UNIVERSITY OF WATERLOO

CS 456/656 Computer Networks

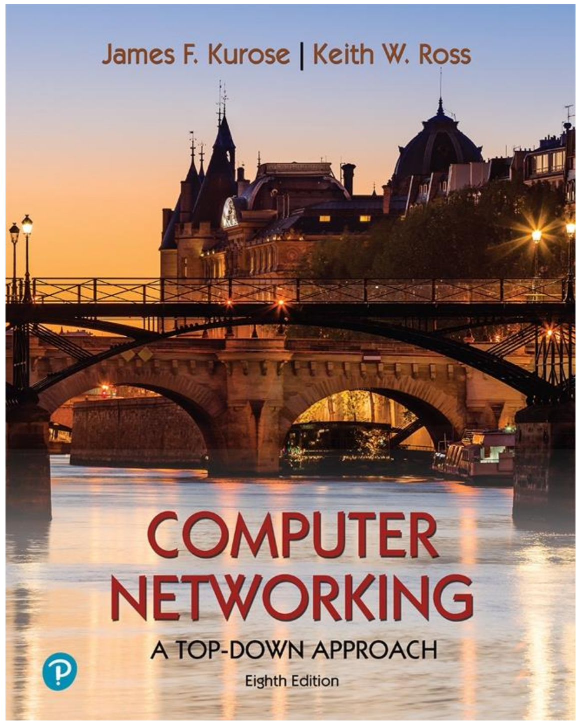
Lecture 14: Link Layer – Part 1

Mina Tahmasbi Arashloo and Uzma Maroof

Fall 2025

A note on the slides

Adapted from the slides that accompany this book.



Computer Networking: A Top-Down Approach
8th edition
Jim Kurose, Keith Ross
Pearson, 2020

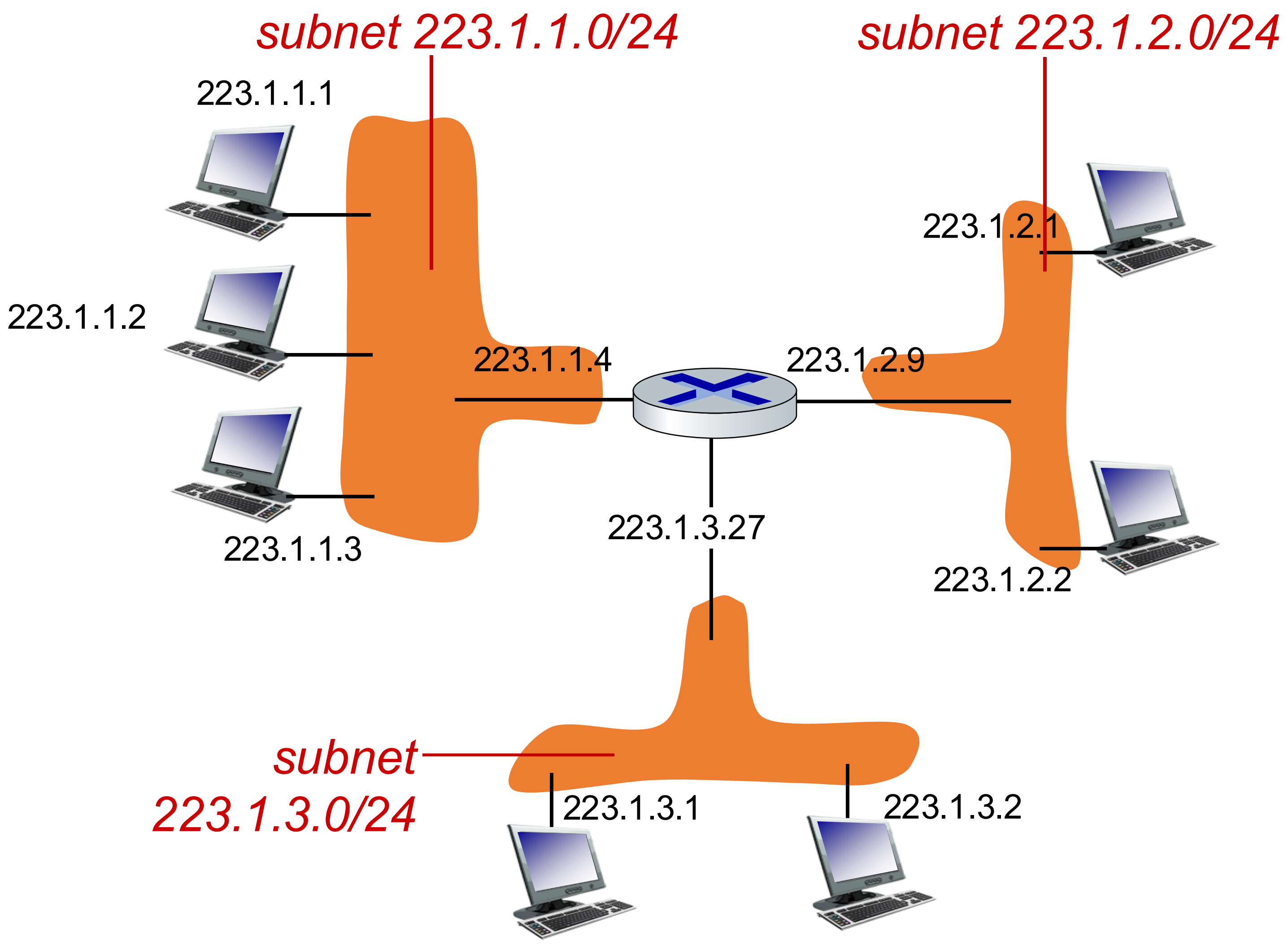
Link layer: roadmap

- Link layer overview
  - Local Area Networks (LANs)
- Switched LANs
- Virtual LANs (VLANs)
- Shared LANs and multiple access protocols

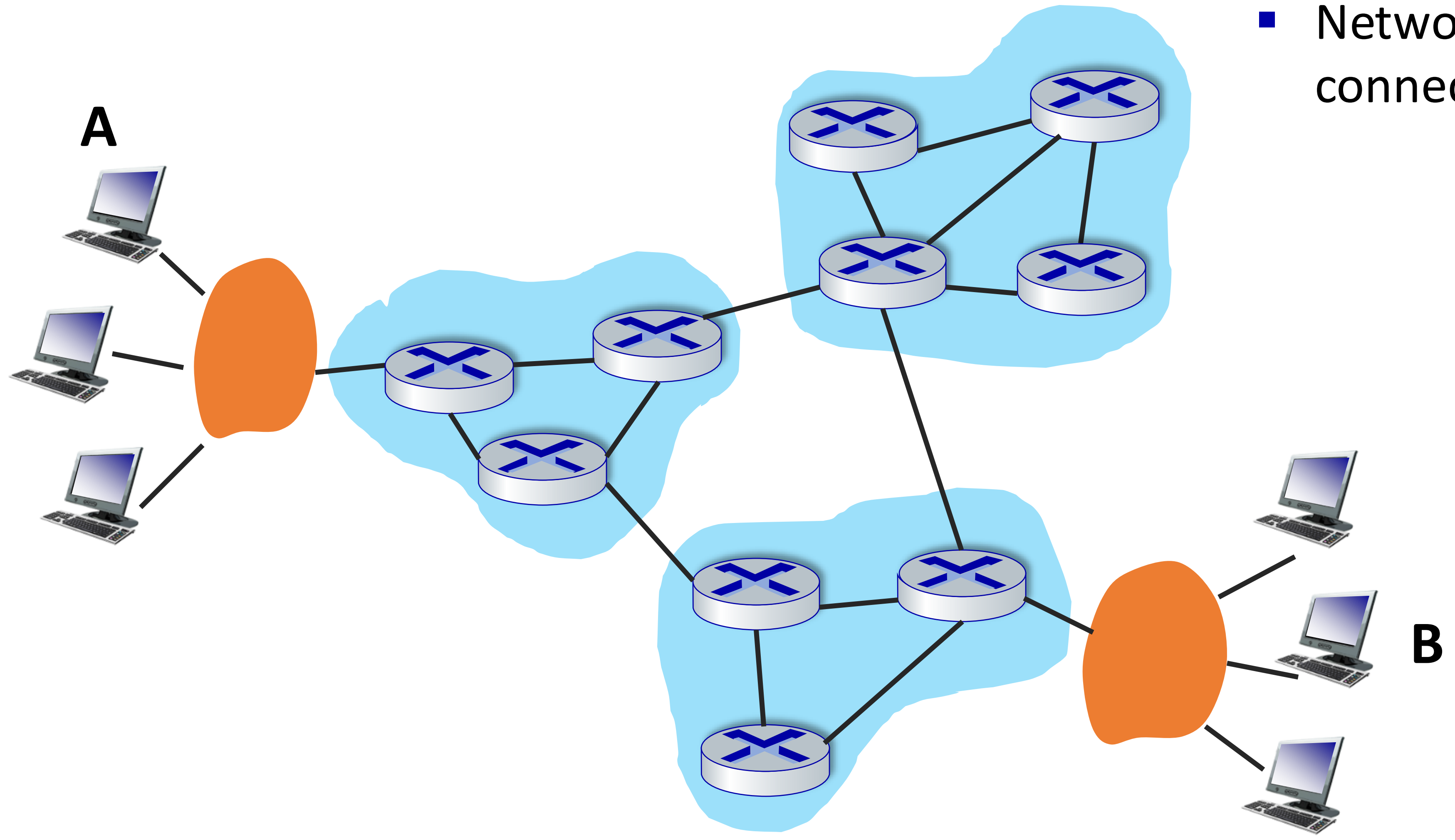
Link layer: roadmap

- Link layer overview
  - Local Area Networks (LANs)
- Switched LANs
- Virtual LANs (VLANs)
- Shared LANs and multiple access protocols

Recall the question: How are interfaces (without intervening router) connected?

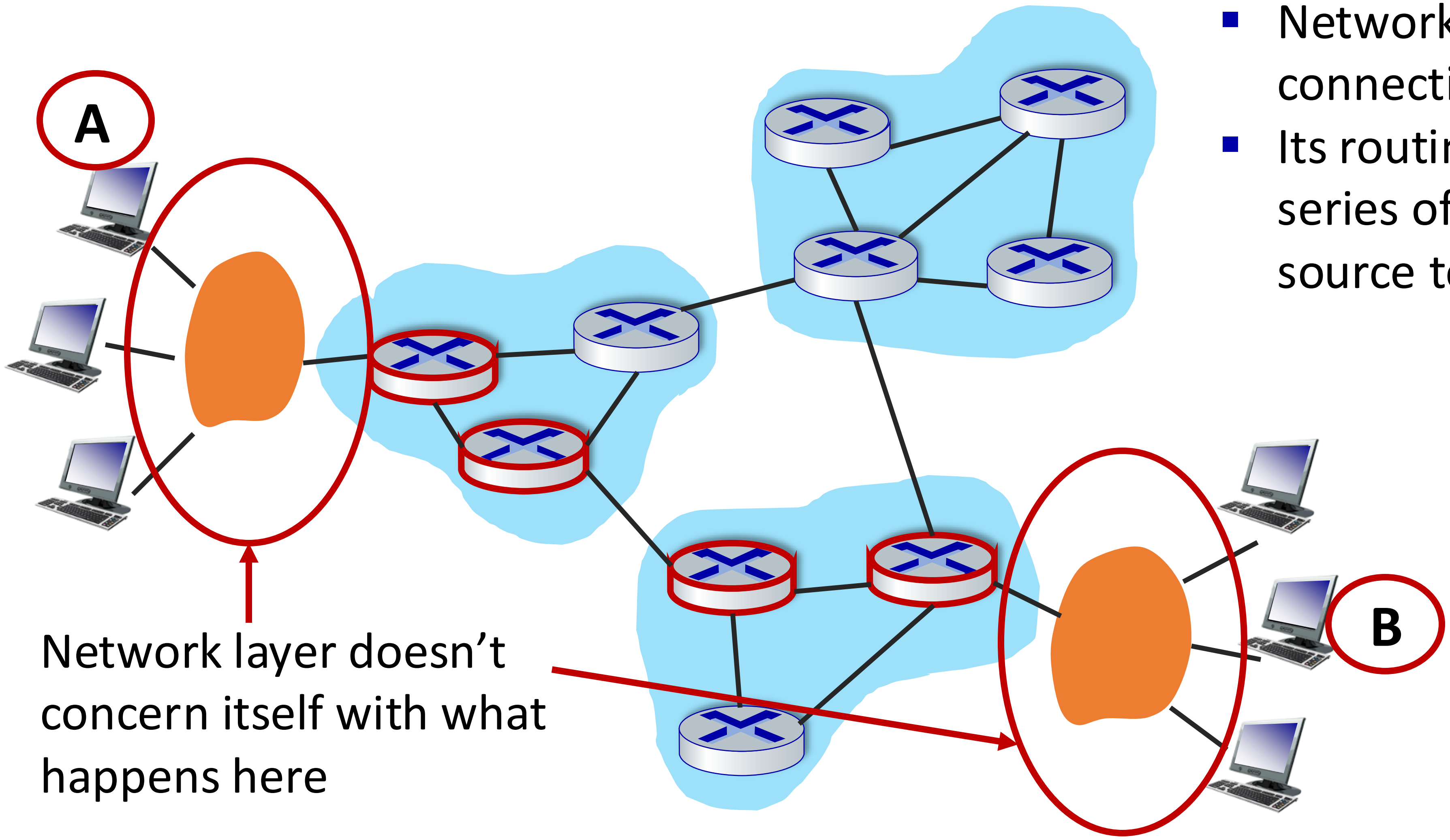


Network layer: global connectivity



- Network layer provides connectivity across networks

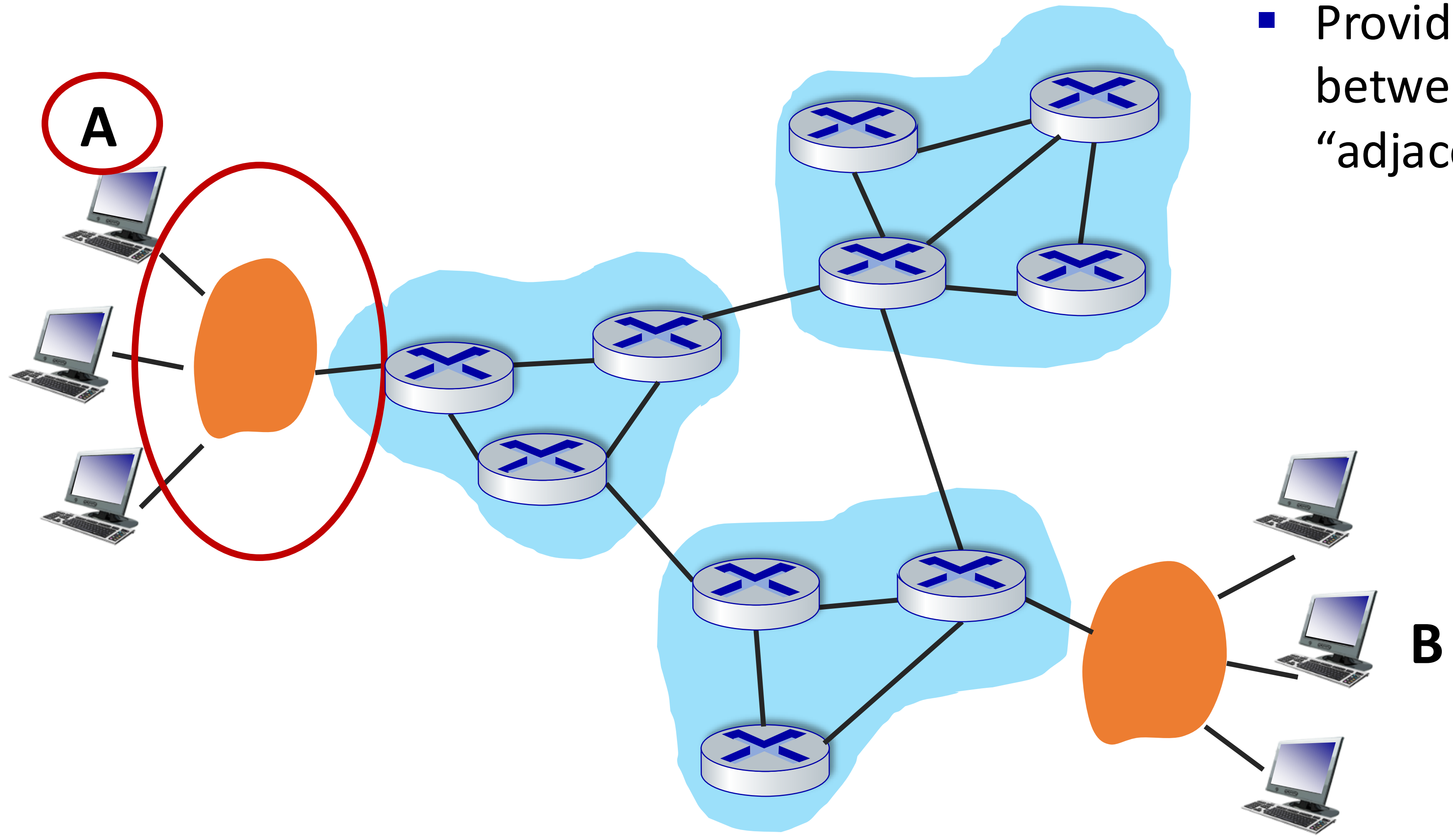
Network layer: global connectivity



- Network layer provides connectivity across networks
- Its routing protocols find a series of routers from a source to a destination

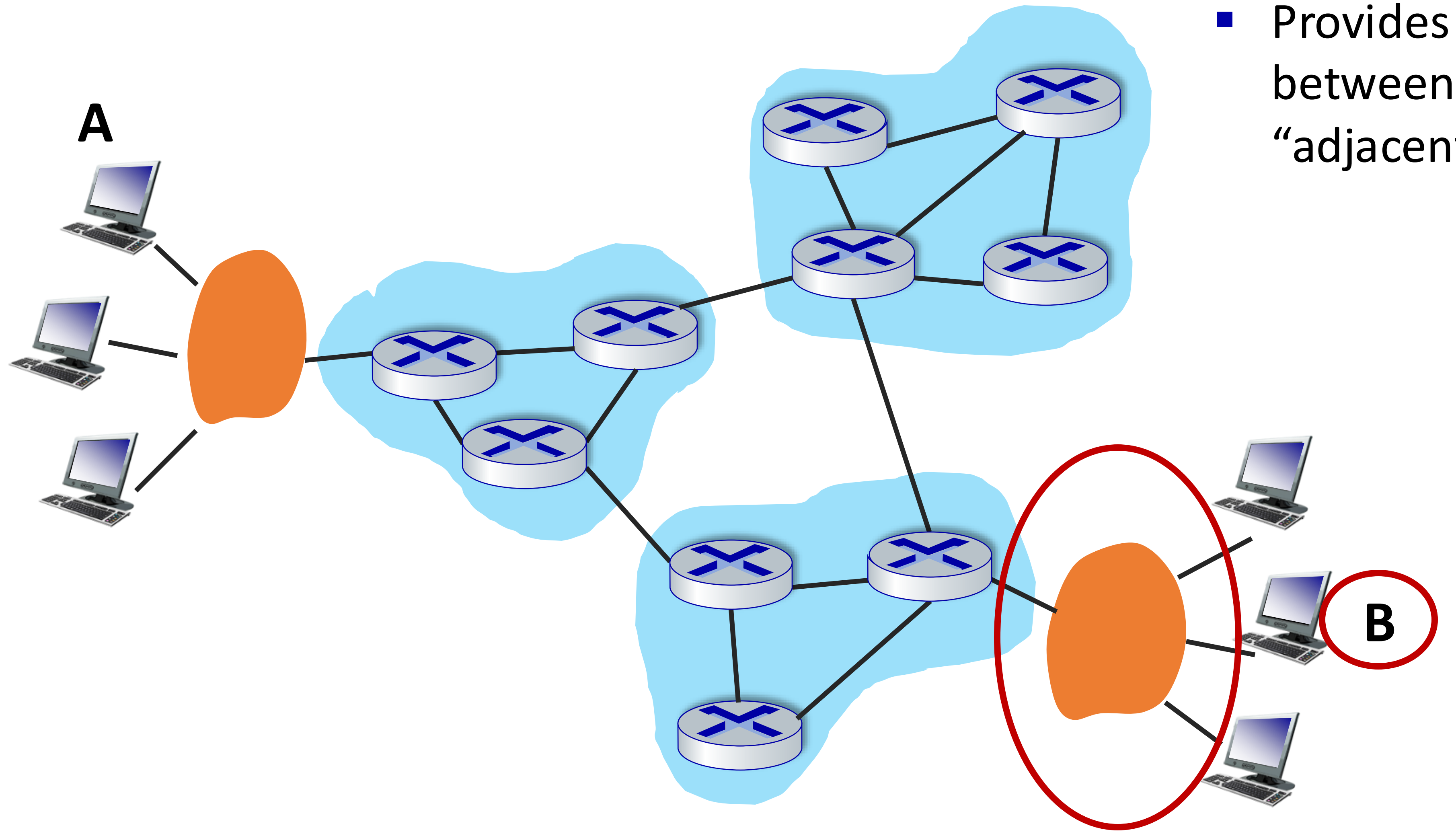
Network layer doesn't concern itself with what happens here

Link layer: local connectivity



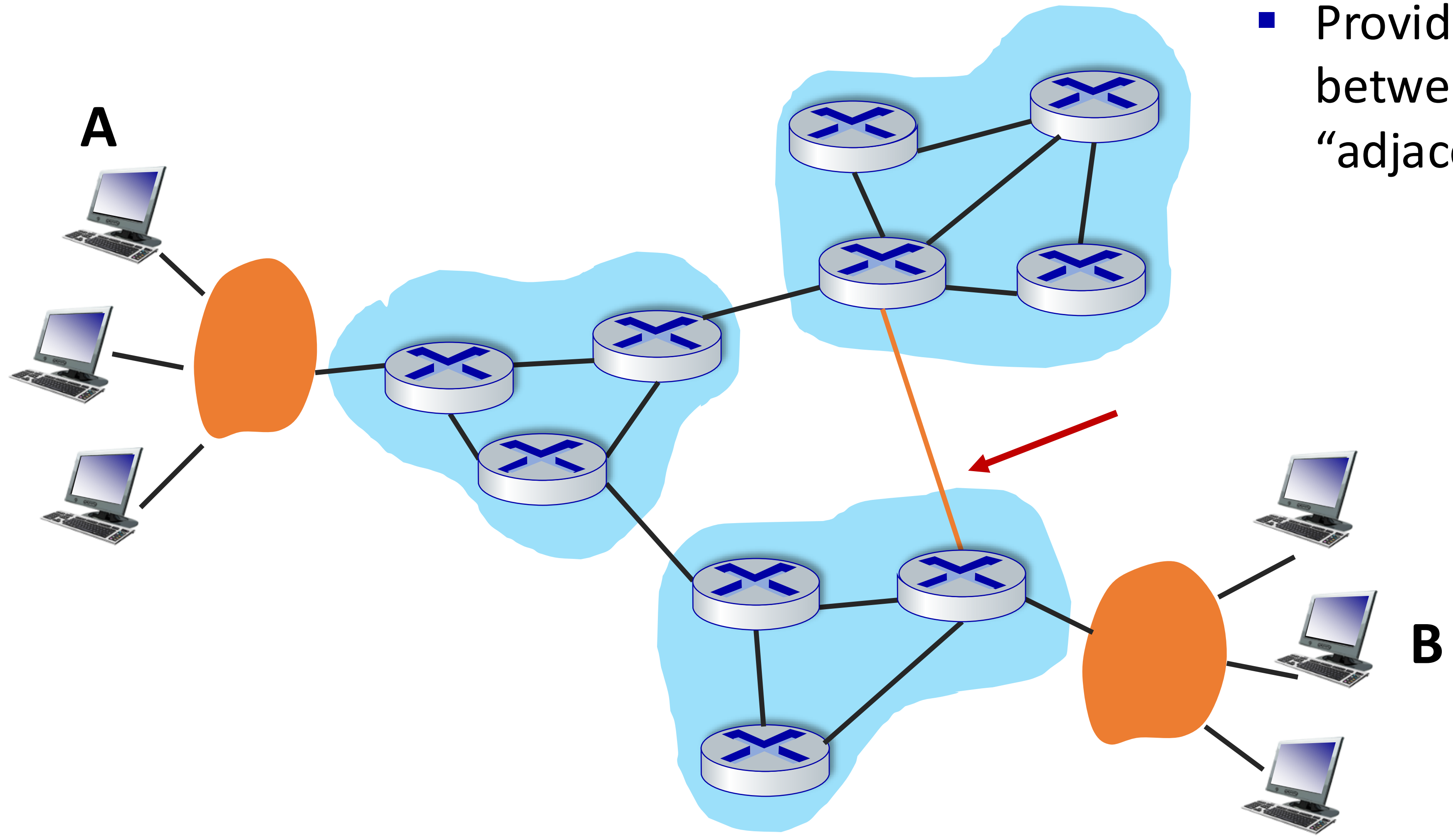
- Provides connectivity between (groups of) physically "adjacent" entities.

Link layer: local connectivity



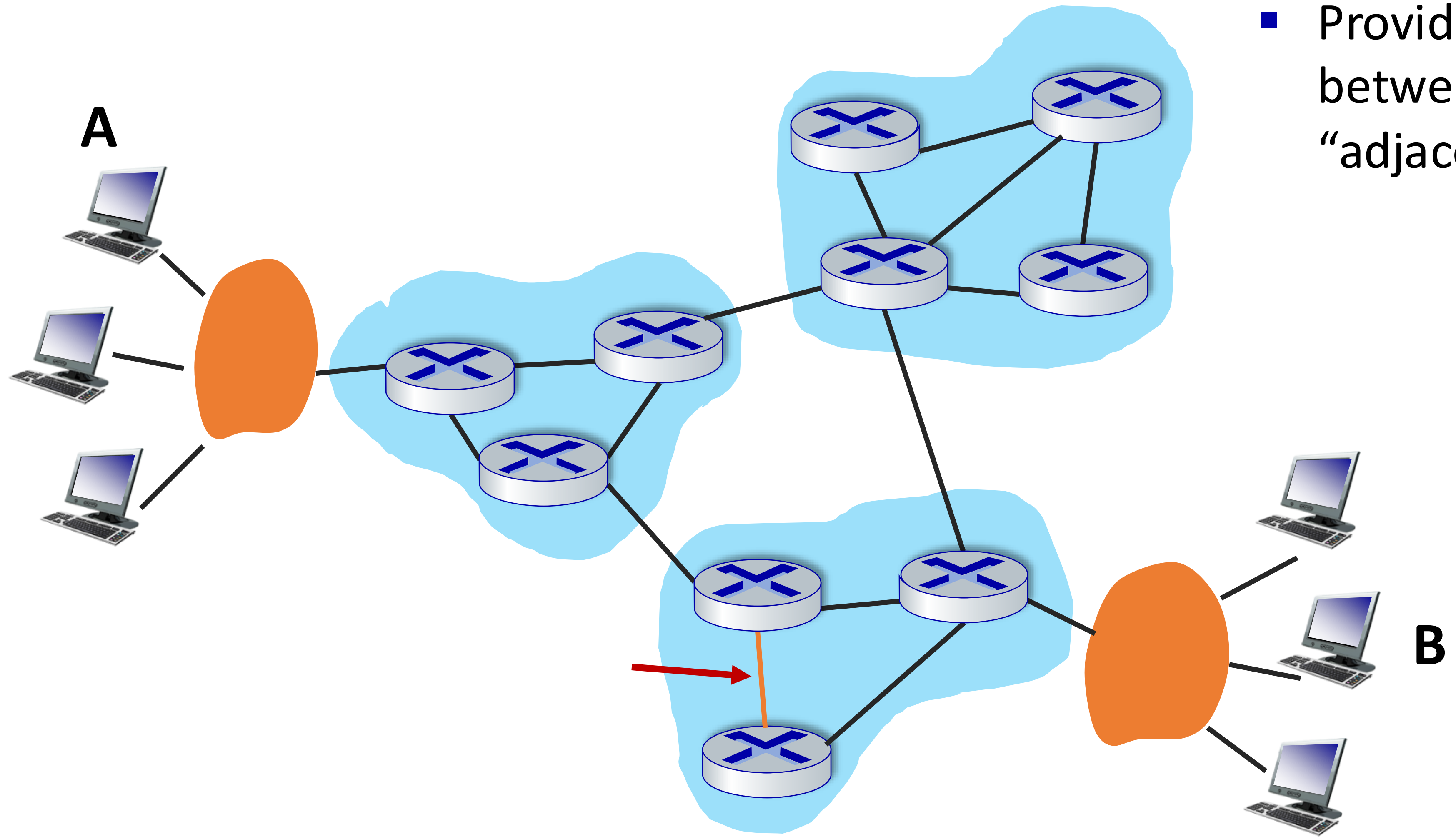
- Provides connectivity between (groups of) physically "adjacent" entities.

Link layer: local connectivity



- Provides connectivity between (groups of) physically "adjacent" entities.

Link layer: local connectivity

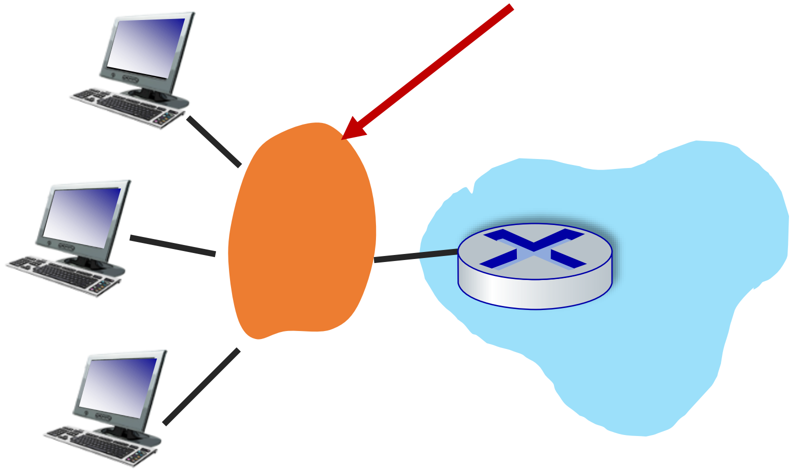


- Provides connectivity between (groups of) physically "adjacent" entities.

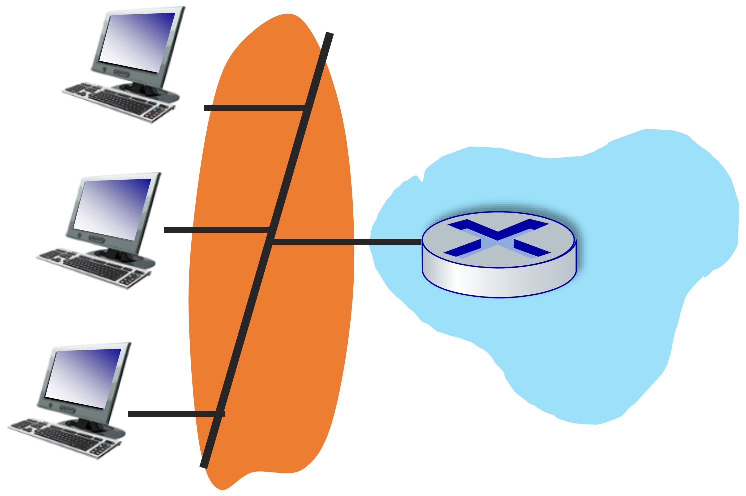
Link layer: local connectivity

Also called a Local Area Network (LAN)

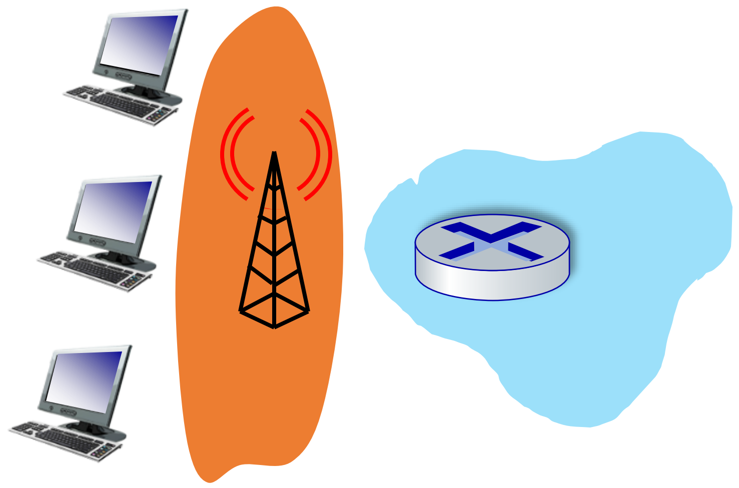
Either "shared link" or a link-layer network



Shared LAN via shared link



shared wire
(e.g., cabled Ethernet)



shared radio
(4G/5G, Wi-Fi, satellite)

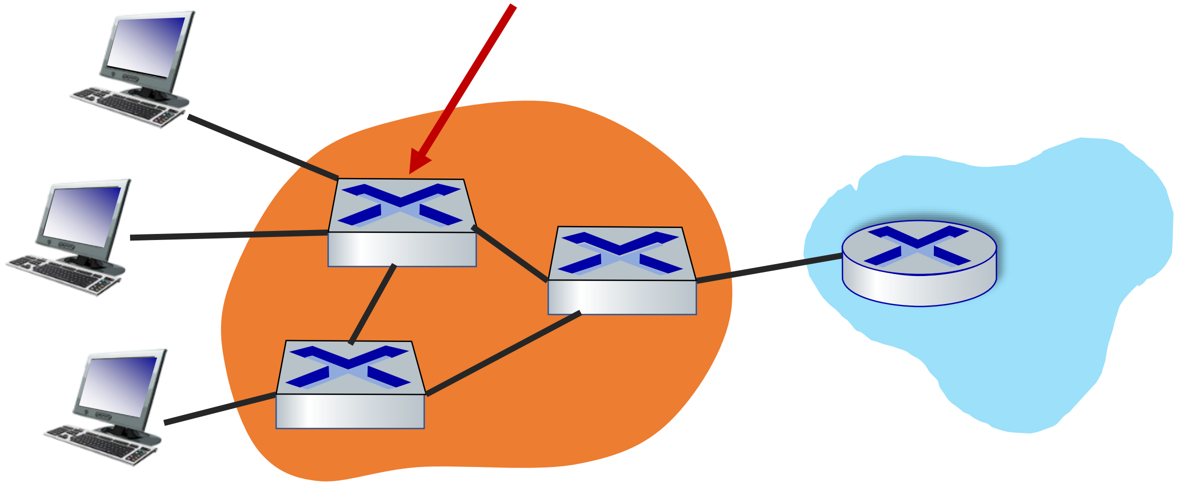
humans at a gathering
(shared air, acoustical)

Switched LAN via link-layer network

Will start with the "switched" LANs first and circle back to "shared links" afterwards.

L2 switch

L2 = Layer 2, aka link layer



Link layer: roadmap

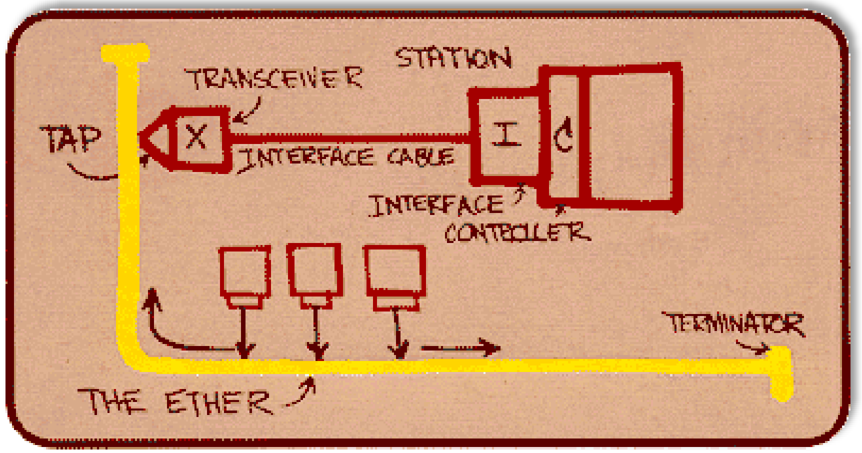
- Link layer overview
  - Local Area Networks (LANs)
- Switched LANs
  - Ethernet and Addressing
  - Address Resolution Protocol (ARP)
  - Switches
- Virtual LANs (VLANs)
- Shared LANs and multiple access protocols

Ethernet

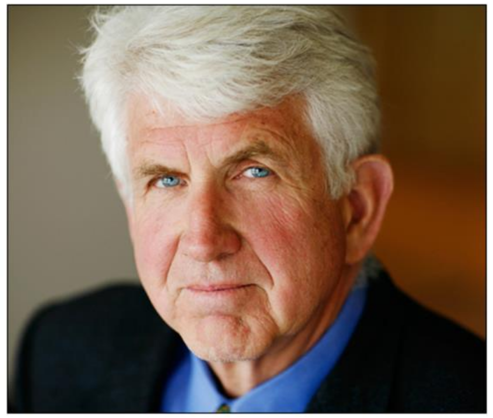
"dominant" wired LAN technology:

- first widely used LAN technology
- kept up with speed race: 10 Mbps – 400 Gbps

Metcalfe's Ethernet sketch



Bob Metcalfe: Ethernet co-designer, 2022 ACM Turing Award recipient

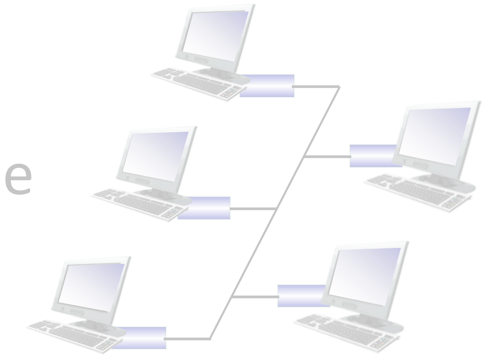


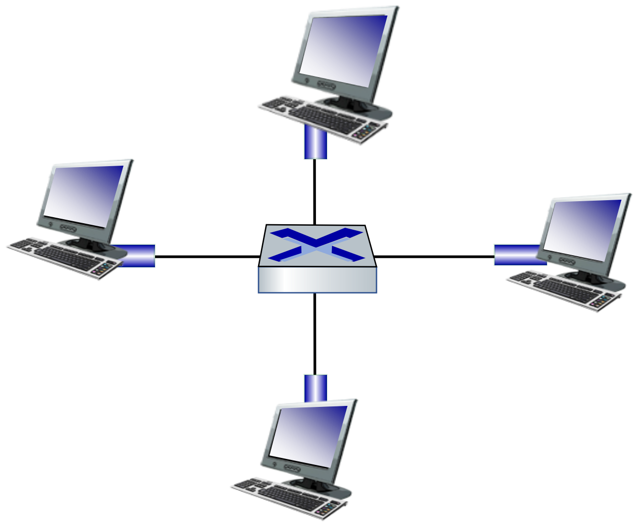
https://www.uspto.gov/learning-and-resources/journeys-innovation/audio-stories/defying-doubters

Ethernet: physical topology

- bus: popular through mid 90s
  - all nodes in same collision domain (can collide with each other)
  - “shared link”: will talk about it later.
- switched: prevails today
  - active link-layer *switch* in center
  - each “spoke” runs a (separate) Ethernet protocol (nodes do not collide with each other)

bus: coaxial cable





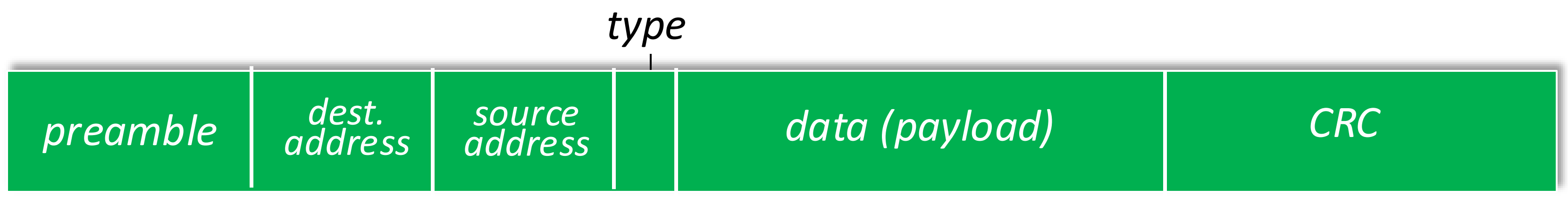
switched

Ethernet: unreliable, connectionless

- connectionless: no handshaking between sending and receiving interfaces
- unreliable: receiving interface doesn’t send ACKs to sending interface
  - data in dropped frames recovered only if initial sender uses higher layer rdt (e.g., TCP), otherwise dropped data lost

Ethernet frame structure

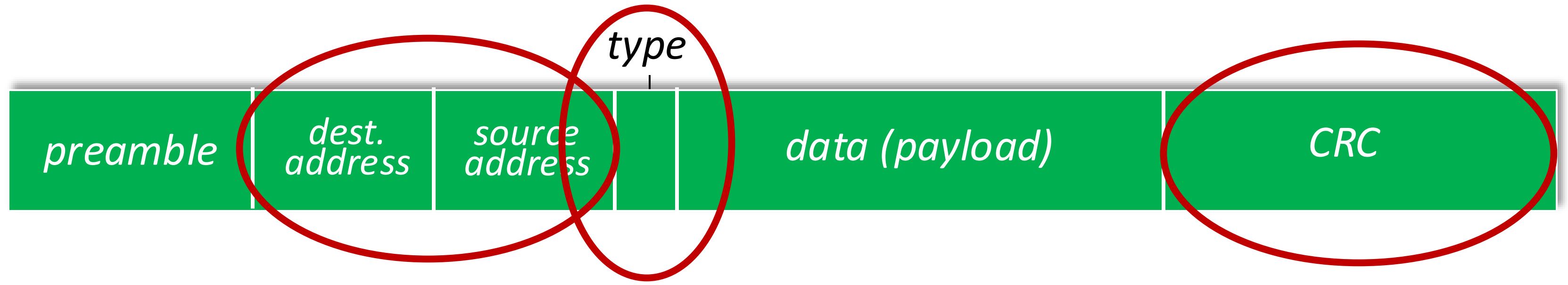
sending interface encapsulates IP datagram (or other network layer protocol packet) in Ethernet frame



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 interface receives frame with matching destination address, or with broadcast address, it passes data in frame to network layer protocol
  - otherwise, interface discards frame.
- type: indicates higher layer protocol
  - mostly IP but others possible, e.g., Novell IPX, AppleTalk
  - used to demultiplex up at receiver
- CRC: cyclic redundancy check at receiving interface
  - error detected: frame is dropped

MAC addresses

- 32-bit IP address:
  - *network-layer* address for interface
  - used for layer 3 (network layer) forwarding
  - e.g.: 128.119.40.136
- MAC (or LAN or physical or Ethernet) address:
  - function: used "locally" to get frame from one interface to another physically-adjacent interface (same subnet, in IP-addressing sense)
  - 48-bit MAC address (for most LANs) burned in the ROM of the interface hardware, also sometimes software settable
  - e.g.: 1A-2F-BB-76-09-AD

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

MAC addresses

Q: Why use a separate set of addresses in the link layer?
A: We will come back to this towards the end of the course.

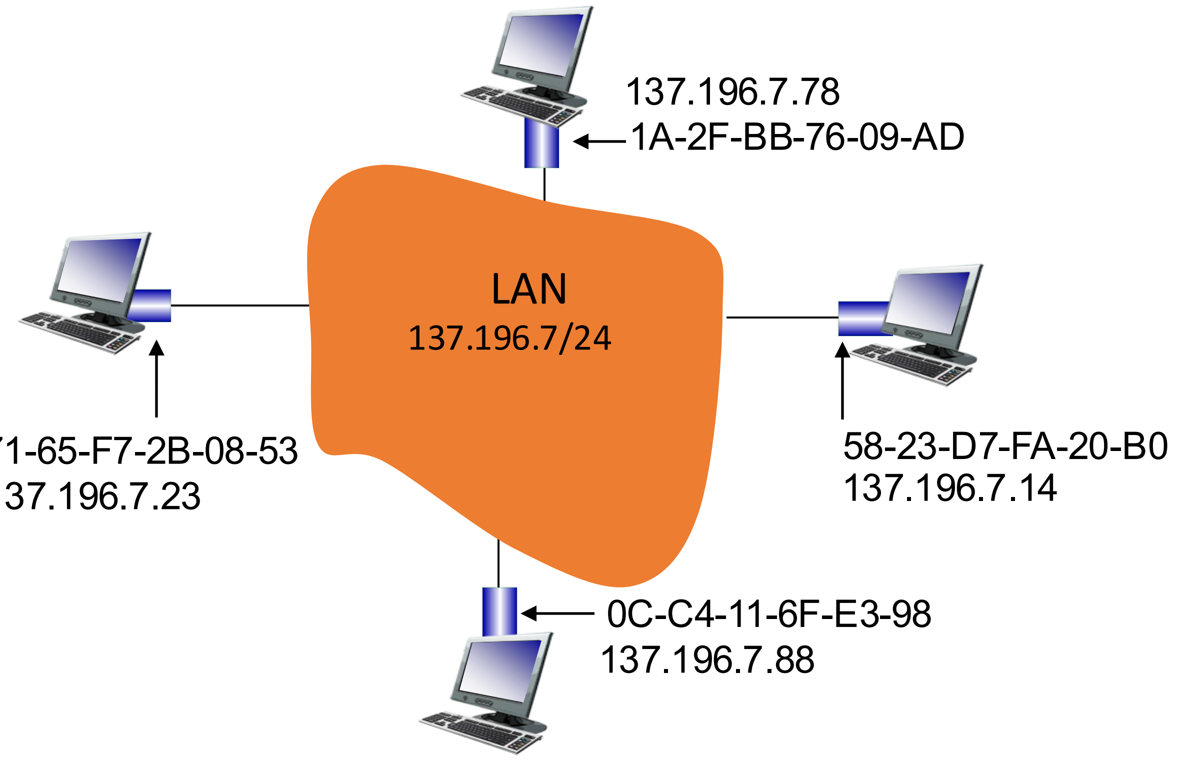
- 32-bit IP address:
  - *network-layer* address for interface
  - used for layer 3 (network layer) forwarding
  - e.g.: 128.119.40.136
- MAC (or LAN or physical or Ethernet) address:
  - function: used "locally" to get frame from one interface to another physically-adjacent interface (same subnet, in IP-addressing sense)
  - 48-bit MAC address (for most LANs) burned in the ROM of the interface hardware, also sometimes software settable
  - e.g.: 1A-2F-BB-76-09-AD

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

MAC addresses

each interface on LAN

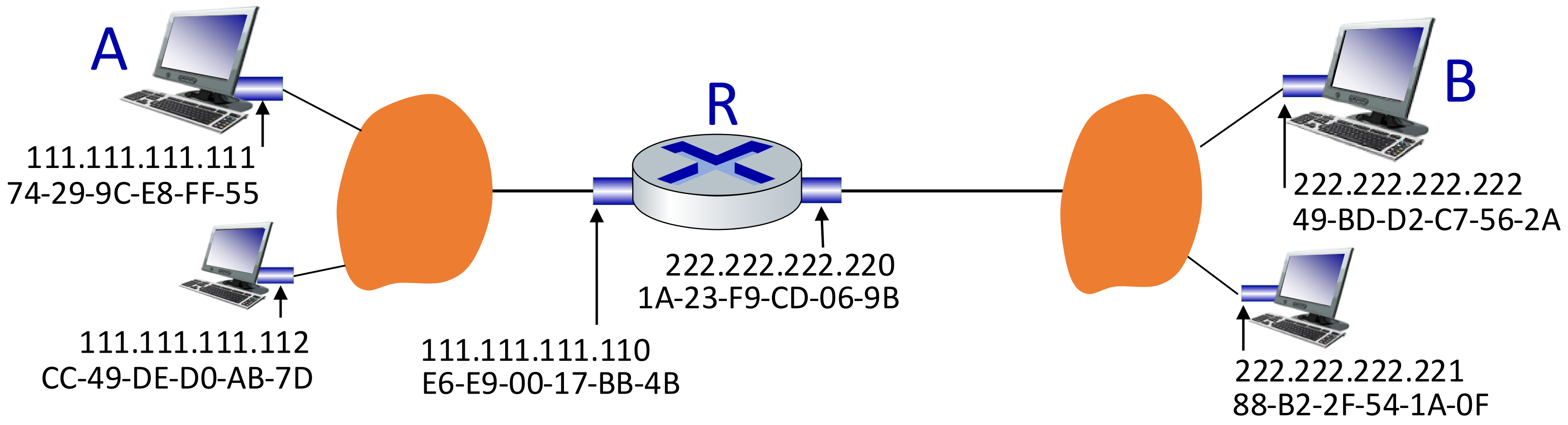
- has unique 48-bit MAC address
- has a locally unique 32-bit IP address (as we've seen)



Routing to another subnet: addressing

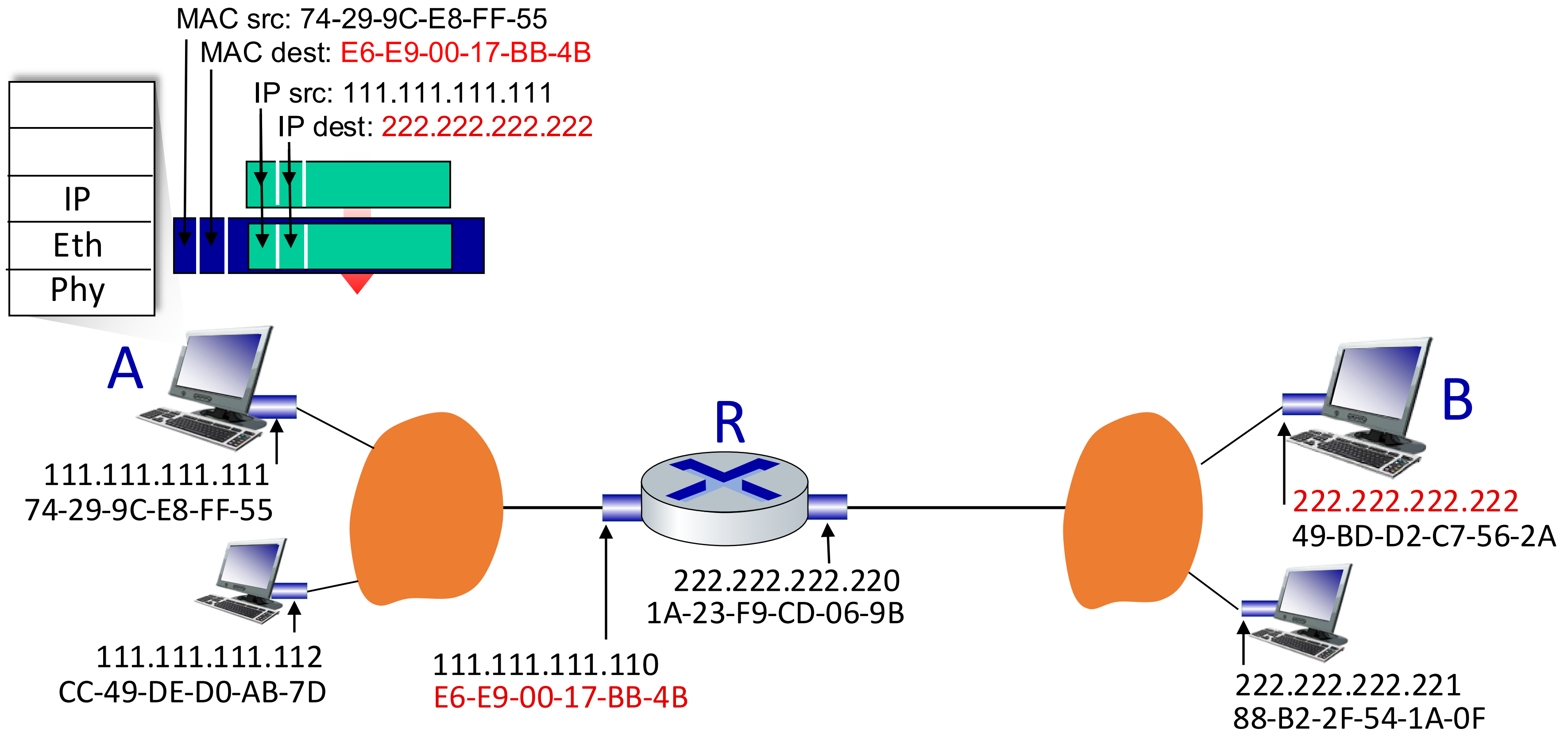
walkthrough: sending a datagram from *A* to *B* via *R*

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



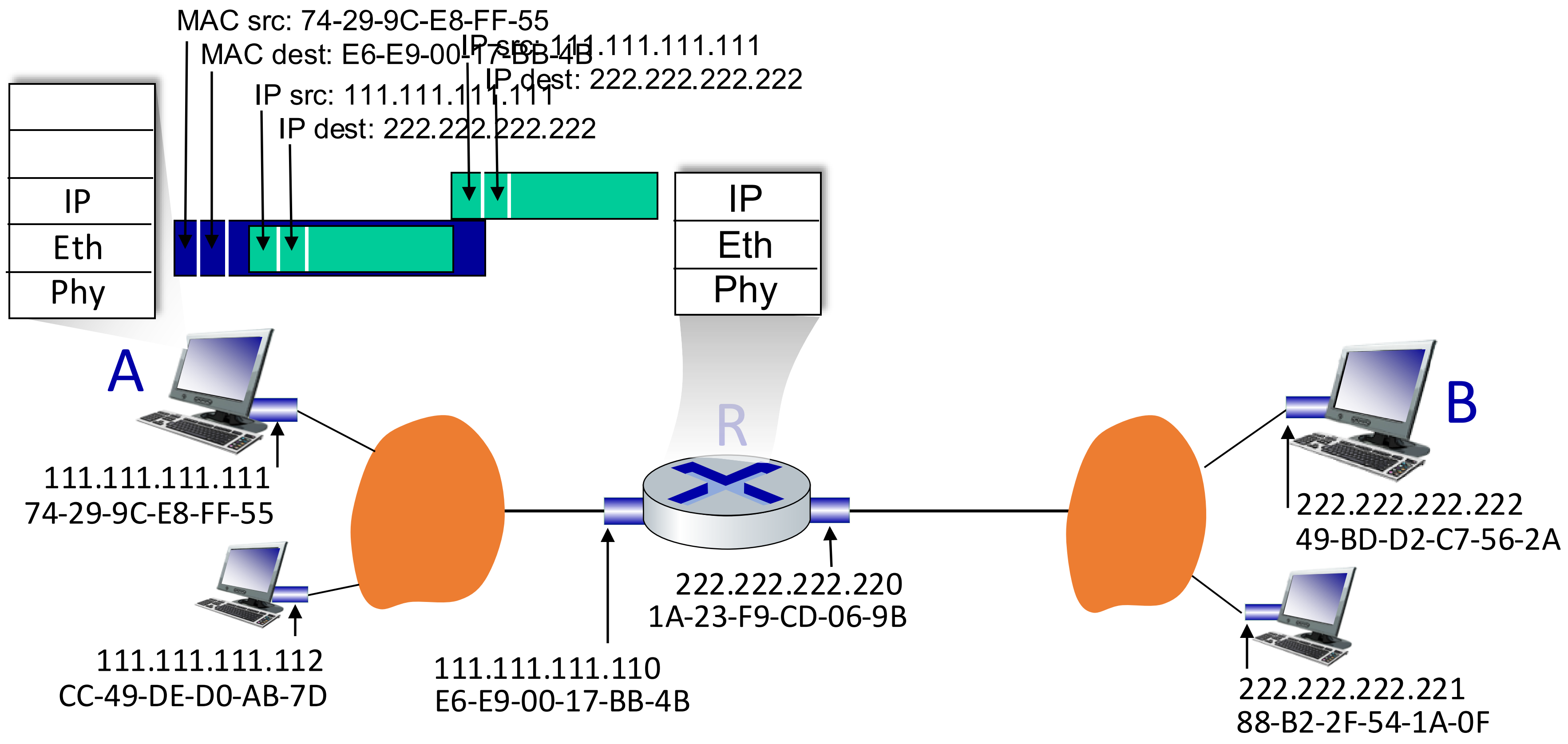
Routing to another subnet: addressing

- 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



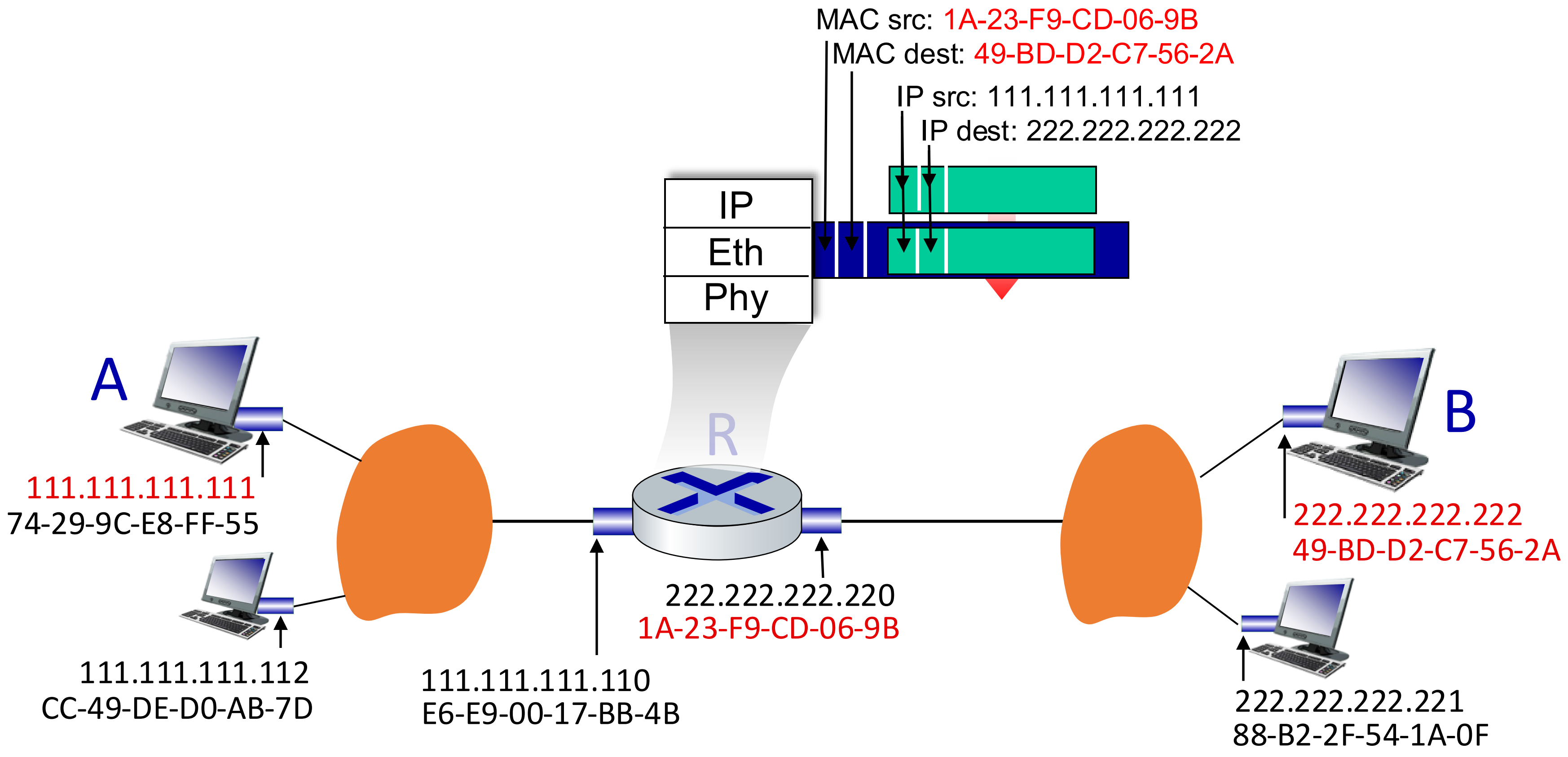
Routing to another subnet: addressing

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



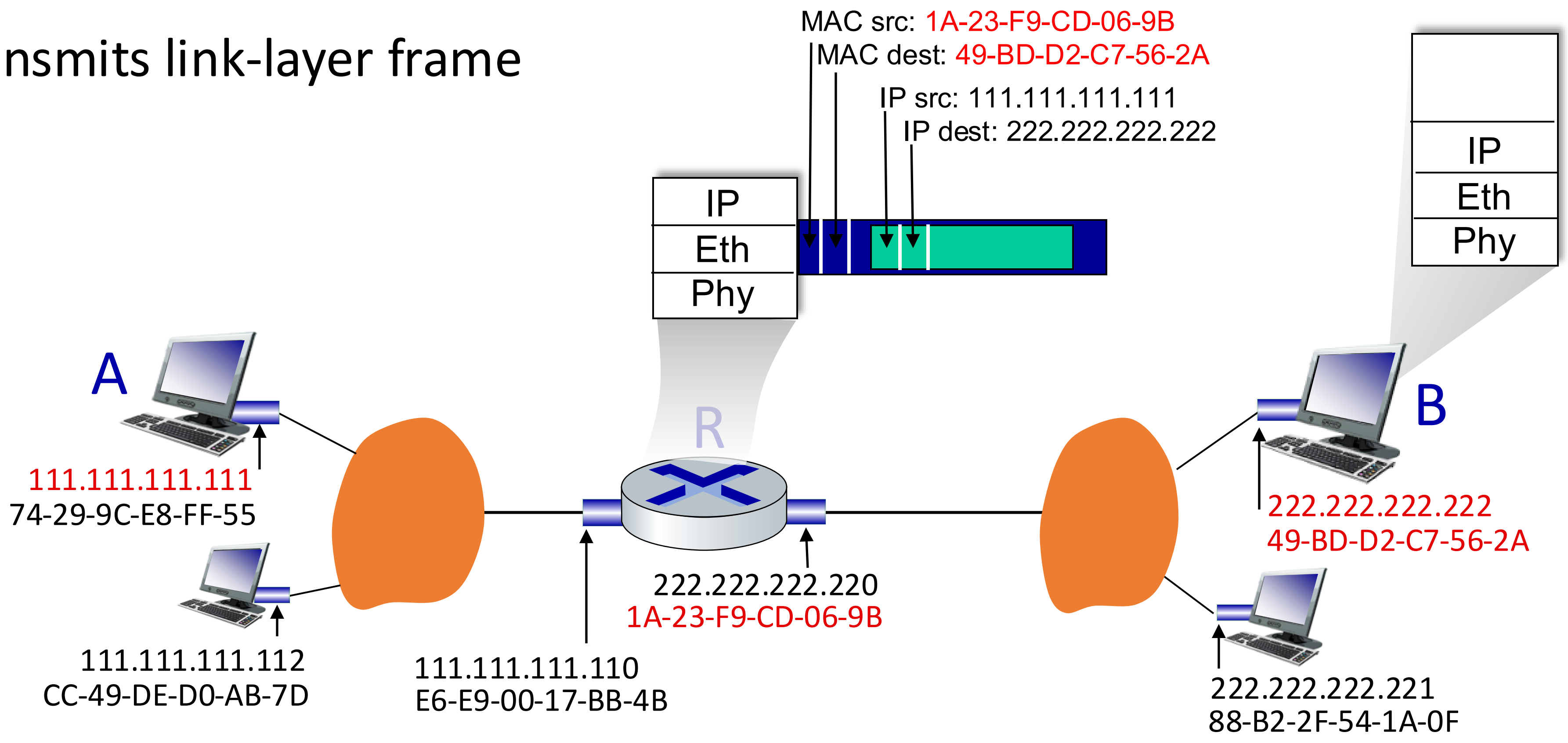
Routing to another subnet: addressing

- 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. Frame destination address: B's MAC address



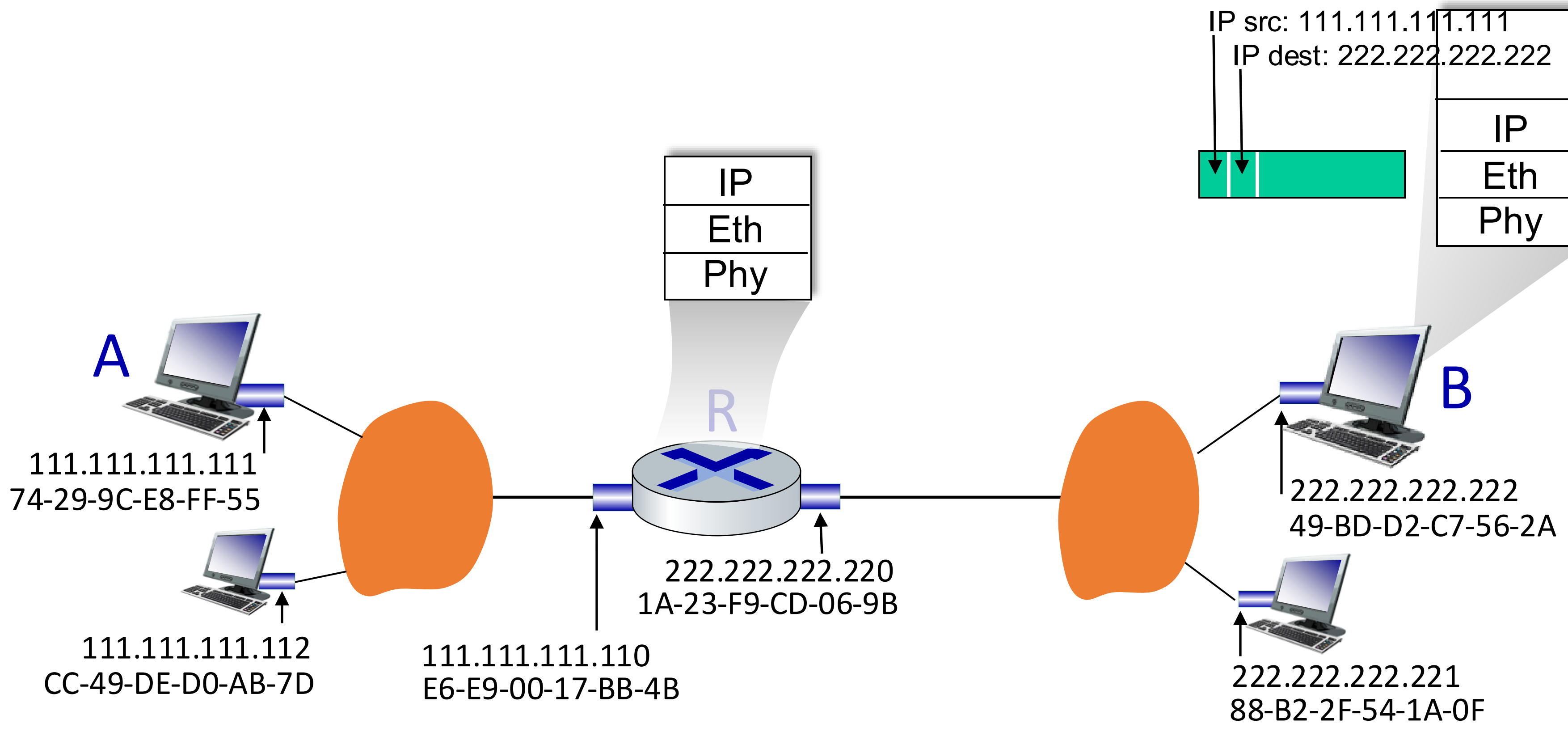
Routing to another subnet: addressing

- 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. Frame destination address: B's MAC address
- transmits link-layer frame



Routing to another subnet: addressing

- B receives frame, extracts IP datagram destination B
- B passes datagram up protocol stack to IP



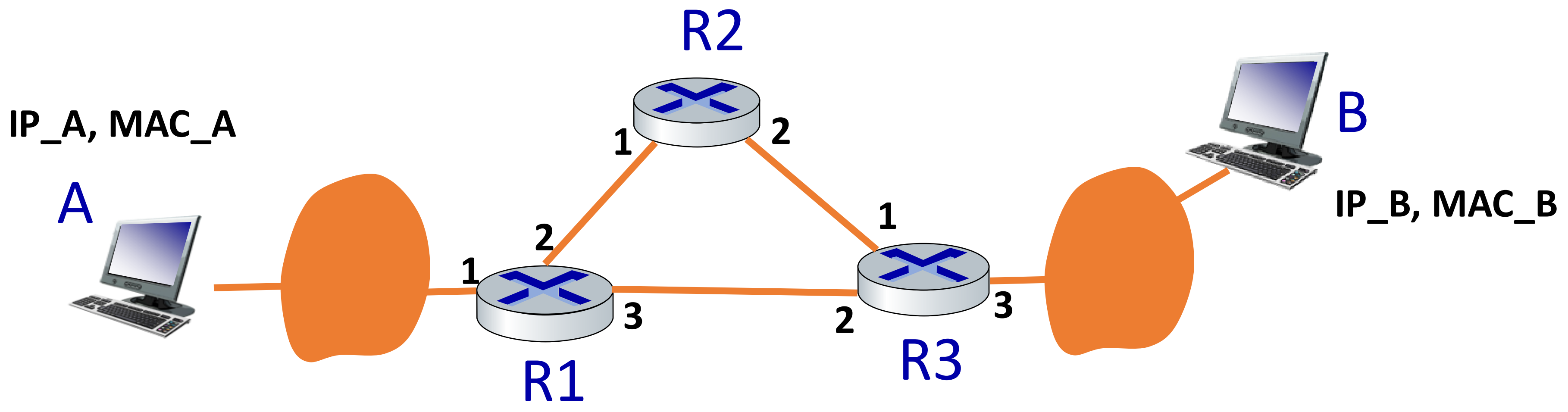
Make sure you know

- The division of labor between the network layer and the link layer
  - Global connectivity vs local connectivity
- What a MAC address is and how it is represented
- The important fields in the Ethernet header
  - Source MAC address, destination MAC address, type

Make sure you know

- How the network layer and the link layer work together to deliver packets end to end
  - E.g., given a packet, its path through the network, and information about the relevant interfaces, you should be able to figure out source and destination IP and MAC addresses as the packet traverses the network.

link layer forwarding exercise



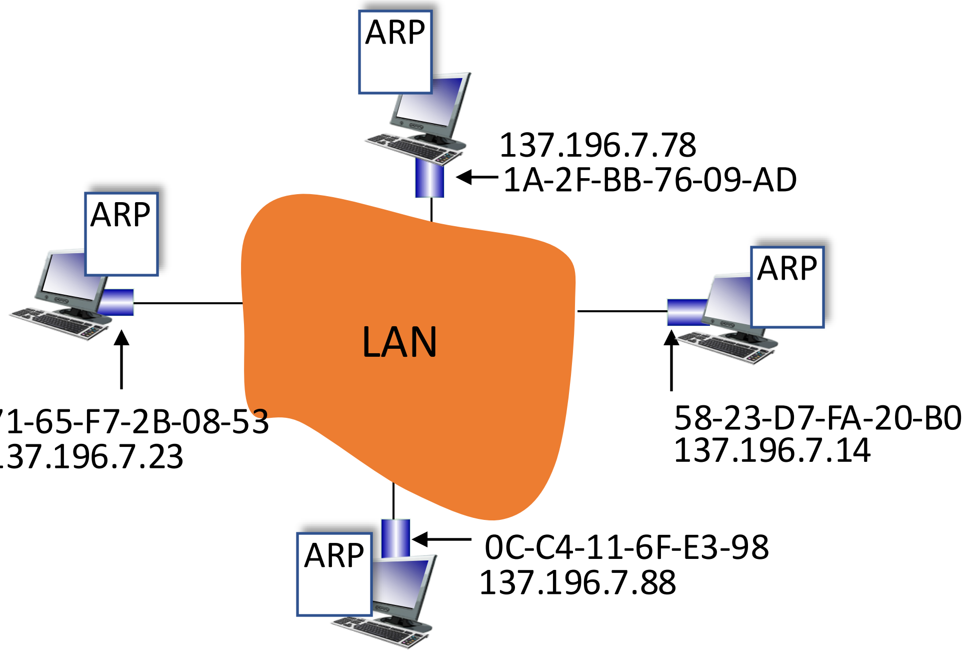
- IP_Ri_j = IP address of interface #j of Ri
- MAC_Ri_j = MAC address of interface #j of Ri
- Routing protocol picks least-cost paths. Link costs:
  - <R1,R2>: 1, <R2,R3>: 2, <R1, R3>: 5
- What are the source and destination IP and MAC address of a packet at each hop going from A to B?

Link layer: roadmap

- Link layer overview
  - Local Area Networks (LANs)
- Switched LANs
  - Ethernet and Addressing
  - Address Resolution Protocol (ARP)
  - Switches
- Virtual LANs (VLANs)
- Shared LANs and multiple access protocols

ARP: address resolution protocol

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



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

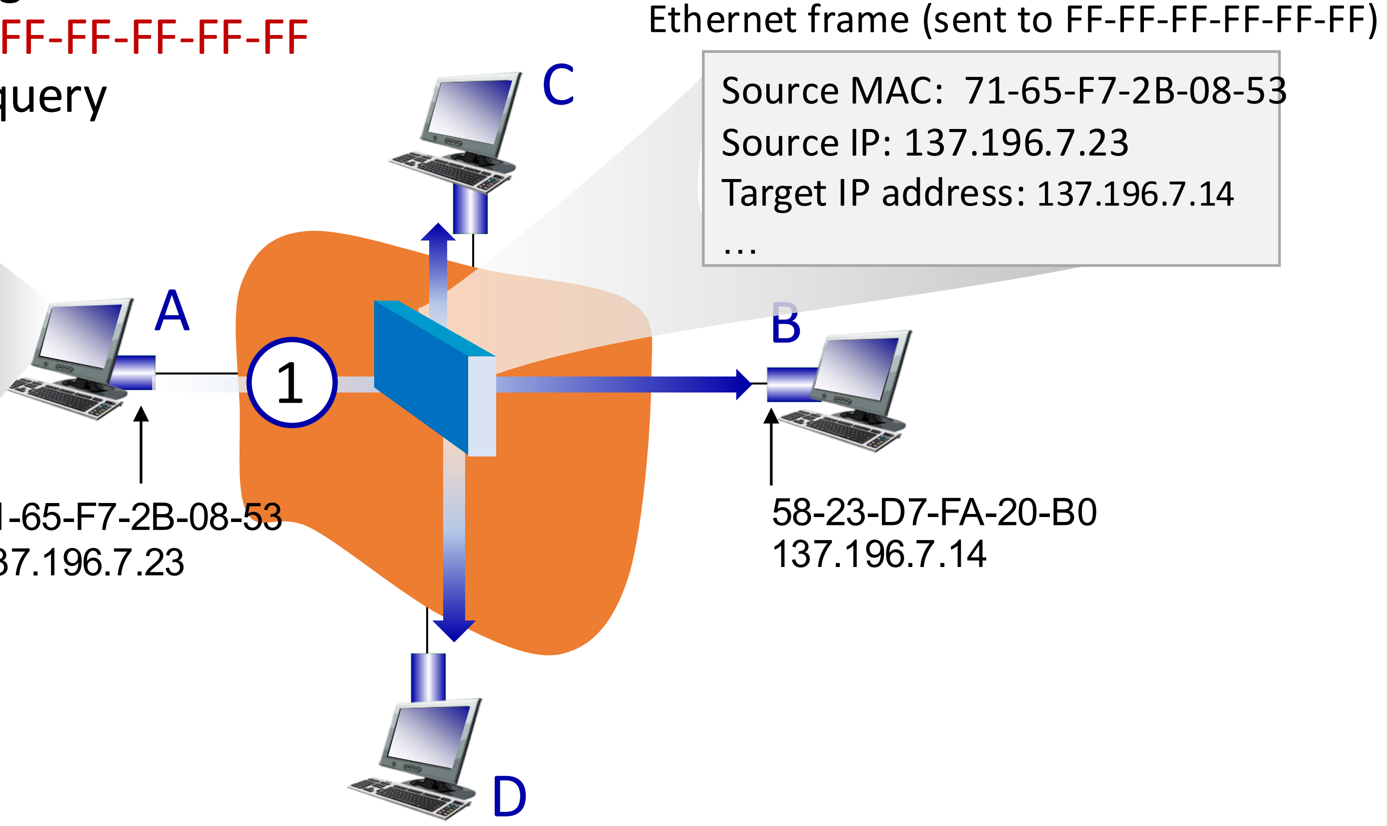
example: A wants to send datagram to B

- 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



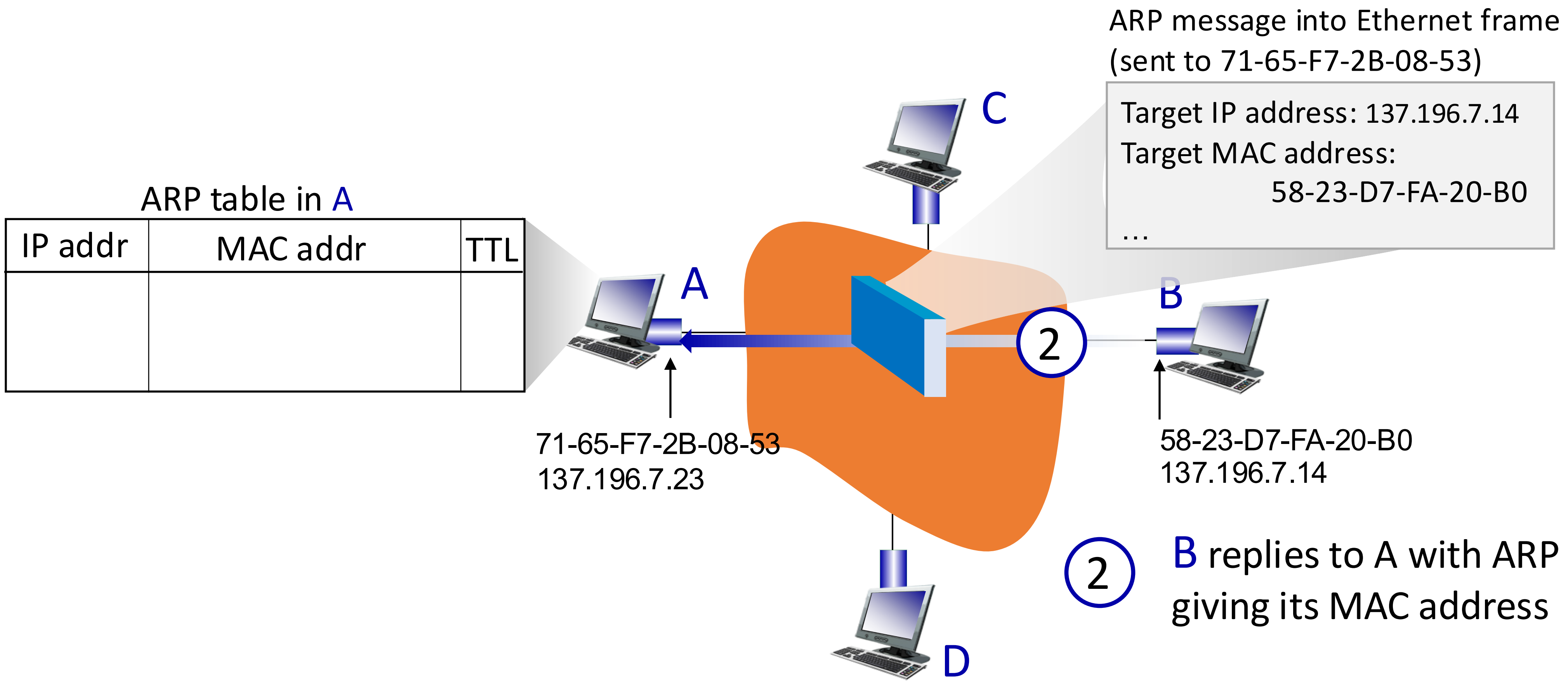
Ethernet frame (sent to FF-FF-FF-FF-FF-FF)
Source MAC: 71-65-F7-2B-08-53
Source IP: 137.196.7.23
Target IP address: 137.196.7.14
…
C
A
1
B
D
71-65-F7-2B-08-53
137.196.7.23
58-23-D7-FA-20-B0
137.196.7.14

ARP operates directly over Ethernet. In an ARP query, the first header is Ethernet, and then ARP.

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



IP addr	MAC addr	TTL

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

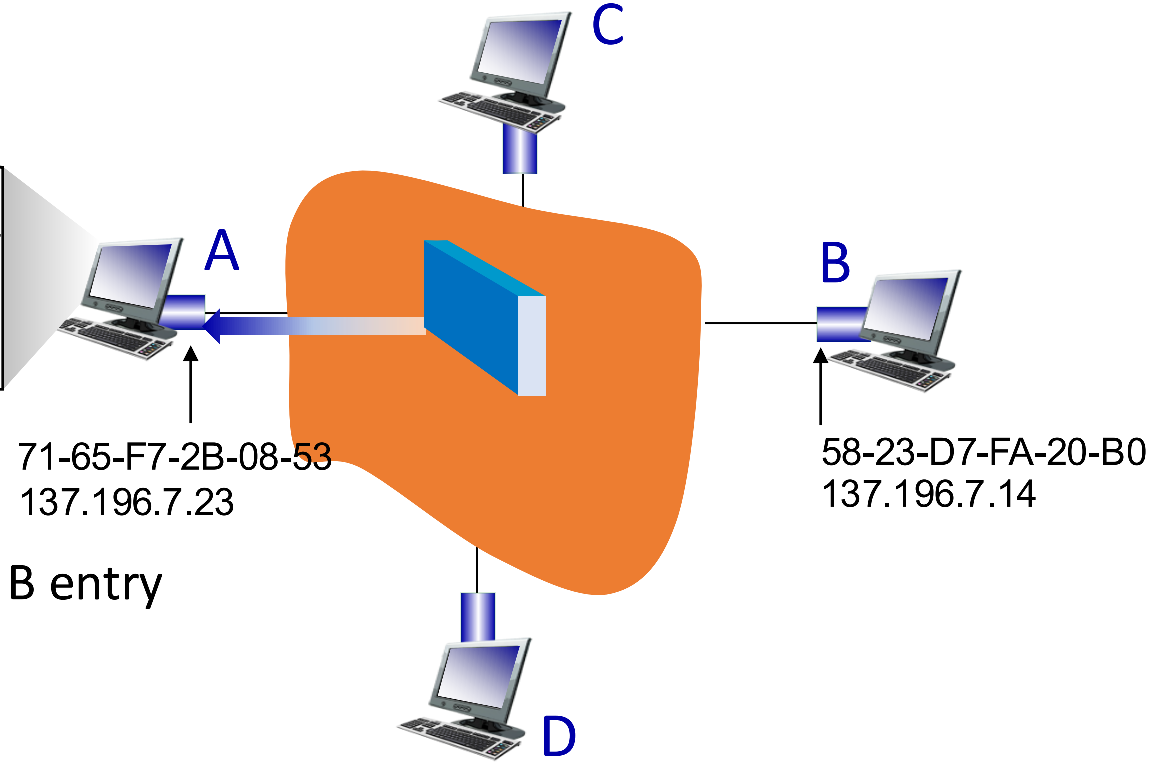
ARP protocol in action

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

Make sure you know

- That ARP operates directly over Ethernet
- How ARP helps end hosts determine the MAC address associated with an IP address.
  - The steps involved address resolution
  - What the source and destination MAC addresses in ARP queries and responses are
  - What is included in ARP queries and responses.

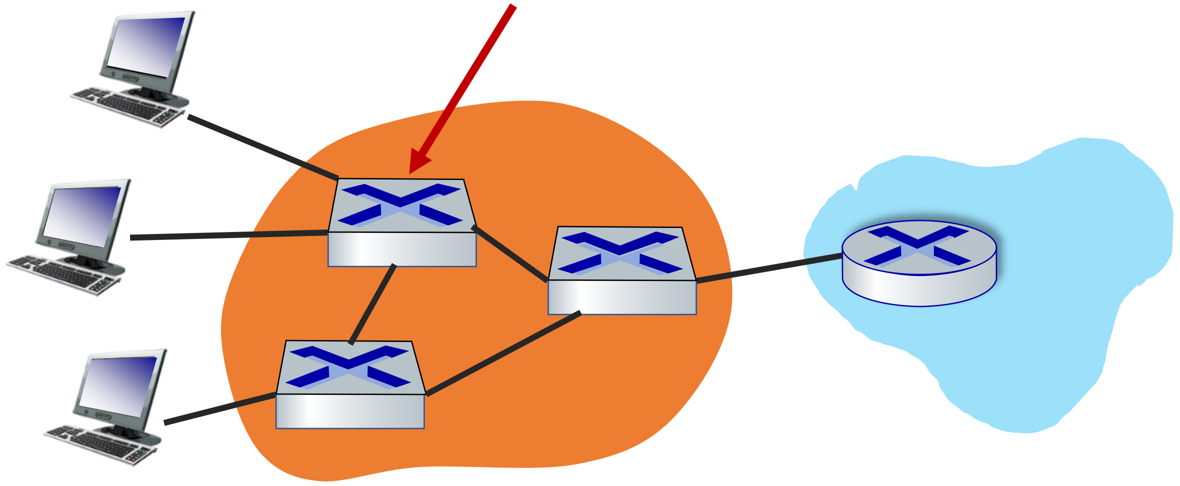
Link layer: roadmap

- Link layer overview
  - Local Area Networks (LANs)
- Switched LANs
  - <u>Ethernet and Addressing</u>
  - <u>Address Resolution Protocol (ARP)</u>
  - <u>Switches</u>
- Virtual LANs (VLANs)
- Shared LANs and multiple access protocols

Link layer: a link-layer network

L2 switch

L2 = Layer 2, aka link layer

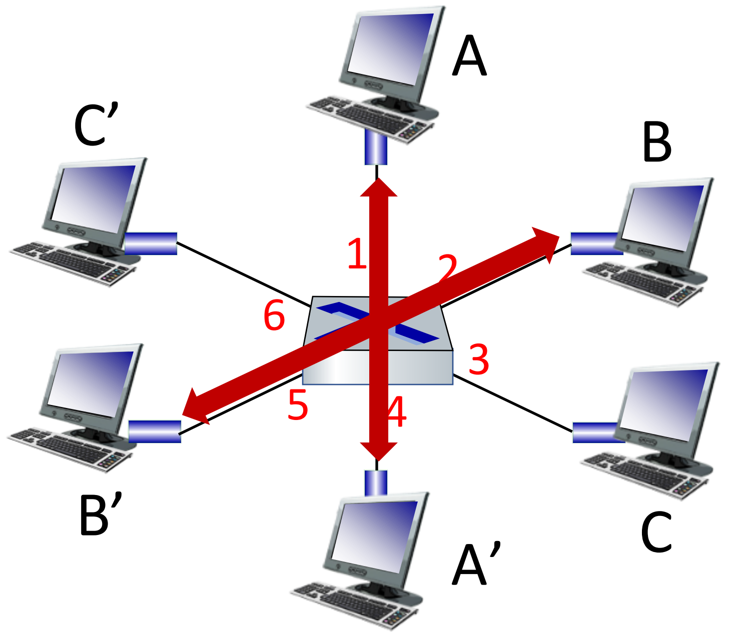


Ethernet switch

- A link-layer switch
  - store, forward Ethernet (or other type of) frames
  - examine incoming frame's MAC address, *selectively* forward frame to one-or-more outgoing links
- transparent: hosts *unaware* of presence of switches
- plug-and-play, self-learning
  - switches do not need to be configured

Ethernet Switch

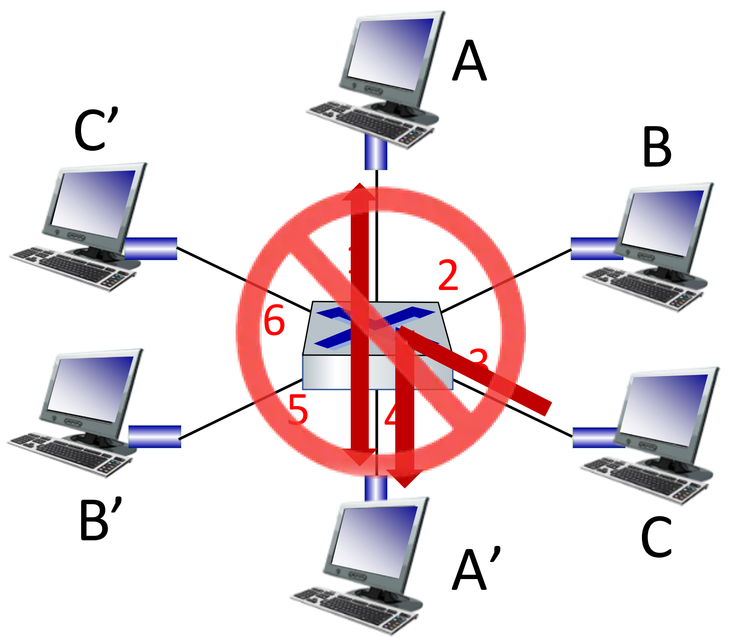
- hosts have dedicated, direct connection to switch
- switches buffer packets
- Ethernet protocol used on *each* incoming link, so:
  - No shared medium, no collisions
  - each link is its own collision domain



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

Ethernet Switch

- hosts have dedicated, direct connection to switch
- switches buffer packets
- Ethernet protocol used on *each* incoming link, so:
  - No shared medium, no collisions
  - each link is its own collision domain



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

Switch forwarding table

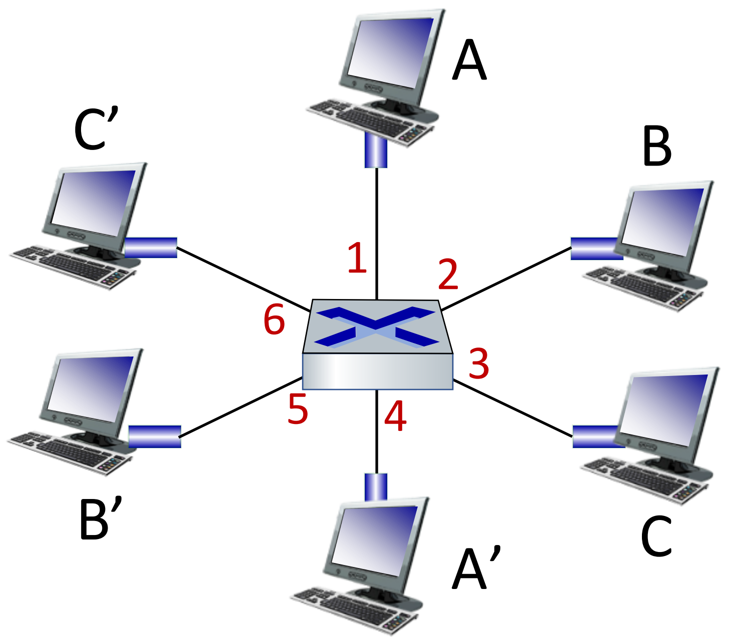
Q: how does switch know A' reachable via interface 4, B' reachable via interface 5?

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

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

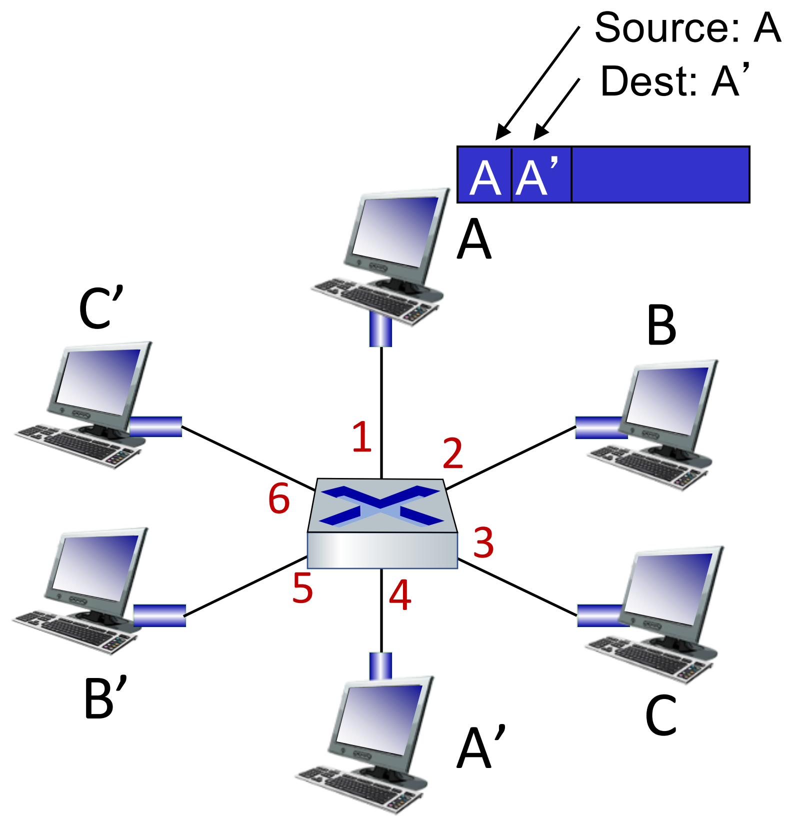
Q: how are entries created, maintained in switch table?

- something like a routing protocol?



Switch: 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



MAC addr	interface	TTL
A	*1*	*60*

Switch table (initially empty)

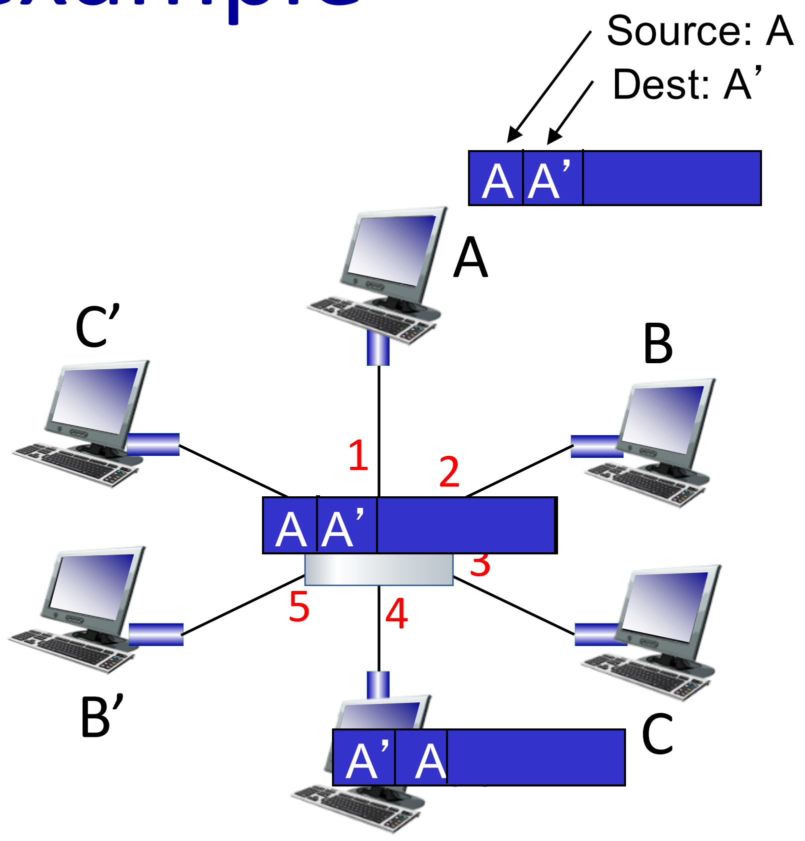
Switch: frame filtering/forwarding

when frame received at switch:

```
1. record incoming link, MAC address of sending host
2. Check there is an entry for the  MAC destination address in the
   switch table
3. if entry found for destination
     then {
     if destination connected to the interface from which frame arrived
         then drop frame
          else forward frame on interface indicated by entry
      }
      else flood  /* forward on all interfaces except arriving interface */
```

Self-learning, forwarding: 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)

Switch: frame filtering/forwarding

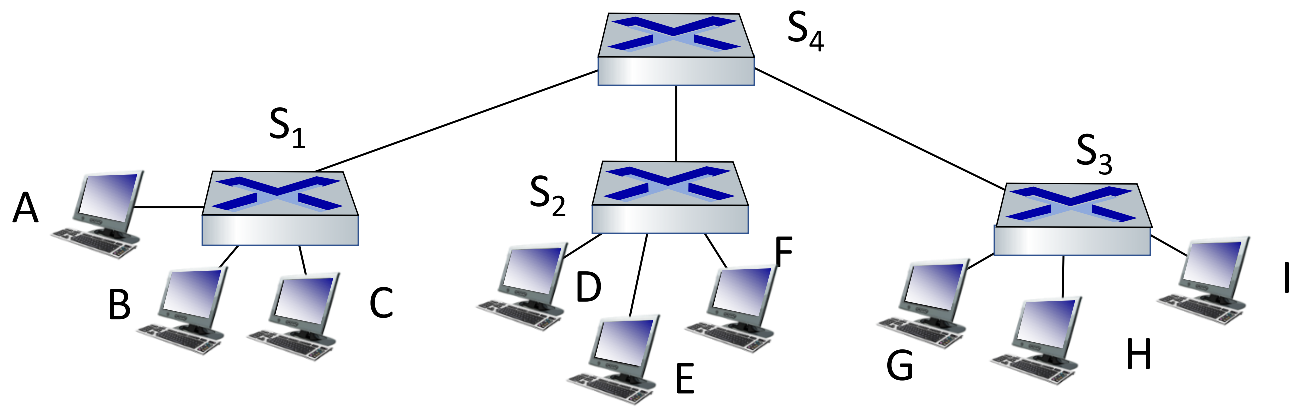
when frame received at switch:

```
1. record incoming link, MAC address of sending host
2. Check there is an entry for the  MAC destination address in the
   switch table
3. if entry found for destination
     then {
     if destination connected to the interface from which frame arrived
        then drop frame
        else forward frame on interface indicated by entry
     }
    else flood  /* forward on all interfaces except arriving interface */
```

Why?

Interconnecting switches

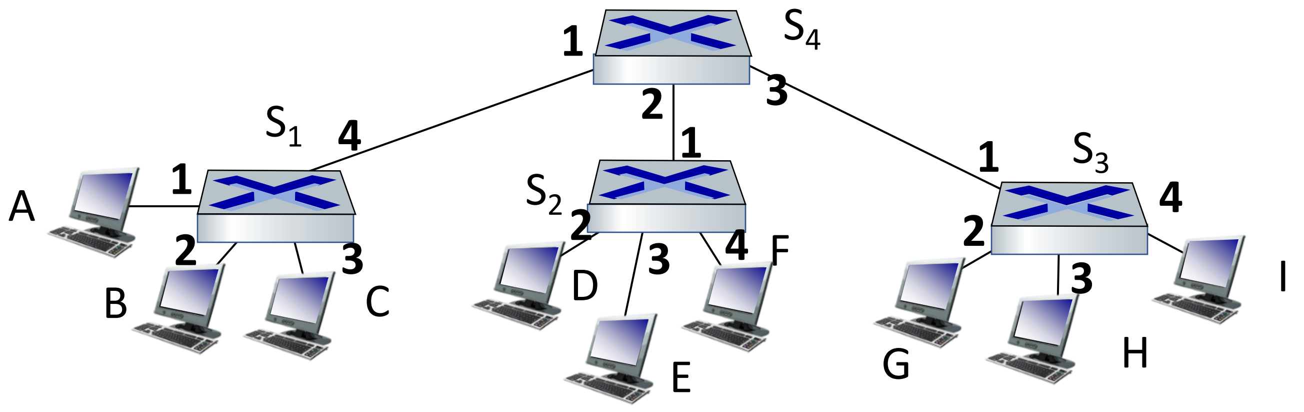
self-learning switches can be connected together:



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!)

Self-learning multi-switch exercise



- MAC_X is the MAC address of end host X
- Suppose C sends frame to I, I responds to C.
- show updates to switch tables as packets are forwarded in S_1, S_2, S_3, S_4

Answer

- C sends a frame with source MAC address MAC_C and dest MAC address MAC_I.
- S1 updates its switch table to record <MAC_C, 3, ttl>
- S1 sends copies of the frame on ports 1, 2, 4
- S4 receives the frame, updates it switch table to record <MAC_C, 1, ttl>
- S4 sends copies of the frame on ports 2 and 3
- S2 receives the frame, updates it table to record <MAC_C, 1, ttl>
- S2 sends copies of the frame on ports 2, 3, 4.

Answer

- S3 receives the frame, updates its table to record <MAC_C, 1, ttl>
- S3 forwards copies of the frame on ports 1, 2, 4
- I receives the frame, responds with a frame with source MAC address MAC_I and dest MAC address MAC_C
- S3 updates its table to record <MAC_I, 4, ttl>
- S3 forwards the frame on port 1
- S4 updates its table to record <MAC_I, 3, ttl>
- S4 forwards the frame on port 1

Answer

- S1 updates its table to record <MAC_I, 4, ttl>
- S1 forwards the frame on port 3

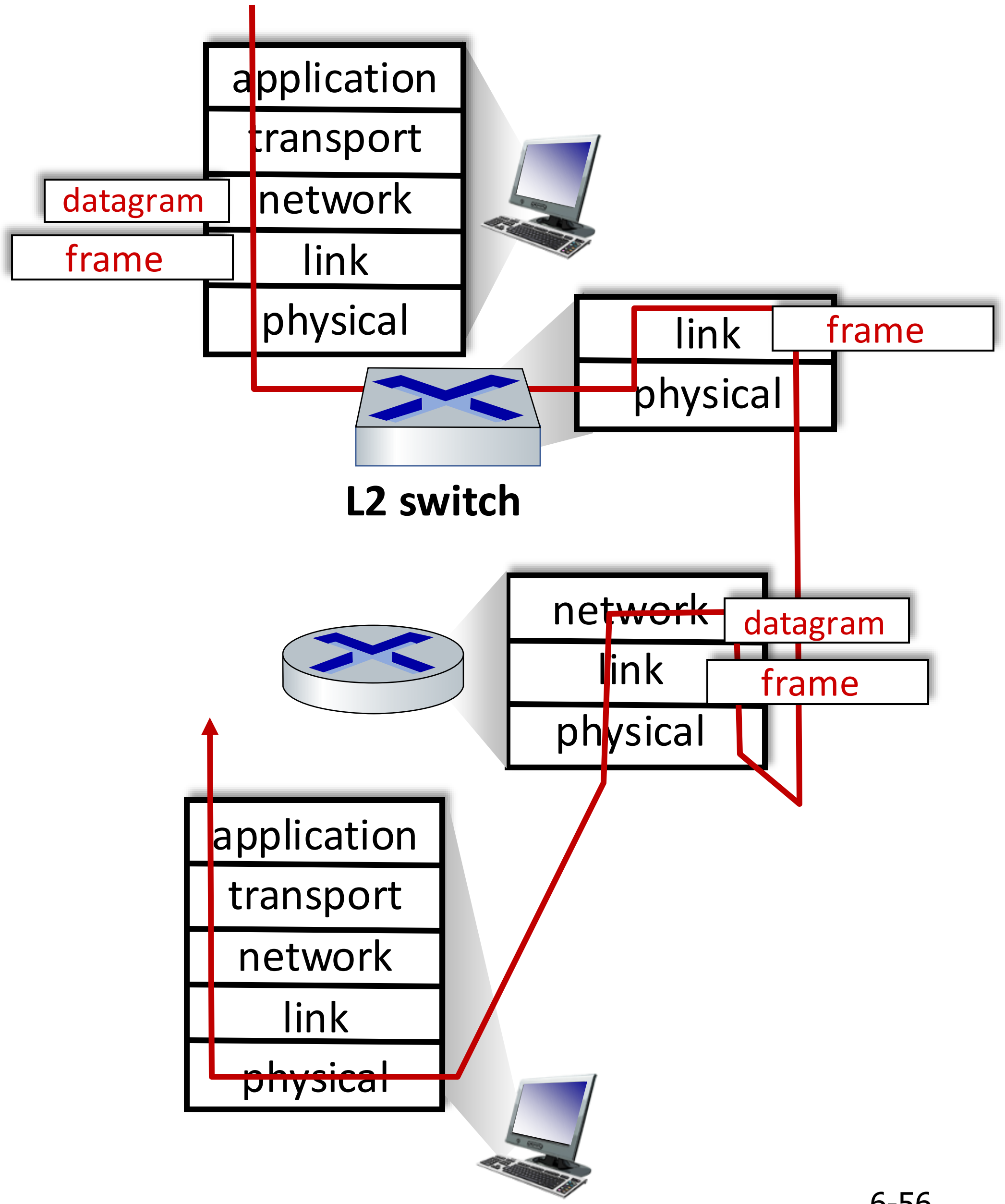
L2 Switches vs. routers

both are store-and-forward:

- *routers*: network-layer devices (examine network-layer headers)
- *L2 switches:* link-layer devices (examine link-layer headers)

both have forwarding tables:

- *routers:* compute tables using routing algorithms, IP addresses
- *L2 switches:* learn forwarding table using flooding, learning, MAC addresses



Make sure you know

- What Ethernet switches are
- How they forward data between the interfaces of a LAN
  - When there is only a single L2 switch
  - Or when there is a network L2 switches
- Specifically, make sure you know how the switch forwards packets
  - The switch table
- And how it learns mappings between MAC addresses and interfaces to reach them.