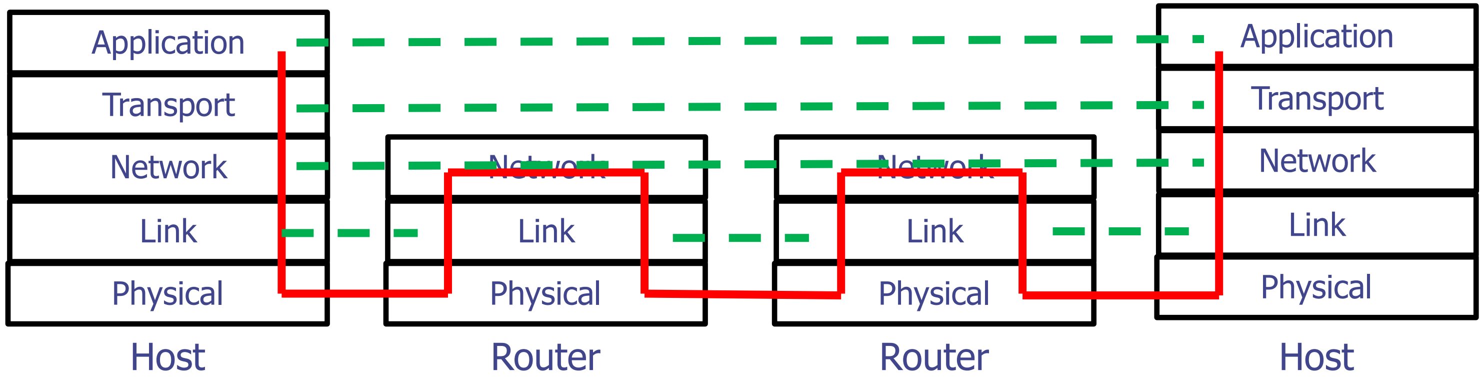
Lecture 9
The Link Layer

The Link Layer

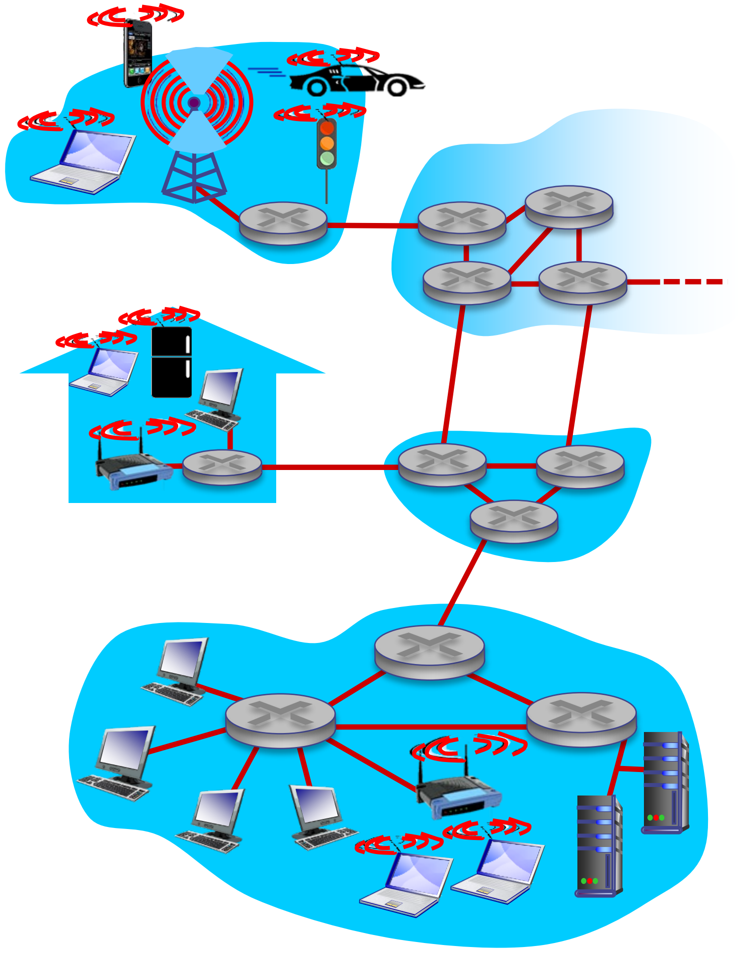


Packet		Layer	Respocibility
Message	M	Application	Message to remote process
Segment	h M	Transport	Process to process delivery
Datagram	h S	Network	Host to host delivery
Frame	h D	Link	**Node to node delivery**
	h F	Physical	

Link layer: introduction

Terminology:

- Hosts and routers: nodes
- Communication channels that connect neighbor nodes along communication path: links
  - Wired or Wireless links
  - LANs
- Layer-2 packet: frame, encapsulates datagram



data-link layer has responsibility of transferring datagrams from one node to *physically adjacent* node over a link

Link layer: context

- Datagram transferred by different link protocols over different links:
  - e.g., Ethernet on first link, frame relay on intermediate links, 802.11 on last link
- Each link protocol provides different services
  - e.g., may or may not provide rdt over link

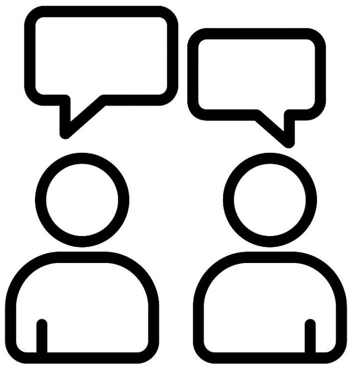
Transportation analogy:

- Trip from Princeton to Lausanne
  - Car: Princeton to JFK
  - Plane: JFK to Geneva
  - Train: Geneva to Lausanne
- Tourist = datagram
- Transport segment = **communication link**
- Transportation mode = **link layer protocol**
- Travel agent = **routing algorithm**

Link layer services

- **Framing, link access:**
  - Encapsulate datagram into frame, adding header, trailer
  - Channel access if shared medium
  - Media Access Control (MAC) addresses used in frame headers to identify source, destination
    - Different from IP address!

- **Reliable delivery between adjacent nodes**
  - We learned how to do this already (Transport Layer)!
  - Seldom used on low bit-error link (fiber, some twisted pair)
  - Wireless links: high error rates



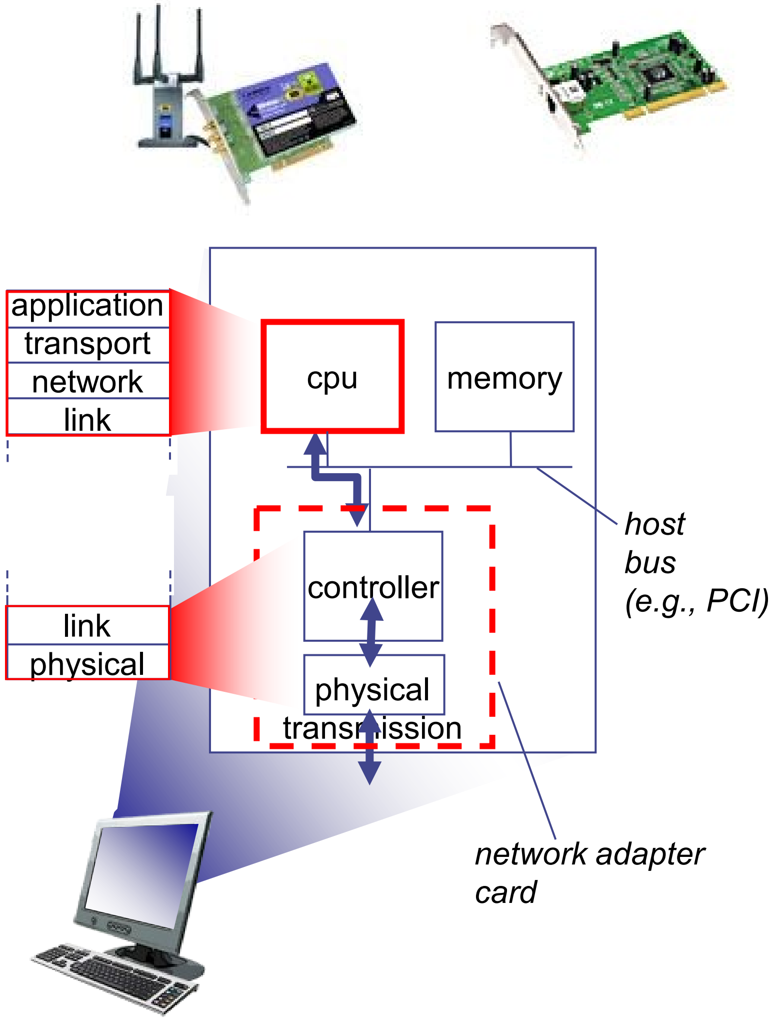
Why both node-to-node and end-to-end reliability?

Link layer services (more)

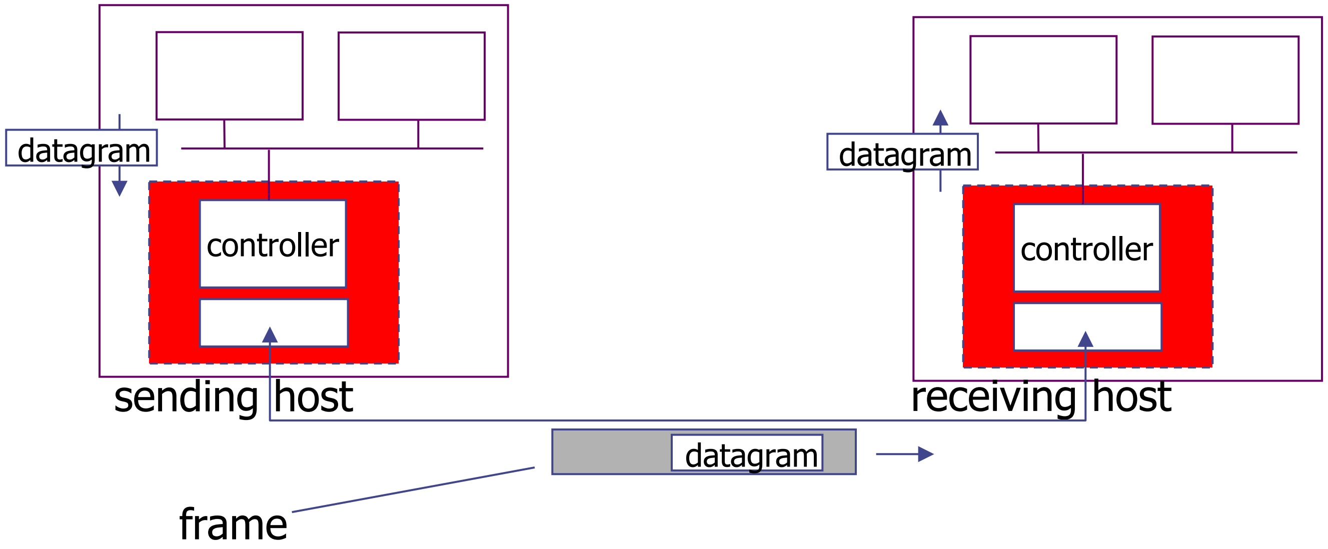
- **Flow control:**
  - pacing between adjacent sending and receiving nodes
- **Error detection:**
  - errors caused by signal attenuation, noise.
  - receiver detects presence of errors:
    - signals sender for retransmission or drops frame
- **Error correction:**
  - receiver identifies and corrects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

Where is the Link Layer Implemented?

- In each and every host and router.
- Link layer implemented in "adaptor" (aka *network interface card* NIC) or on a chip
  - Ethernet card, 802.11 card; Ethernet chipset
  - Implements link, physical layer
- Attaches into host's system buses
- Combination of hardware, software, firmware



Adaptors Communicating



Sending side:

- encapsulates datagram in frame
- adds error checking bits, rdt, flow control, etc.

Receiving side

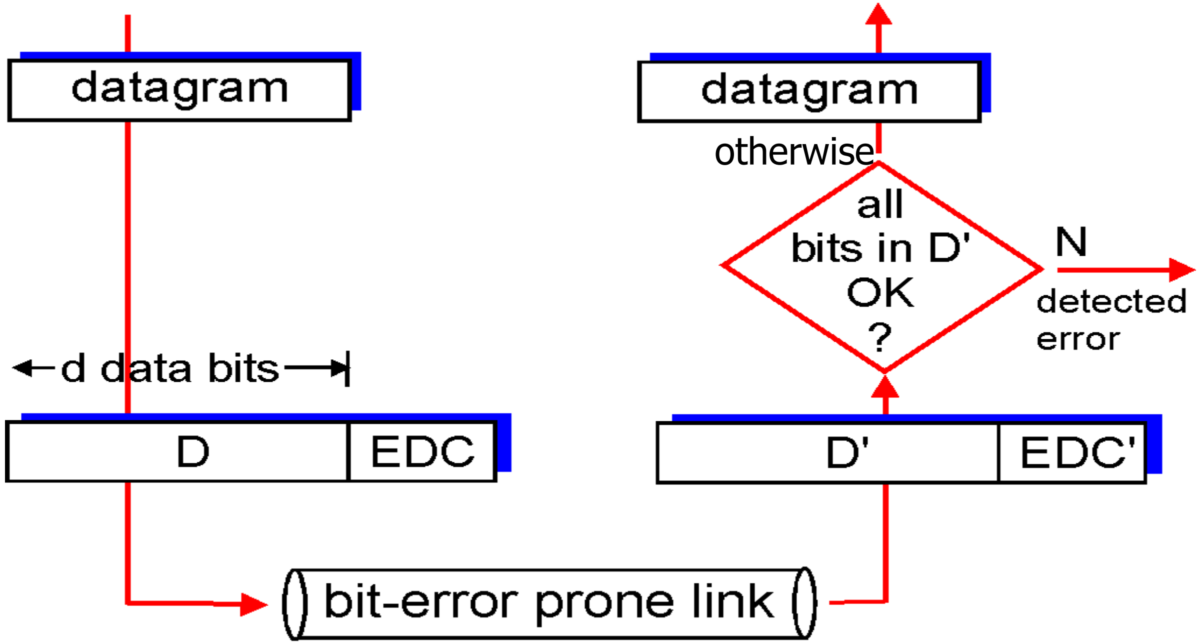
- looks for errors, rdt, flow control, etc.
- extracts datagram, passes to upper layer at receiving side

9.2 Error Detection and Correction

Error detection

EDC = Error Detection and Correction bits (redundancy)
D = Data protected by error checking, may include header fields

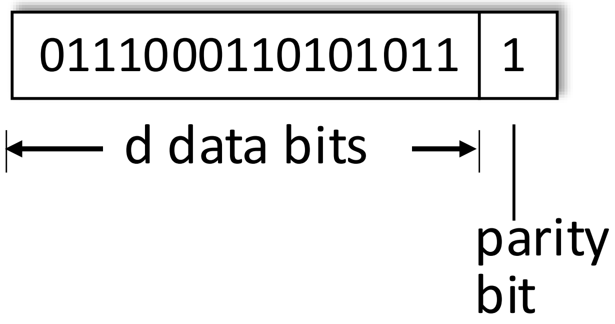
- Error detection not 100% reliable!
  - protocol may miss some errors, but rarely
  - larger EDC field yields better detection and correction



Parity checking

Single bit parity:

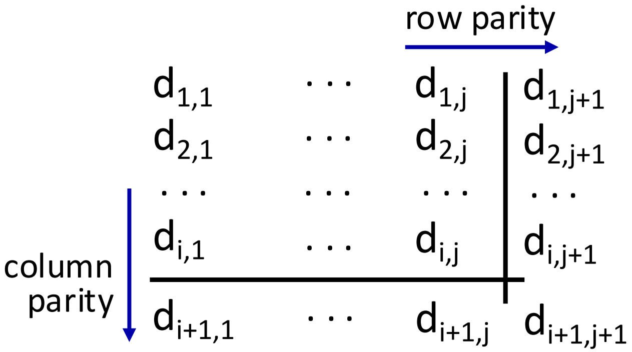
- Detect single bit errors

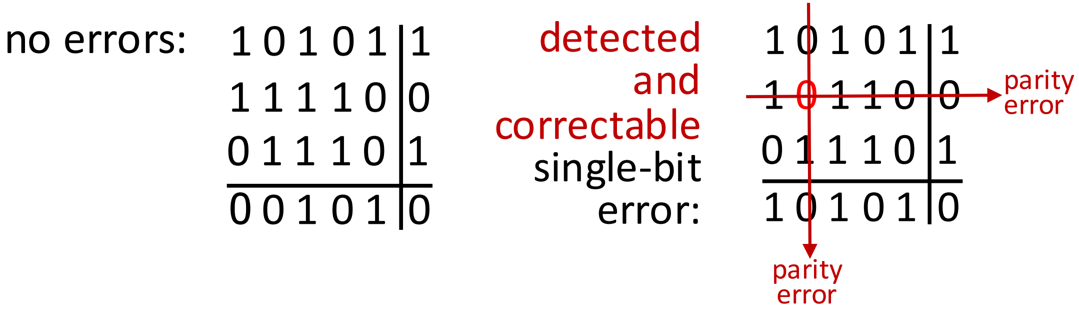


Even parity: set parity bit so there is an even number of 1's

Two-dimensional bit parity:

- Detect and correct single bit errors





SDU

CRC and Error Detection Overview

1. The sender treats the data bits as a long binary number
2. It divides this number by a fixed bit pattern called the generator
3. The remainder of this division is the **CRC value**
4. CRC is appended and then checked at reciever
5. If the bit-string including the CRC value is exactly divisible by generator pattern then all is fine!

Data Communication Method	Error Correction Technique	Explanation
Serial links (UART, RS-232)	Parity checking	Detects single-bit errors
Ethernet	CRC (Cyclic Redundancy Check)	Detects burst errors efficiently
Wireless (Wifi, 4G/5G	CRC + FEC	The costlier the retransmission the stronger Forward Error Correction method is justified

9.3 MAC and ARP

Multiple Acces Links and Protocols

- Point-to-point link between Ethernet switch and host
- Broadcast (shared wire or medium)
  - Old-fashioned Ethernet
  - Industrial fieldbuses
  - 802.11 wireless LAN, 4G/4G. satellite

shared wire (e.g., cabled Ethernet)

shared radio: WiFi

- Core problem:
  - Single shared broadcast channel
  - Two or more simultaneous transmissions by nodes: interference
    - *Collision* if node receives two or more signals at the same time

- **Learning these principles is a self-study task!**
  Recommendation:
  - Go to: https://gaia.cs.umass.edu/kurose_ross/videos/6/
  - Watch the lecture video on "Multiple Access Links and Protocols" (redirects to youtube)

MAC Addresses and ARP

- Relevant for modern Ethernet

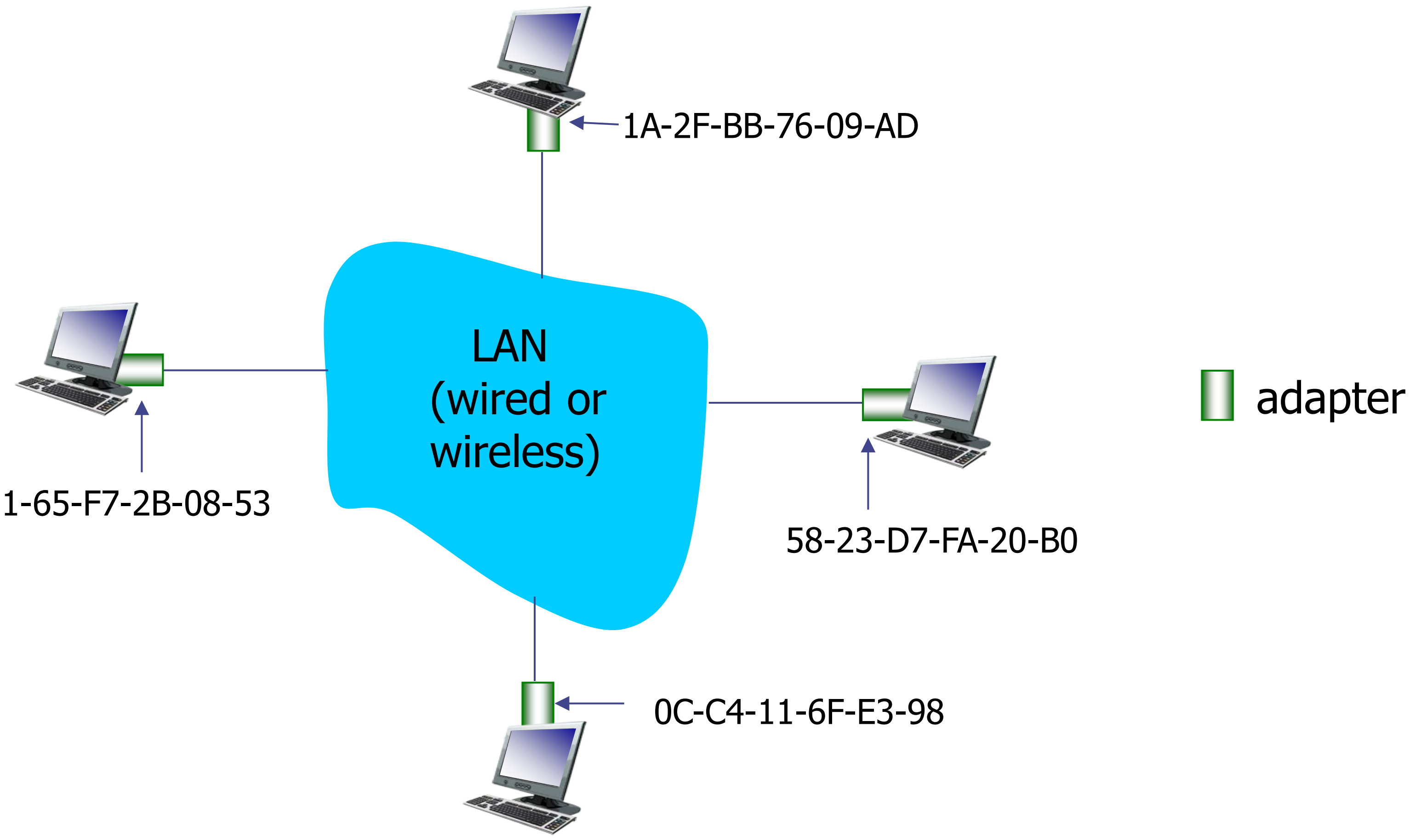
- 32-bit IP address:
  - network-layer address for interface
  - used for layer 3 (network layer) forwarding

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

hexadecimal (base 16) notation
(each “numeral” represents 4 bits)

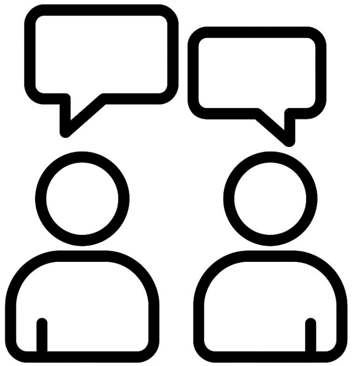
LAN Addresses and ARP

Each adapter on Local Area Network has unique **LAN** address



LAN Addresses (more)

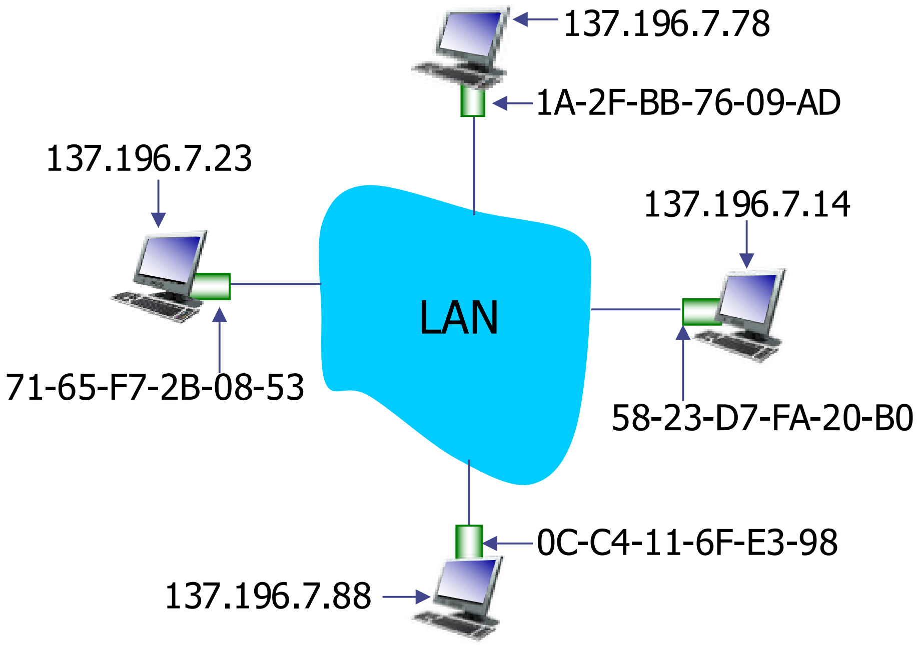
- MAC address allocation administered by IEEE
- Manufacturer buys portion of MAC address space (to assure uniqueness)
- Analogy:
  - MAC address: like Social Security Number
  - IP address: like postal address
- MAC flat address ➜ portability
  - Can move LAN card from one LAN to another



Why do we need both IP and MAC addresses?

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)

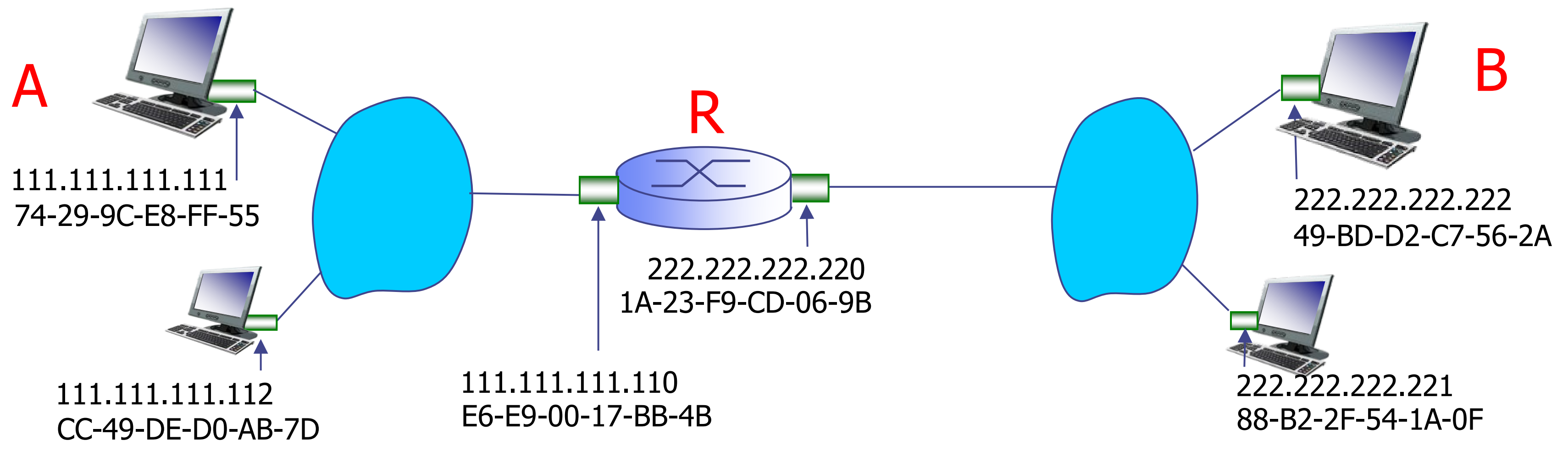
ARP Protocol: Same LAN

1. A wants to send datagram to B
   - B’s MAC address not in A’s ARP table.
2. A **broadcasts** ARP query packet, containing B's IP address
   - Destination MAC address = FF-FF-FF-FF-FF-FF
   - All nodes on LAN receive ARP query
3. B receives ARP packet, replies to A with its (B's) MAC address
   - Frame sent to A’s MAC address (unicast)
4. A caches IP/MAC address pair in its ARP table
   - Only until information becomes old → Soft state: information that times out (unless refreshed)
5. ARP is “plug-and-play”:
   - Nodes create their ARP tables without intervention from net administrator

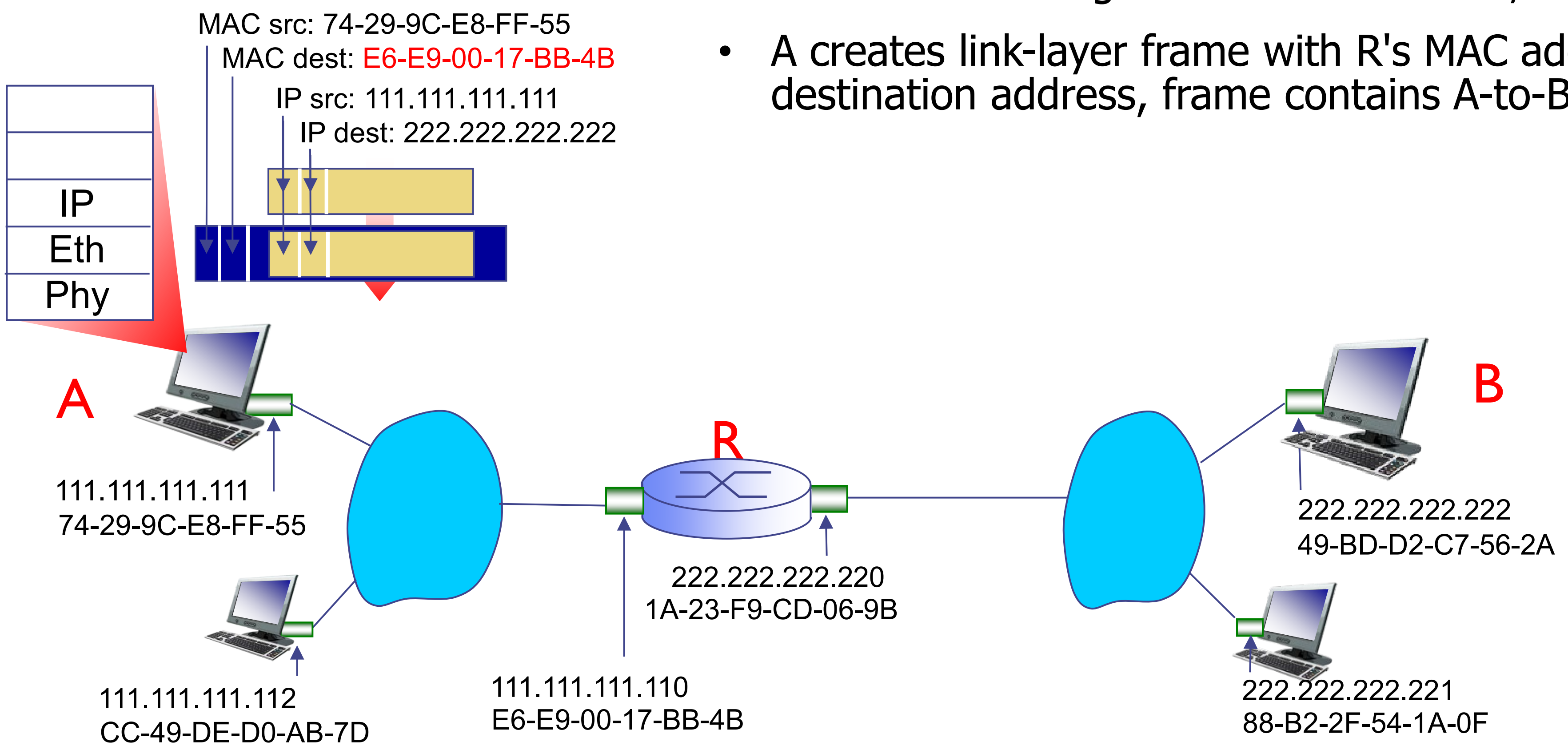
Addressing: Routing to Another LAN

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 the IP address of first hop router, R **(how?)**
- Assume A knows R's MAC address **(how?)**



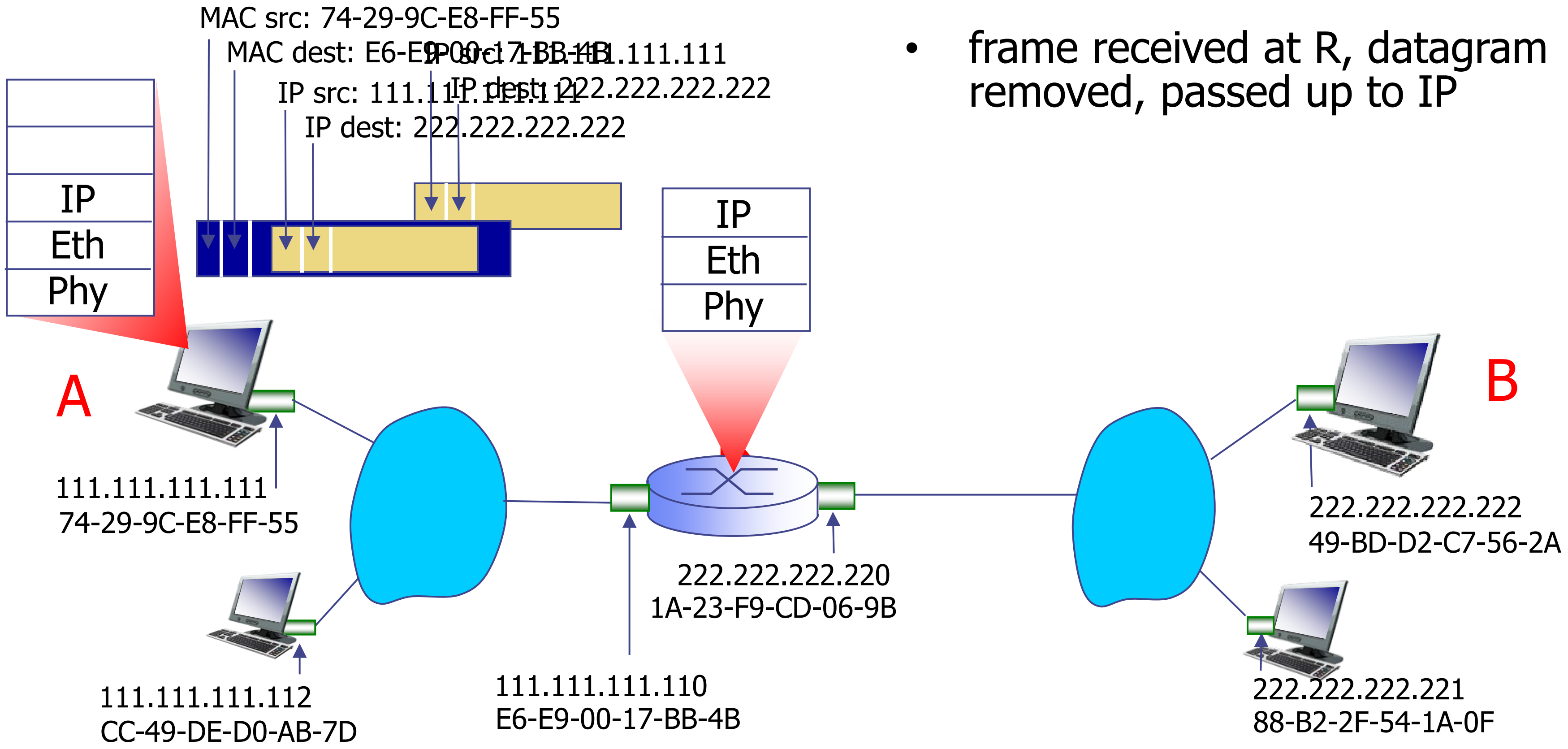
Addressing: Routing to Another LAN



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

Addressing: Routing to Another LAN

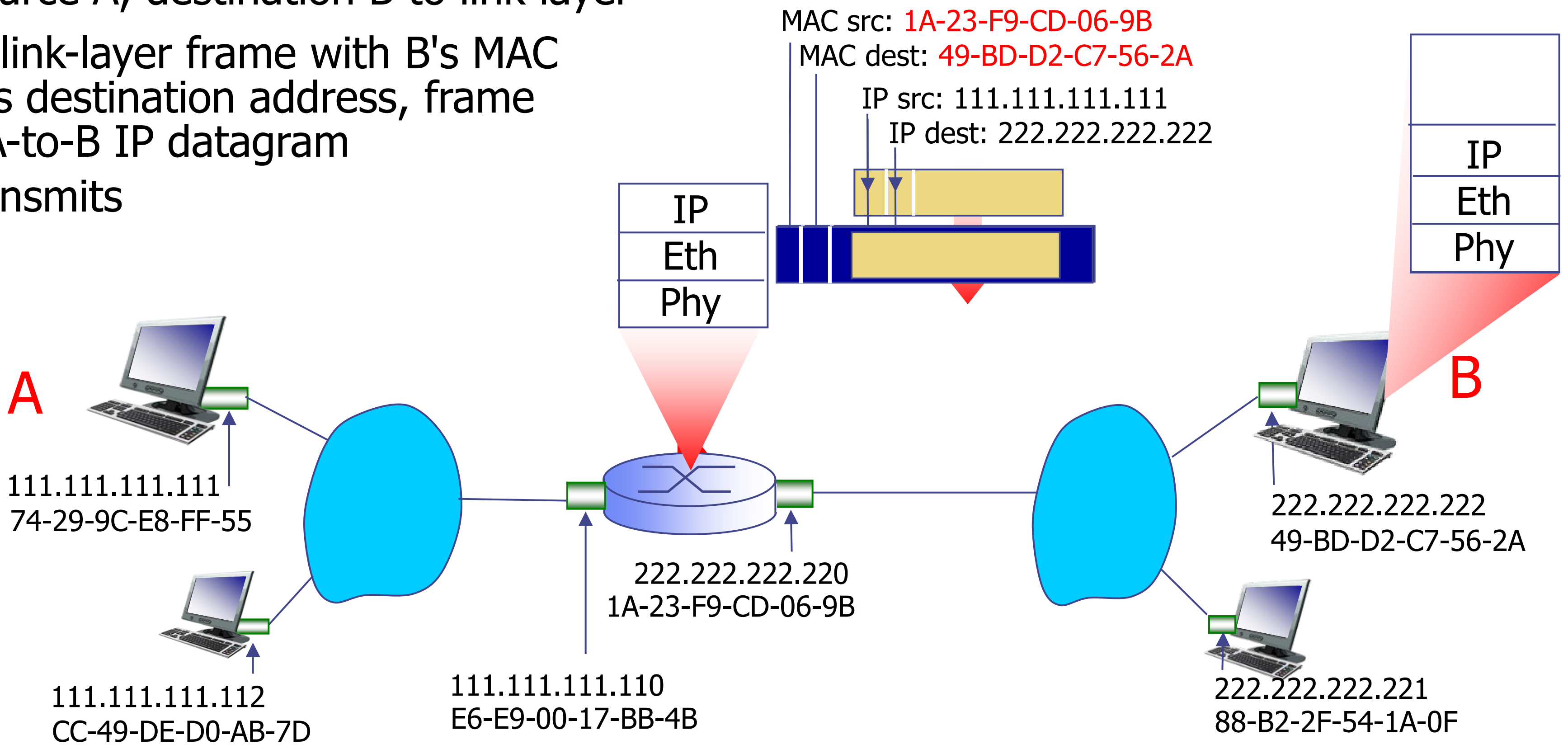
- frame sent from A to R
- frame received at R, datagram removed, passed up to IP



SDU

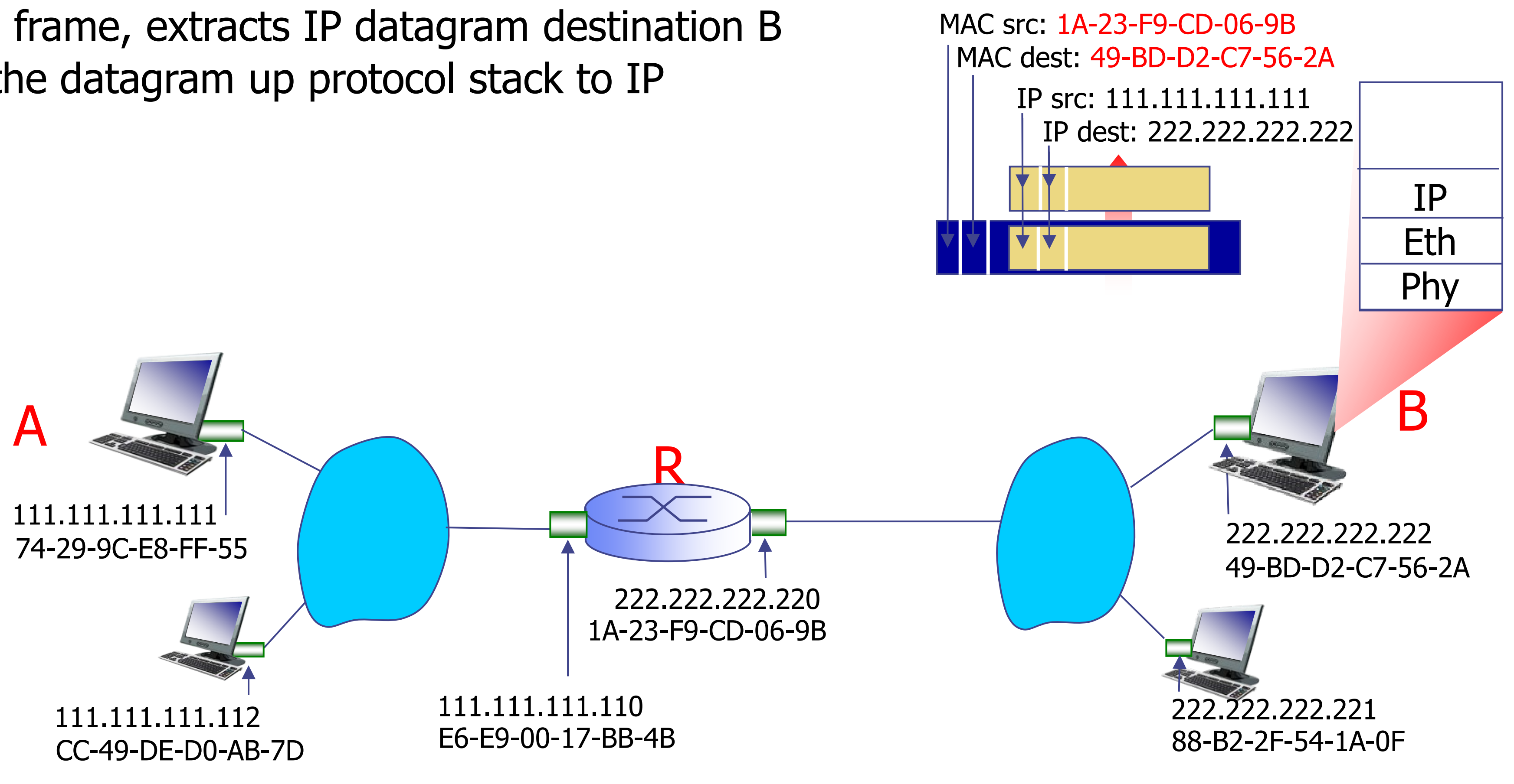
Addressing: Routing to Another LAN

- R determines outgoing interface, passes datagram with IP source A, destination B to link layer
- R creates link-layer frame with B's MAC address as destination address, frame contains A-to-B IP datagram
- ... and transmits



Addressing: Routing to Another LAN

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

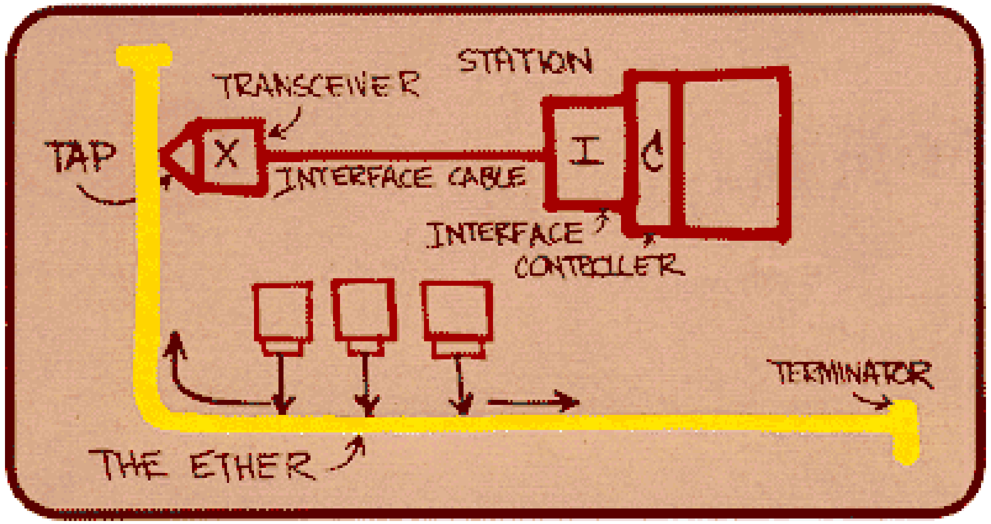


* Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/interactive/

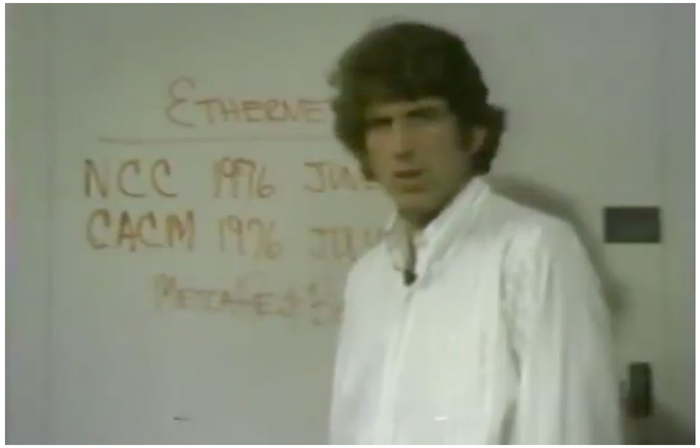
Ethernet

"Dominant" wired LAN technology:

- Single chip, multiple speeds (e.g., Broadcom BCM5761)
- First widely used LAN technology
- Simpler, cheap
- Kept up with speed race: 10 Mbps – 10 Gbps



Metcalfe's Ethernet sketch (mid 1970's)

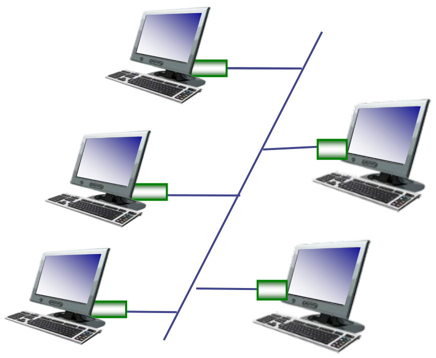


https://youtu.be/Fj7r3vYAjGY?si=eOIdK1R_5FRcdRcY

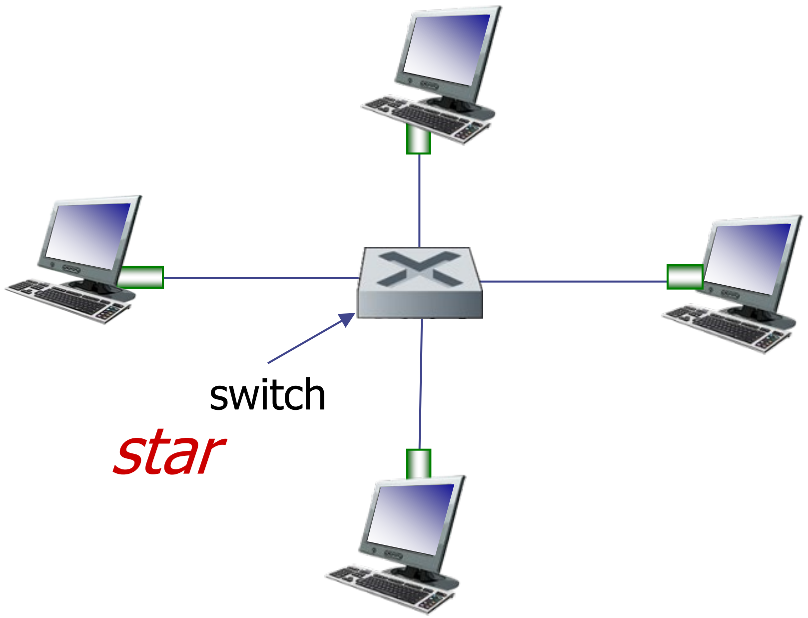
Ethernet: Physical Topology

- **Bus:** popular through mid 90s
  - All nodes in same collision domain (can collide with each other)

- **Star:** prevails today
  - Active **switch** in center
  - Each connection point runs a (separate) Ethernet protocol (nodes do not collide with each other)



bus: coaxial cable



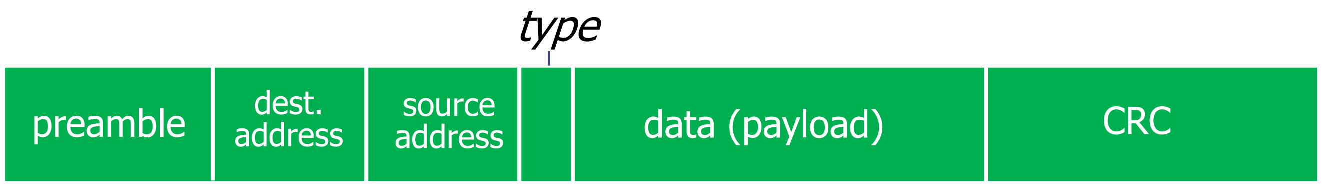
star

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

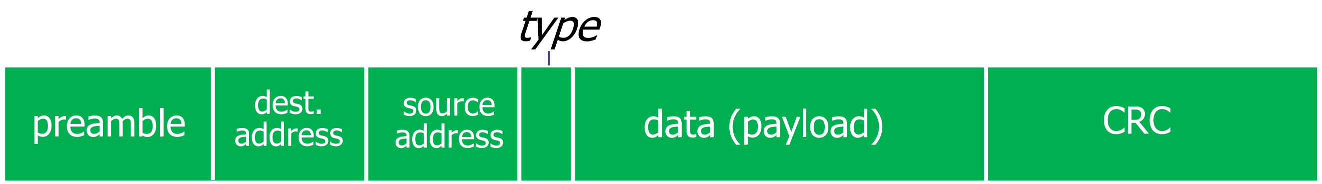


Ethernet Frame Structure (more)

- **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 but others possible, e.g., ARP)

- **CRC:** cyclic redundancy check at receiver
  - Error detected: frame is dropped



Ethernet: Unreliable, Connectionless

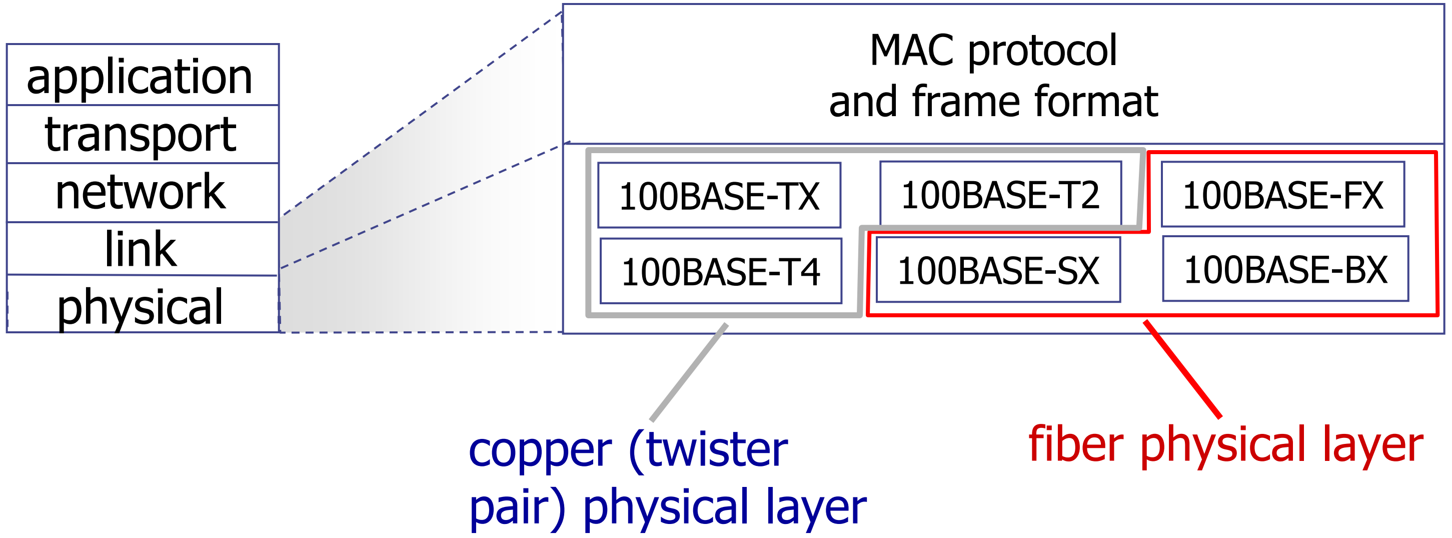
- **Connectionless:** no handshaking between sending and receiving NICs
- **Unreliable:** receiving NIC doesn't send acks or nacks to sending NIC
    - Data in dropped frames recovered only if initial sender uses higher layer *reliable data transfer* (e.g., TCP), otherwise dropped data lost

- Ethernet's MAC protocol: unslotted **CSMA/CD with backoff**
    - **However:** In modern Access control is handled by switches
        - Separate physical connections
        - MAC forwarding tables

802.3 Ethernet Standards: Link & Physical layers

Many different Ethernet standards

- Common MAC protocol and frame format
- Different speeds: 2 Mbps, 10 Mbps, 100 Mbps, 1Gbps, 10 Gbps, 40 Gbps (cable)
- Different physical layer media: fiber, cable
- Twisted pair cables naming: CATx (newest is CAT8)

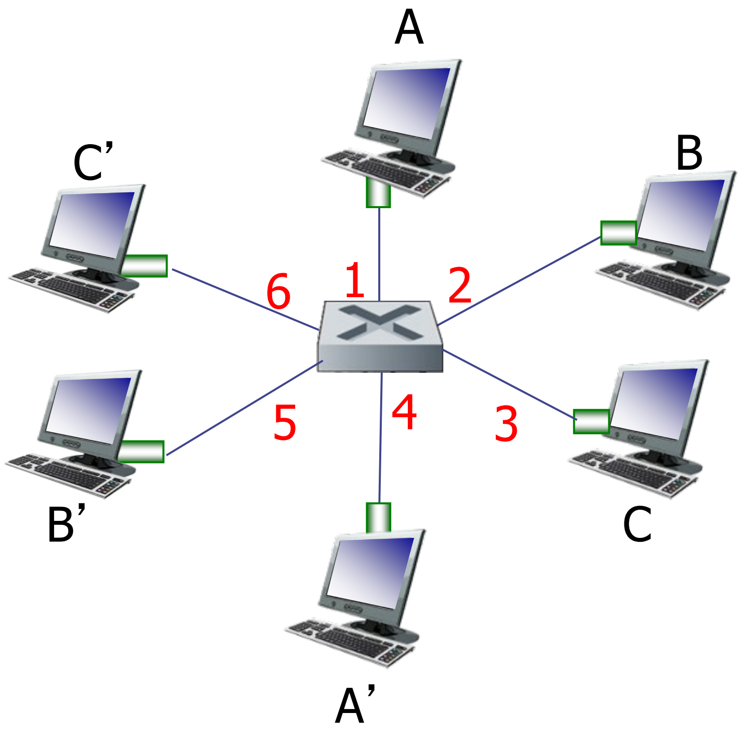


Ethernet Switch

- **Link-layer device**
  - 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
- **Transparent**
  - Hosts are unaware of presence of switches
- **Plug-and-play, self-learning**
  - Switches do not need to be configured **(unless they are Managed switches)**

Switch: Multiple Simultaneous 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’ can transmit simultaneously, without collisions



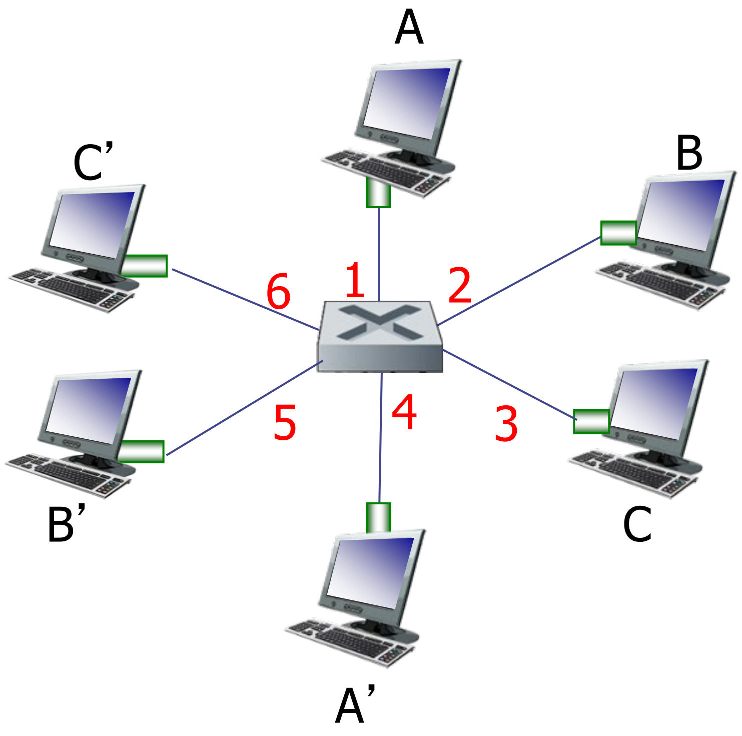
switch with six interfaces (1,2,3,4,5,6)

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/forwarding table!



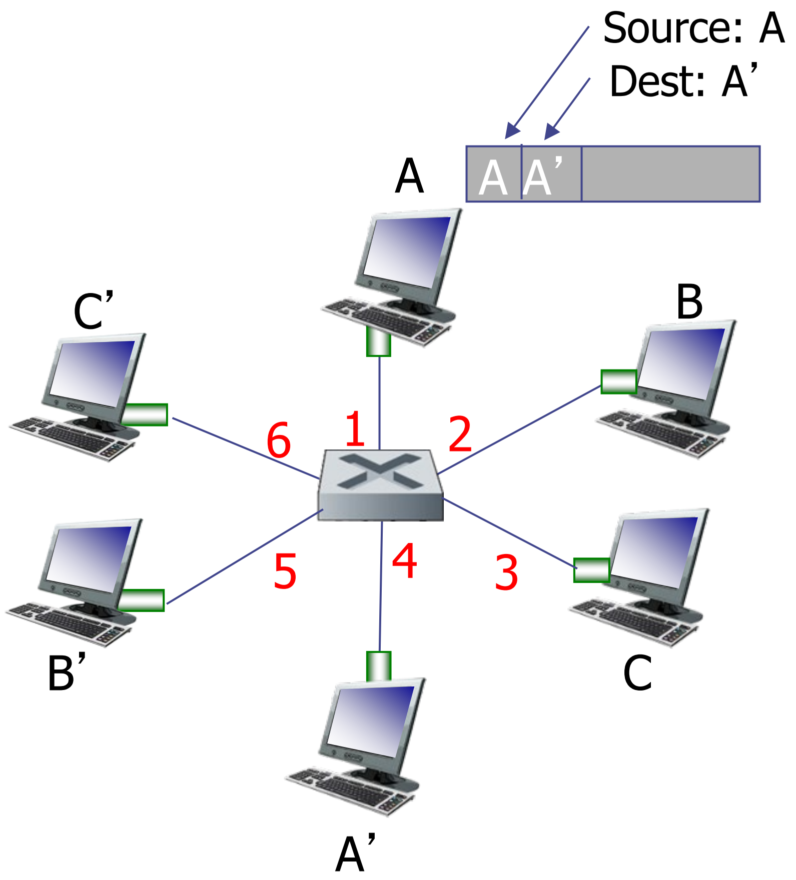
switch with six interfaces (1,2,3,4,5,6)

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

- something like a routing protocol?

Switch: Self-learning

- The 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

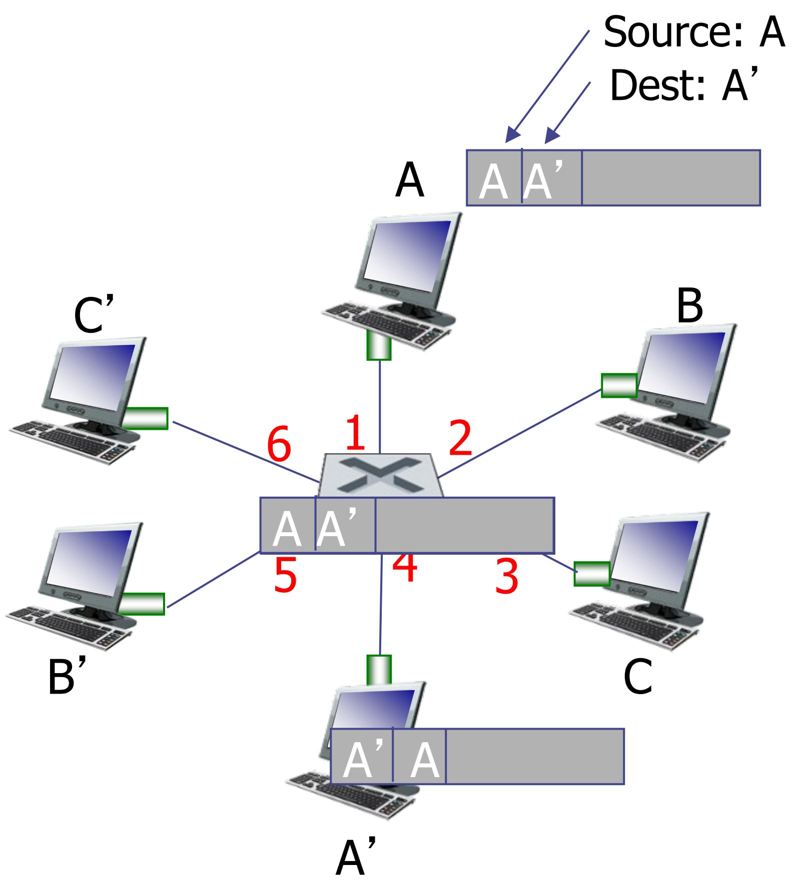
When a frame is received at switch:

```
1. Record incoming link, MAC address of sending host
2. Index switch table using MAC destination address
3. if entry found for the 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 */
```

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)

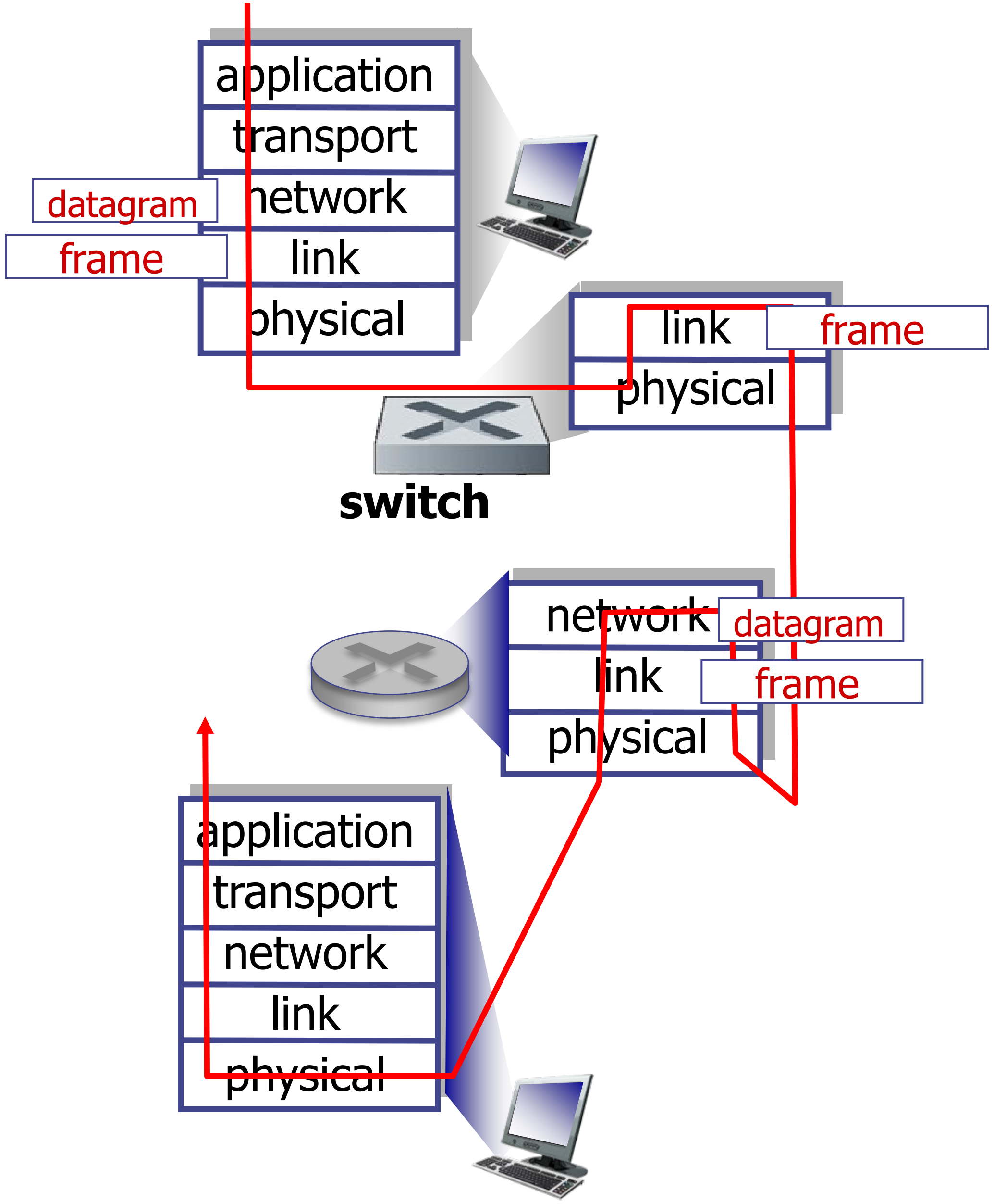
Switches vs. Routers

Both are store-and-forward:

- *Routers:* network-layer devices (examine network-layer headers)
- *Switches:* link-layer devices (examine link-layer headers)

Both have forwarding tables:

- *Routers:* compute tables using routing algorithms → learning IP addresses
- *Switches:* learn forwarding table using flooding → learning MAC addresses



Next week is about Wireless and Mobile Networks.
We will cover parts of chapter 7 in the book (page 561-627)
Consider reading after the lecture.

Do the interactive exercises:

- ERROR DETECTION AND CORRECTION: TWO DIMENSIONAL PARITY
  - Note: Columns first, rows second
- LINK LAYER (AND NETWORK LAYER) ADDRESSING AND FORWARDING

Do Wireshark Lab 8.