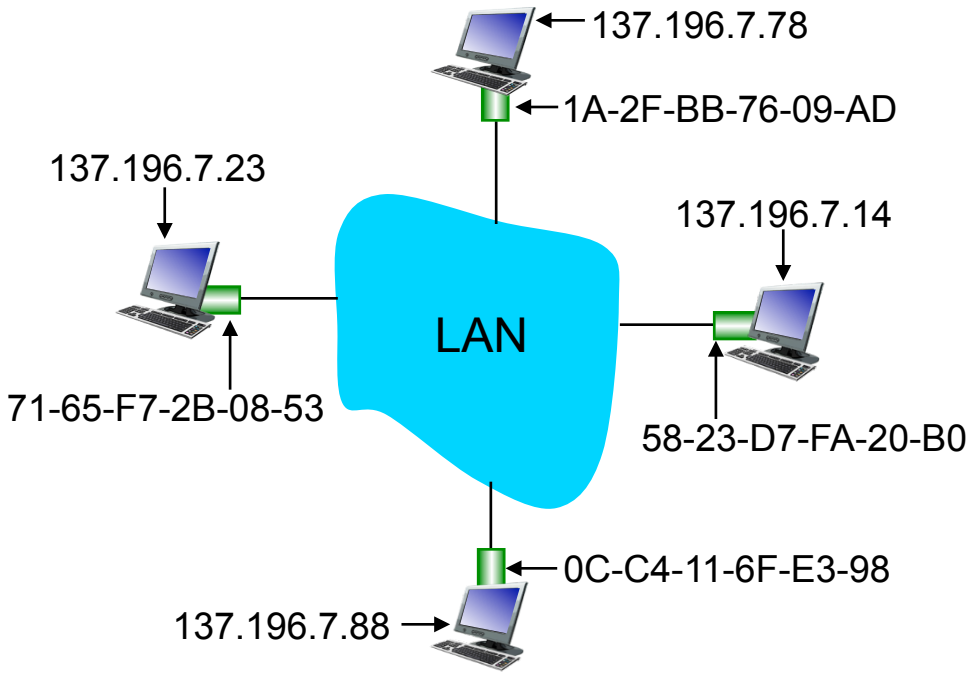
  (a) MAC address: like Social Security Number

  (b) IP address: like postal address

- MAC flat address ➜ portability
  - can move LAN card from one LAN to another
- IP hierarchical address NOT portable
  - depends on IP subnet to which node is attached

ARP: Address Resolution Protocol

Question: how to determine interface's MAC address, knowing its IP address?



- Each IP node (Host, Router) on LAN has ARP table
- ARP Table: IP/MAC address mappings for some LAN nodes

< IP address; MAC address; TTL>

  - TTL (Time To Live): time after which address mapping will be forgotten (typically 20 min)

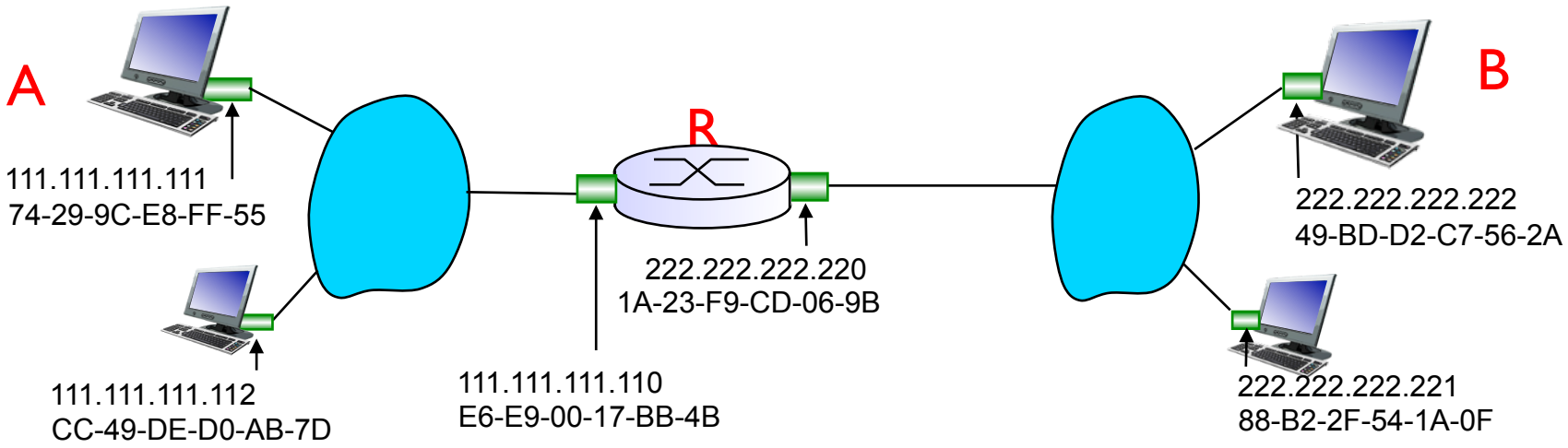
ARP protocol: Same LAN (network)

- A wants to send datagram to B, and B's MAC address not in A's ARP table.
- A broadcasts ARP query packet, containing B's IP address
  - Dest MAC address = FF-FF-FF-FF-FF-FF
  - all machines on LAN receive ARP query
- B receives ARP packet, replies to A with its (B's) MAC address
  - frame sent to A's MAC address (unicast)

- A caches (saves) IP-to-MAC address pair in its ARP table until information becomes old (times out)
  - soft state: information that times out (goes away) unless refreshed
- ARP is "plug-and-play":
  - nodes create their ARP tables without intervention from net administrator

Routing to another LAN

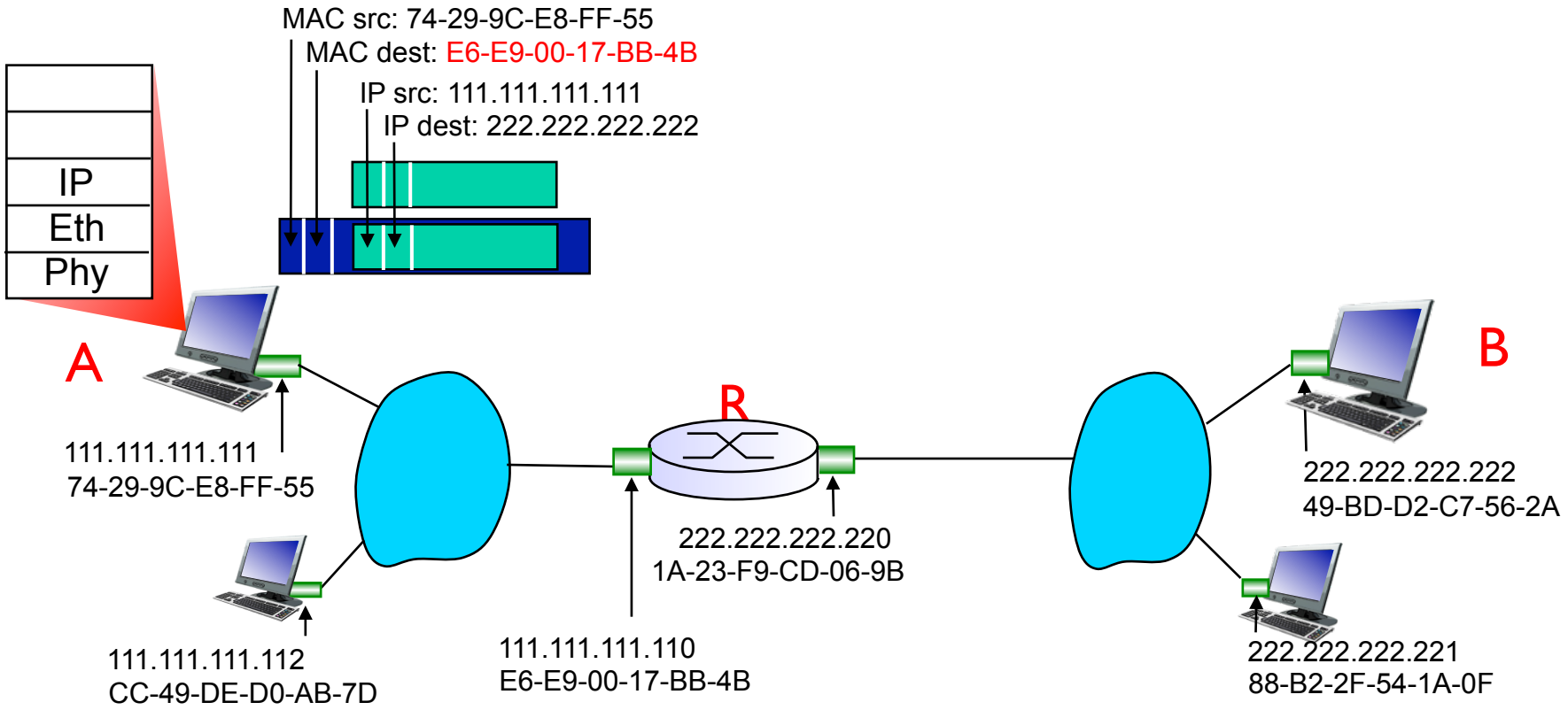
walkthrough: send datagram from A to B via R



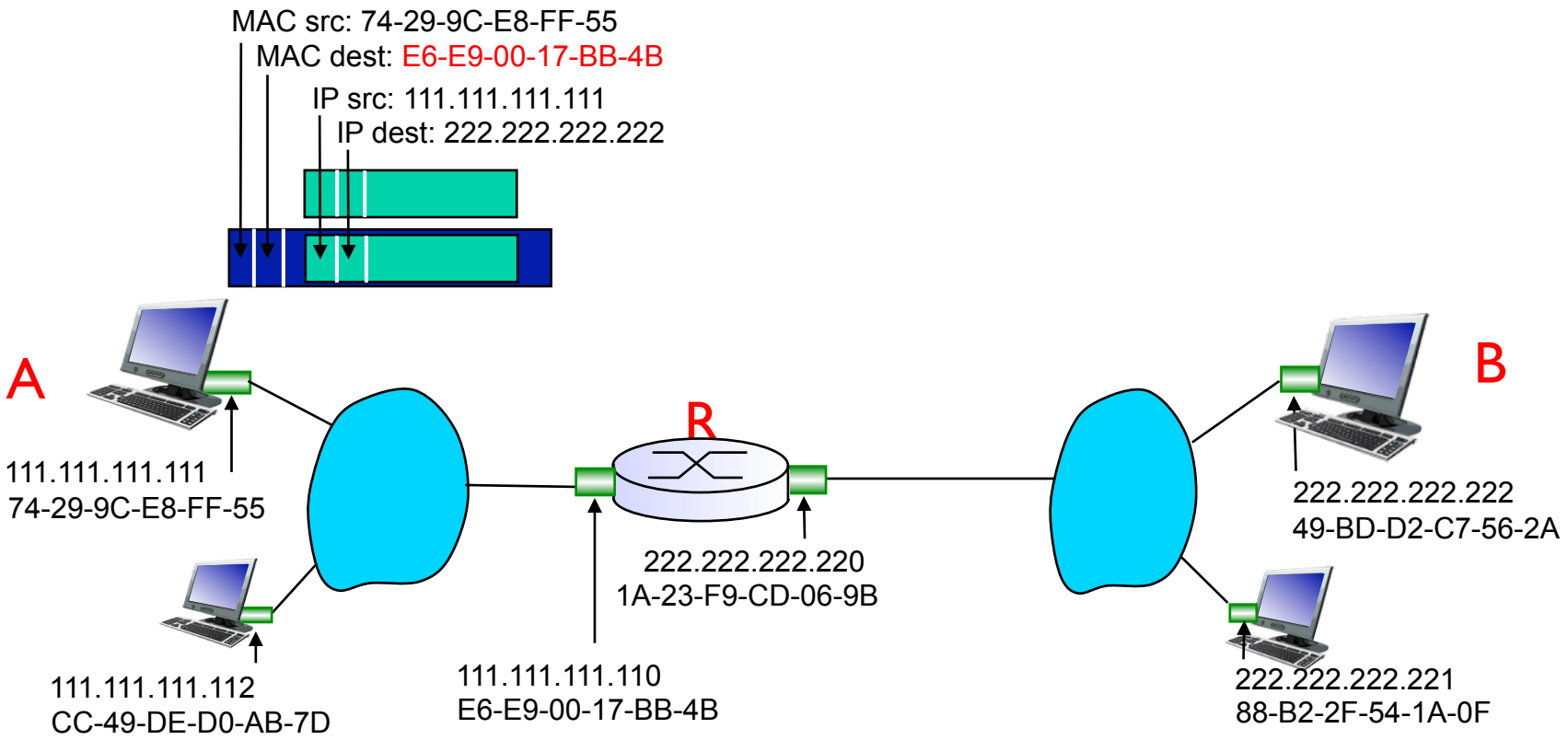
- focus on addressing – at IP (datagram) and MAC layer (frame)
- assume A knows B's IP address
- assume A knows IP address of first hop router, R (how?)
- assume A knows R's MAC address (how?)

Routing to another LAN

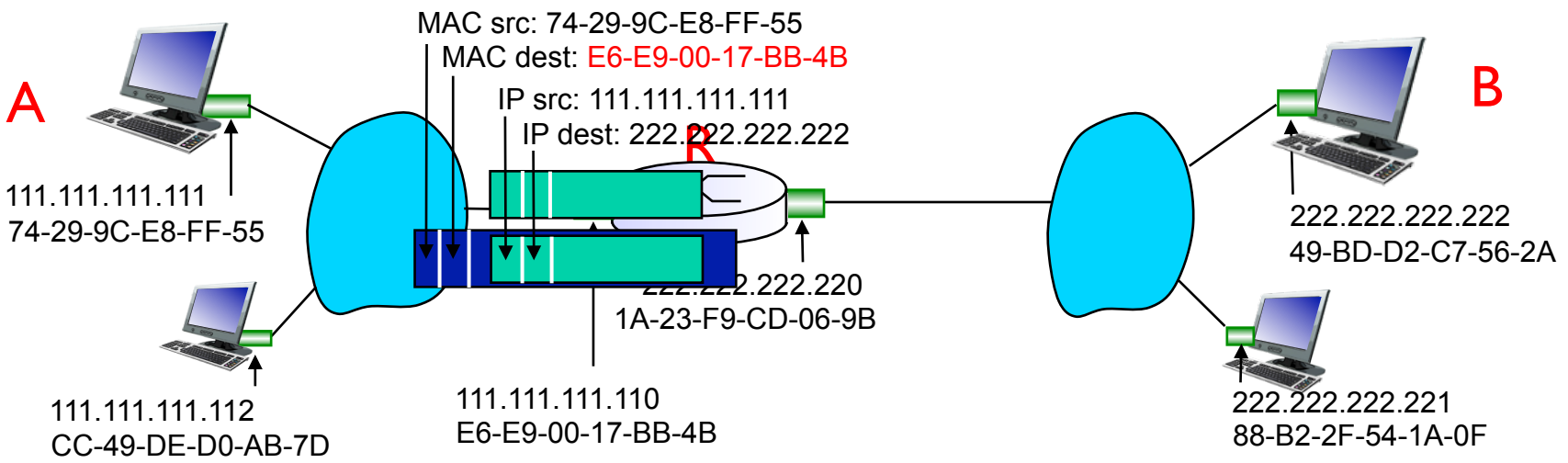
- A creates IP datagram with IP source A, destination B
- A creates link-layer frame with R's MAC address as dest, frame contains A-to-B IP datagram



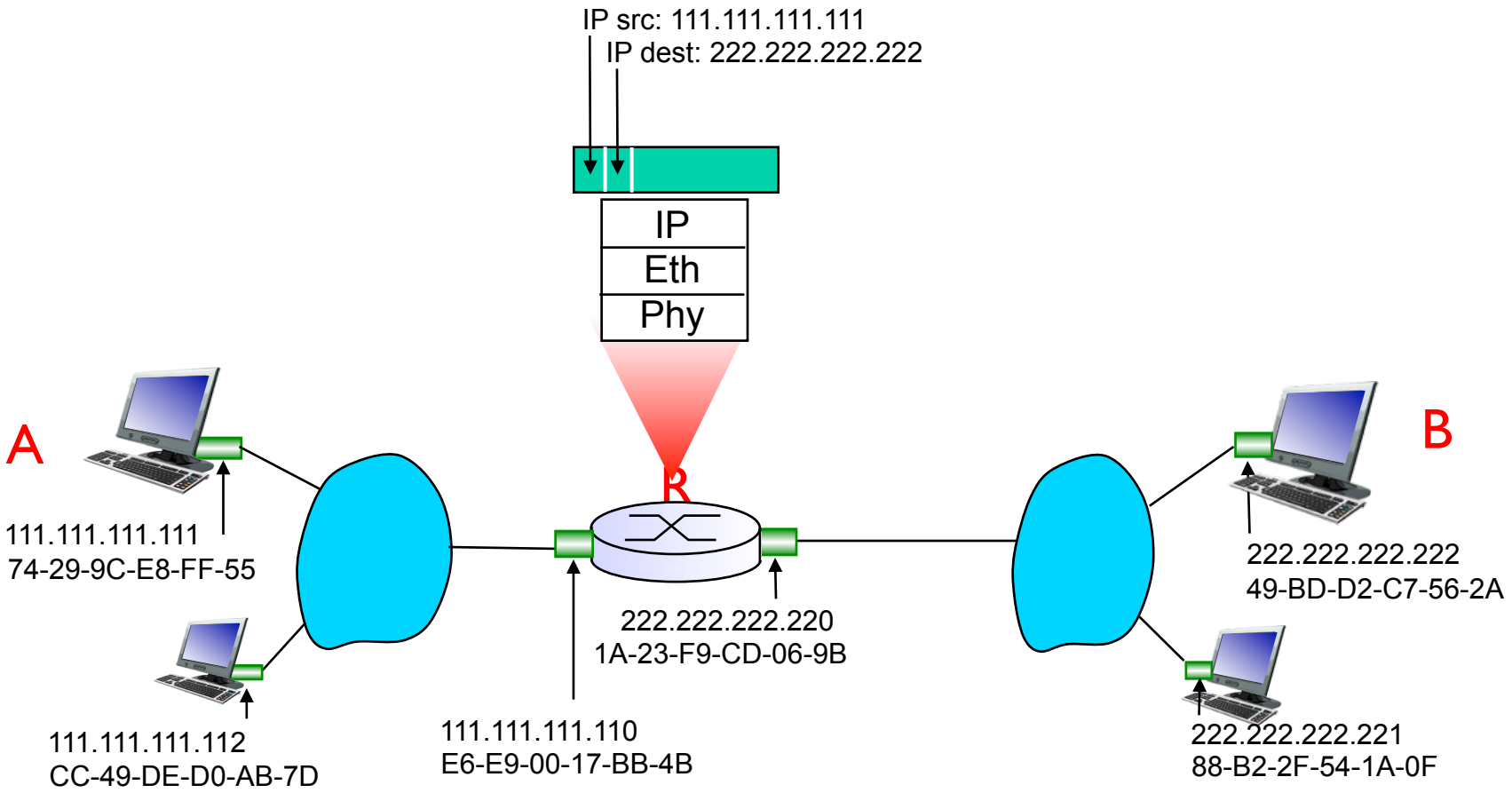
Routing to another LAN



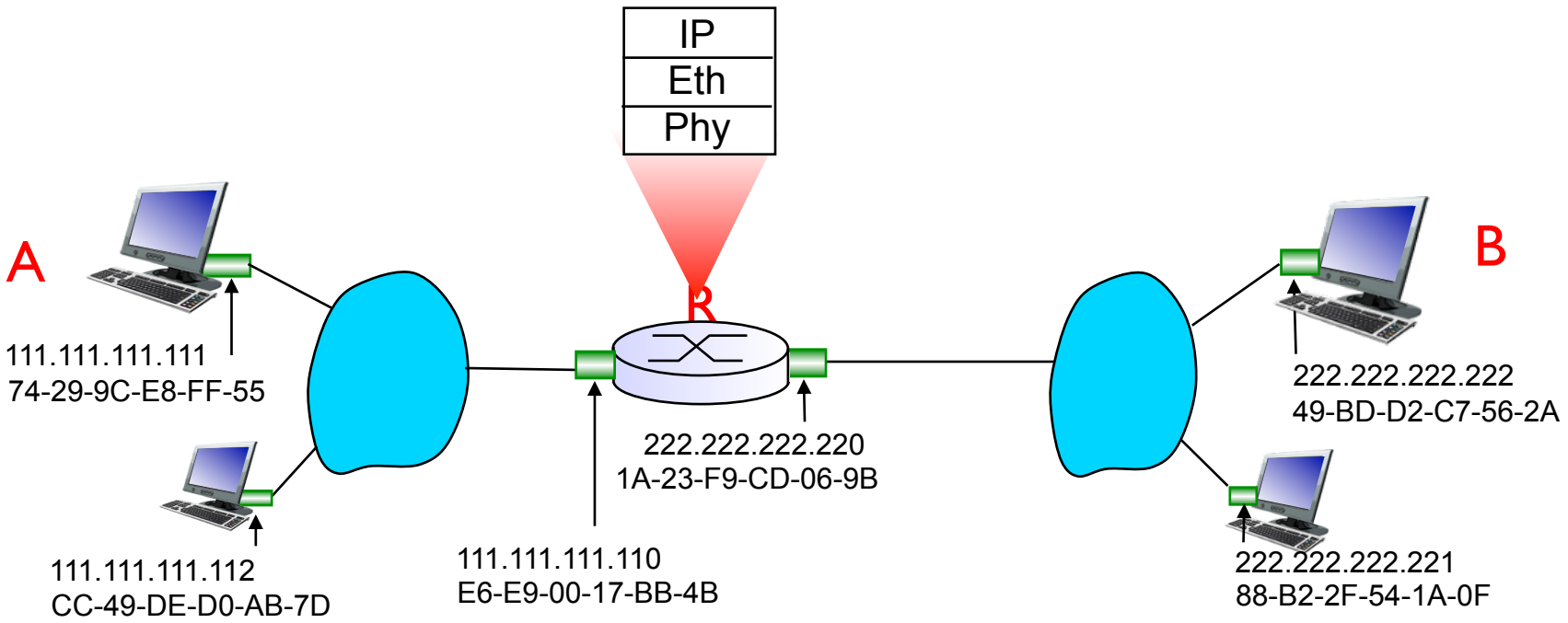
Routing to another LAN



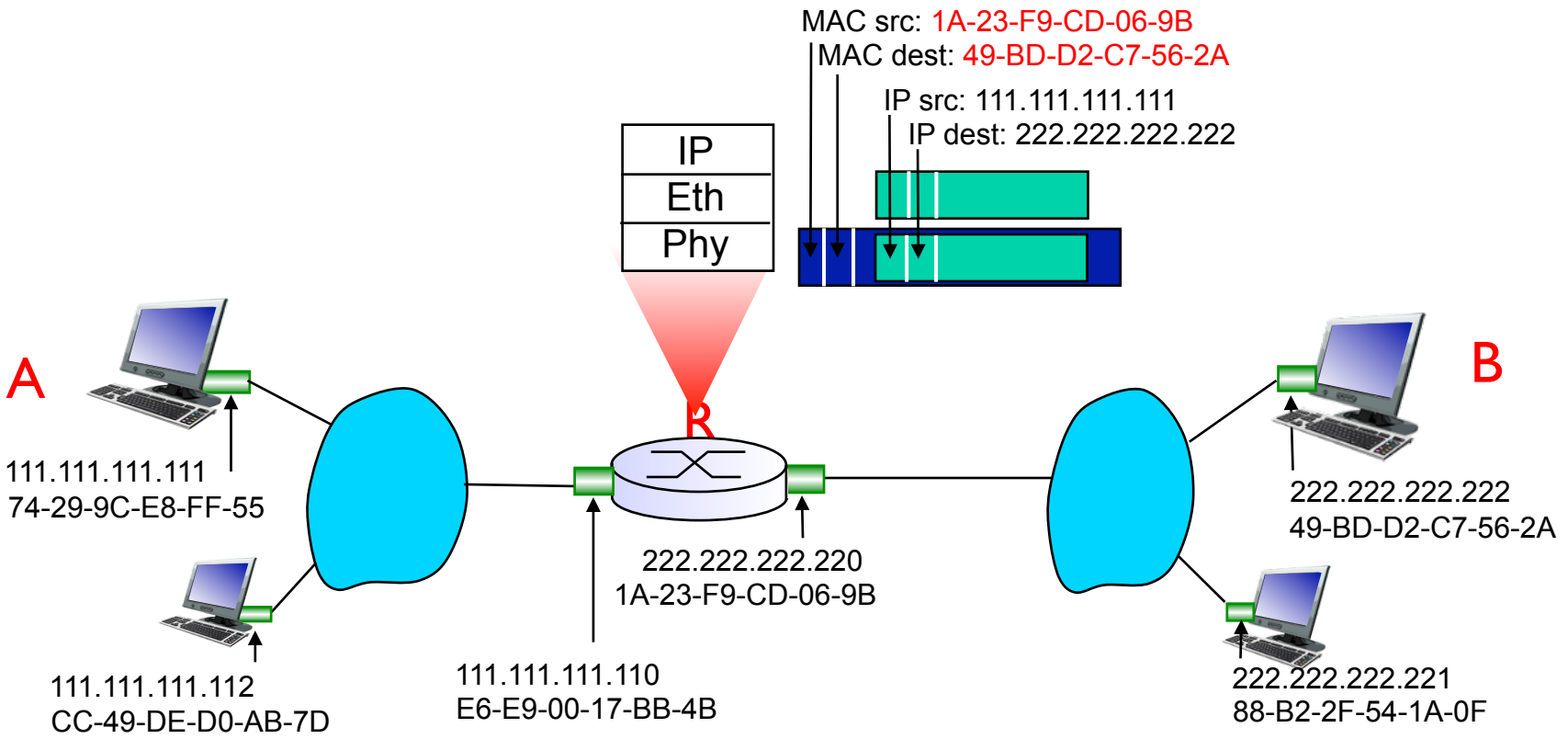
Routing to another LAN



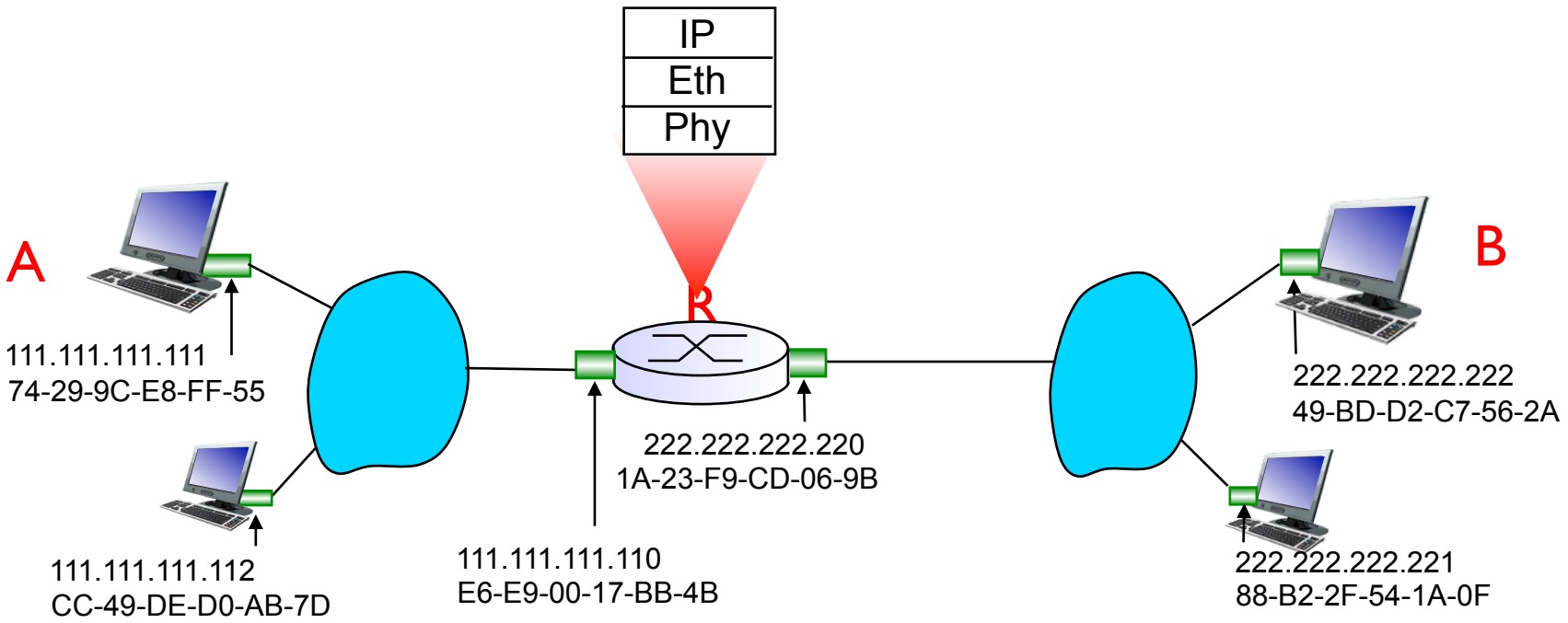
Routing to another LAN



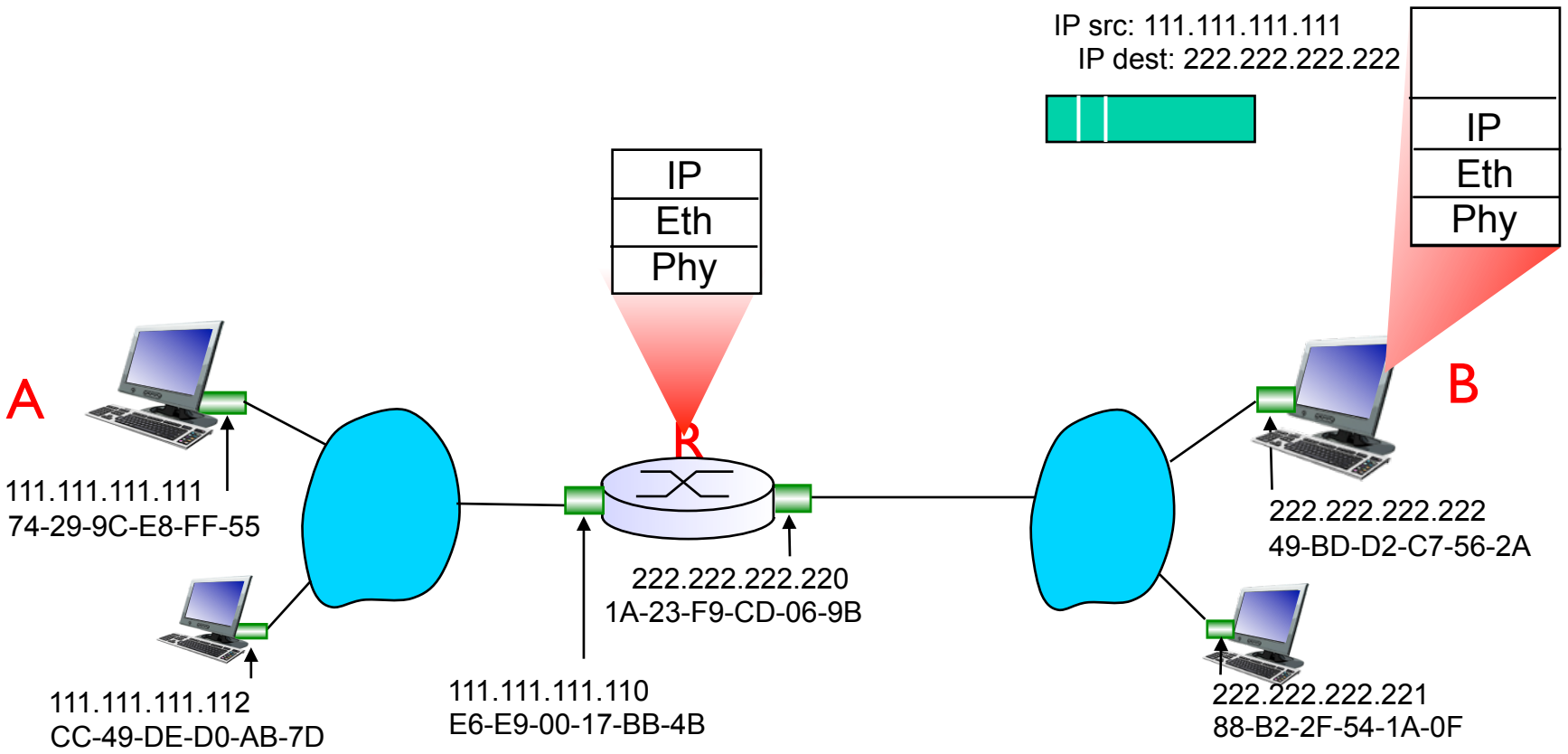
Routing to another LAN



Routing to another LAN



Routing to another LAN



ARP Spoofing/Poisoning

- ARP relies on "authentication by assertion".
  - Anyone who claims to know the mapping between IP/MAC addresses is always right.

- When someone requests an address mapping resolution, the attacker responds by injecting some other value (e.g., theirs).

- What can you do by lying about an address?

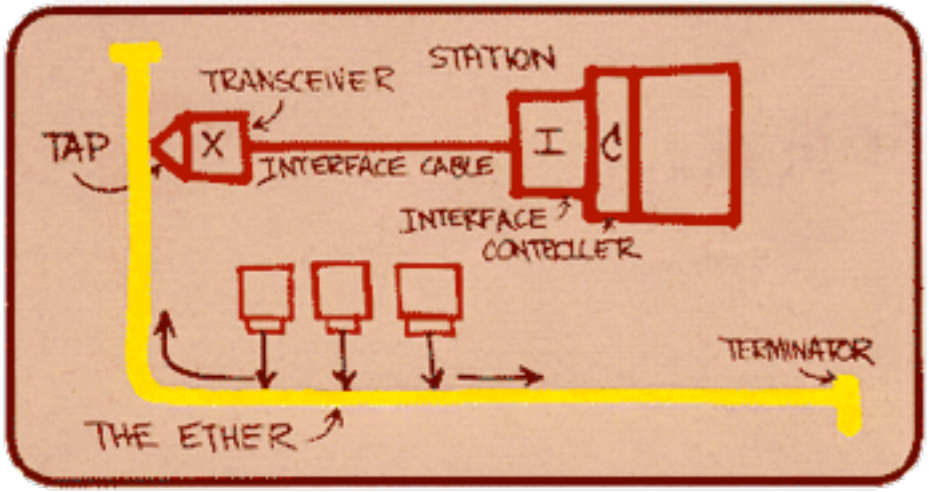
Link Layer

- 5.1 Introduction and services
- 5.2 Error detection and correction
- 5.3 Multiple access protocols
- 5.4 LANs
  - addressing, ARP
  - Ethernet
  - switches
  - VLANS
- 5.5 link virtualization: MPLS
- 5.6 data center networking
- 5.7 a day in the life of a web request

Ethernet

"Dominant" wired LAN technology:

- cheap $20 for 100Mbs!
- first widely used LAN technology
- Simpler, cheaper than token LANs and ATM
- Kept up with speed race: 10 Mbps – 10 Gbps



Metcalfe's Ethernet sketch

Pieces of History

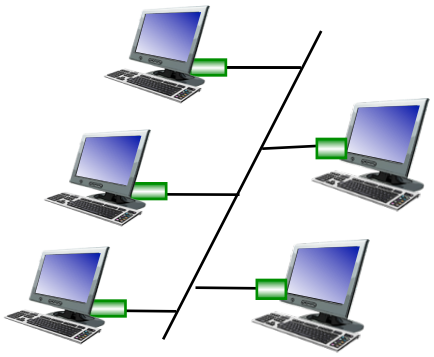
- Original Ethernet connected by 10Base5 cable
  - The "yellow garden hose" of networking
- Markings every 2.5 meters for "vampire taps"
  - Difficult to install

Errors

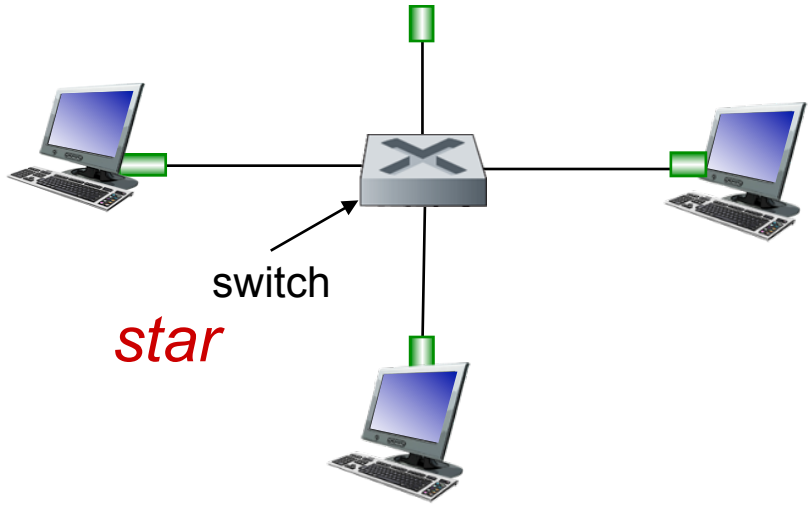
- With long pieces of wire connecting multiple machines, a single error (cable break, bad tap, loose connector) can mean trouble for everyone.
  - How long does a garden hose last before a leak occurs?
- You can determine the location of these errors by sending a special message across the wire and timing its echo.
- This technique is known as "Time Domain Reflectometry"

Star topology

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



bus: coaxial cable



Ethernet Frame Structure

Sending adapter encapsulates IP datagram (or other network layer protocol packet) in Ethernet frame

Preamble:

- 7 bytes with pattern 10101010 followed by one byte with pattern 10101011
- used to synchronize receiver, sender clock rates
  - a lecture on manchester encoding,etc will come later...

Ethernet Frame Structure (more)

- Addresses: 6 bytes
  - if adapter receives frame with matching destination address, or with broadcast address (eg ARP packet), it passes data in frame to net-layer protocol
  - otherwise, adapter discards frame
- Type: indicates the higher layer protocol (mostly IP but others may be supported such as Novell IPX and AppleTalk)
- CRC: checked at receiver, if error is detected, the frame is simply dropped

Unreliable, connectionless service

- Connectionless: No handshaking between sending and receiving adapter.
- Unreliable: receiving adapter doesn't send acks or nacks to sending adapter
  - stream of datagrams passed to network layer can have gaps (missing datagrams)
  - gaps will be filled if app is using TCP
  - otherwise, app will see the gaps
- Ethernet's MAC protocol: unslotted CSMA/CD with binary backoff.

Ethernet CSMA/CD algorithm

1. Adaptor receives datagram from net layer & creates frame

2. If adapter senses channel idle, it starts to transmit frame. If it senses channel busy, waits until channel idle and then transmits

3. If adapter transmits entire frame without detecting another transmission, the adapter is done with frame !

4. If adapter detects another transmission while transmitting, aborts and sends jam signal

5. After aborting, adapter enters exponential backoff: after the mth collision, adapter chooses a K at random from $\{0,1,2,\ldots,2^m-1\}$. Adapter waits K·512 bit times and returns to Step 2

Ethernet's CSMA/CD (more)

Jam Signal: make sure all other transmitters are aware of collision; 48 bits

Bit time: .1 microsec for 10 Mbps Ethernet ;
for K=1023, wait time is about 50 msec

Exponential Backoff:

- Goal: adapt retransmission attempts to estimated current load
  - heavy load: random wait will be longer
- first collision: choose K from {0,1}; delay is K· 512 bit transmission times
- after second collision: choose K from {0,1,2,3}…
- after ten collisions, choose K from {0,1,2,3,4,…,1023}

CSMA/CD efficiency

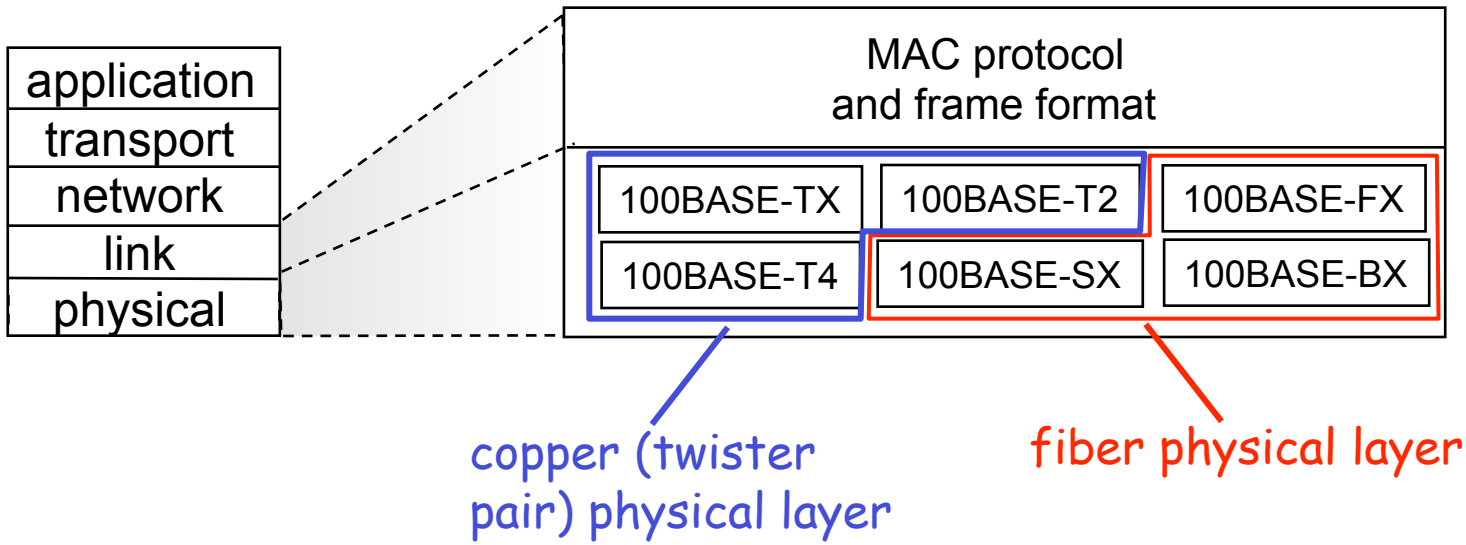
- T_{prop} = max prop between 2 nodes in LAN
- t_{trans} = time to transmit max-size frame

$$\text{efficiency} = \frac{1}{1 + 5t_{prop} / t_{trans}}$$

- efficiency goes to 1
  - as t_{prop} goes to 0
  - as t_{trans} goes to infinity
- better performance than ALOHA: and simple, cheap, decentralized!

802.3 Ethernet Standards: Link

- many different Ethernet standards
  - common MAC protocol and frame format
  - different speeds: 2 Mbps, 10 Mbps, 100 Mbps, 1Gbps, 10G bps
  - different physical layer media: fiber, cable



Gbit Ethernet

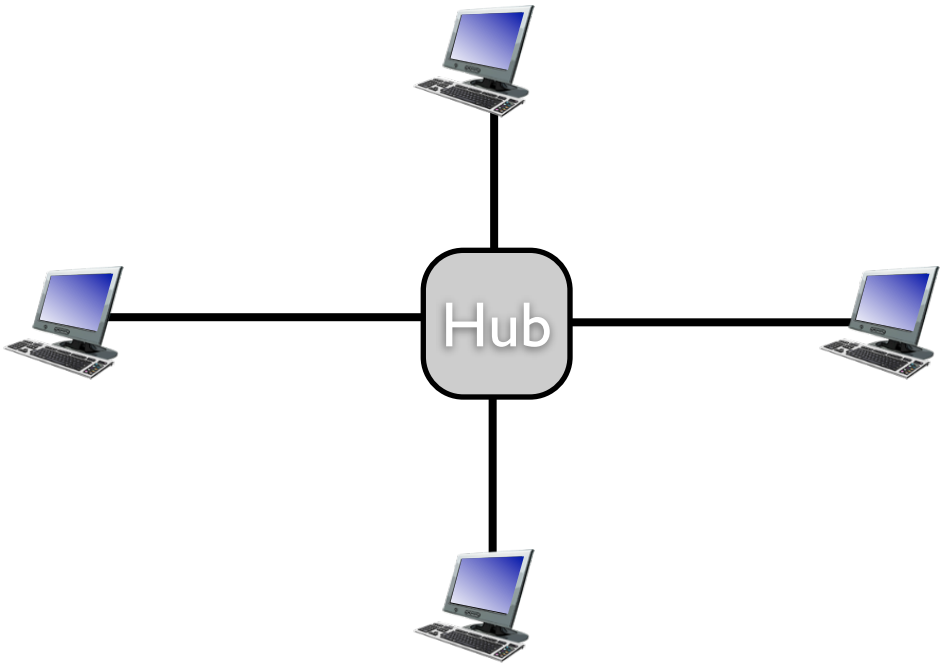
- uses standard Ethernet frame format
- allows for point-to-point links and shared broadcast channels
- in shared mode, CSMA/CD is used; short distances between nodes required for efficiency
- uses hubs, called here "Buffered Distributors"
- Full-Duplex at 1 Gbps for point-to-point links
- 10 Gbps now !

Link Layer

- 5.1 Introduction and services
- 5.2 Error detection and correction
- 5.3 Multiple access protocols
- 5.4 LANs
  - addressing, ARP
  - Ethernet
  - switches
  - VLANS
- 5.5 link virtualization: MPLS
- 5.6 data center networking
- 5.7 a day in the life of a web request

Hubs

- … physical-layer ("dumb") repeaters:
  - bits coming in one link go out all other links at same rate
  - all nodes connected to hub can collide with one another
  - no frame buffering
  - no CSMA/CD at hub: host NICs detect collisions

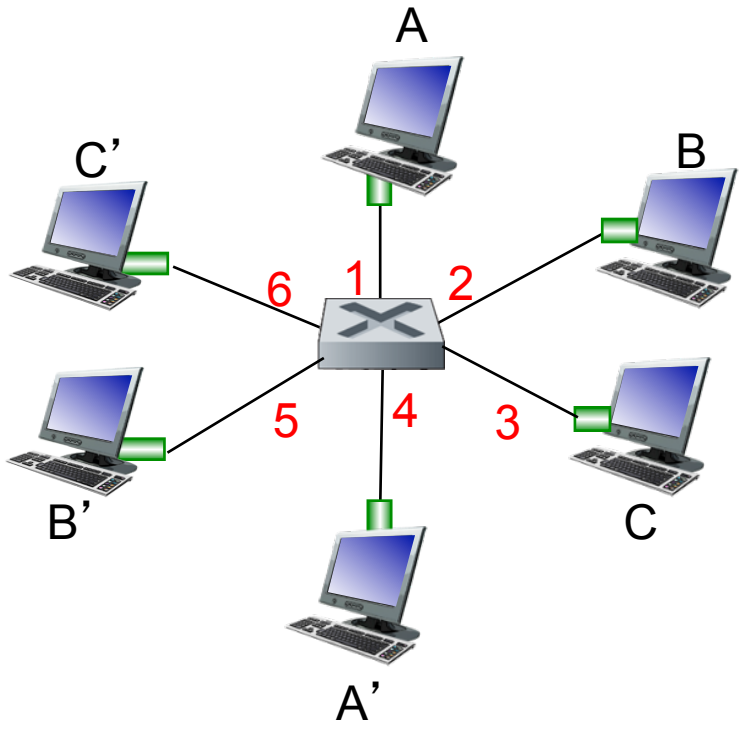


Switch

- *link-layer device*: smarter than hubs, take 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 to access segment
- transparent
  - hosts are unaware of presence of switches
- plug-and-play, self-learning
  - switches do not need to be configured

Switch: Multiple Transmissions

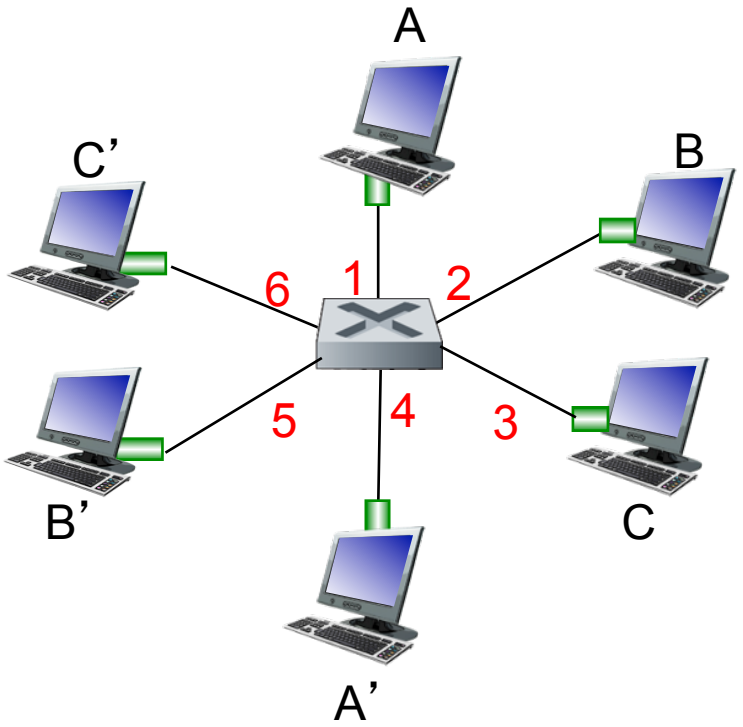
- hosts have dedicated, direct connection to switch
- switches buffer packets
- Ethernet protocol used on each incoming link, but no collisions; full duplex
  - each link is its own collision domain
- switching: A-to-A' and B-to-B' simultaneously, without collisions
  - not possible with dumb hub



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

Switch Table

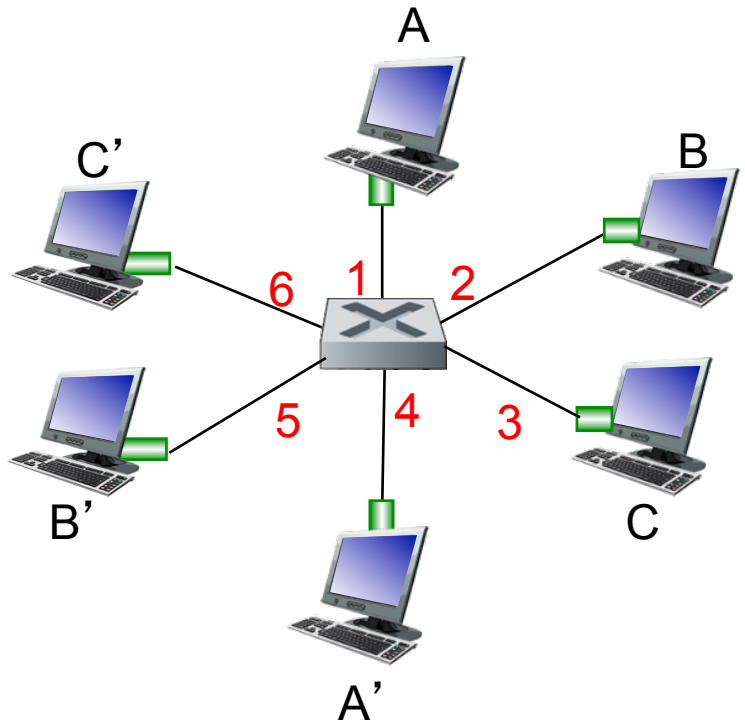
- Q: how does switch know that A' reachable via interface 4, B' reachable via interface 5?
- A: each switch has a switch table, each entry:
  - (MAC address of host, interface to reach host, time stamp)
  - looks like a routing table!
- Q: how are entries created, maintained in switch table?
  - something like a routing protocol?



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

Self learning

- switch learns which hosts can be reached through which interfaces
  - when frame received, switch "learns" location of sender: incoming LAN segment
  - records sender/location pair in switch table



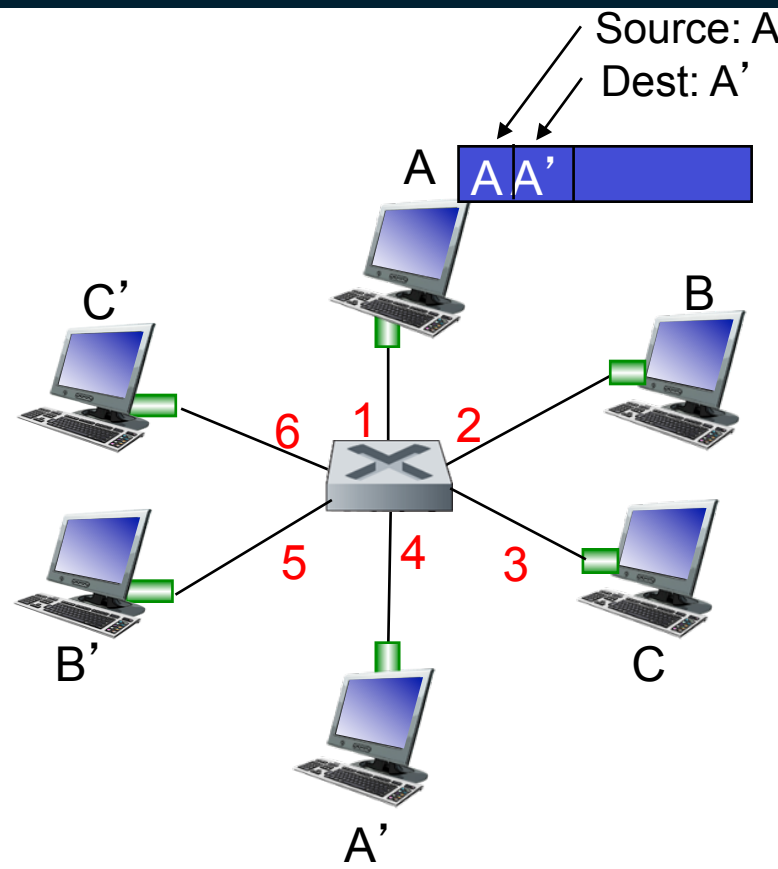
switch with six interfaces (1,2,3,4,5,6)

MAC addr	interface	TTL
A	*1*	*60*

Switch table (initially empty)

Self learning

- switch learns which hosts can be reached through which interfaces
  - when frame received, switch "learns" location of sender: incoming LAN segment
  - records sender/location pair in switch table



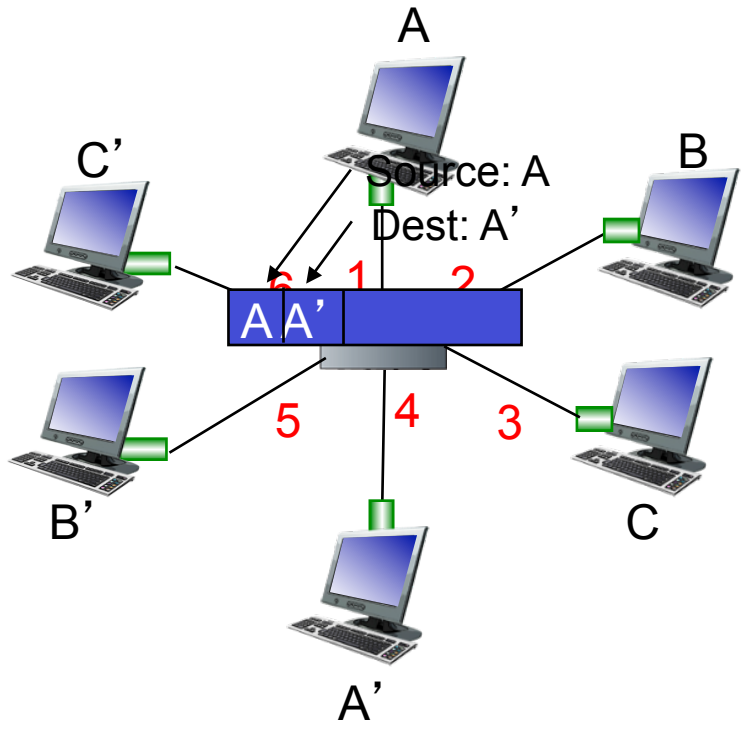
switch with six interfaces (1,2,3,4,5,6)

MAC addr	interface	TTL
A	*1*	*60*

Switch table (initially empty)

Self learning

- switch learns which hosts can be reached through which interfaces
  - when frame received, switch "learns" location of sender: incoming LAN segment
  - records sender/location pair in switch table



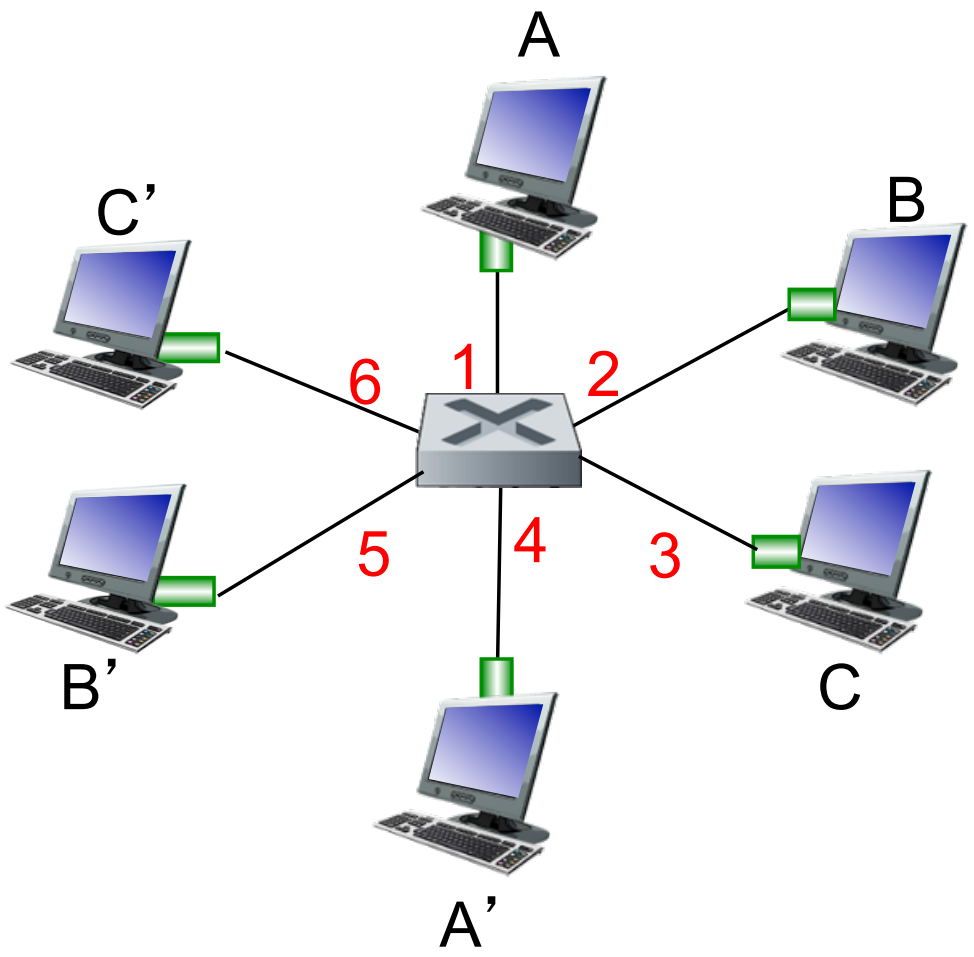
switch with six interfaces (1,2,3,4,5,6)

MAC addr	interface	TTL
A	*1*	*60*

Switch table (initially empty)

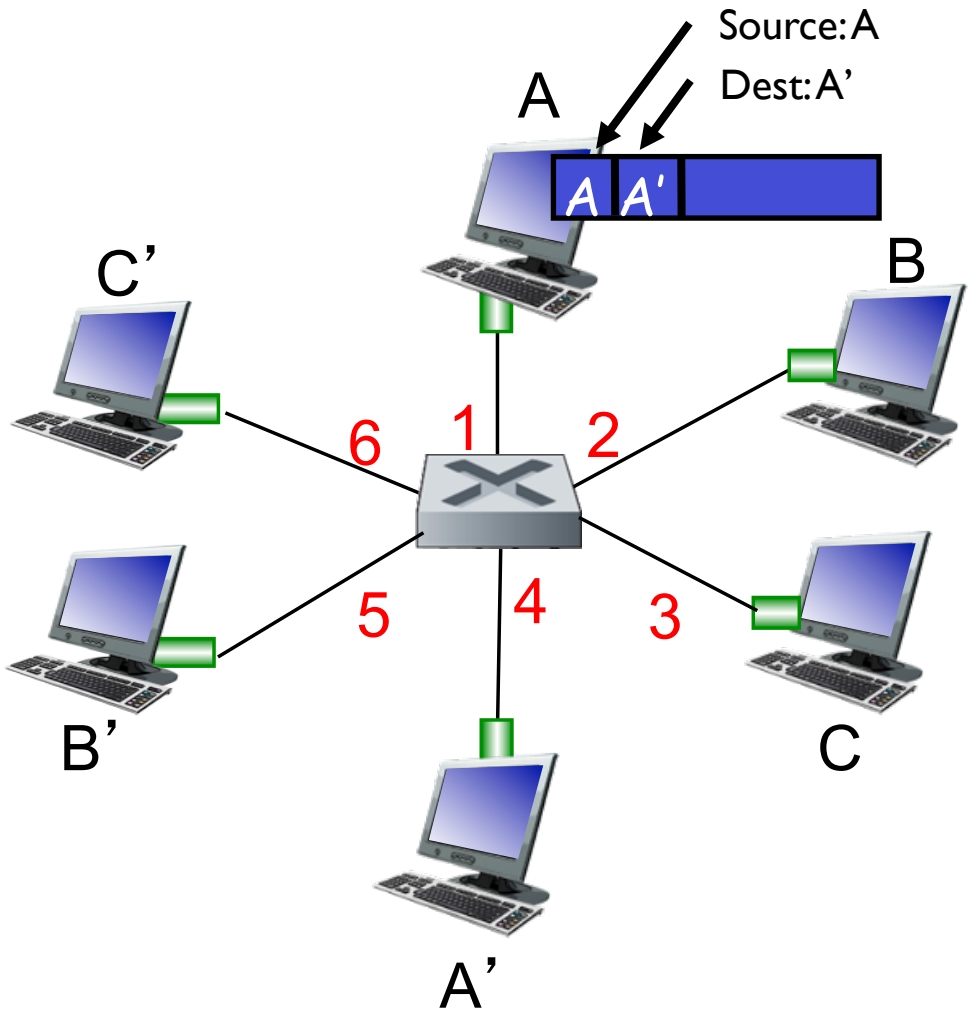
Self-Learning, Forwarding: Example

- Frame Destination unknown: *flood*
- Destination A location known: *selective send*



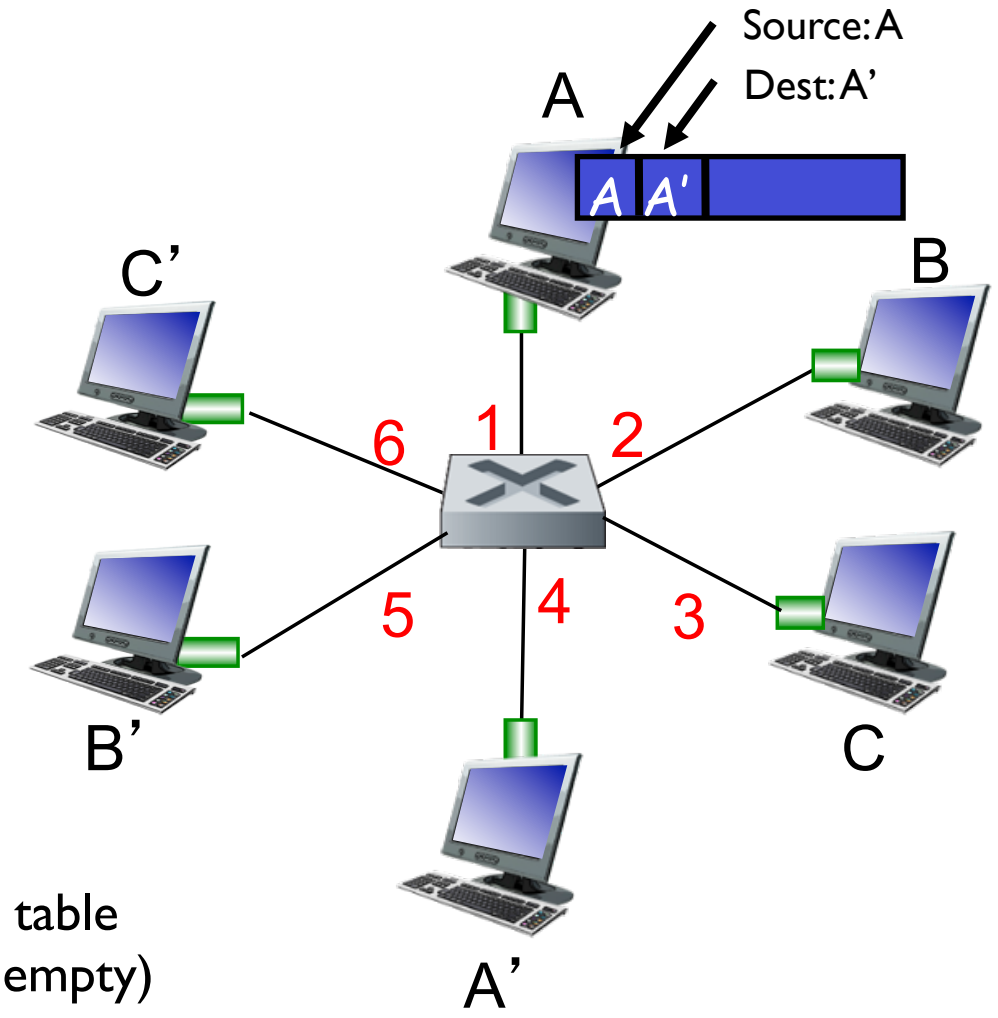
Self-Learning, Forwarding: Example

- Frame Destination unknown: *flood*
- Destination A location known: *selective send*



Self-Learning, Forwarding: Example

- Frame Destination unknown: *flood*
- Destination A location known: *selective send*

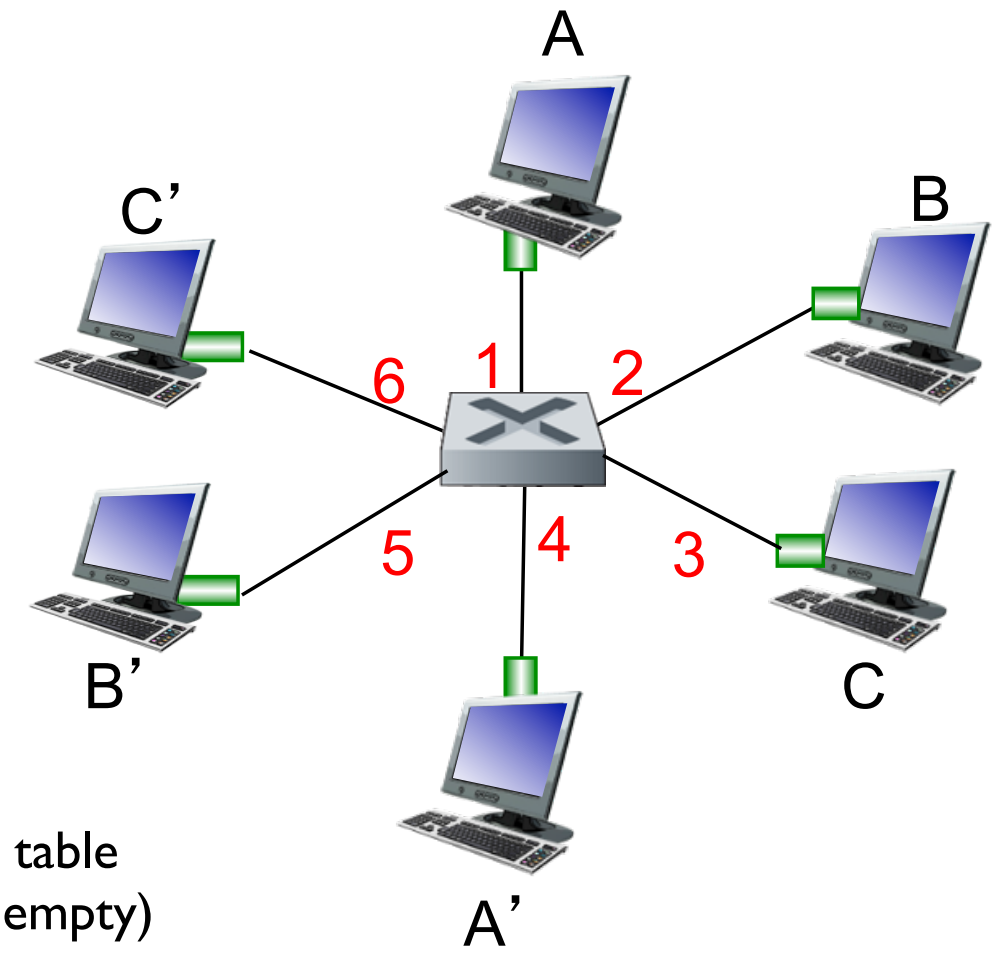


MAC addr	interface	TTL

Switch table
(initially empty)

Self-Learning, Forwarding: Example

- Frame Destination unknown: *flood*
- Destination A location known: *selective send*

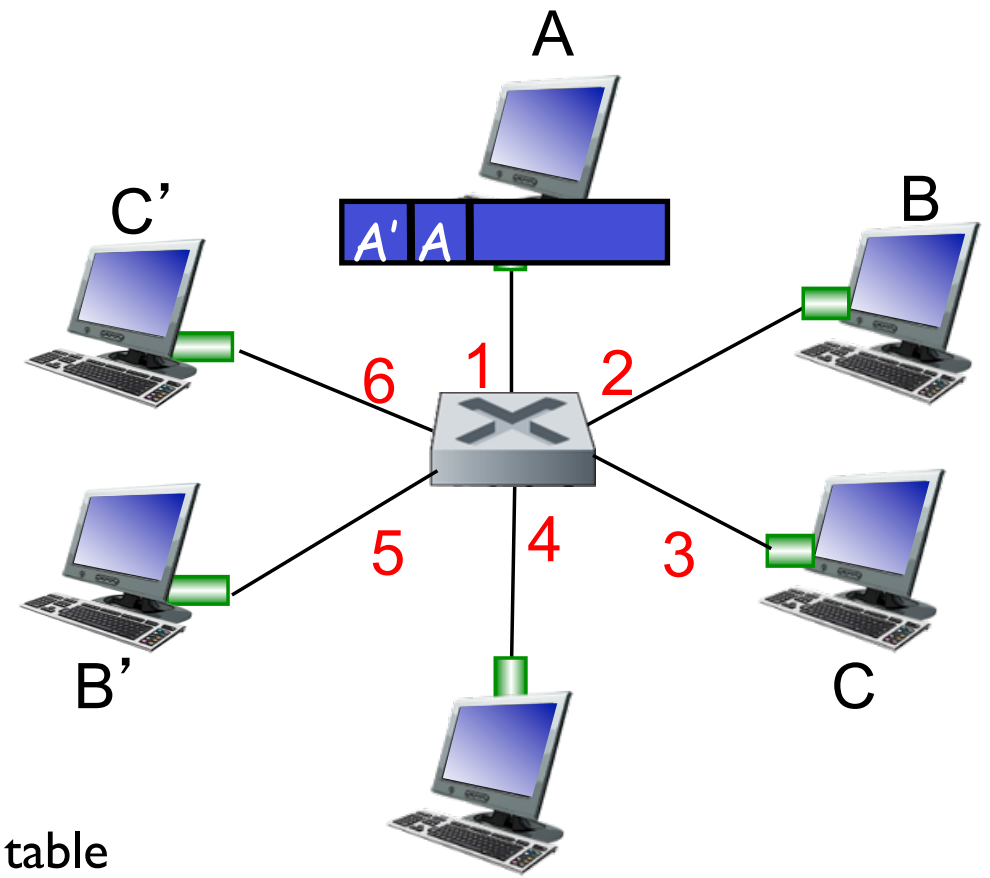


MAC addr	interface	TTL
A	1	60

Switch table
(initially empty)

Self-Learning, Forwarding: Example

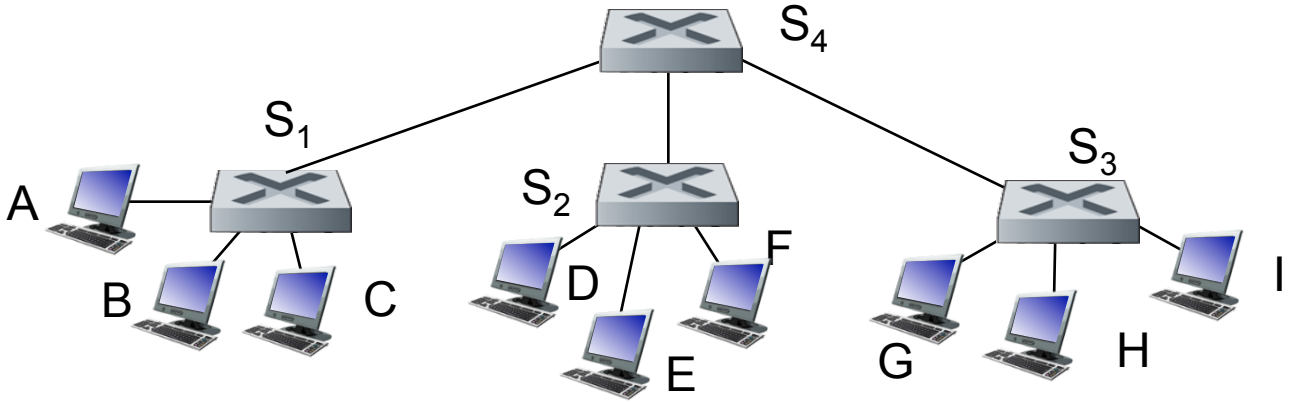
- Frame Destination unknown: *flood*
- Destination A location known: *selective send*



MAC addr	interface	TTL
A	1	60
A'	4	60

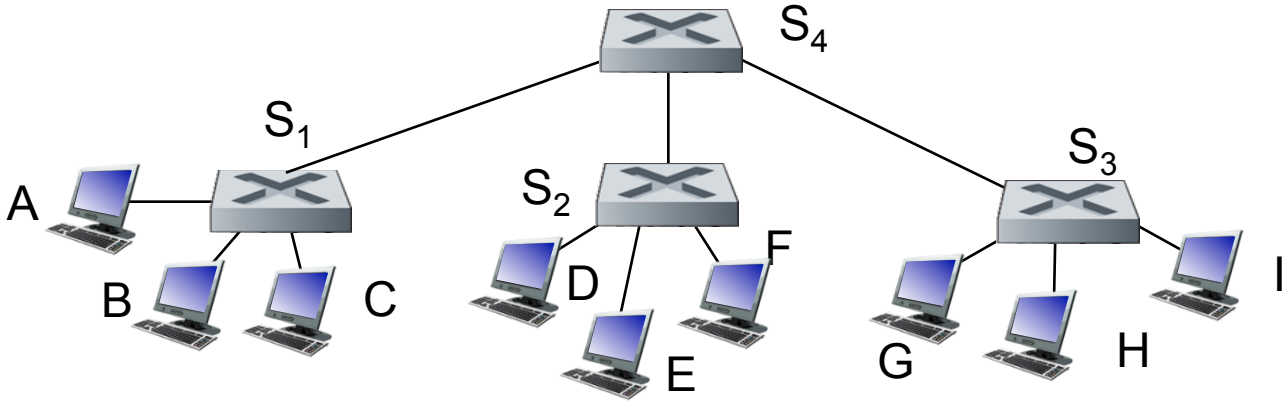
Switch table
(initially empty)

Interconnecting Switches



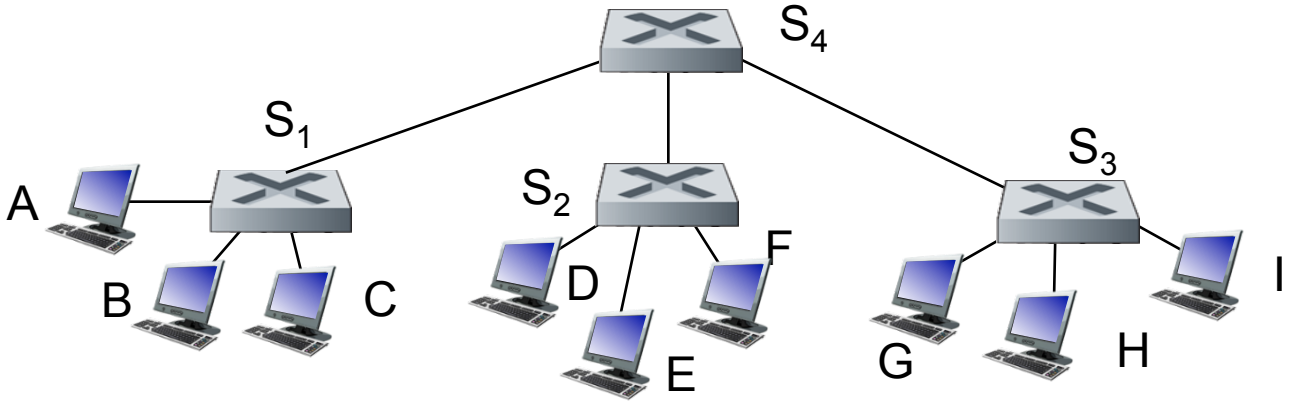
Interconnecting Switches

- Switches can be connected together



Interconnecting Switches

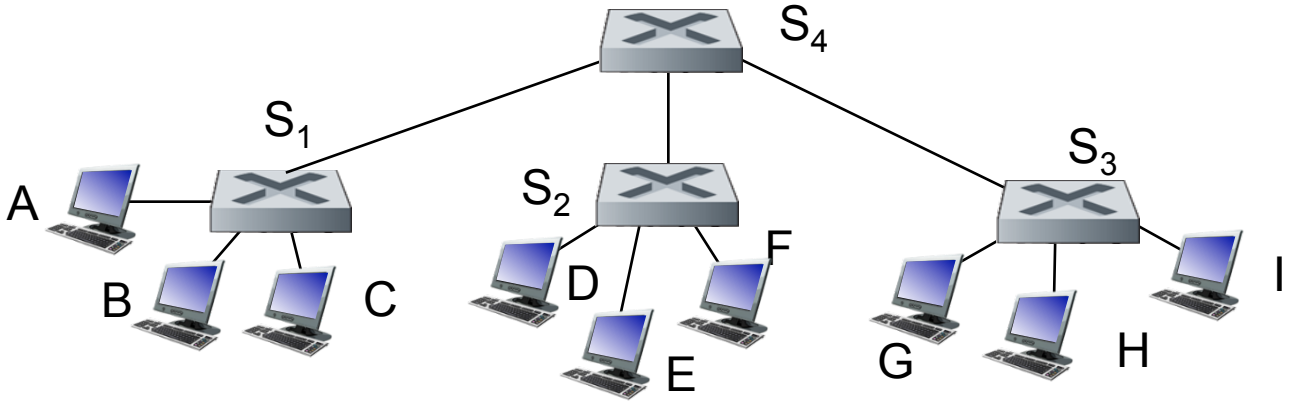
- Switches can be connected together



- Q: sending from A to G - how does S1 know to forward frame destined to F via S4 and S3?

Interconnecting Switches

- Switches can be connected together

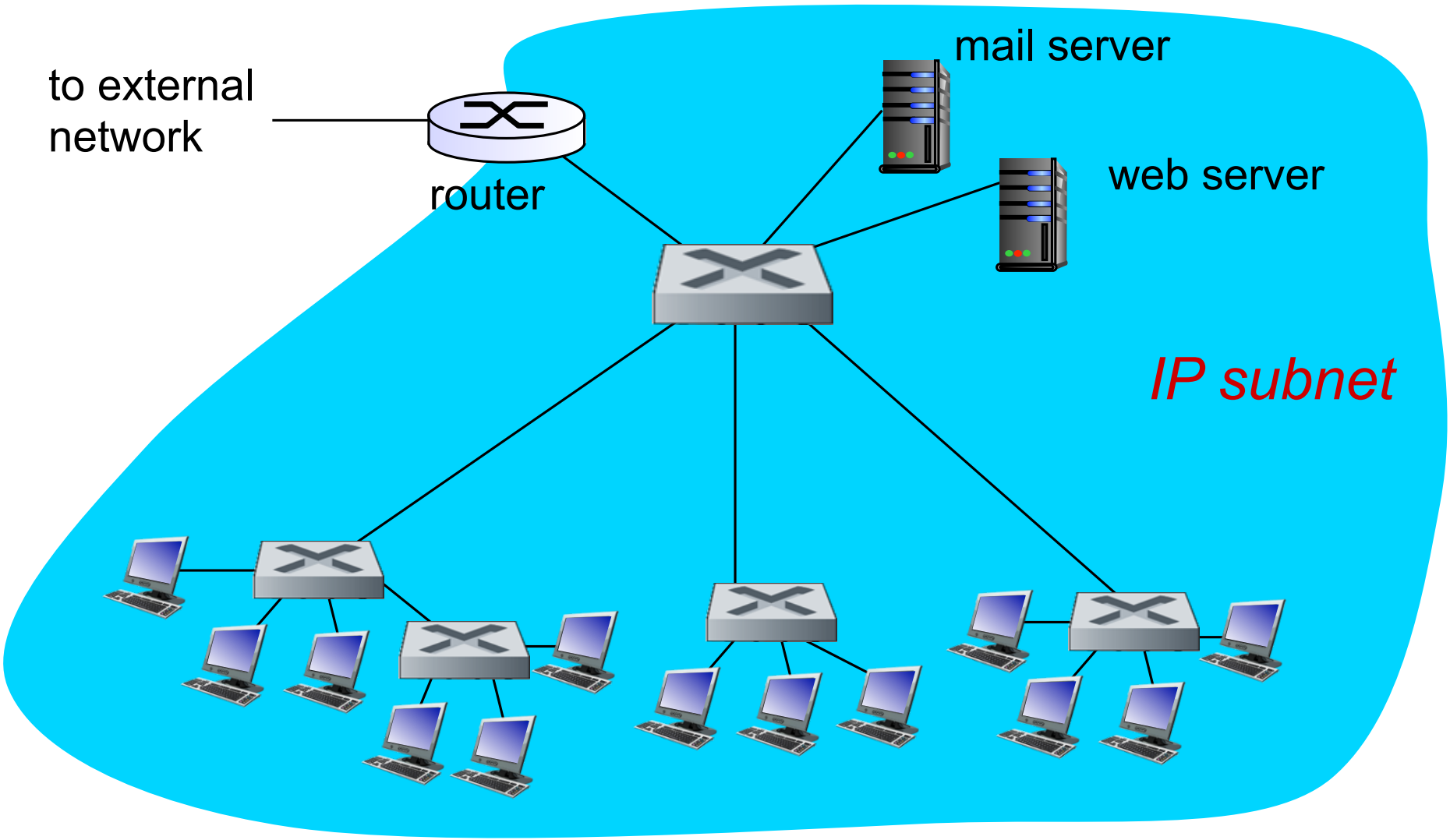


- Q: sending from A to G - how does S1 know to forward frame destined to F via S4 and S3?
- A: self learning! (works exactly the same as in single-switch case!)

More on Switches

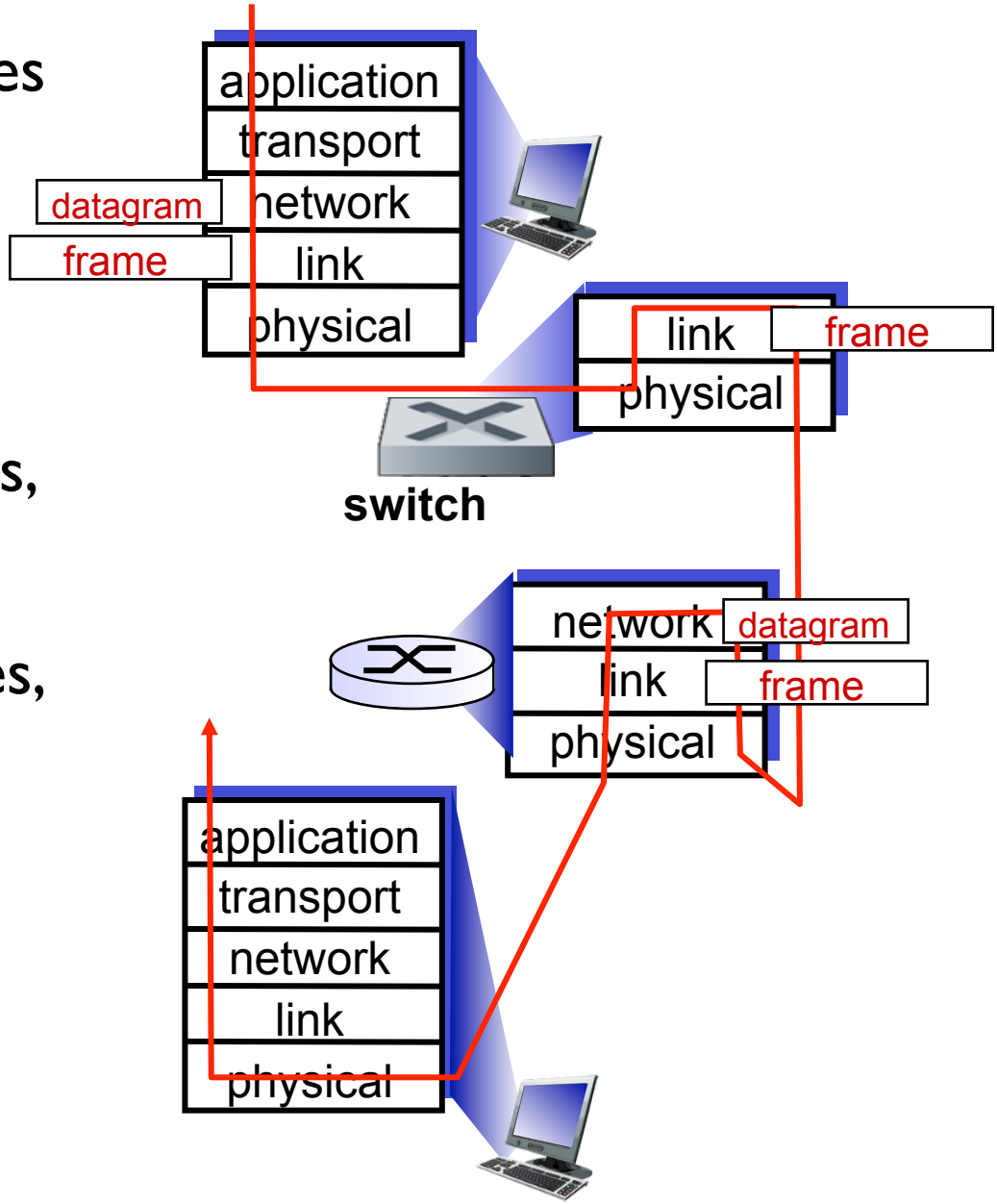
- cut-through switching: frame forwarded from input to output port without first collecting entire frame
  - slight reduction in latency
- combinations of shared/dedicated, 10/100/1000 Mbps interfaces

Institutional Network

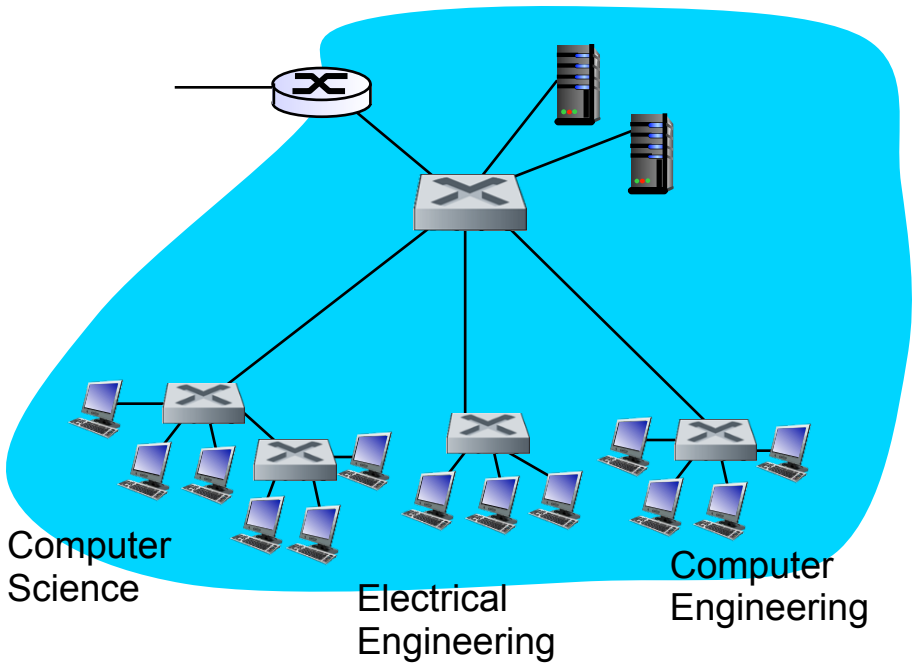


Switches vs. Routers

- both store-and-forward devices
  - routers: network layer devices (examine network layer headers)
  - switches are link layer devices
- routers maintain routing tables, implement routing algorithms
- switches maintain switch tables, implement filtering, learning algorithms



VLANs: Motivation



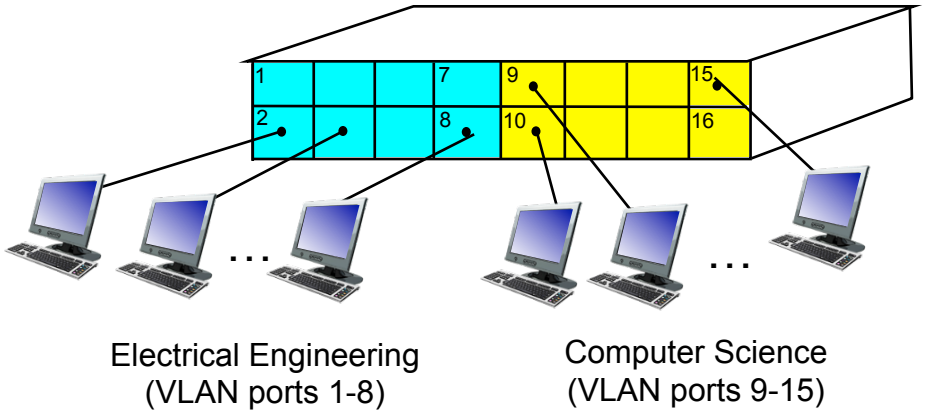
- Consider the following scenario:
  - CS user moves office to EE, but wants connect to CS switch?
  - single broadcast domain:
  - all layer-2 broadcast traffic (ARP, DHCP, unknown location of destination MAC address) must cross entire LAN
  - security/privacy, efficiency issues

VLANs

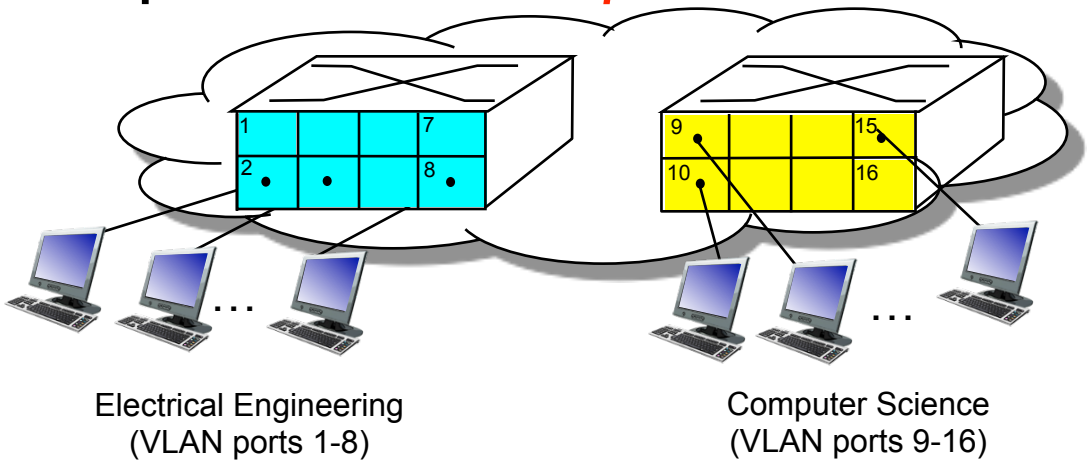
Virtual Local Area Network

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

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

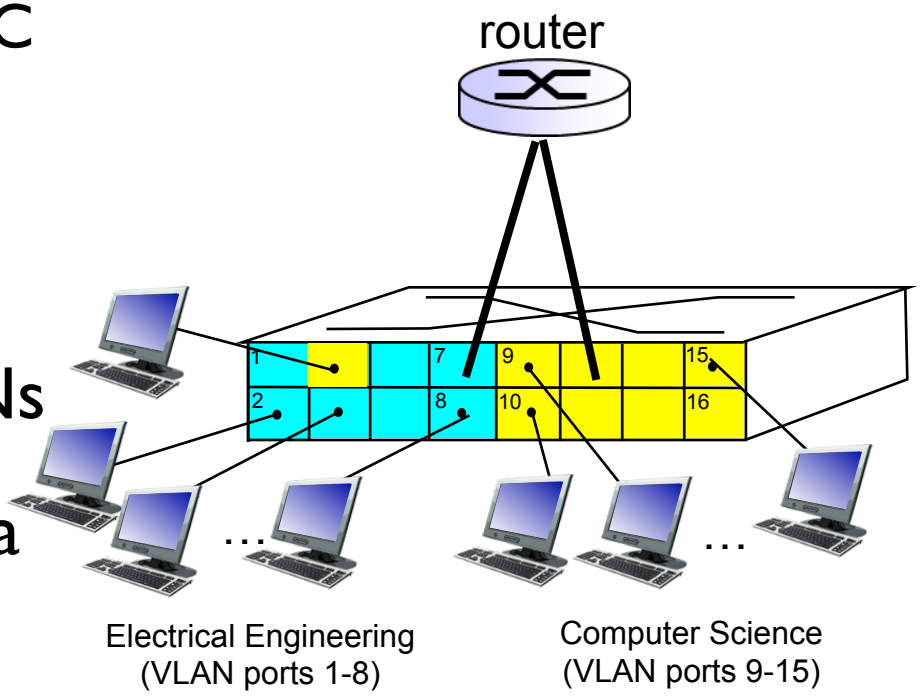


- … operates as *multiple* virtual switches

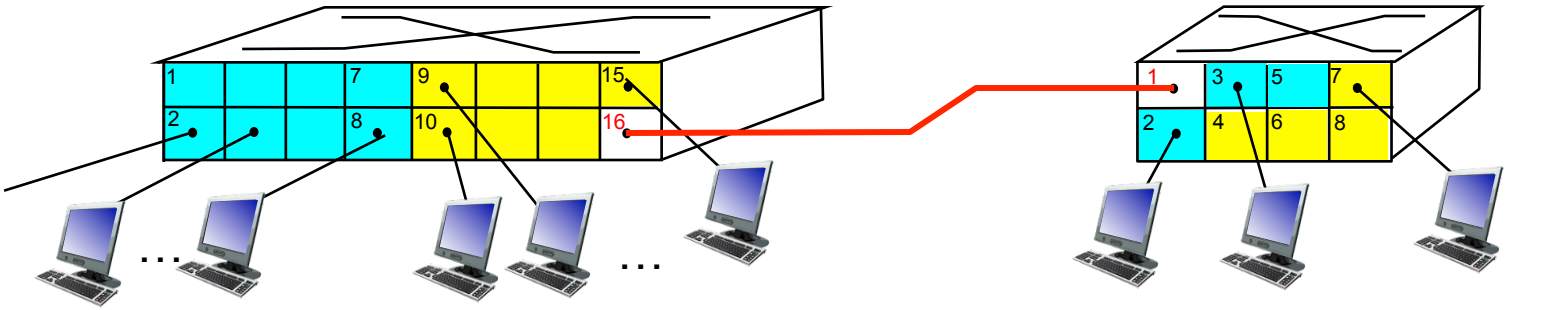


Port-Based VLAN

- *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

Next Time...

- Fall Break!
- No more material before the midterm.
  - Sounds like a great time to start studying...