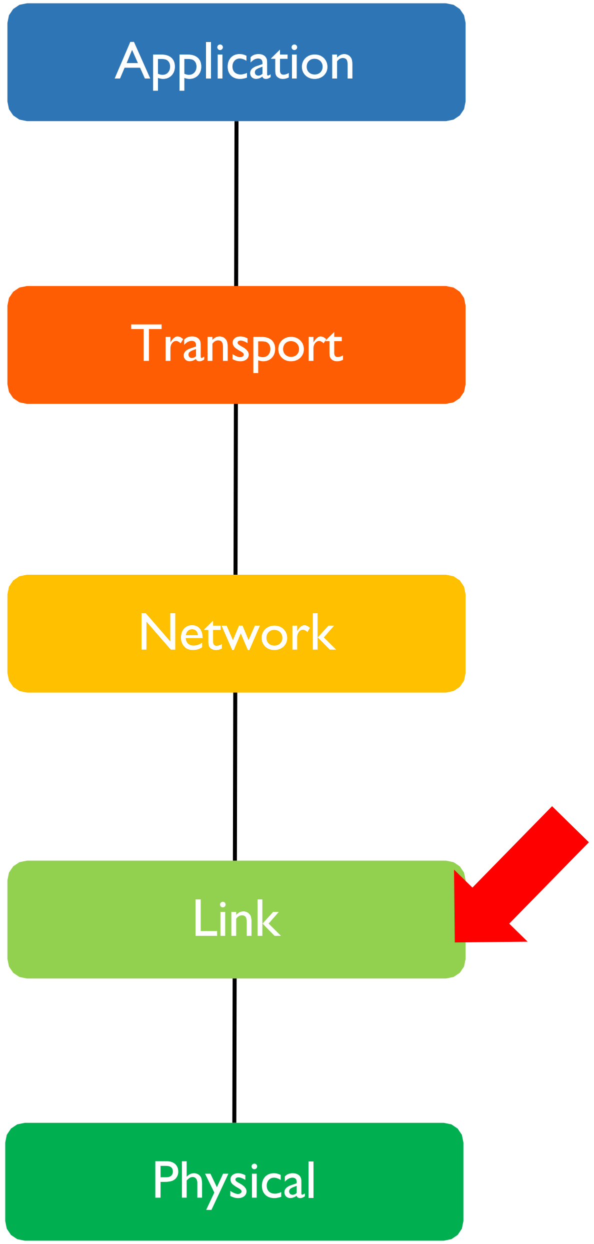
Lesson 08-01: MAC and ~~Cheese~~ Ethernet

CS 356 Computer Networks

Mikyung Han

mhan@cs.utexas.edu

Internet Reference Model



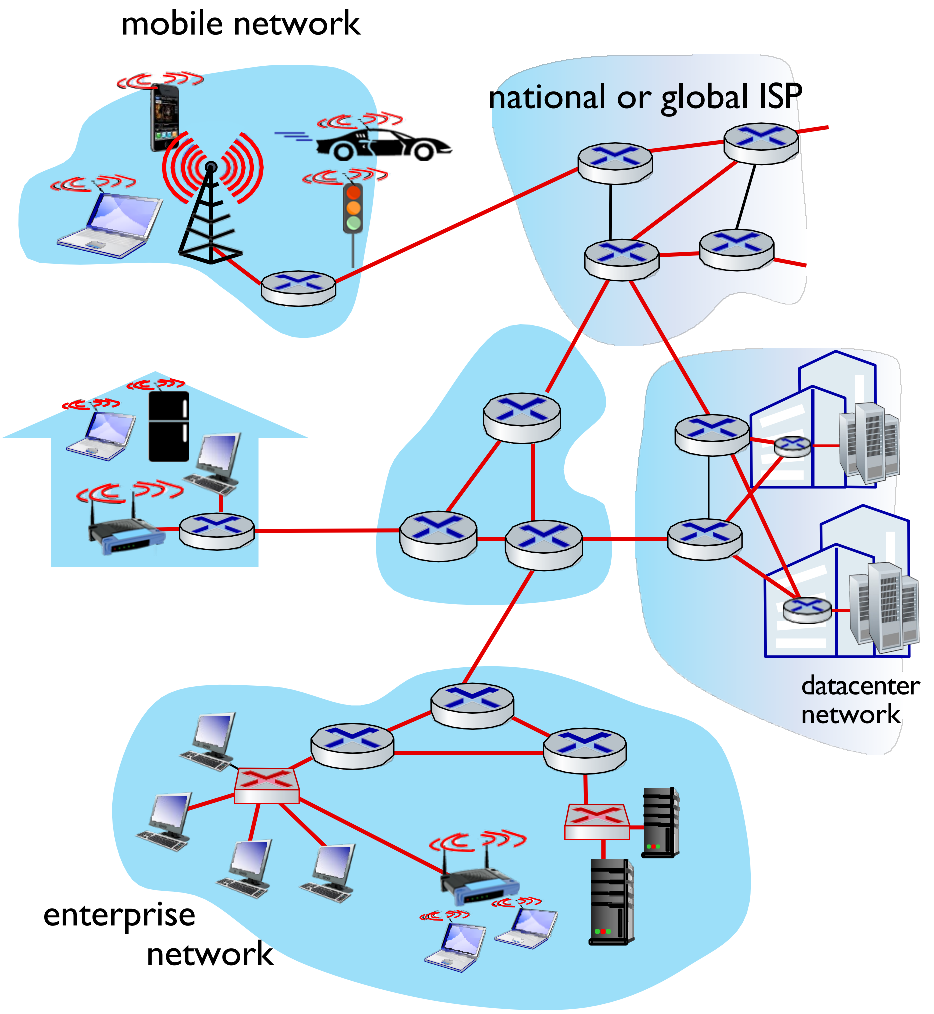
Example Protocols		Responsible for
FTP, HTTP, SMTP	Application	application specific needs
TCP, UDP	Transport	process to process data transfer
IP	Network	host to host data transfer across different network
Ethernet, WiFi	Link	data transfer between physically adjacent nodes
802.3 PHY	Physical	bit-by-bit or symbol-by-symbol delivery

Outline

1. Link Layer Intro

Link layer L2 Terminology

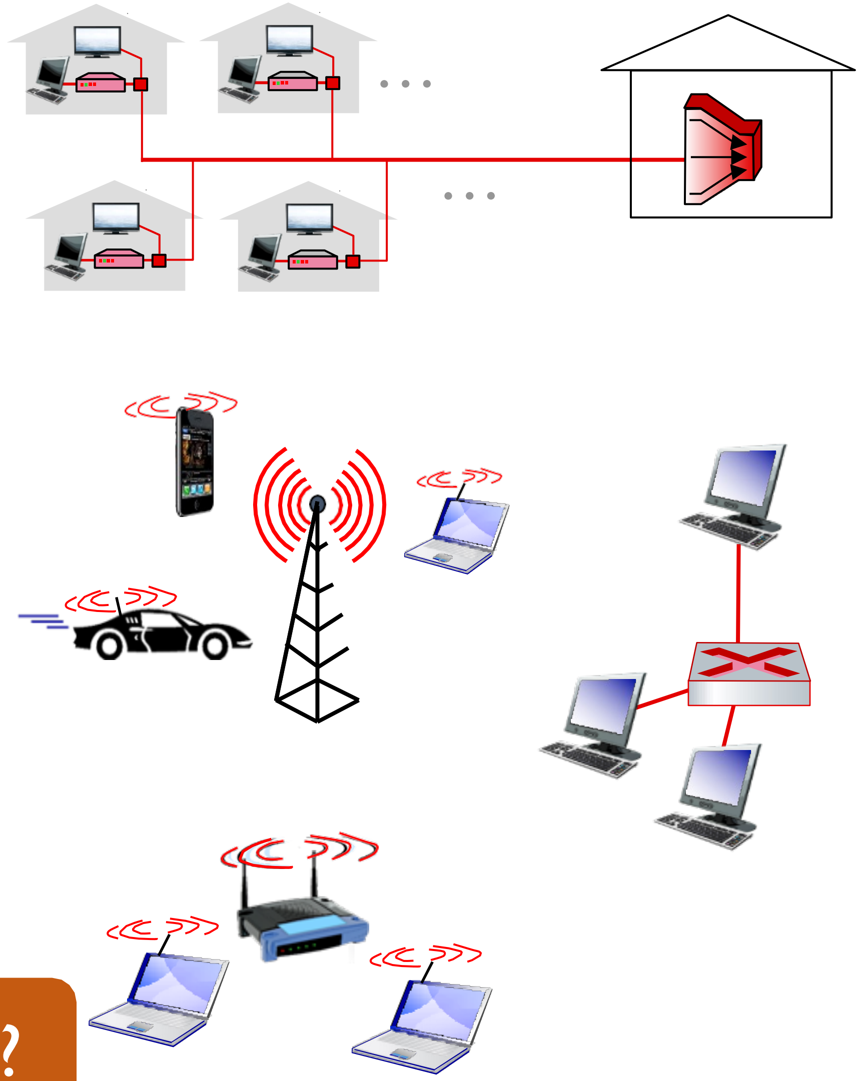
- hosts and routers: [redacted]
- communication channels that connect adjacent nodes along communication path: [redacted]
  - Wired or wireless
  - LANs
- Packet in L2 is called [redacted], encapsulates [redacted]



L2 is responsible for transferring packets between physically adjacent nodes

What services does L2 provide?

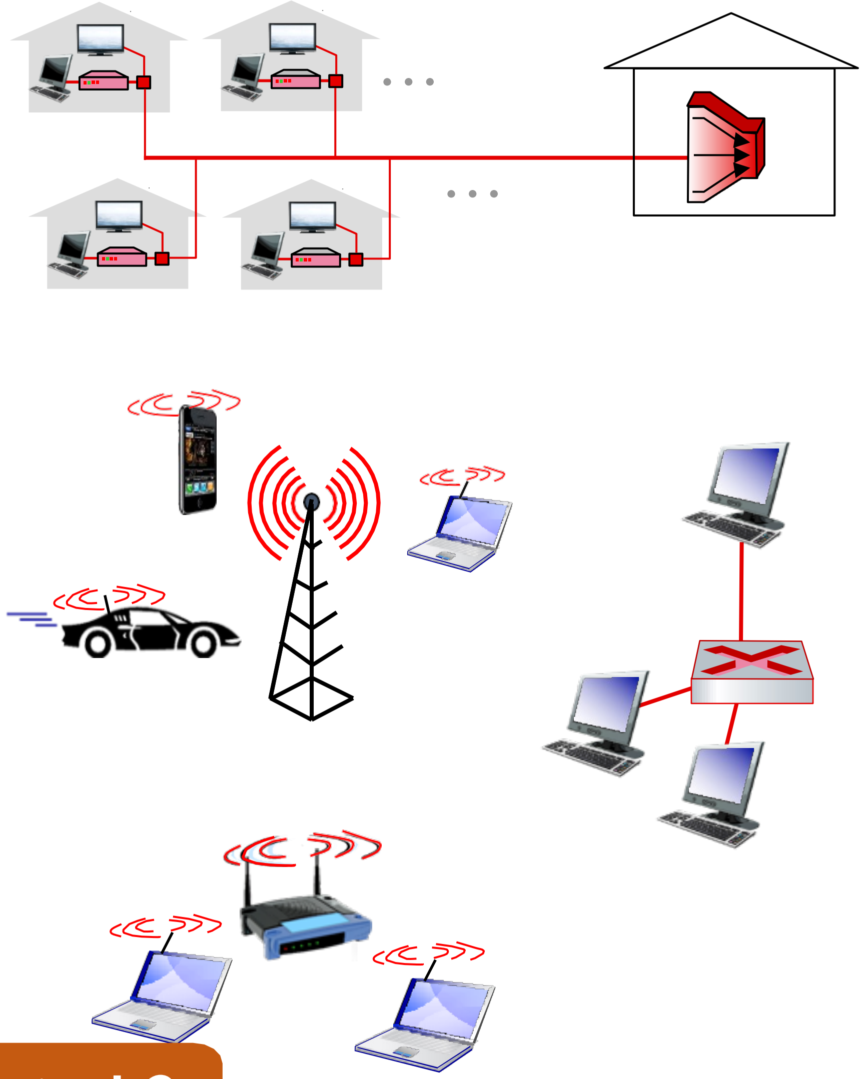
- framing, link access:
  - encapsulate datagram into frame, adding header, trailer
  - channel access if shared medium
  - “MAC” addresses in frame headers identify source, destination (different from IP address!)
- reliable delivery between adjacent nodes
  - seldom used on low bit-error links
  - wireless links: high error rates



Why both link-level and end-to-end reliability?

What services does L2 provide?

- **flow control:**
  - pacing between adjacent sending and receiving nodes
- **error detection:**
  - errors caused by signal attenuation, noise.
  - receiver detects errors, signals retransmission, or drops frame
- **error correction:**
  - receiver identifies and corrects bit error(s) without retransmission
- **half-duplex and full-duplex:**
  - with half duplex, nodes at both ends of link can transmit, but not at same time



We have seen these before - Same principle holds in L2

Outline

1. Link Layer Intro
2. Addressing: MAC recap

For network layer, IP address is used

- e.g.: IPv4 32-bit IP address
  - network-layer address for interface
  - used for layer 3 (network layer) forwarding
  - e.g.: 128.119.40.136

Link layer has a different address: MAC addresses

- MAC (or LAN or physical or Ethernet) address:
  - function: used "locally" to get frame from one interface to another physically-connected interface (same subnet, in IP-addressing sense)
  - 48-bit MAC address (for most LANs) burned in NIC ROM, also sometimes software settable
  - e.g.: 1A-2F-BB-76-09-AD

hexadecimal (base 16) notation
(each "numeral" represents 4 bits)

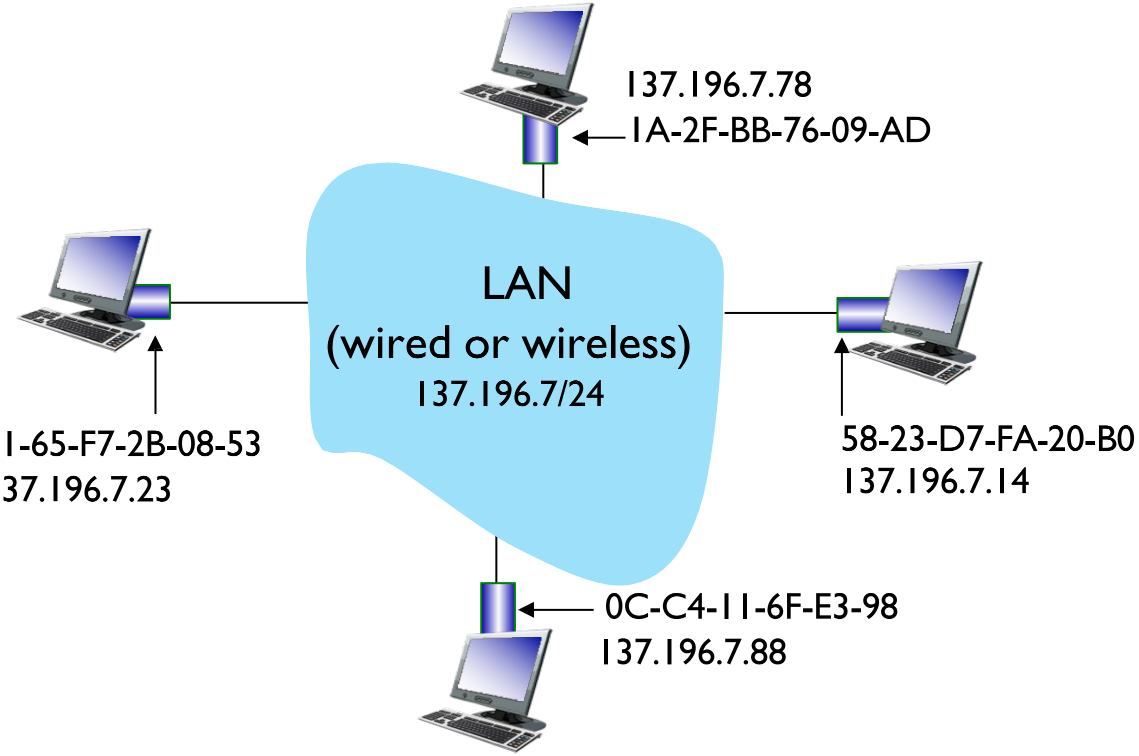
Why MAC address in addition to IP addr?

- MAC address allocation administered by IEEE
- manufacturer buys portion of MAC address space (to assure uniqueness)
- MAC's flat address allows portability
  - can move interface from one LAN to another
- IP address are NOT portable
  - depends on IP subnet to which node is attached
- Analogy:
  - MAC address: like Social Security Number
  - IP address: like postal address

MAC addresses

each interface on LAN

- has unique 48-bit MAC address
- has a locally unique 32-bit IP address (as we've seen)

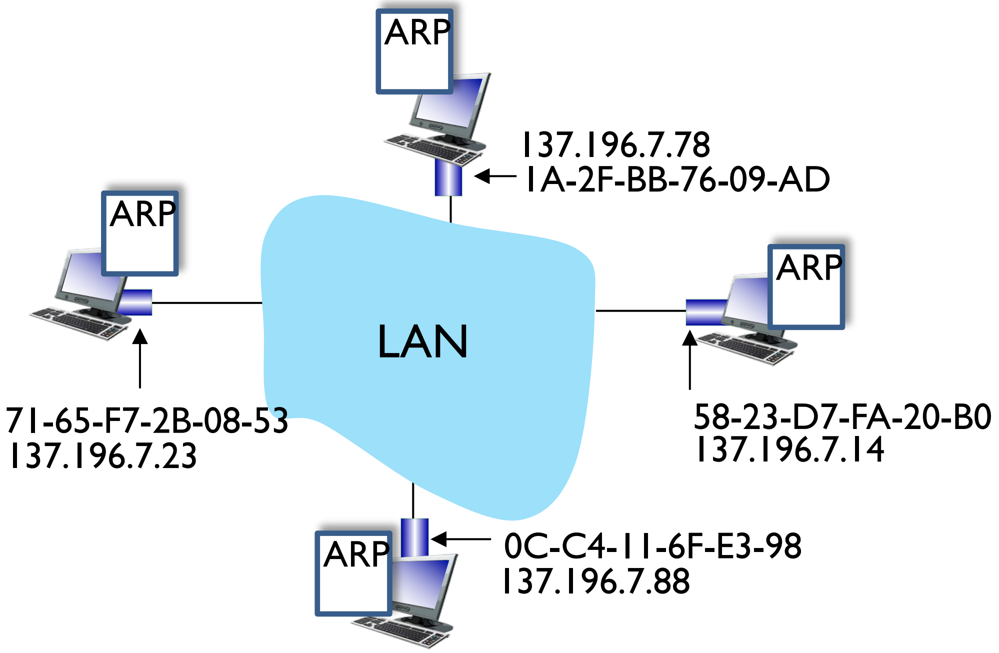


How to learn MAC address given IP address?

Outline

1. Link Layer Intro
2. Addressing: MAC Recap
3. Address Resolution Protocol

ARP: Address Resolution Protocol



ARP table: each IP node (host, router) on LAN has 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 in action

example: A wants to send datagram to B

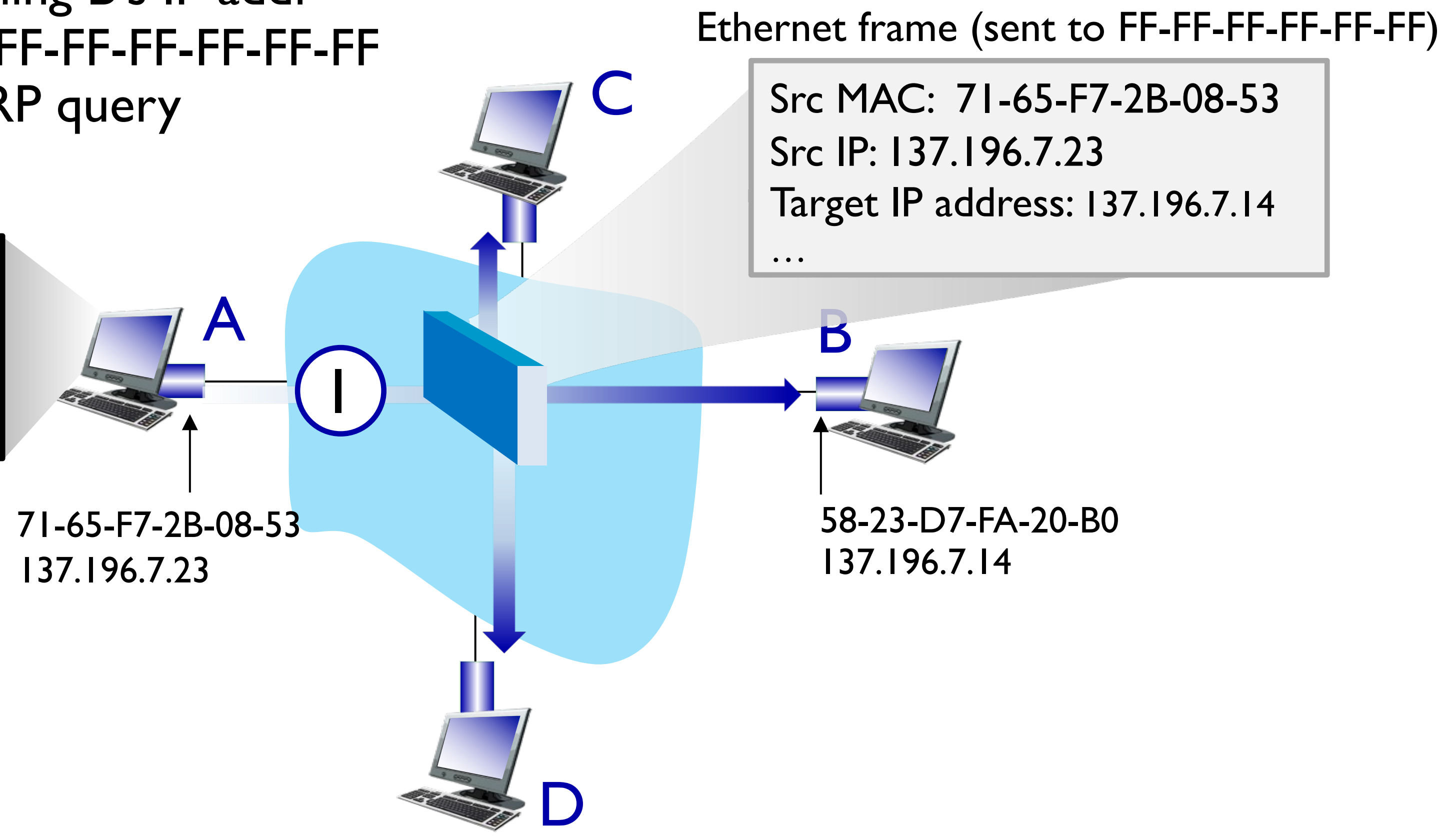
- B’s MAC address not in A’s ARP table, so A uses ARP to find B’s MAC address

1 A broadcasts ARP query, containing B's IP addr

- destination MAC address = FF-FF-FF-FF-FF-FF
- all nodes on LAN receive ARP query

ARP table in A

IP addr	MAC addr	TTL



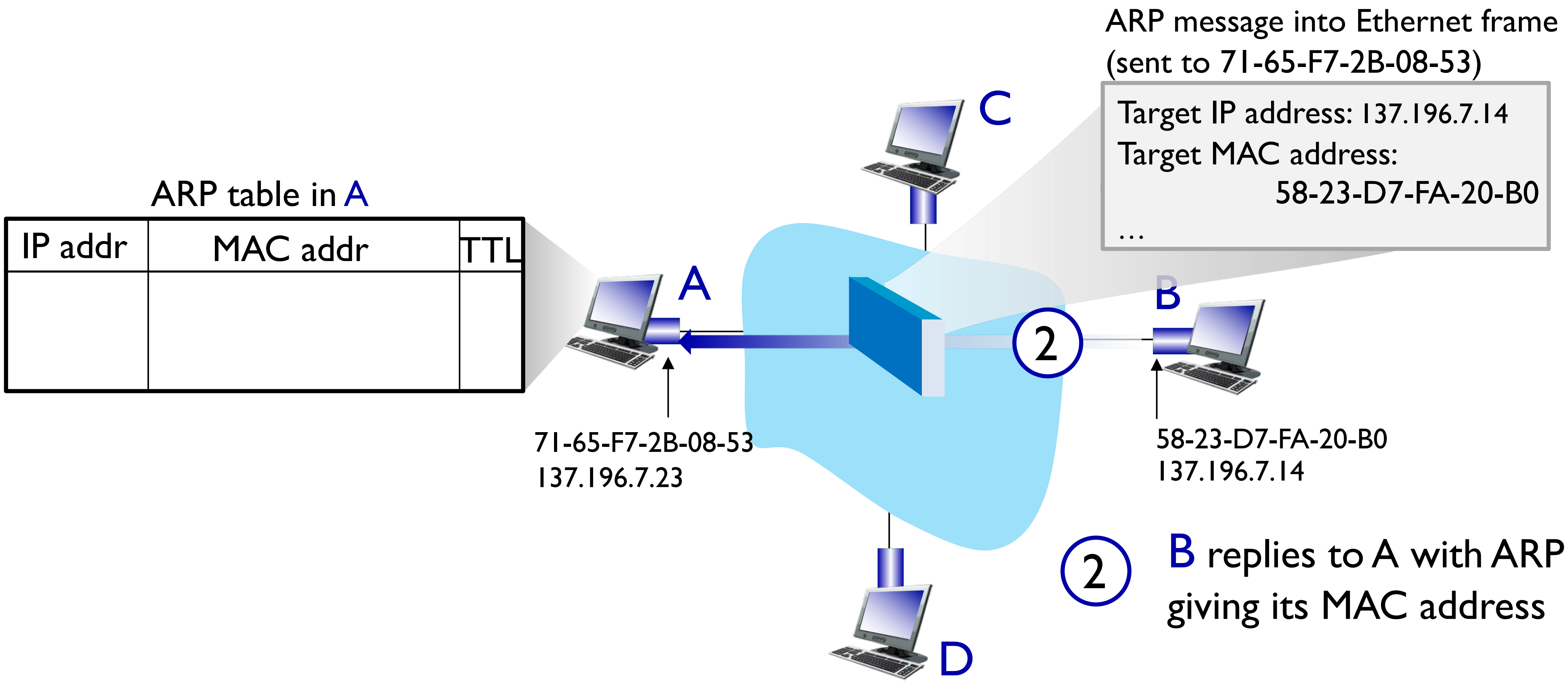
Ethernet frame (sent to FF-FF-FF-FF-FF-FF)

Src MAC: 71-65-F7-2B-08-53
Src IP: 137.196.7.23
Target IP address: 137.196.7.14
…

ARP protocol in action

example: A wants to send datagram to B

- B's MAC address not in A's ARP table, so A uses ARP to find B's MAC address



ARP table in A

IP addr	MAC addr	TTL

ARP message into Ethernet frame
(sent to 71-65-F7-2B-08-53)

Target IP address: 137.196.7.14
Target MAC address:
58-23-D7-FA-20-B0
…

2 B replies to A with ARP response, giving its MAC address

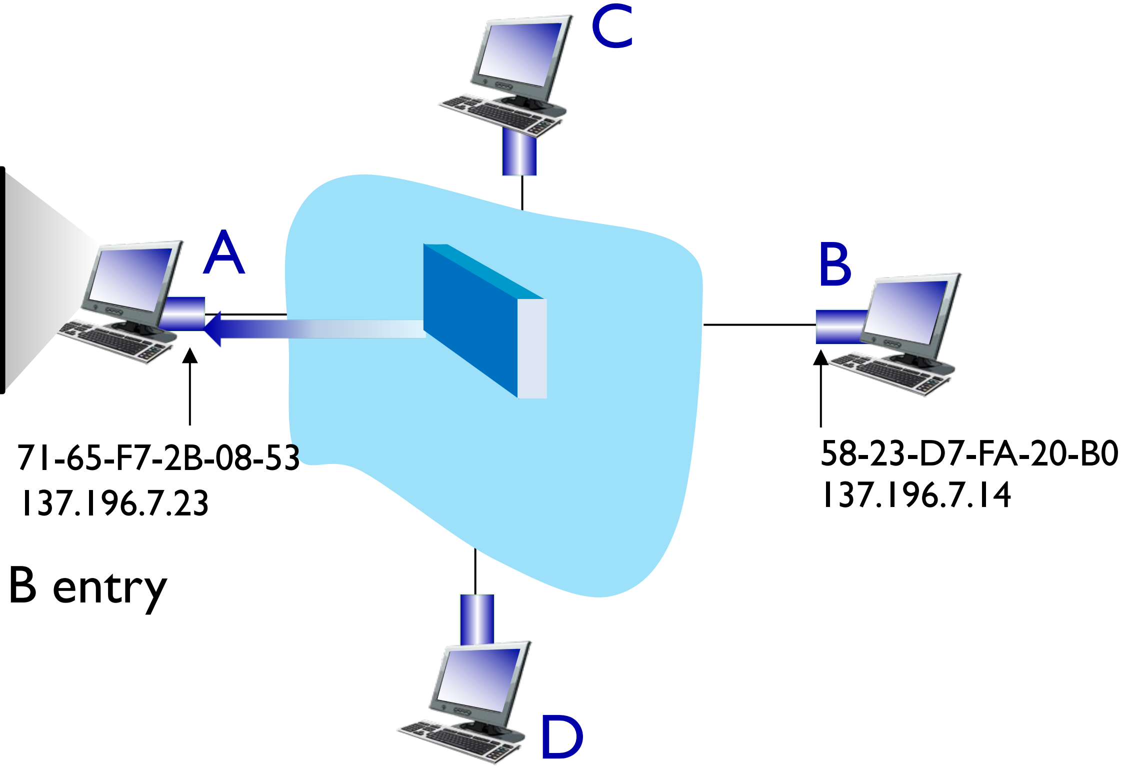
ARP protocol in action

example: A wants to send datagram to B

- B's MAC address not in A's ARP table, so A uses ARP to find B's MAC address

ARP table in A

IP addr	MAC addr	TTL
137.196.7.14	58-23-D7-FA-20-B0	500



3 A receives B's reply, adds B entry into its local ARP table

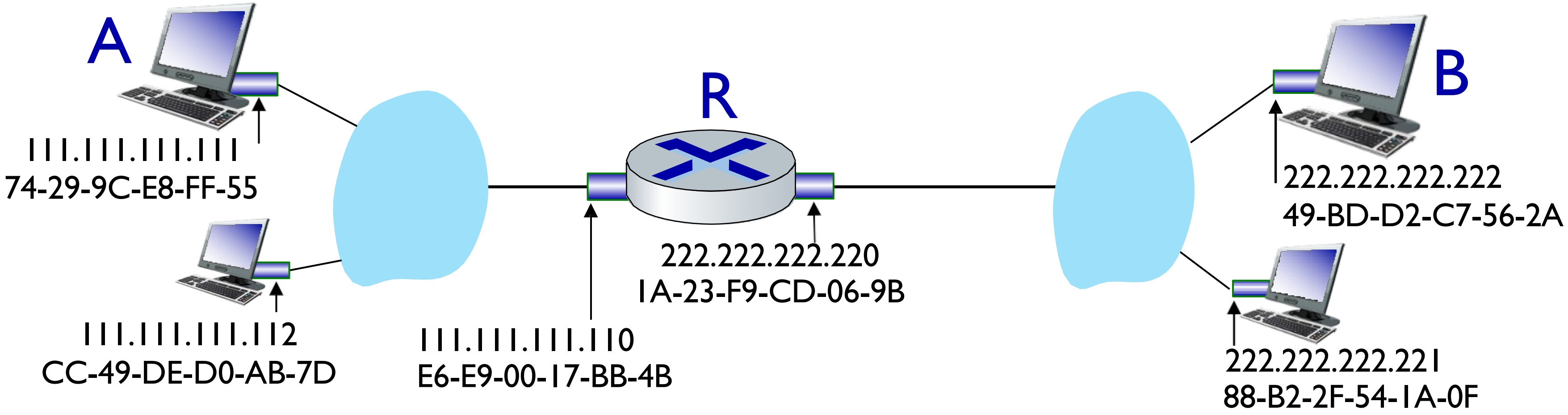
Outline

1. Link Layer Intro
2. Addressing: MAC
3. Address Resolution Protocol
4. Routing revisited with Addressing

Routing to another subnet revisited with addressing

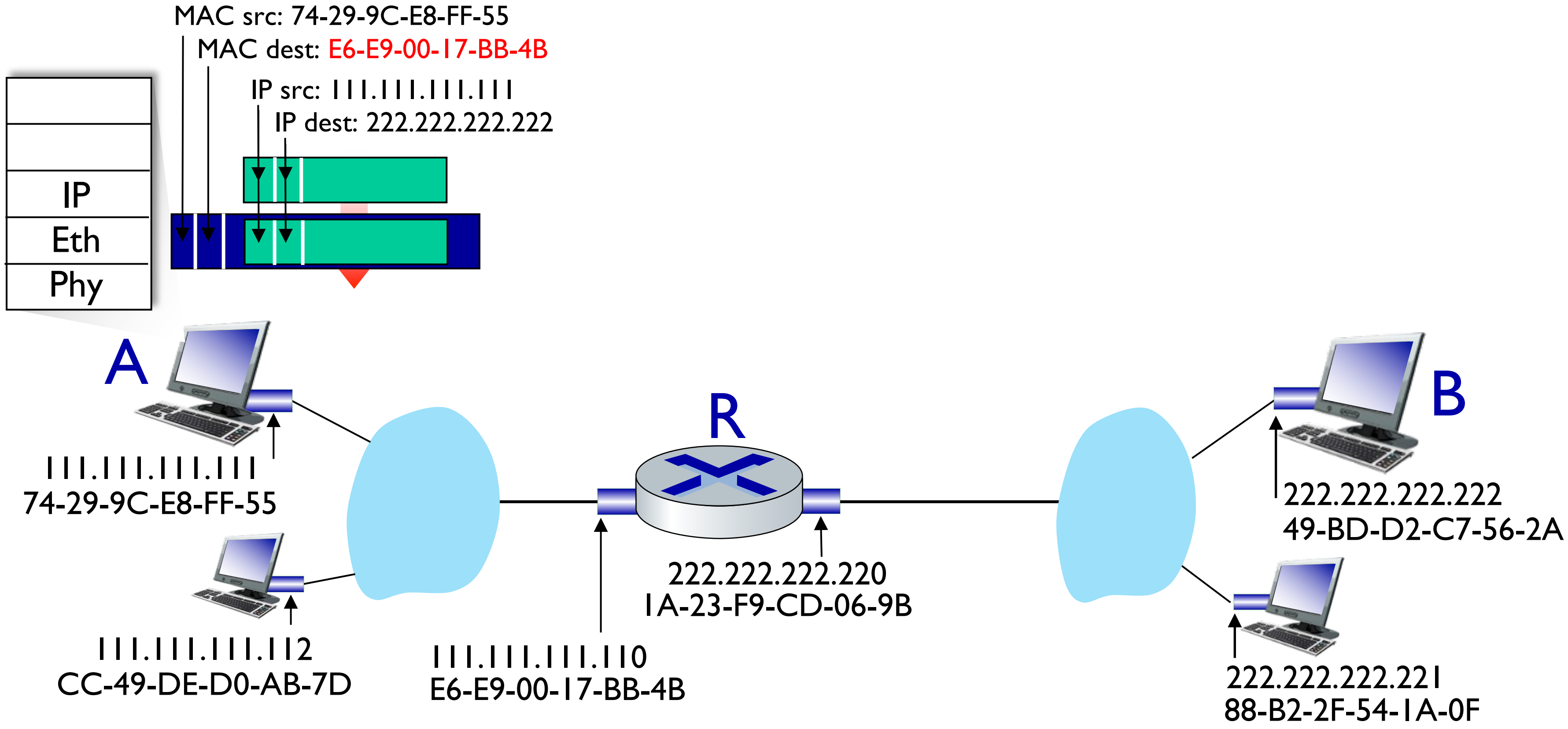
Sending a datagram from A to B via R

- assume that:
  - A knows B's IP address (how?)
  - A knows IP address of first hop router, R (how?)
  - A knows R's MAC address (how?)



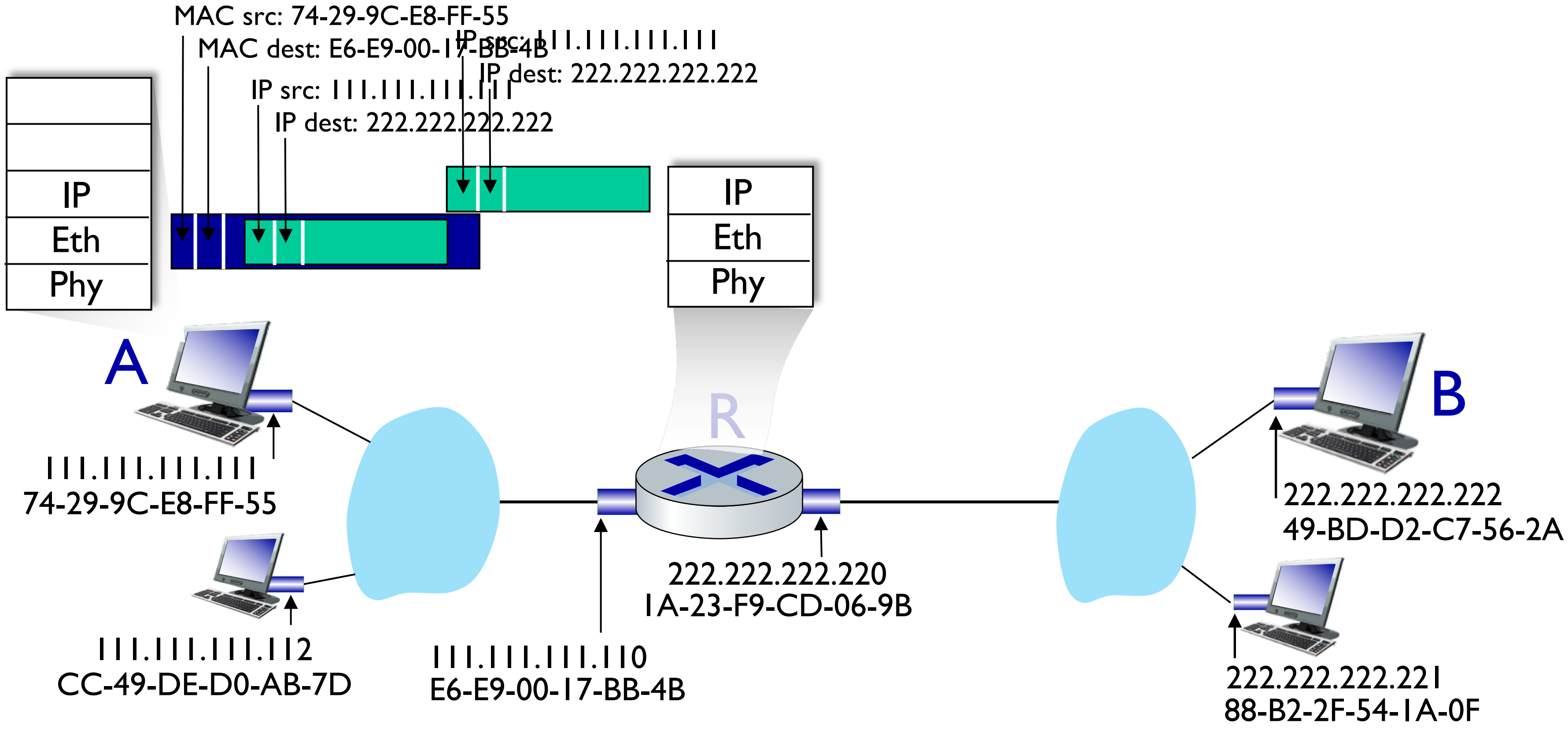
Routing to another subnet revisited with addressing

- A creates IP datagram with IP source A, destination B
- A creates link-layer frame containing A-to-B IP datagram
  - R's MAC address is frame’s destination



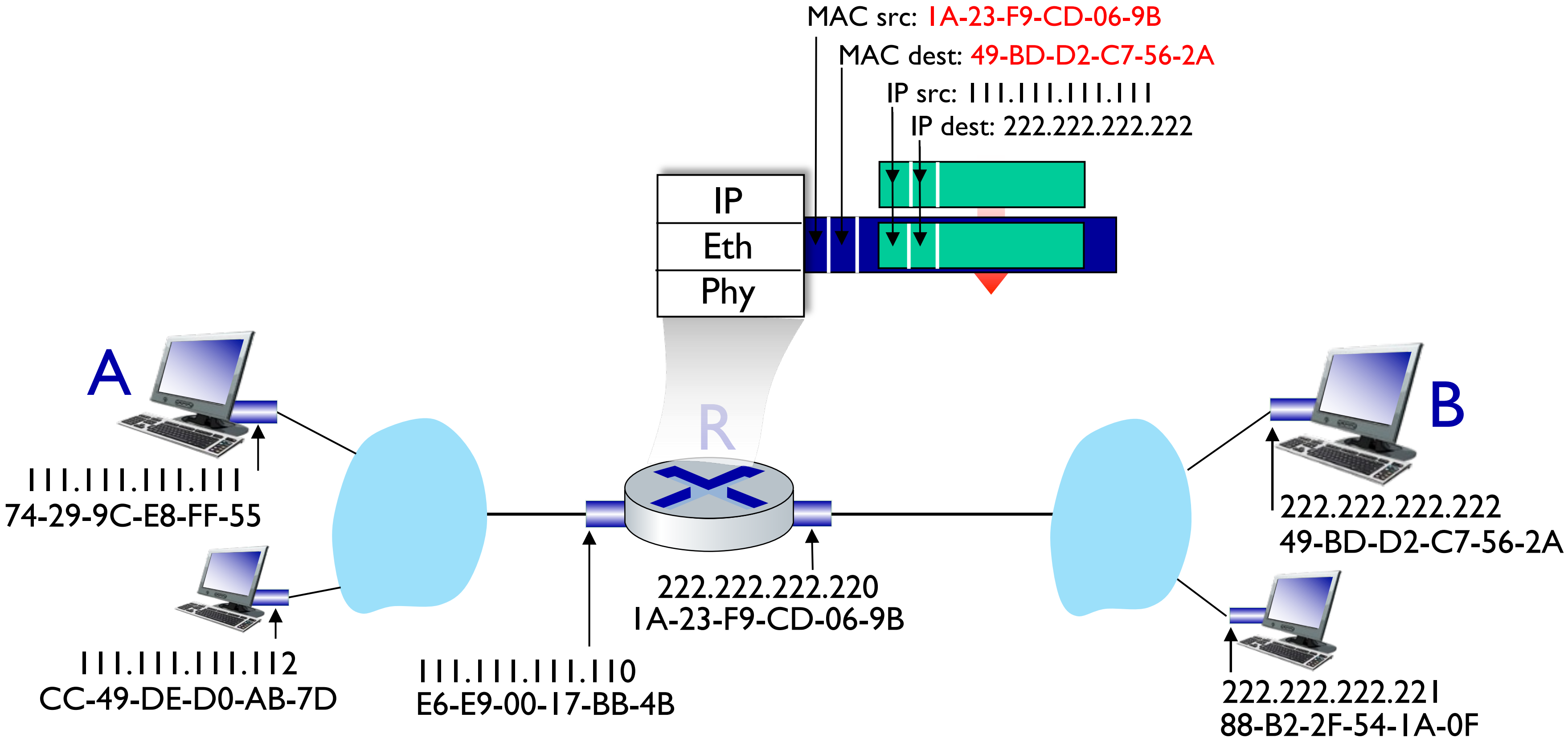
Routing to another subnet revisited with addressing

- frame sent from A to R
- frame received at R, datagram removed, passed up to IP



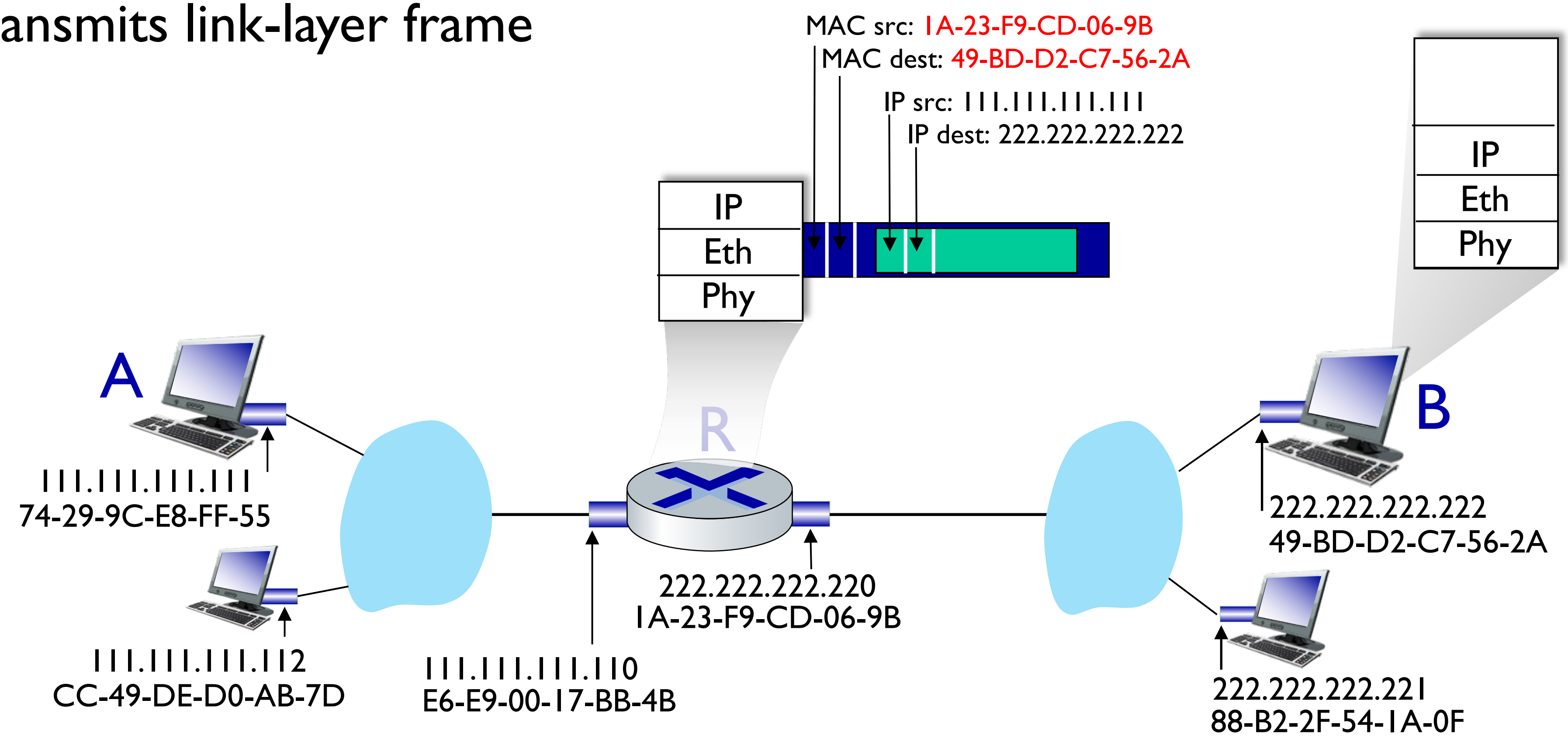
Routing to another subnet revisited with addressing

- R determines outgoing interface, passes datagram with IP source A, destination B to link layer
- R creates link-layer frame containing A-to-B IP datagram. What is MAC src/dest?



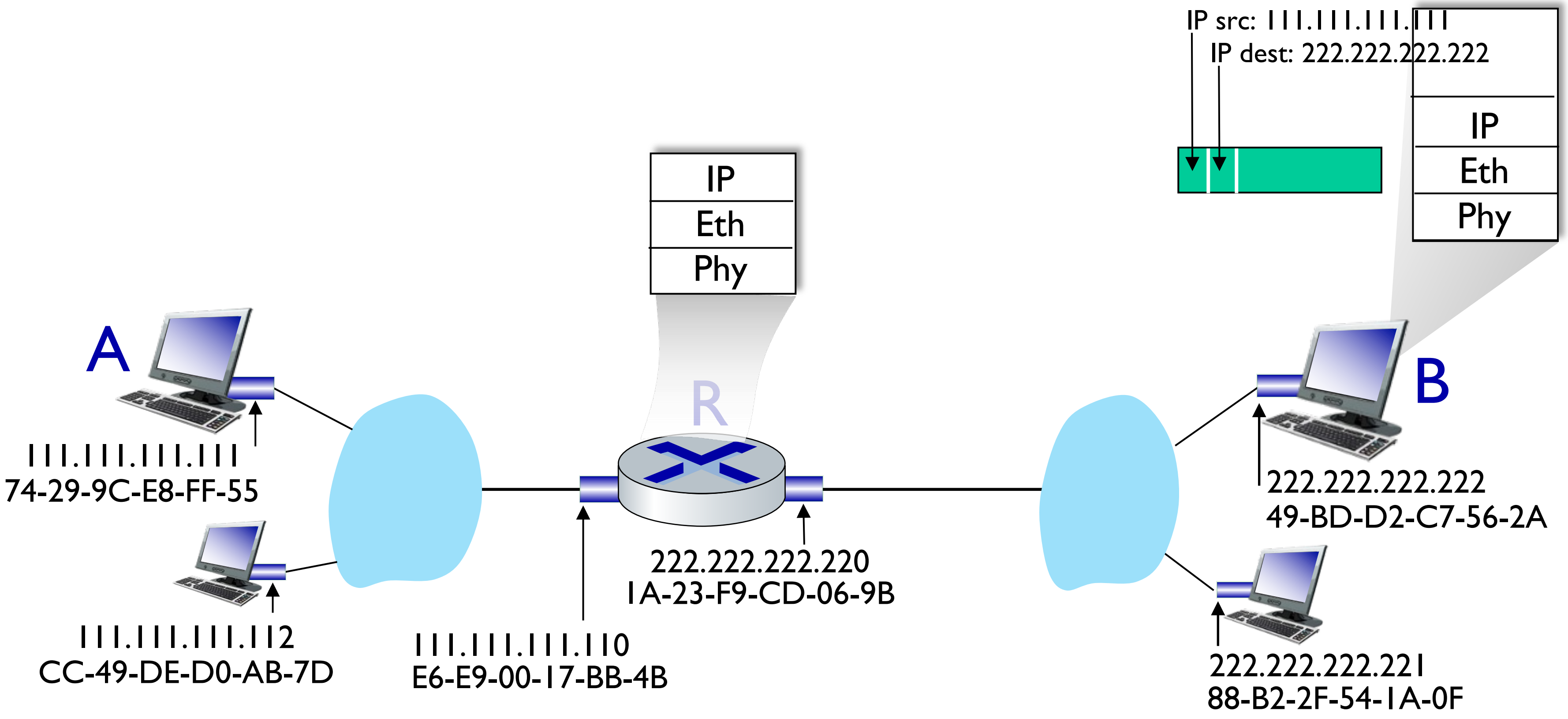
Routing to another subnet revisited with addressing

- R determines outgoing interface, passes datagram with IP source A, destination B to link layer
- R creates link-layer frame containing A-to-B IP datagram.
- transmits link-layer frame



Routing to another subnet revisited with addressing

- B receives frame, extracts IP datagram destination B



True/False?

- IP src and dst changes per hop
- MAC src and dst changes per hop

- Network layer header may change completely from one link to another
- Link Layer header may change completely from one link to another

In-class Exercise!

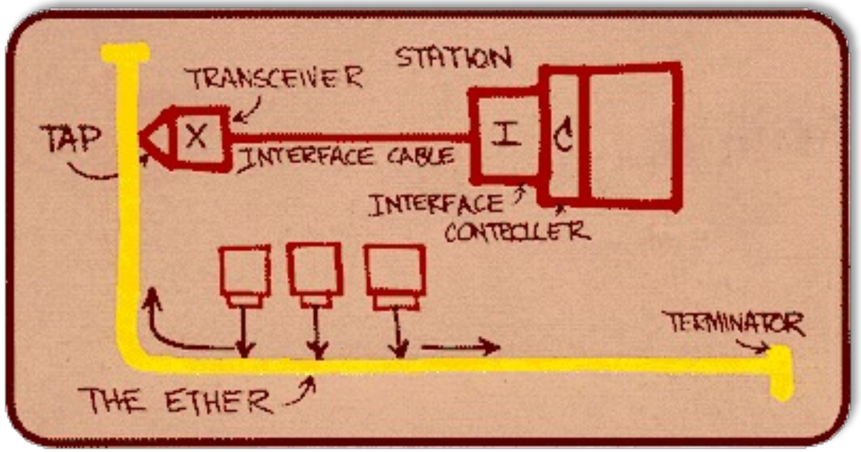
Outline

1. Link Layer Intro
2. Addressing: MAC
3. Address Resolution Protocol
4. Routing revisited with Addressing
5. Ethernet

Ethernet

"dominant" wired LAN technology:

- first widely used LAN technology
- simpler, cheap
- kept up with speed race: 10 Mbps – 400 Gbps
- single chip, multiple speeds (e.g., Broadcom BCM5761)



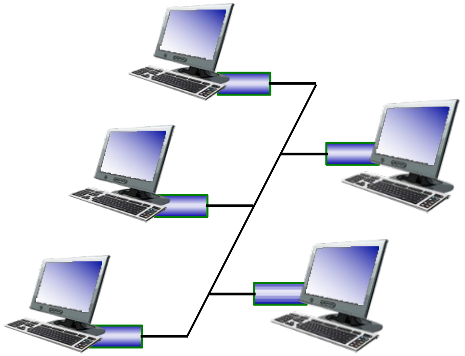
Metcalfe's Ethernet sketch

https://www.uspto.gov/learning-and-resources/journeys-innovation/audio-stories/defying-doubters

Ethernet: physical topology

- bus: popular through mid 90s
  - all nodes in same collision domain (can collide with each other)

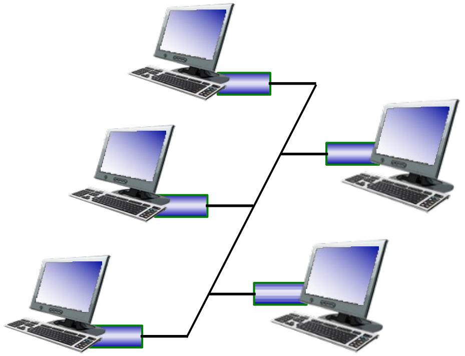
bus: coaxial cable

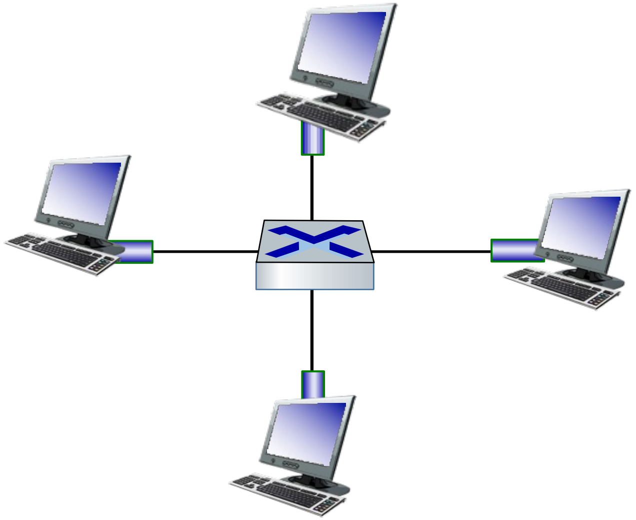


Ethernet: physical topology

- 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

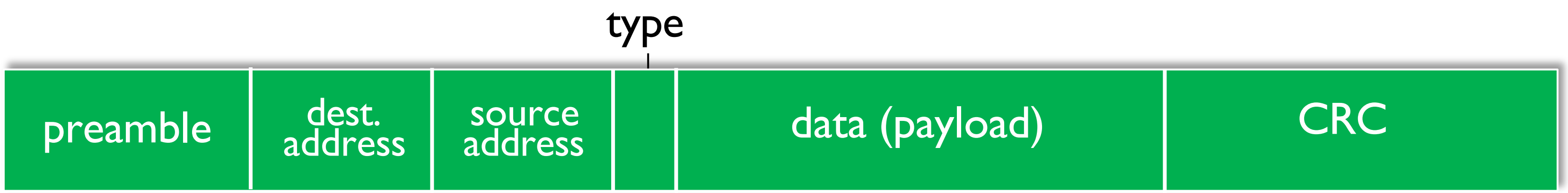
Why switch is better than bus?

- Switch is full-duplex while bus is not
- Switch is point-to-point while bus is shared medium (broadcast)
- Switch is more secure as it only sends to the necessary recipients

Overall switch provides better performance, efficiency and scalability

Ethernet frame structure

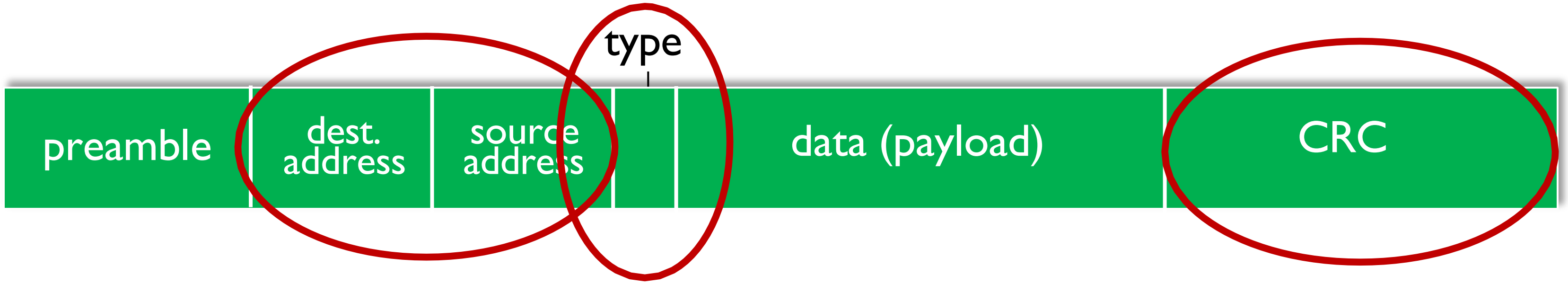
sending interface encapsulates IP datagram (or other network layer protocol packet) in Ethernet frame



preamble:

- used to synchronize receiver, sender clock rates
- 7 bytes of 10101010 followed by one byte of 10101011

Ethernet frame structure (more)



- **addresses:** 6 byte source, destination MAC addresses
  - if adapter receives frame with matching destination address, or with broadcast address (e.g.,ARP packet), it passes data in frame to network layer protocol
  - otherwise, adapter discards frame
- **type:** indicates higher layer protocol
  - mostly IP
  - used to demultiplex up at receiver
- **CRC:** cyclic redundancy check at receiver
  - error detected: frame is dropped

Ethernet: unreliable, connectionless

- connectionless: no handshaking between sending and receiving NICs
- unreliable: receiving NIC doesn't send ACKs to sending NIC
  - data in dropped frames recovered only if initial sender uses higher layer rdt (e.g.,TCP), otherwise dropped data lost
- Ethernet's MAC protocol: unslotted CSMA/CD with binary backoff

Outline

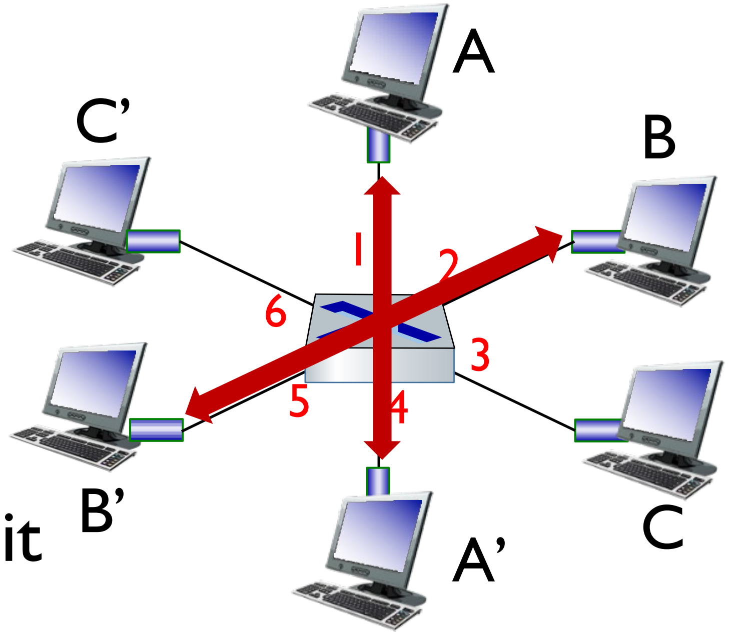
1. Link Layer Intro
2. Addressing: MAC
3. Address Resolution Protocol
4. Routing revisited with Addressing
5. Ethernet
6. Switches

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 to access segment
- transparent: hosts unaware of presence of switches
- plug-and-play, self-learning
  - switches do not need to be configured

Switch: multiple simultaneous transmissions

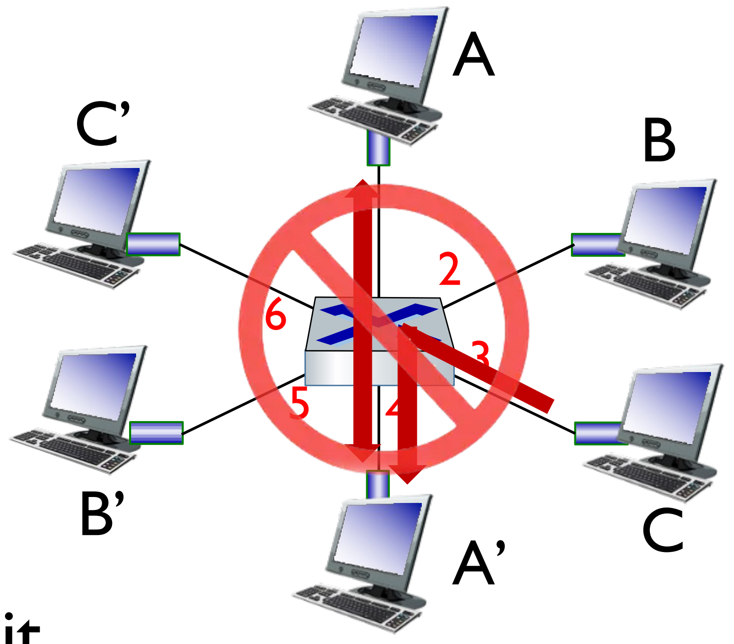
- hosts have dedicated, direct connection to switch
- Ethernet protocol used on each incoming link
  - no collisions; full duplex
  - each link is its own collision domain
- switching: A-to-A' and B-to-B' can transmit simultaneously, without collisions



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 (overlapping spokes cannot!)



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

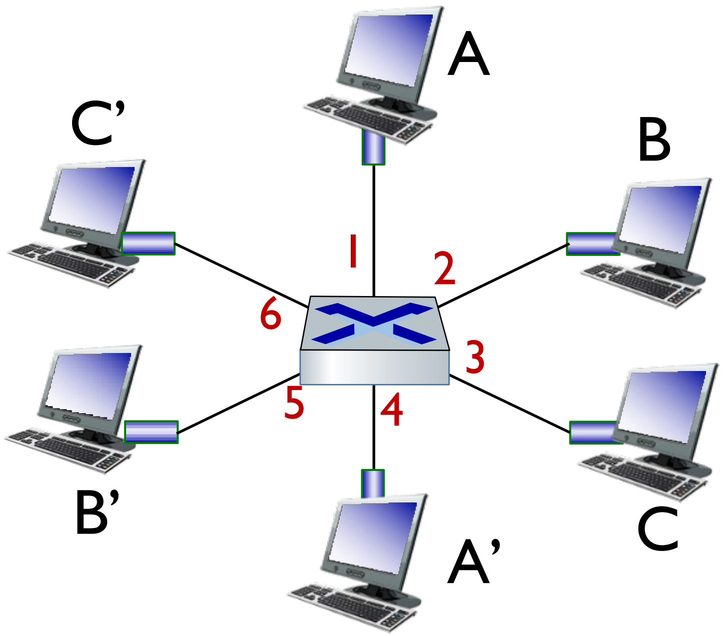
Switch's forwarding table

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

A: each switch has a switch table

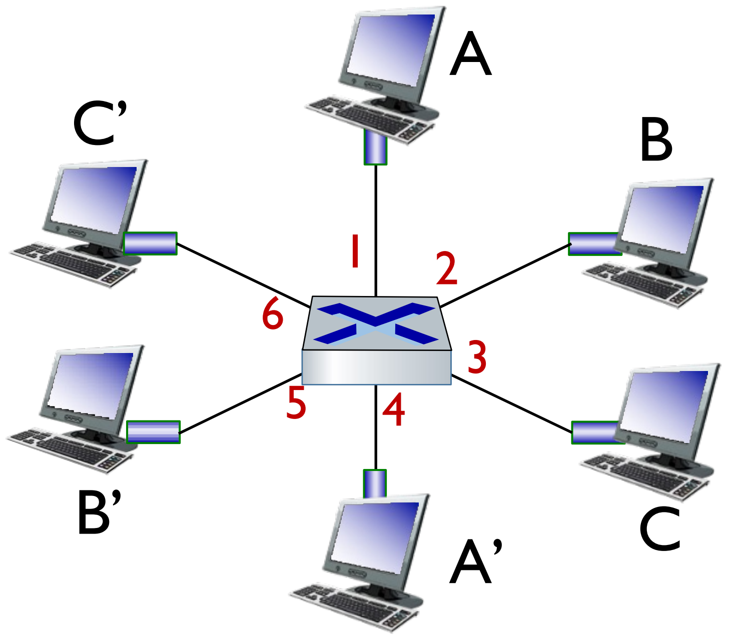
where each entry:

- (MAC address of host, interface to reach host, time stamp)
- looks like a routing table!



How does a switch know which interface to forward to?

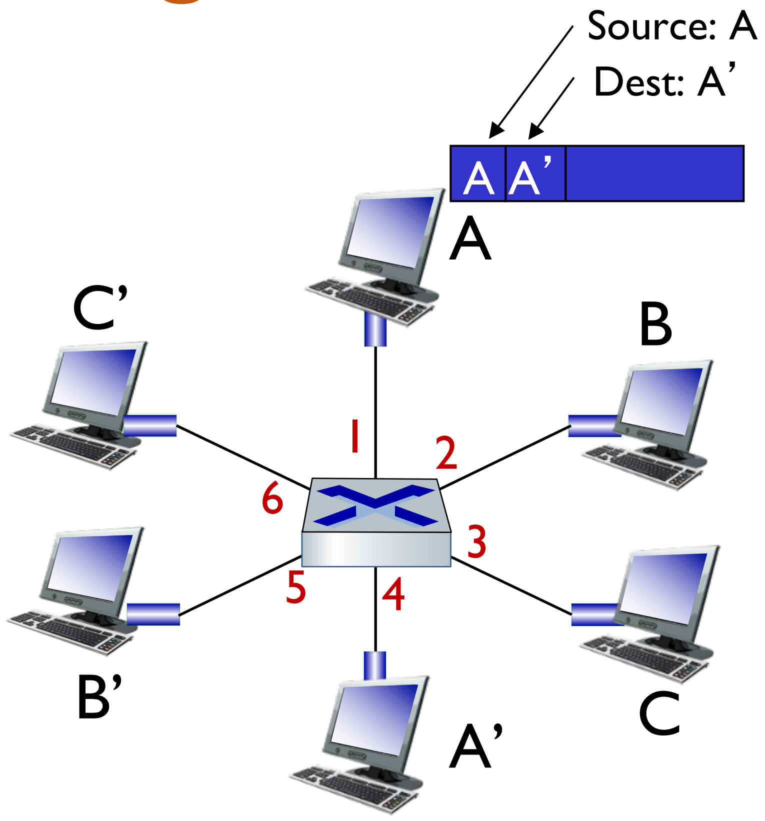
Q: Who fills out the switch table?



Self! Thus, it’s called self-learning switches

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

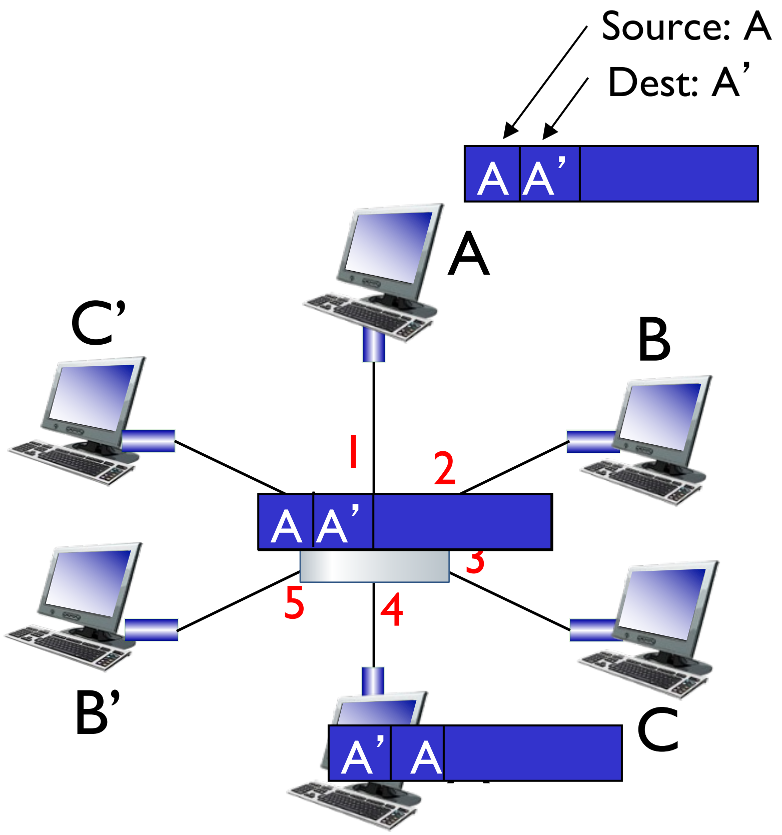


MAC addr	interface	TTL
A	1	60

Switch table (initially empty)

Self-learning switch example

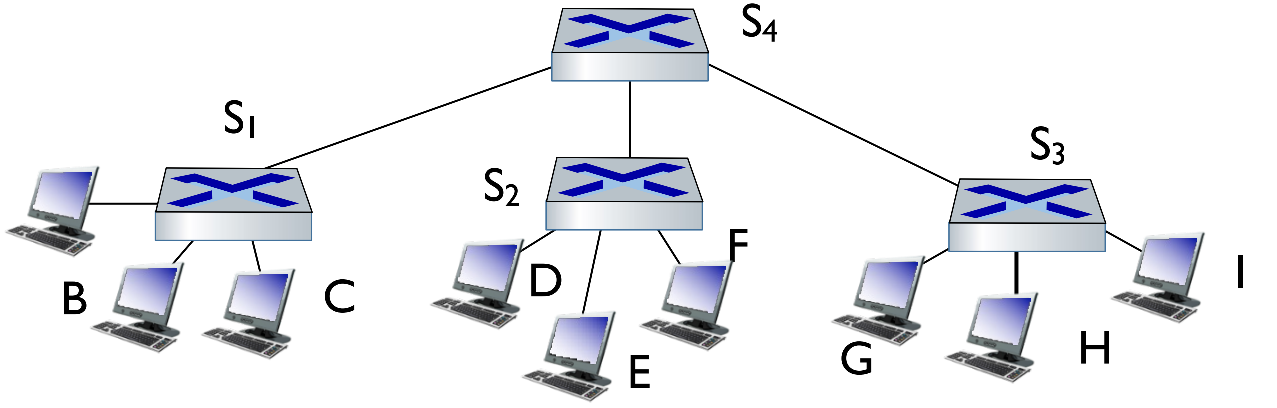
- frame destination, A', location unknown: flood
- destination A location known: selectively send on just one link



MAC addr	interface	TTL
A	1	60
A'	4	60

switch table
(initially empty)

Self-learning switches can be connected together

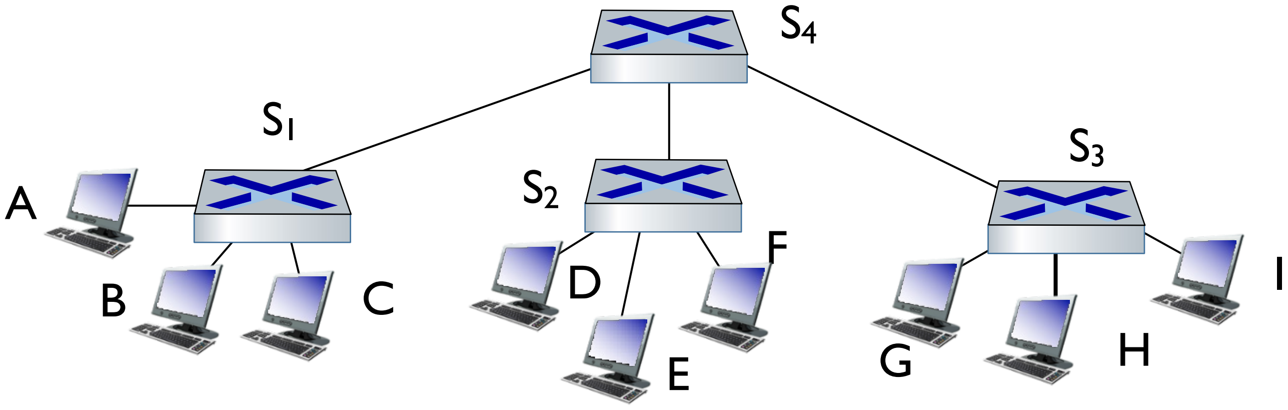


Q: Sending from A to G - how does S_1 know to forward frame destined to G via S_4 and S_3?

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

In-class Ex: Self-learning multi-switch example

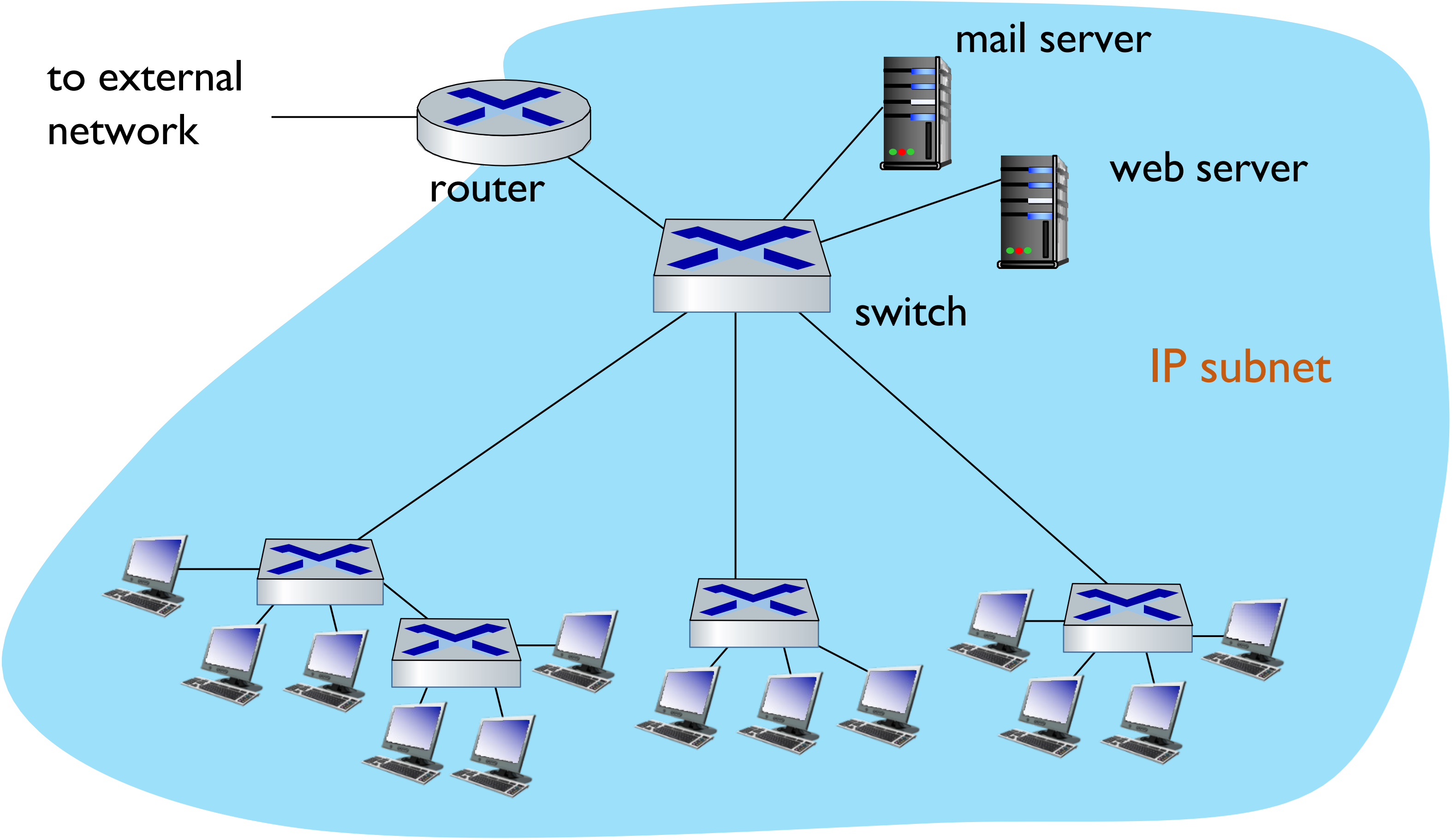
Suppose C sends frame to I, I responds to C



Show switch tables and packet forwarding in S_1, S_2, S_3, S_4

Switch vs. router

Small institutional network example



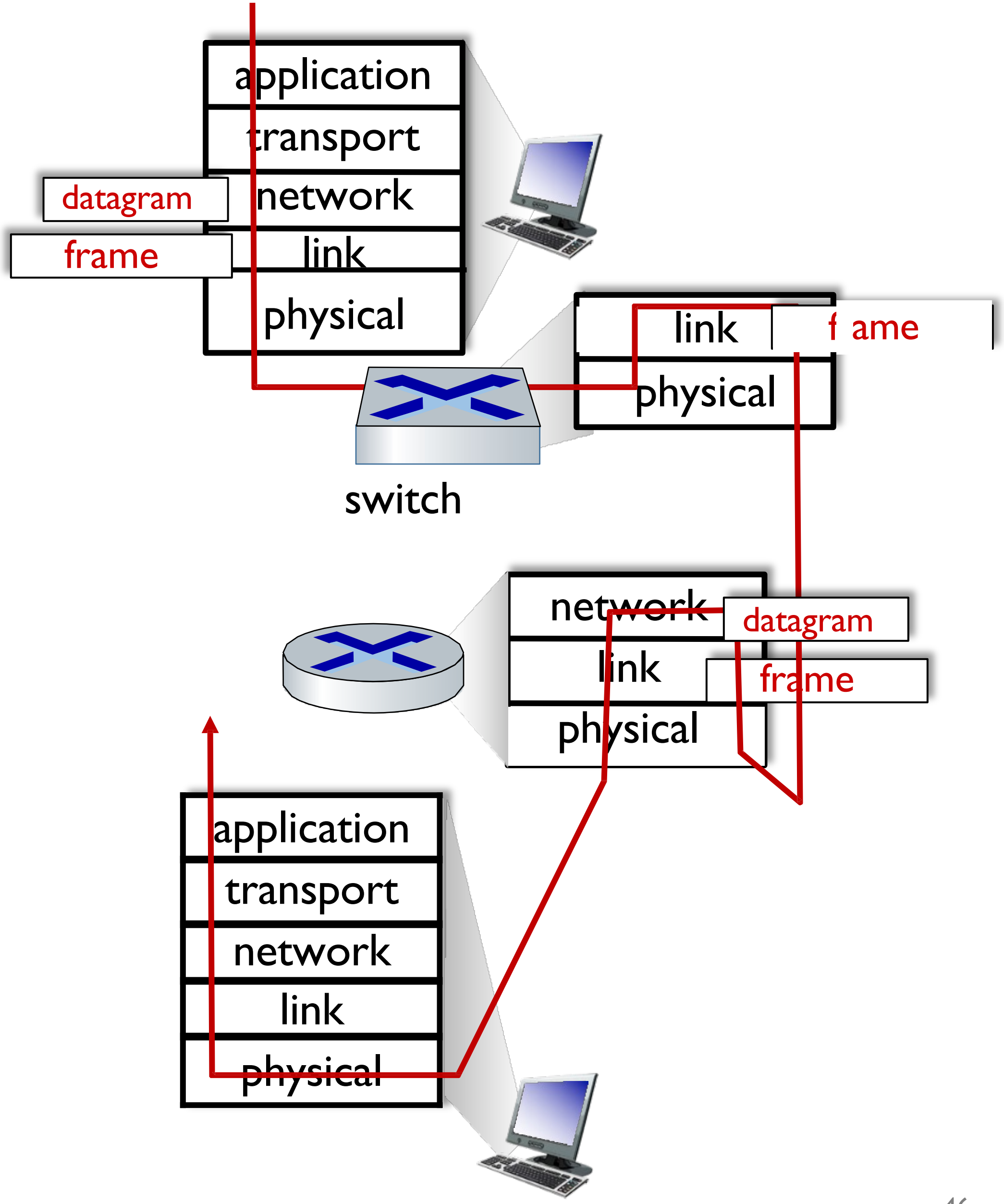
Switches vs. routers

both are store-and-forward:

- routers: network-layer devices examines network-layer headers
- switches: link-layer devices examines link-layer headers

both have forwarding tables:

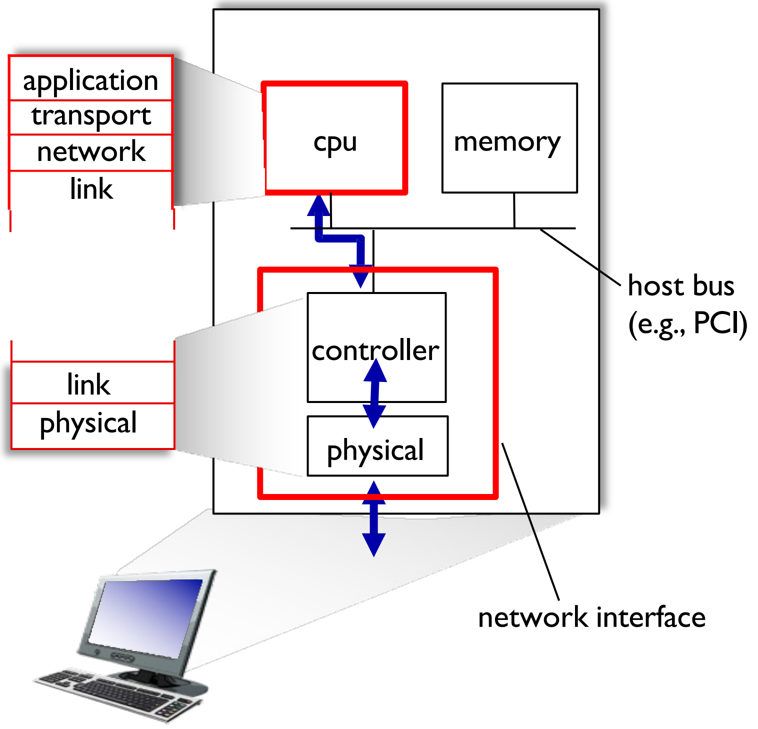
- routers: compute tables using routing algorithms (IP addresses)
- switches: learn forwarding table using flooding, self-learning (MAC addresses)



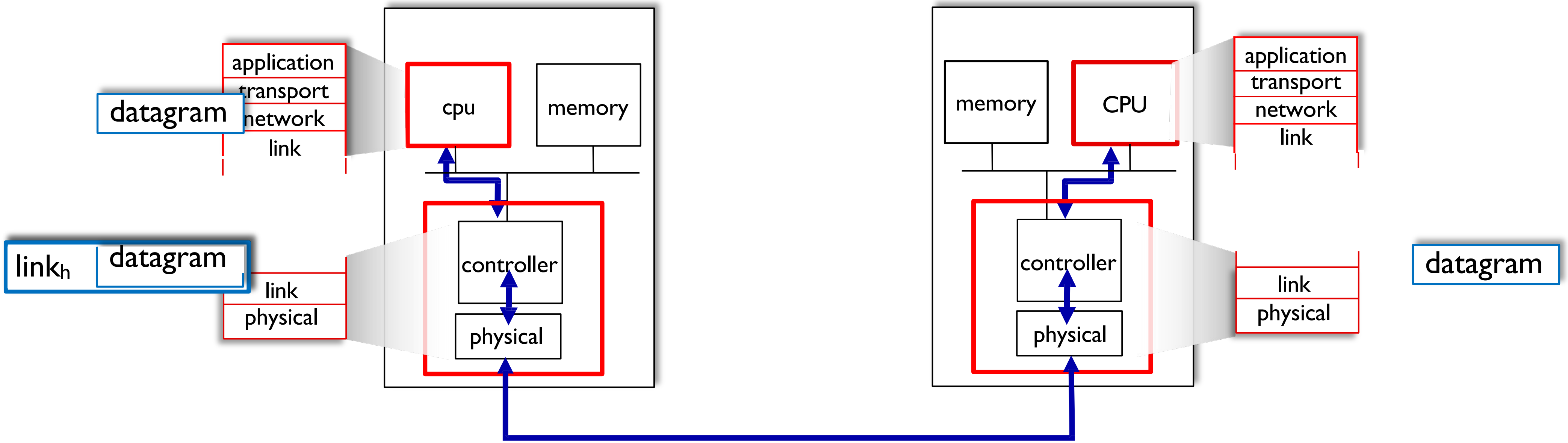
Backup Slides

Where is L2 implemented?

- in each-and-every host
- link layer implemented in network interface card (NIC) or on a chip
  - Ethernet, WiFi card or chip
  - implements link, physical layer
- attaches into host's system buses
- combination of hardware, software, firmware



Interfaces communicating



sending side:

- encapsulates datagram in frame
- adds error checking bits, reliable data transfer, flow control, etc.

receiving side:

- looks for errors, reliable data transfer, flow control, etc.
- extracts datagram, passes to upper layer at receiving side

Switch self-learning/flooding algo

```
when  frame received at switch:

  1. record incoming link, MAC address of sending host
  2. index switch table using MAC destination address
  3. if entry found for destination
       then {
        if destination on segment from which frame arrived
            then drop frame
        else forward frame on interface indicated by entry
        }
      else flood  /* forward on all interfaces except arriving interface */
```

Acknowledgements

Slides are adopted from Kurose’ Computer Networking Slides