

Computer Networks

Lecture 08

Data-Link Layer

By
Dr. Khaled ELMenshawy

Lecture

08

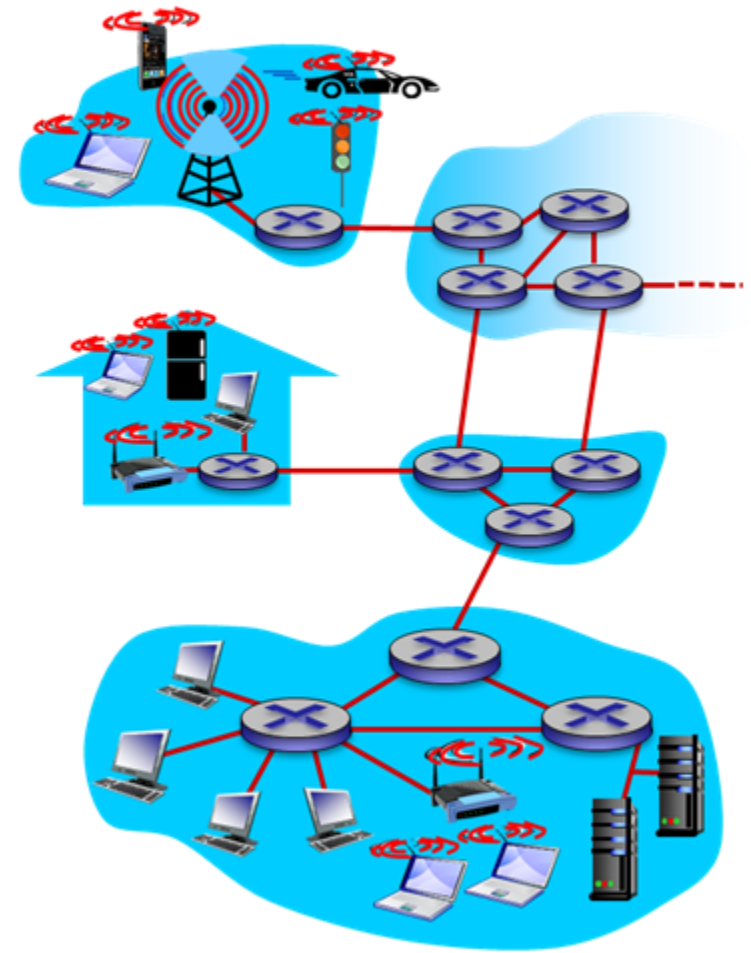
LEARNING OUTCOMES

- Understand link layer functions and services .
- Understand link layer services.
- Recognize where the link layer implemented.
- Understand error detection.
- Differentiate between the types of error detection (Parity checking and CRC).
- Define MAC and Differentiate between the types of MAC protocols (Slotted ALOHA, Pure ALOHA, CSMA/CD, CSMA/CA, and taking turns).
- Understand address resolution protocol and how it works.
- Understand Ethernet and recognize ethernet frame structure.
- Understand Switches and VLANs.

Overview of Link layer

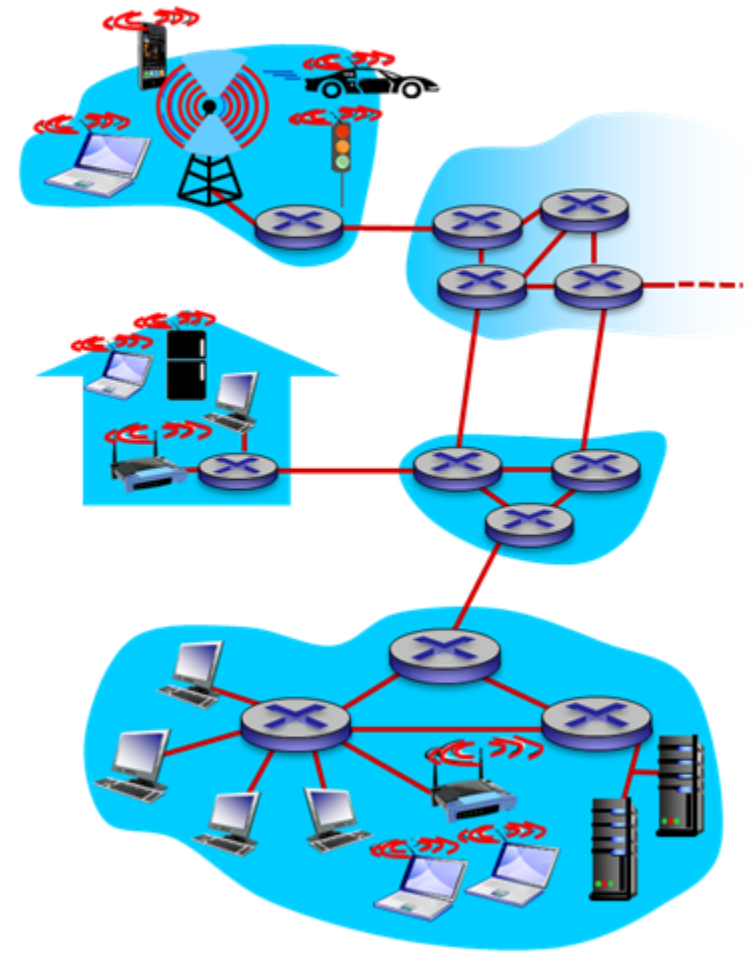
What is the Data-Link layer?

- It is responsible for transferring a datagram from one node to a physically adjacent node over a link.
- Layer-2 encapsulates packet (Datagram) into the frame, adding a header and trailer.
- “MAC” addresses are used in frame headers to identify the source and destination.
- It ensures error-free transmission of information (FCS).

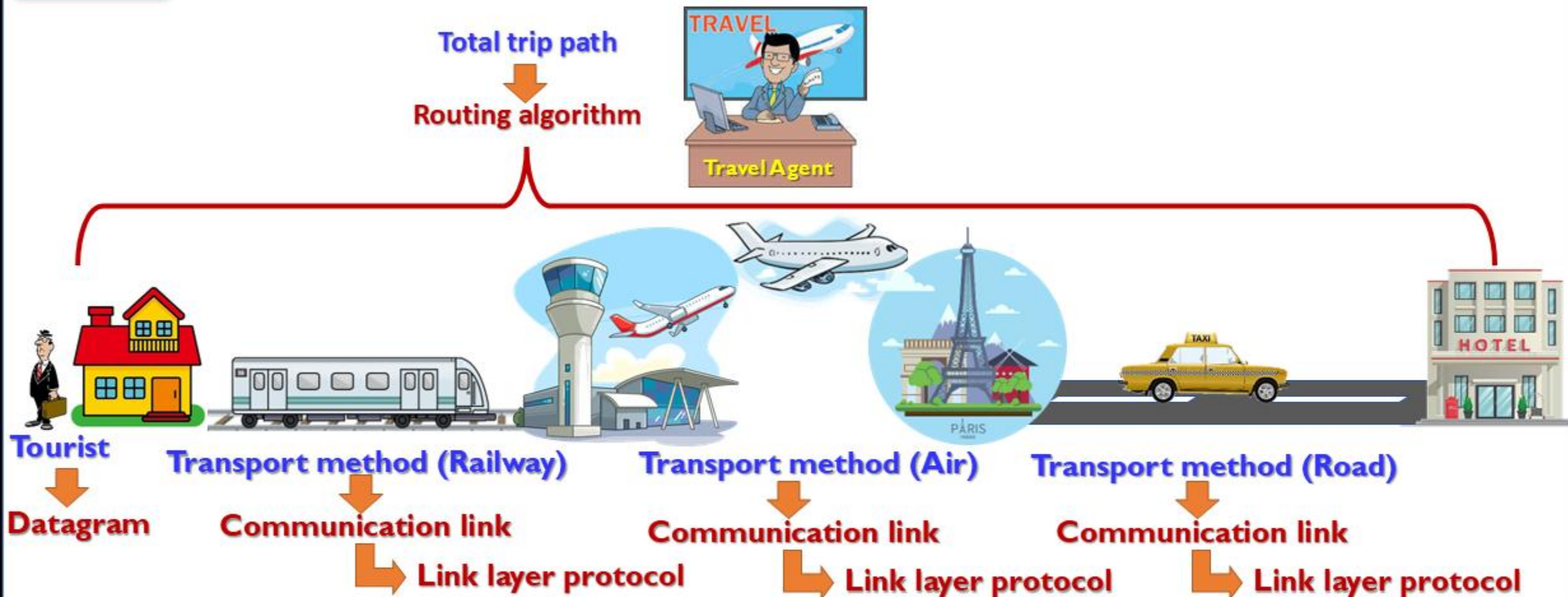


Terminology:

- Hosts and routers: **Nodes**
- Communication channels that connect adjacent nodes along communication path: **Links**
 - Wired links
 - Wireless links
 - LANs
- Layer-2 packet: **Frame, Encapsulates Datagram**



Overview of Link layer



Overview of Link layer

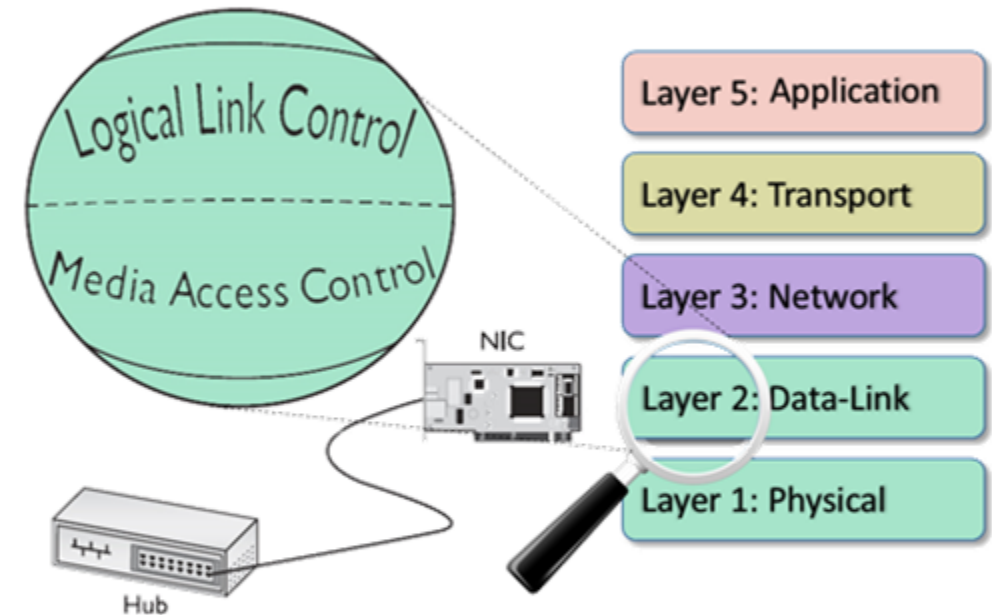
Data-Link layer

Logical Link Control (LLC)

- It talks to the system's operating system (usually via device drivers).
- It deals with multiplexing, acknowledgments, flow control and error detection.

Media Access Control (MAC)

- It provides Physical addressing (MAC addressing), Multiple access methods (CSMA/CD, CSMA/CA), LAN switching (packet switching), and Virtual LANs (VLAN).



➤ **Framing, link access:**

- Encapsulate datagram into frame, adding header, trailer.
- Channel access if shared medium (e.g., wireless medium).
- “MAC” addresses used in frame headers to identify source, destination.

➤ **Reliable delivery between adjacent nodes:**

- Seldom used on low bit-error link (fiber, some twisted pair).
- Wireless links: high error rates.

➤ **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:**

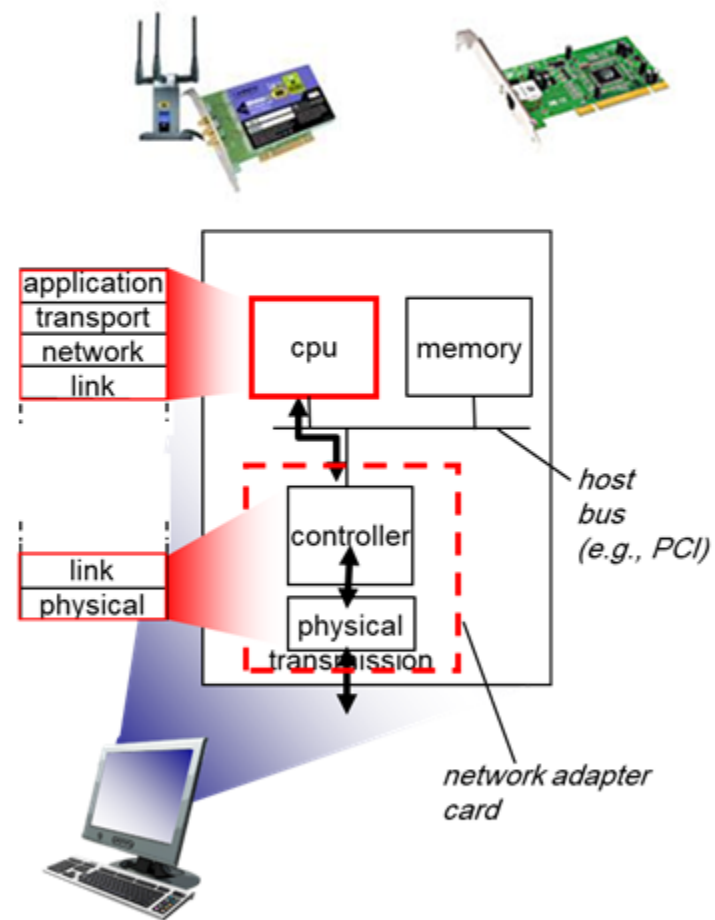
- Receiver identifies and *corrects* bit error(s) without resorting to retransmission.

➤ **Half-duplex and full-duplex:**

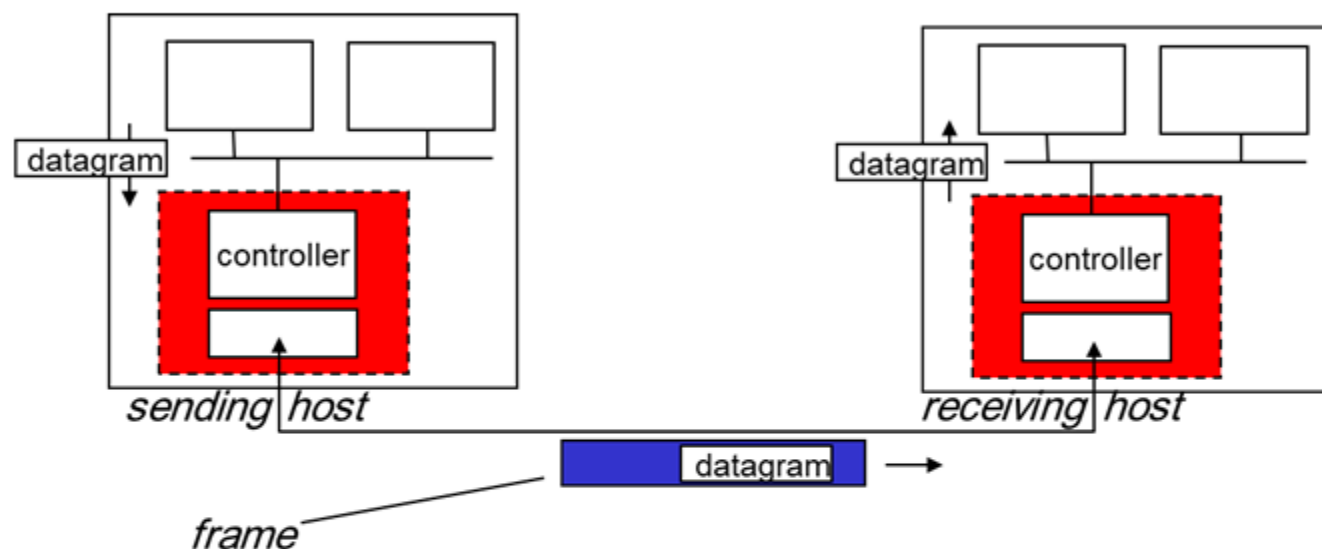
- With half duplex, nodes at both ends of link can transmit, but not at same time.

Where is the link layer implemented?

- In each and every host.
- 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



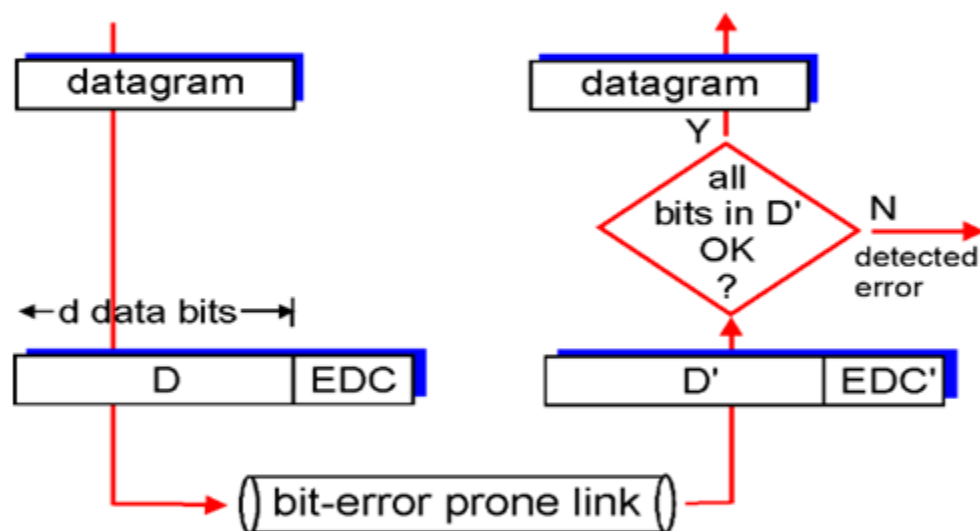
➤ Sending side:

- 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.

- EDC = Error Detection and Correction bits (redundancy)
- D = Data protected by error checking, may include header fields
- Error detection not 100% reliable!
 - Larger EDC field yields better detection and correction



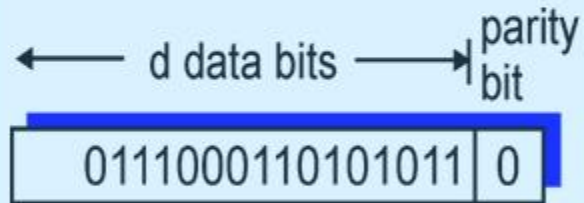
Parity checking

- **XOR Gate:**
Used in Parity checking

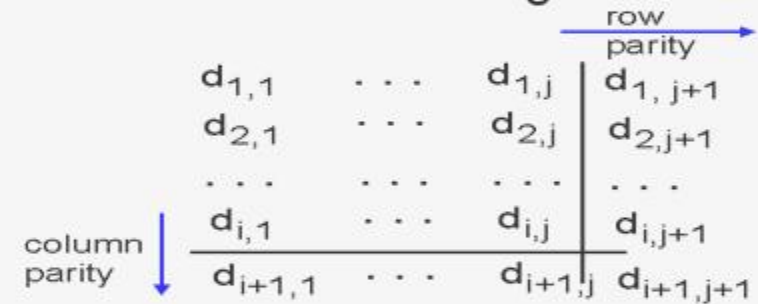


X	Y	Z
0	0	0
1	0	1
0	1	1
1	1	0

- **Single bit parity:**
detect single bit errors



- **Two-dimensional bit parity:**
detect and correct single bit errors



```

1 0 1 0 1 | 1
1 1 1 1 0 | 0
0 1 1 1 0 | 1
-----
0 0 1 0 1 | 0
no errors

```

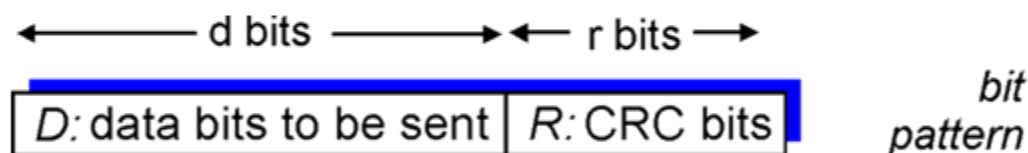
```

1 0 1 0 1 | 1
1 0 1 1 0 | 0 parity error
0 1 1 1 0 | 1
-----
1 0 1 0 1 | 0 parity error
correctable
single bit error

```

Cyclic redundancy check

- More powerful error-detection coding.
- View data bits = D (as a binary number).
- Generator = G (choose $R+1$ bit pattern, based on reducible polynomial).
- CRC bits = R bits (e.g., $R=32$ bits in Ethernet and 8 bits in Bluetooth) $R=G-1$.
- $\langle D, R \rangle$ exactly divisible by G .
 - Receiver knows G , divides $\langle D, R \rangle$ by G . If non-zero remainder: error detected
- Widely used in practice (Ethernet, 802.11 Wi-Fi, ATM)



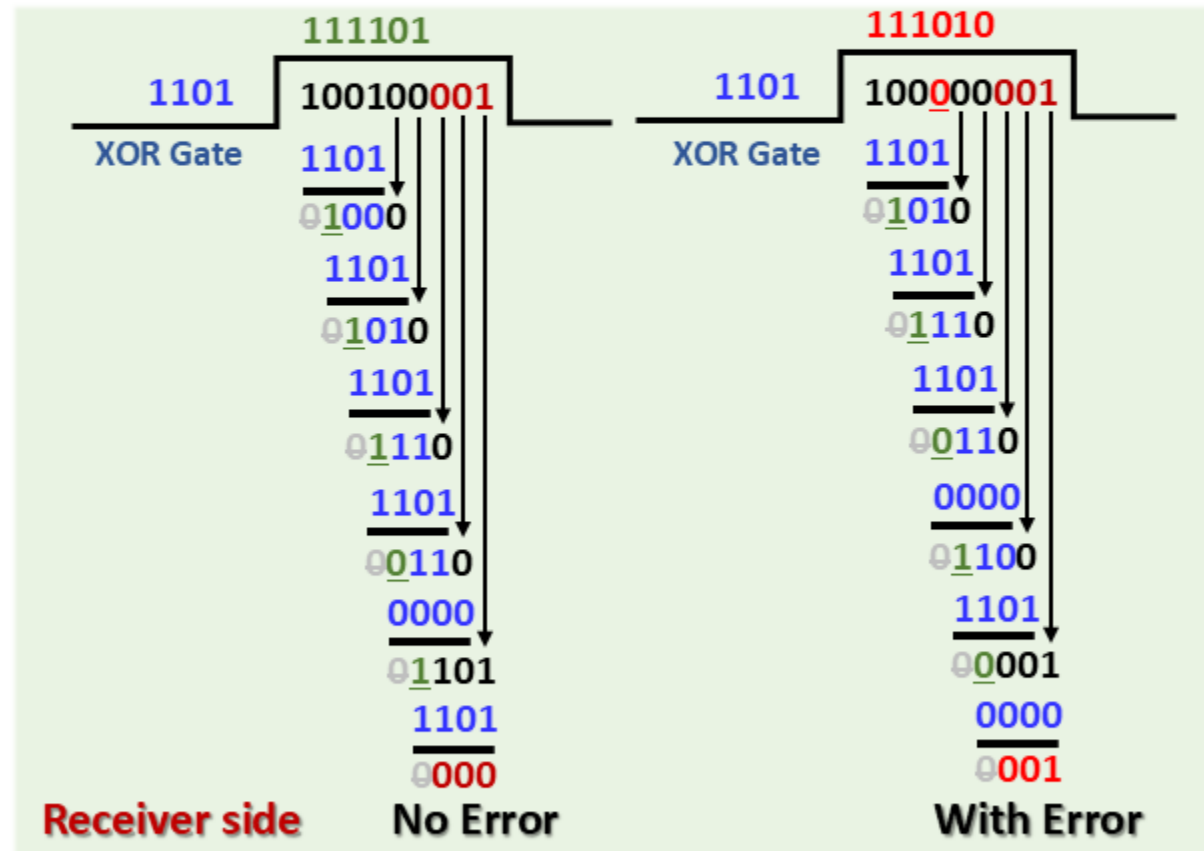
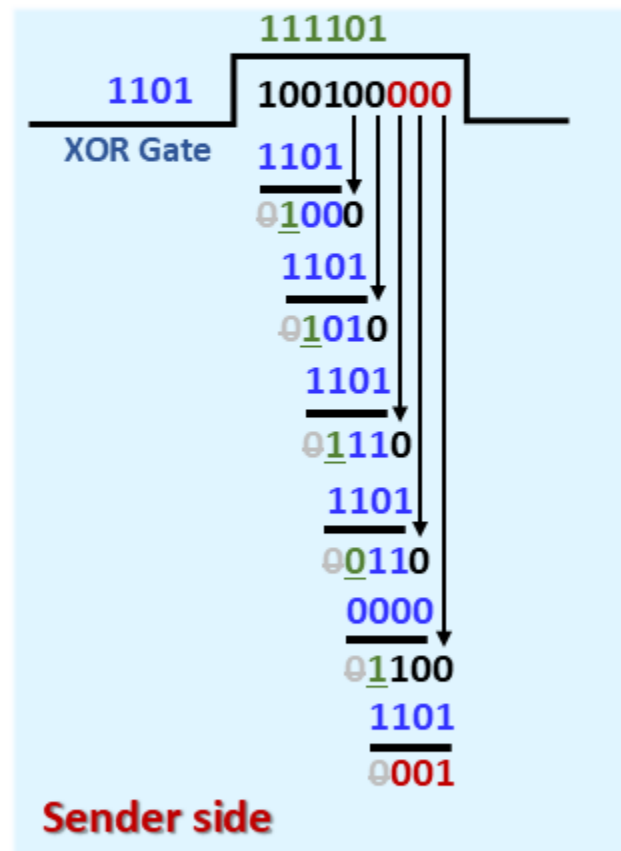
Cyclic redundancy check

➤ Data to be sent:

- 100100

➤ Generator polynomial:

- $x^3 + x^2 + 1$
 - Bit-Length= 4 bits
 - Bit-Value= 1101

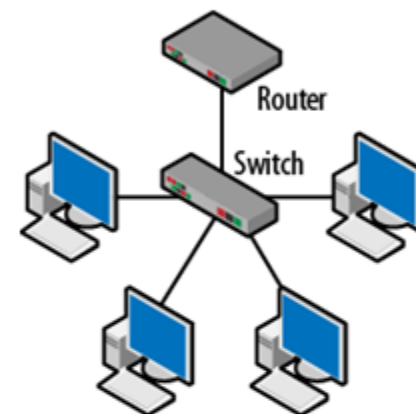
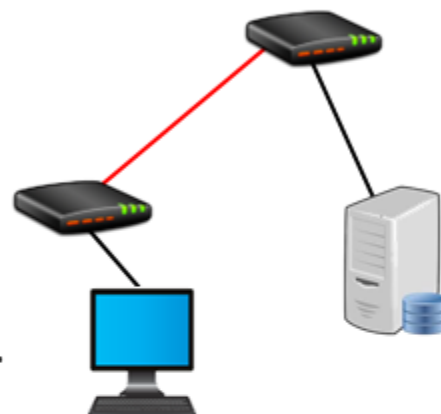
➤ $R = G - 1 = 4 - 1 = 3$ bits

Multiple access links, protocols

Two types of "links"

Point-to-Point

- Point-to-Point Protocol (PPP) for dial-up access.
- Point-to-Point link between Ethernet switch, host.



Shared Medium BROADCAST

- Hybrid fiber coax (HFC)
- 802.11 wireless LAN.
- **Collision** if node receives two or more signals at the same time.



Shared Wire
(e.g., cabled Ethernet)



Shared RF
(satellite)



Shared RF
(e.g., 802.11 WiFi)



Humans at a party
(shared air, acoustical)

➤ **Multiple Access Protocol:**

- Distributed algorithm that determines how nodes share channel, i.e., determine when node can transmit.
- Communication about channel sharing must use channel itself.
 - *No out-of-band channel for coordination.*

➤ **An ideal multiple access protocol:**

Given: broadcast channel of rate R bps

1. When one node wants to transmit, it can send at rate R .
2. When M nodes want to transmit, each can send at average rate R/M
3. Fully decentralized:
 - No special node to coordinate transmissions.
 - No synchronization of clocks, slots.
4. Simple



Shared RF
(e.g., 802.11 WiFi)



Shared Wire
(e.g., cabled Ethernet)

MAC protocols: taxonomy

Three broad classes:

➤ **Channel partitioning:**

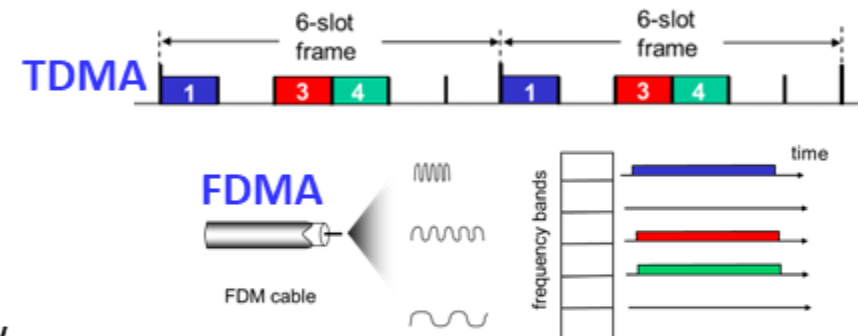
- Divide channel into smaller “pieces” (time slots, frequency, code).
- Allocate piece to node for exclusive use.

➤ **Random access:**

- Channel not divided (transmit at full channel data rate R), allow collisions.
- MAC protocol specifies how to detect and recover from collisions (e.g., slotted ALOHA, ALOHA, CSMA, CSMA/CD, CSMA/CA).

➤ **“taking turns”:**

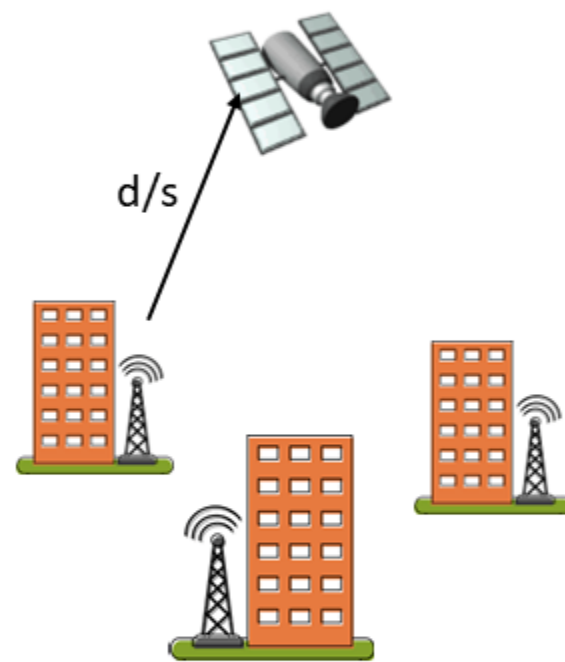
- Nodes take turns, but nodes with more to send can take longer turns.



MAC protocols: Slotted ALOHA

Assumptions:

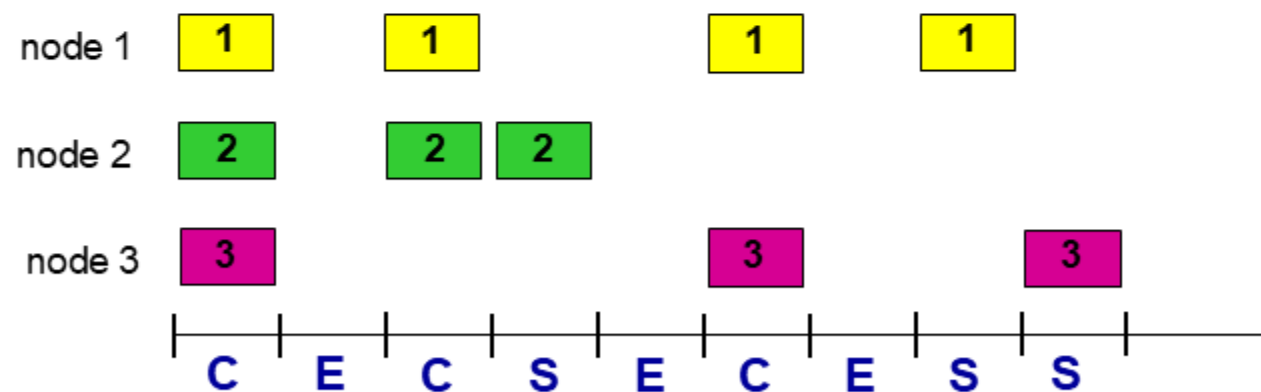
- All frames divided into equal size (L).
- Time divided into equal size slots (time to transmit 1 frame $T_{\text{trans.}} = L/R$ where L is frame length and R is the data rate).
- All nodes are Promiscuous mode enabled (can send and receive all frames including their frames).
- Nodes start to transmit only slot beginning ($T_{\text{trans.}} \geq T_{\text{prop.}}$, $T_{\text{prop.}} = 2d/s$ where d is distance to satellite and s is the speed of light).
- Nodes are synchronized
- If 2 or more nodes transmit in slot, all nodes detect collision.



Promiscuous mode enabled nodes

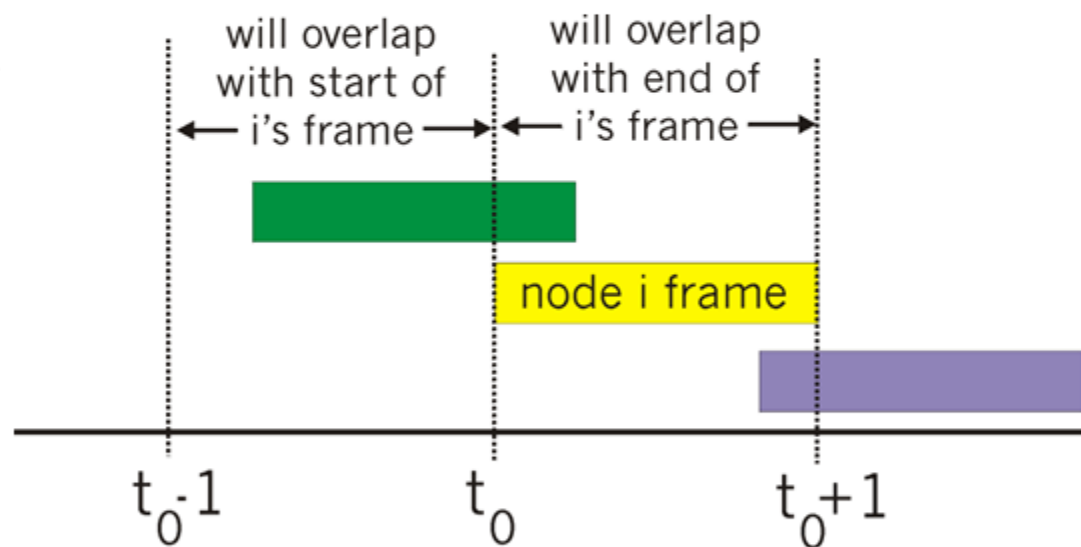
MAC protocols: Slotted ALOHA**Operation:**

- When node obtains fresh frame, transmits in next slot.
- If no collision: node can send new frame in next slot.
- If collision: node retransmits frame in each subsequent slot with prob. P until success, at best: channel used for useful transmissions 37% of time.



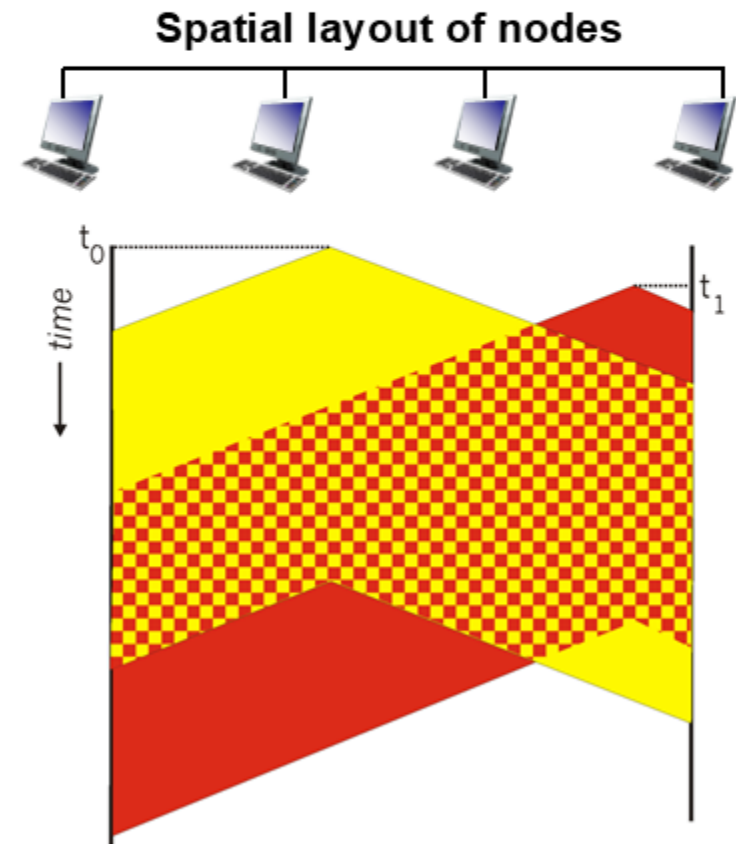
MAC protocols: Pure (Unslotted) ALOHA

- Unslotted Aloha: simpler, no synchronization.
- When frame first arrives transmit immediately.
- Collision probability increases:
 - Frame sent at t_0 collides with other frames sent in $[t_0-1, t_0+1]$
 - At best: channel used for useful transmissions 18% of time, worse than slotted Aloha.



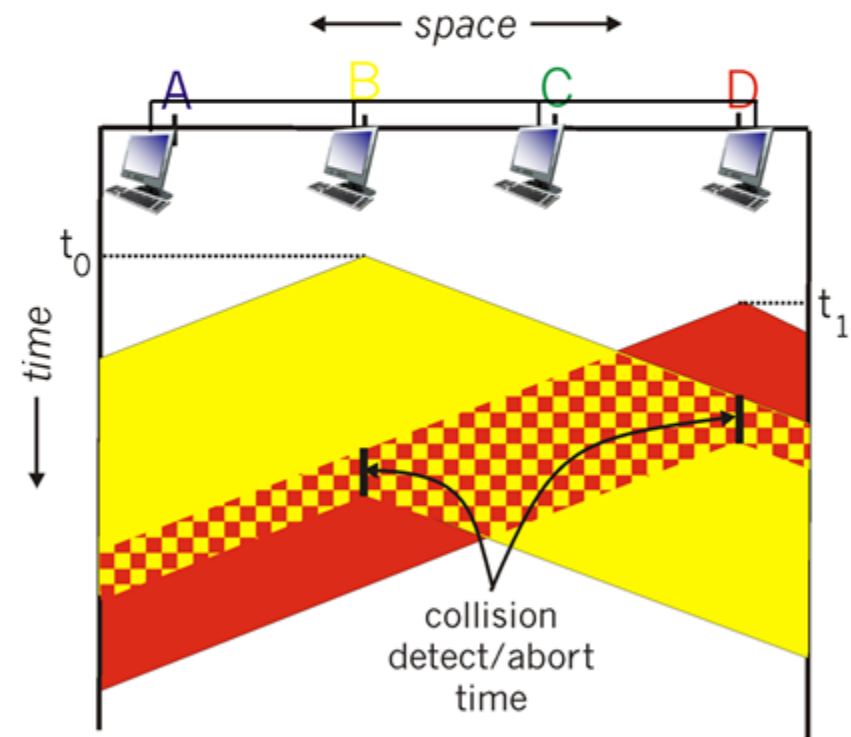
MAC protocols: CSMA (carrier sense multiple access)

- **CSMA**: listen before transmit:
 - If channel sensed idle: transmit entire frame.
 - If channel sensed busy, defer transmission.
- **Collisions can still occur**: propagation delay means two nodes may not hear each other's transmission.
 - **Collision**: entire packet transmission time wasted.



MAC protocols: CSMA /CD (Collision Detection)

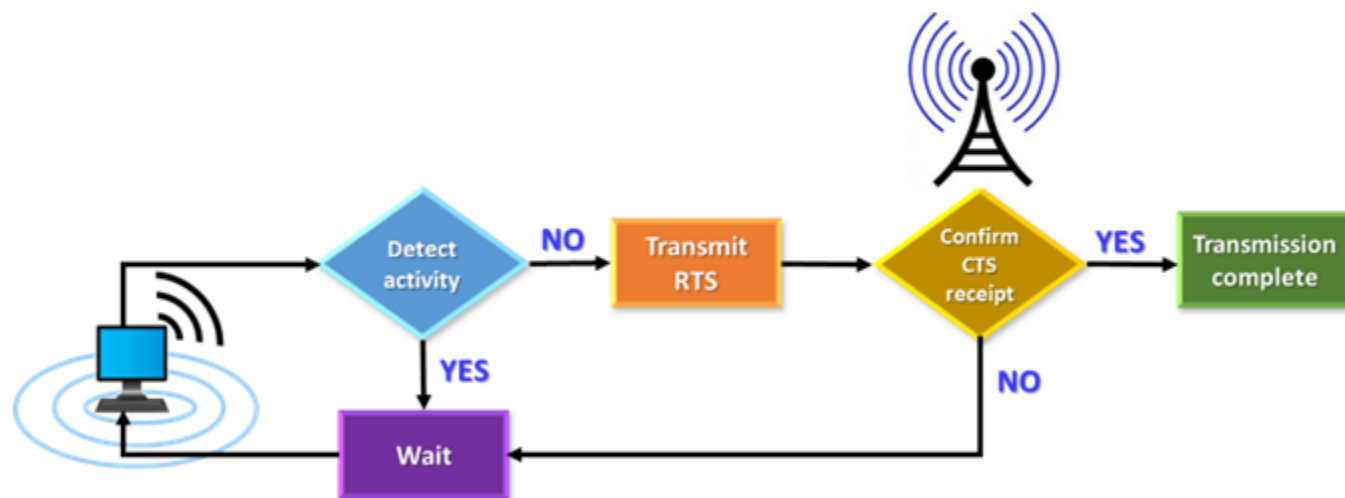
- **CSMA/CD**: listen before transmit (Promiscuous mode enabled):
 - If channel sensed idle: transmit entire frame.
 - If channel sensed busy, defer transmission.
- **Collisions can still occur**:
 - If NIC detects another transmission while transmitting, aborts and sends jam signal.
 - After aborting, NIC enters binary (exponential) backoff:
 - After m^{th} collision, NIC chooses K at random from $\{0, 1, 2, \dots, 2^m - 1\}$. NIC waits $K \times 512$ -bit times.
- Better performance than ALOHA: and simple, cheap, decentralized.



MAC protocols: CSMA /CA (Collision Avoidance)

➤ **CSMA/CA**: Widely used for wireless network:

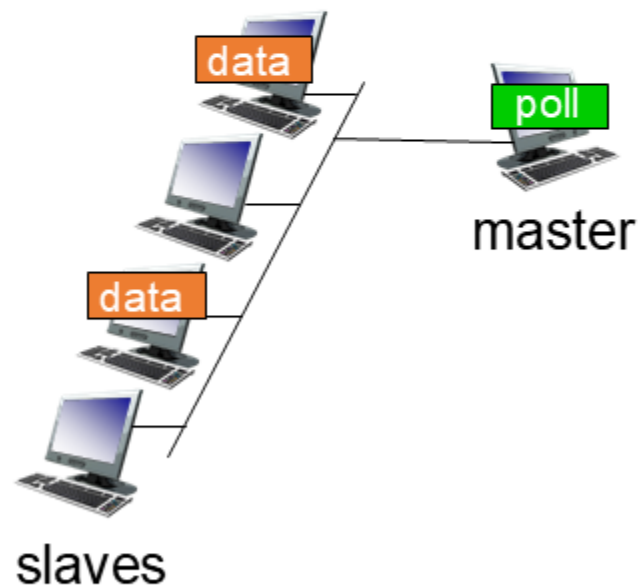
- If channel sensed busy, defer transmission.
- If channel sensed idle:
 - Sends *Request to Send (RTS)* message to AP.
 - If the AP sends busy message, the node waits a period of time called *InterFrame Space (IFS)* time.
 - If the AP sends, *Clear to Send (CTS)* message the node transmits the frames.



MAC protocols: "Taking turns"

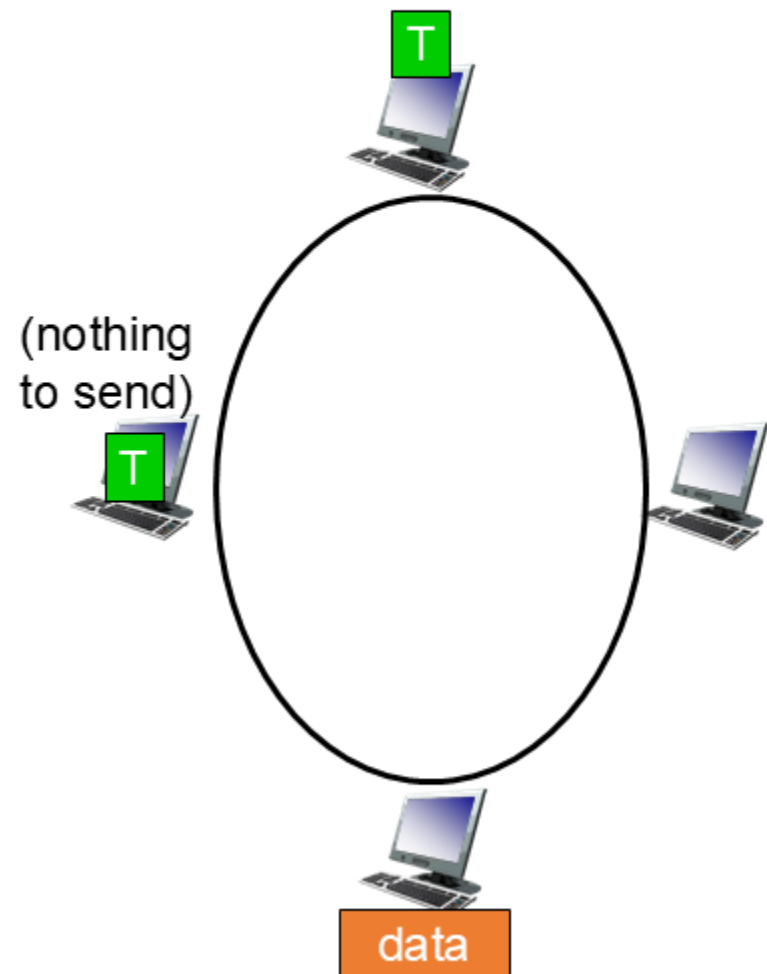
Polling:

- Master node "invites" slave nodes to transmit in turn.
- Typically used with "dumb" slave devices
- Concerns:
 - Polling overhead
 - Latency
 - Single point of failure (master)



MAC protocols: "Taking turns"Token passing:

- Control **token** passed from one node to next sequentially.
- The node that has the token can transmit.
- **Concerns:**
 - Token overhead
 - Latency
 - Single point of failure (token)

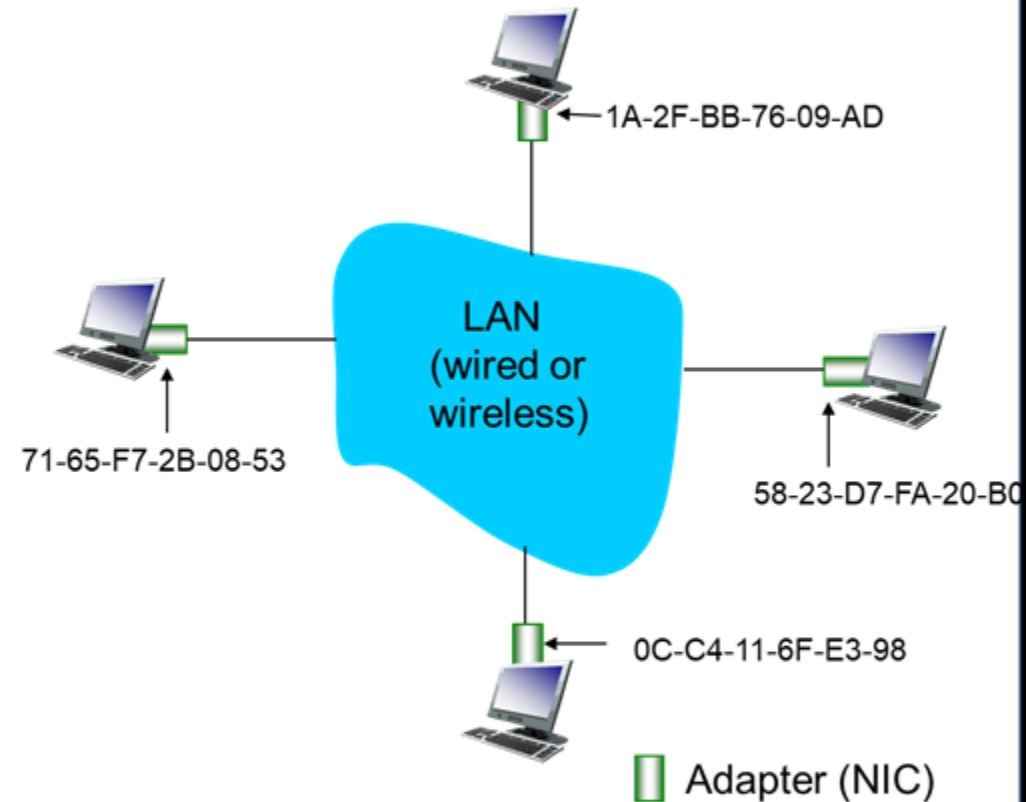


➤ **32-bit IP address:**

- *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 NIC ROM, also sometimes software settable. MAC address allocation administered by IEEE e.g.: 1A-2F-BB-76-09-AD

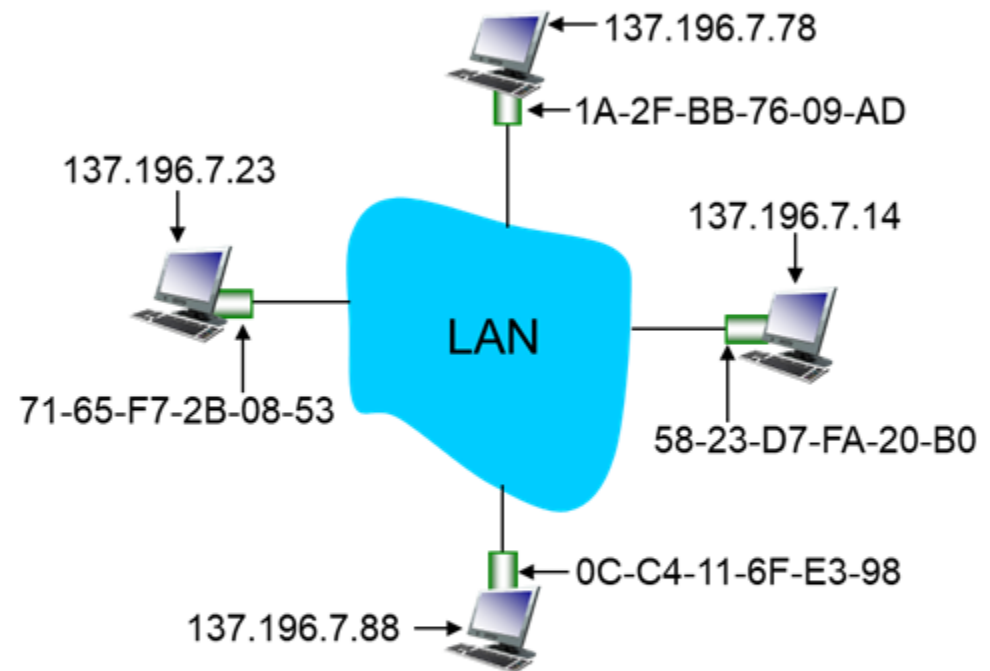


ARP table: each IP node (host, router) on LAN has table

- 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).



Question:

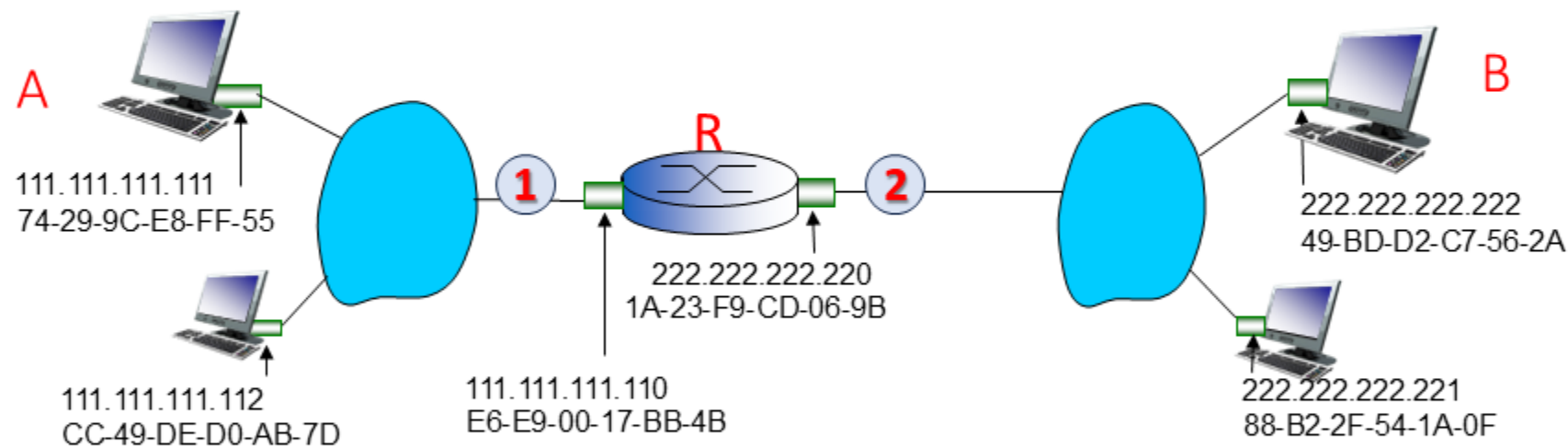
How to determine interface's MAC address, knowing its IP address?

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 (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.
5. **ARP is “plug-and-play”:**
 - Nodes create their ARP tables without intervention from net administrator.

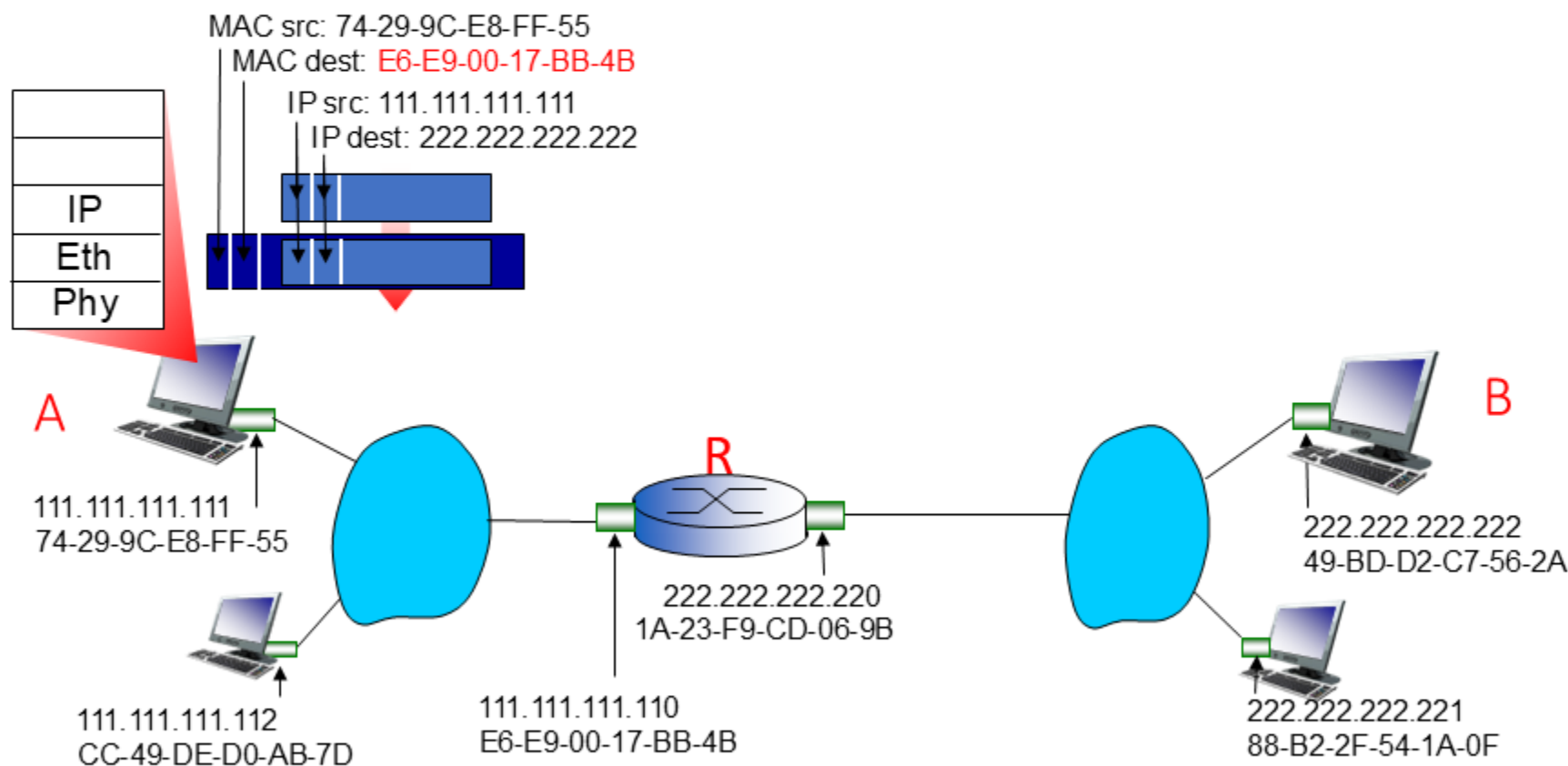
ARP protocol

ARP: address resolution protocol

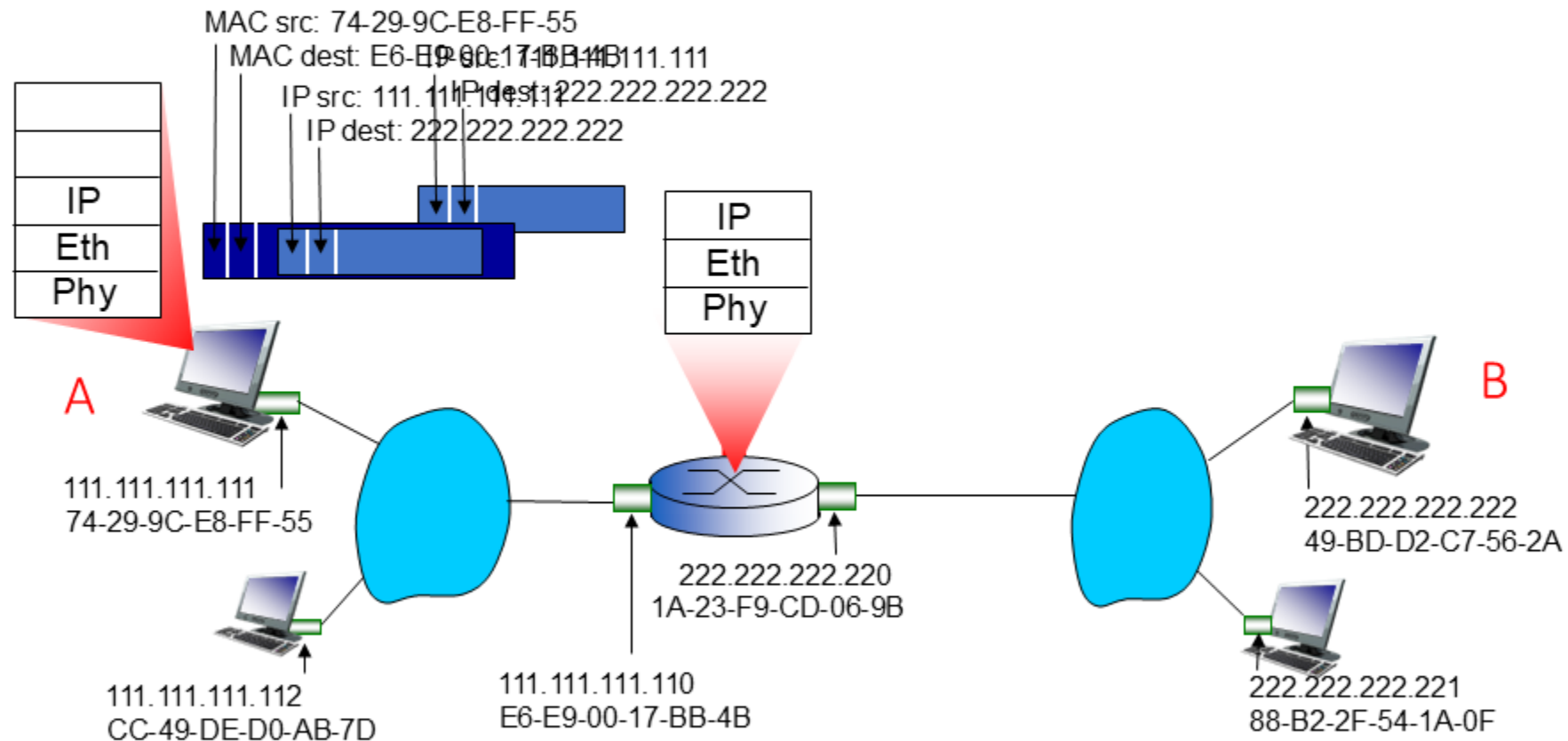
Send datagram from A to B via R



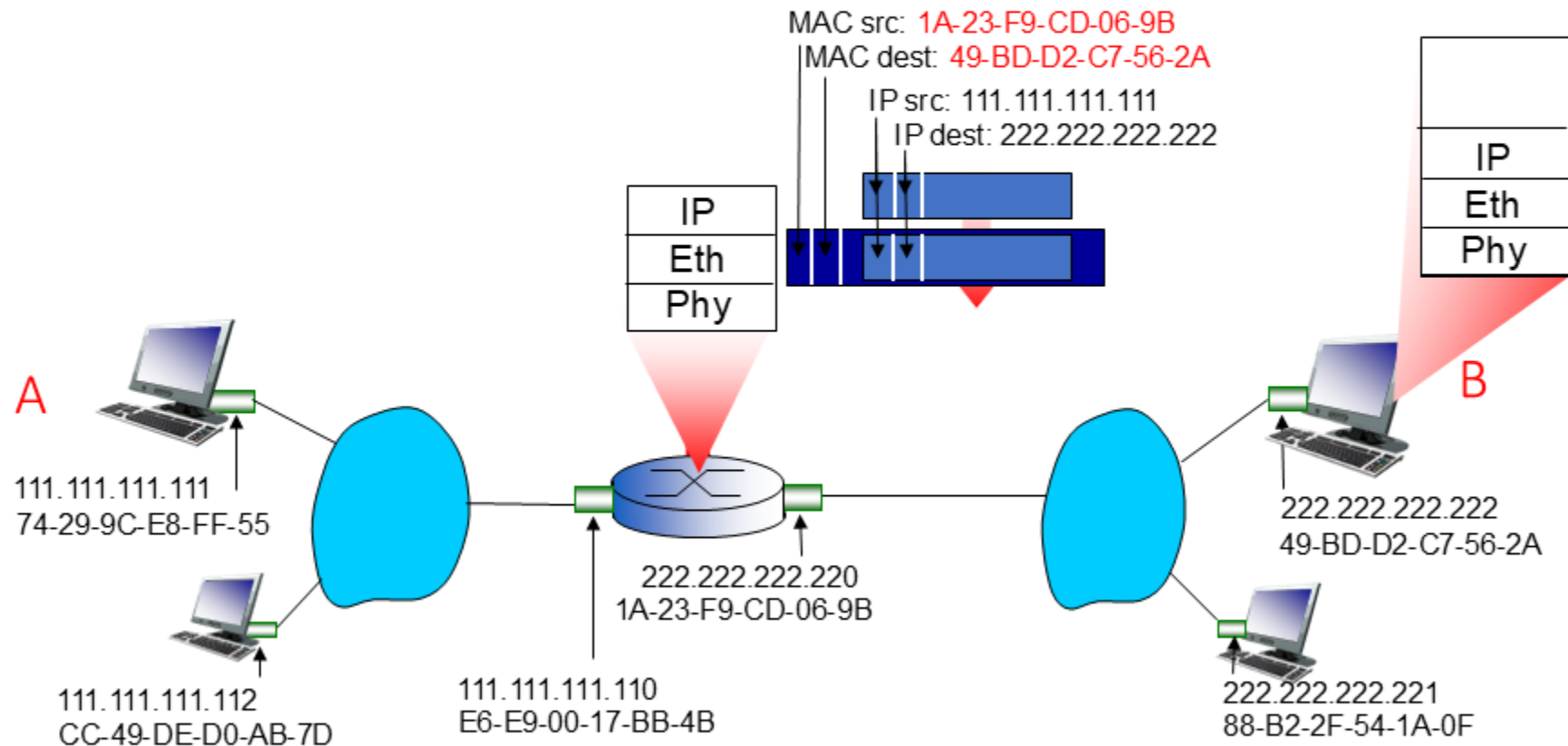
- **A** creates IP datagram with IP source **A**, destination **B**
- **A** creates link-layer frame with R's MAC address as destination address, frame contains **A-to-B** IP datagram.



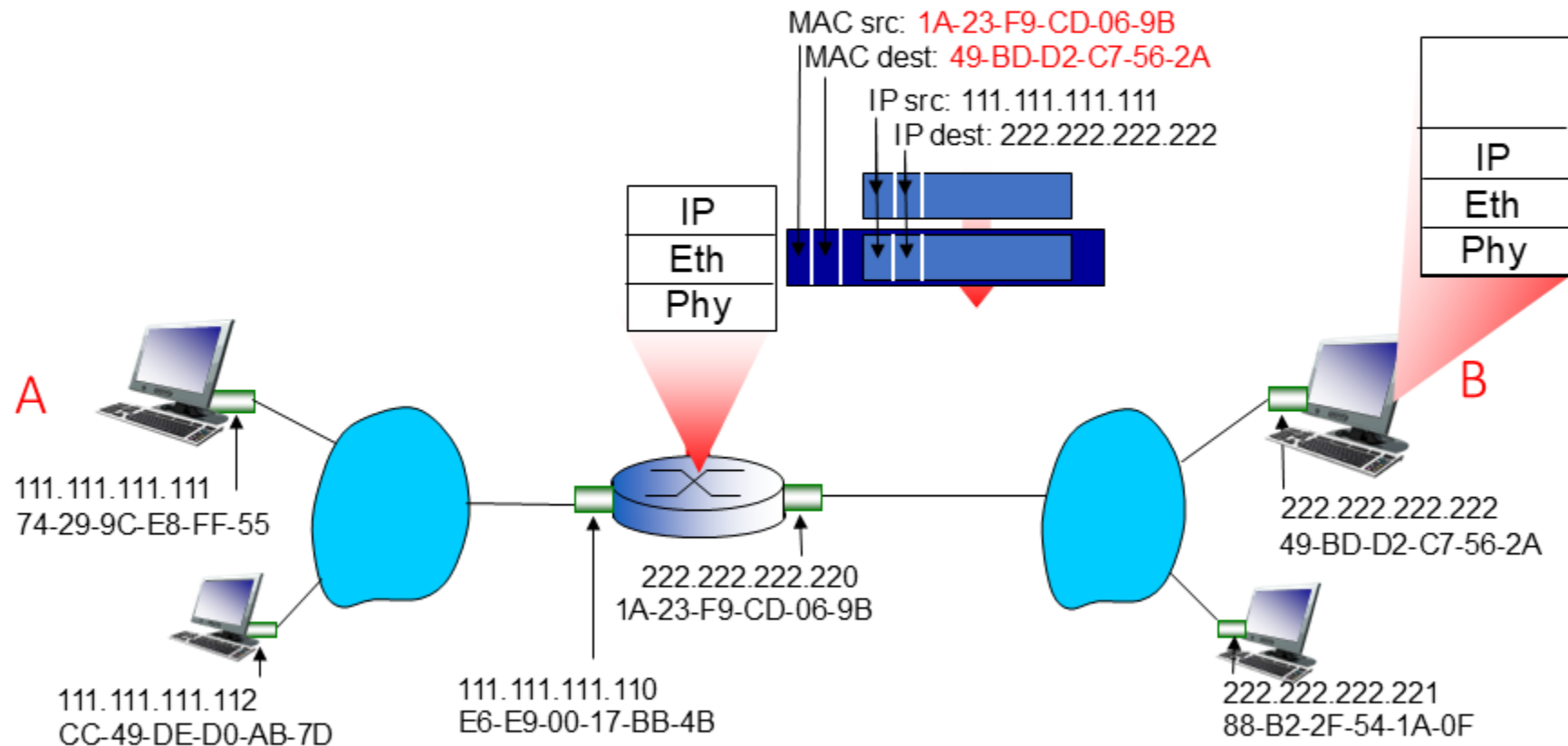
- Frame sent from **A** to **R**
- Frame received at **R**, datagram removed, passed up to IP.



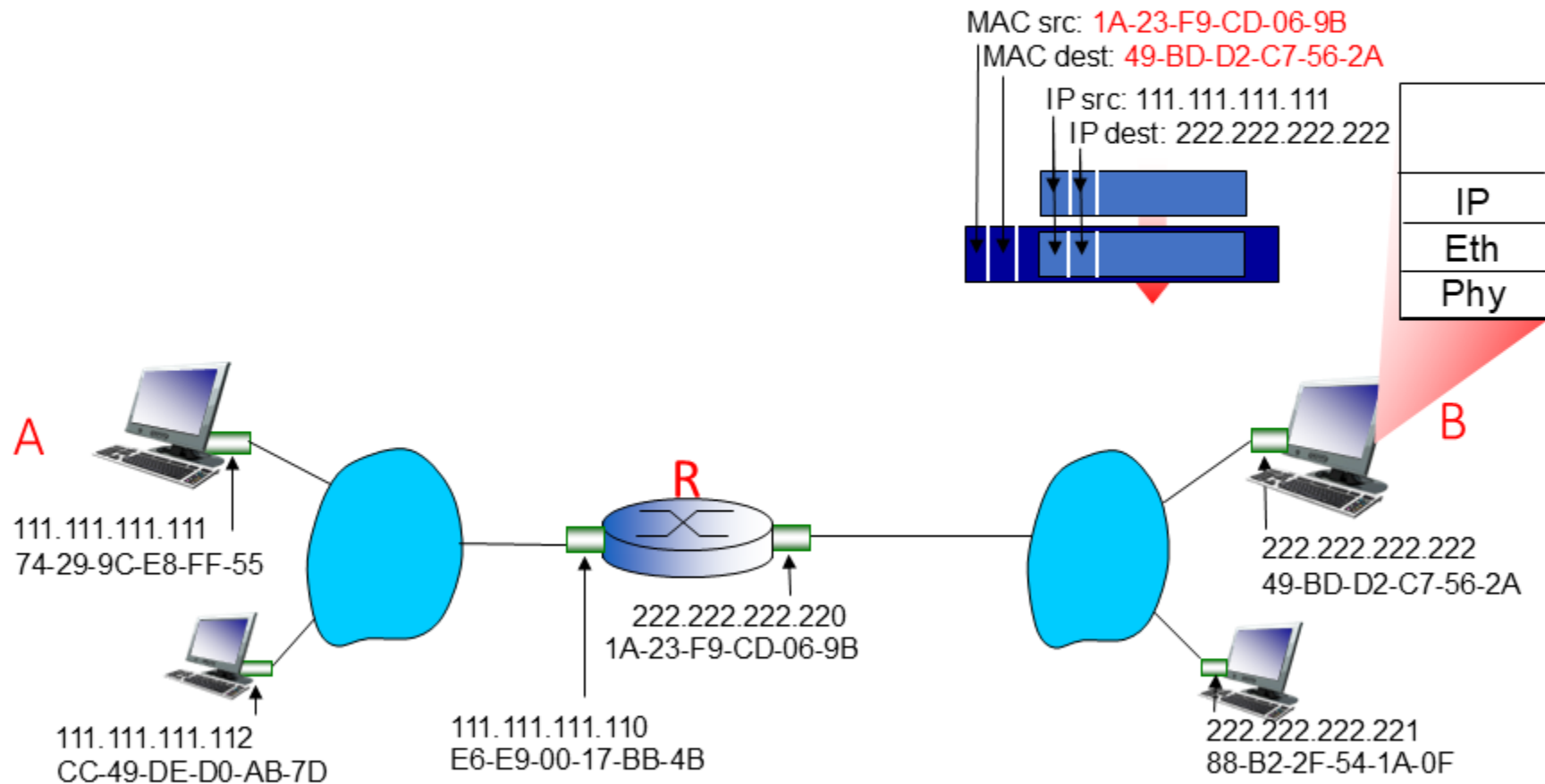
- **R** forwards datagram with IP source **A**, destination **B**
- **R** creates link-layer frame with **B**'s MAC address as destination address, frame contains **A-to-B** IP datagram.



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

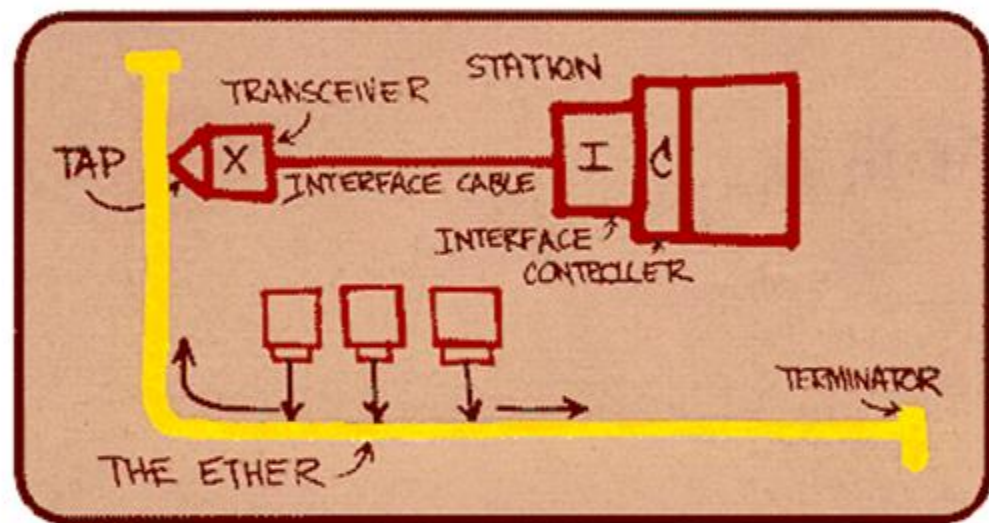


- **R** forwards datagram with IP source **A**, destination **B**
- **R** creates link-layer frame with **B**'s MAC address as destination address, frame contains **A-to-B** IP datagram.



“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 .
- Developed from BUS topology to other topologies such as star and hybrid.

*Metcalfe's Ethernet sketch*

Ethernet frame structure

**Preamble:**

- 7 bytes with pattern 10101010 followed by one byte with pattern 10101011
- Used to synchronize receiver, sender clock rates

Addresses: 6-byte source, destination MAC addresses

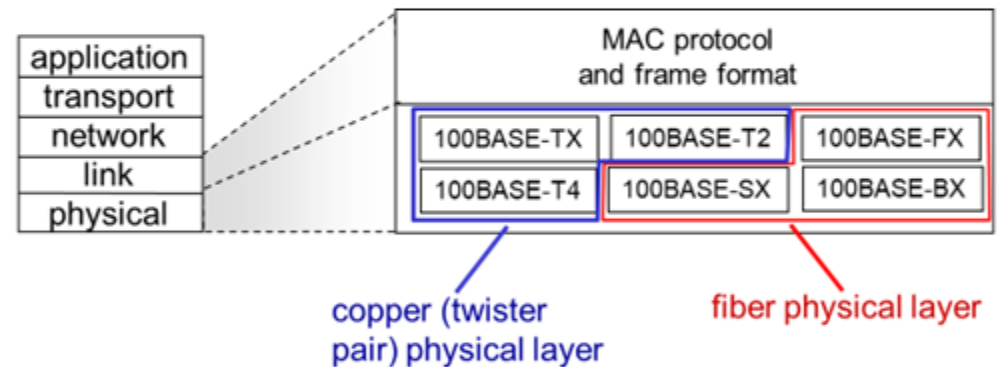
- If adapter receives frame with matching destination address, or with broadcast address (e.g., ARP packet), it passes data in frame to network layer protocol
- Otherwise, adapter discards frame

Type: indicates higher layer protocol (mostly IP but others possible, e.g., Novell IPX, AppleTalk).

CRC: cyclic redundancy check at receiver (error detected: frame is dropped).

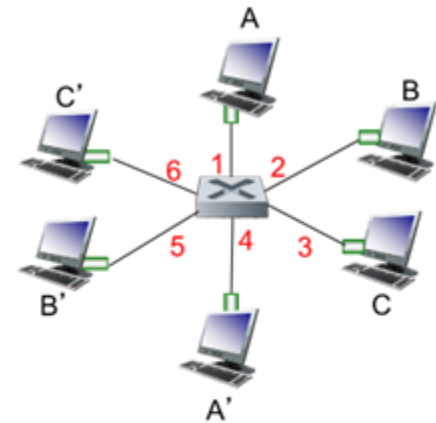
802.3 Ethernet standards: link & physical layers

- **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 **rdt** (e.g., TCP), otherwise dropped data lost
- Ethernet's MAC protocol: unslotted **CSMA/CD with binary backoff**
- **Many** different Ethernet standards.
 - Common MAC protocol and frame format.
 - Different speeds: 2 Mbps, 10 Mbps, 100 Mbps, 1Gbps, 10 Gbps, 40 Gbps.
 - Different physical layer media: fiber, cable.



➤ **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



MAC	Interface	TTL
CC-49-DE-D0-AB-7D	16	60

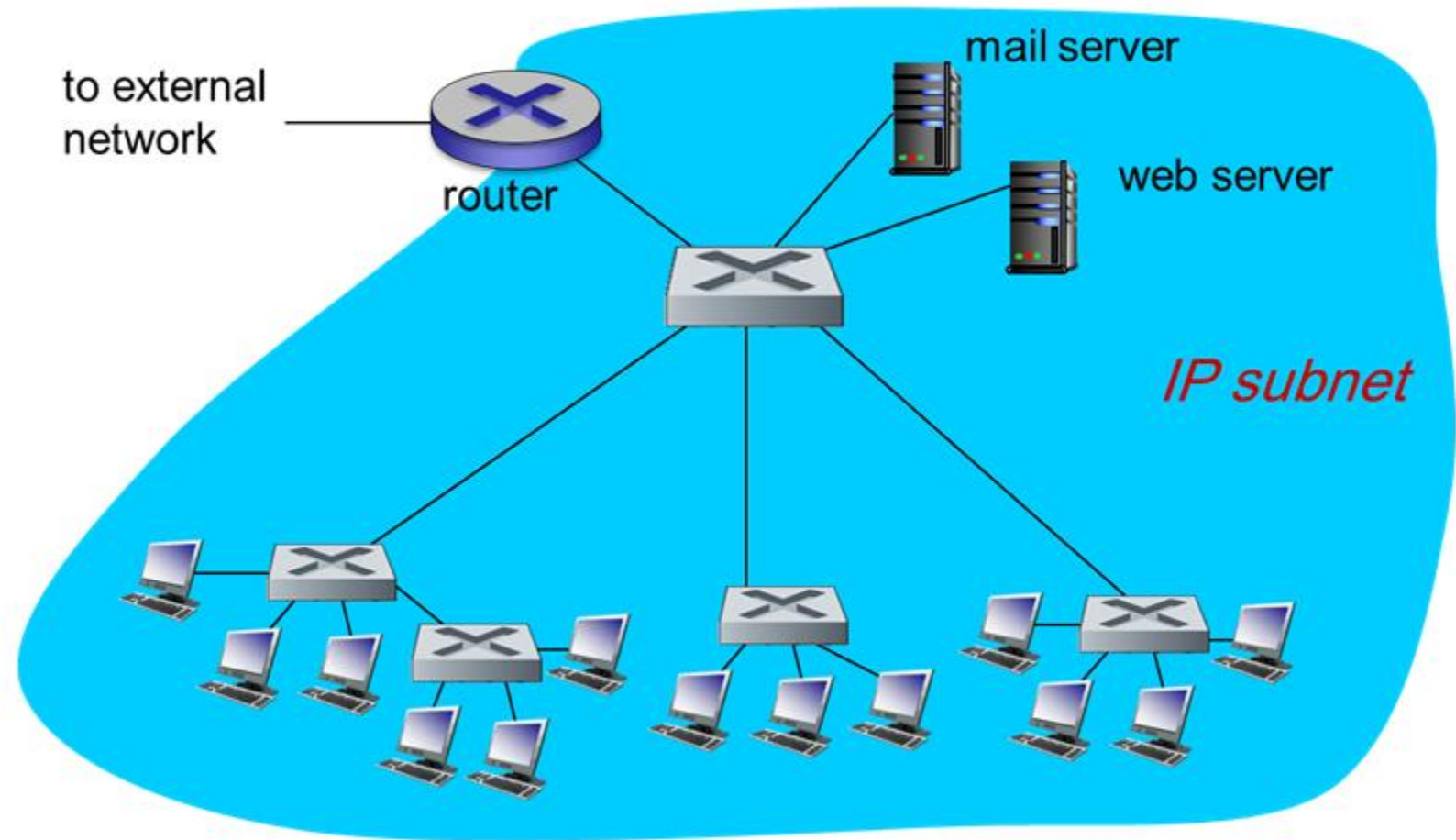
➤ **Transparent**

- Hosts are unaware of presence of switches

➤ **Plug-and-play, self-learning**

- Switches do not need to be configured.





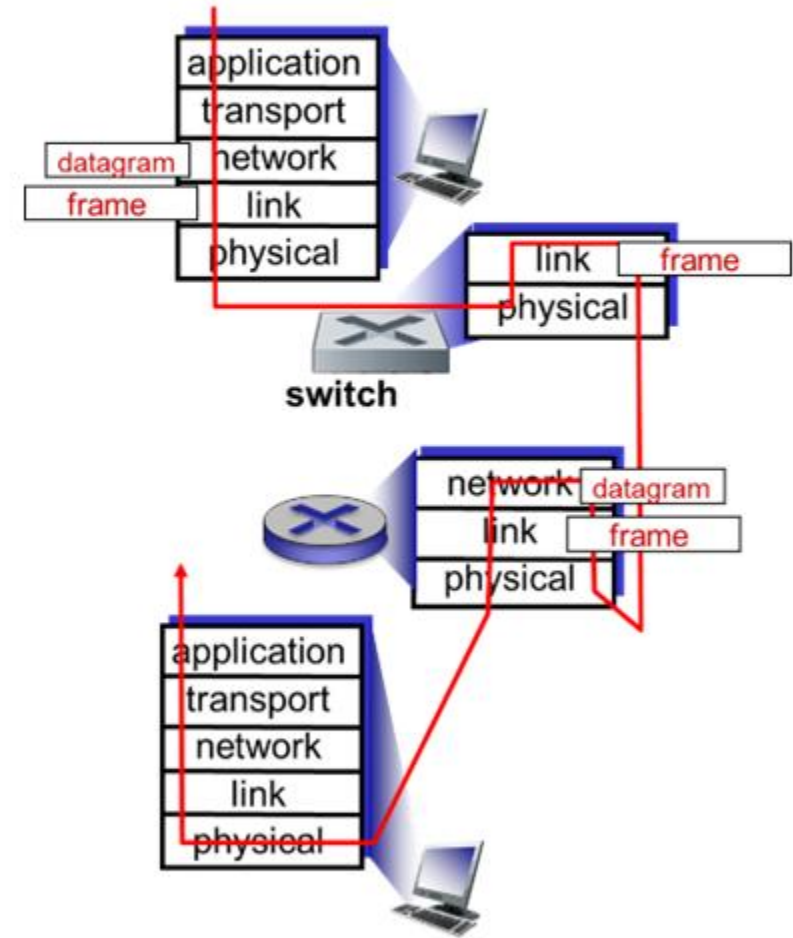
Switches vs. routers

Both are store-and-forward:

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

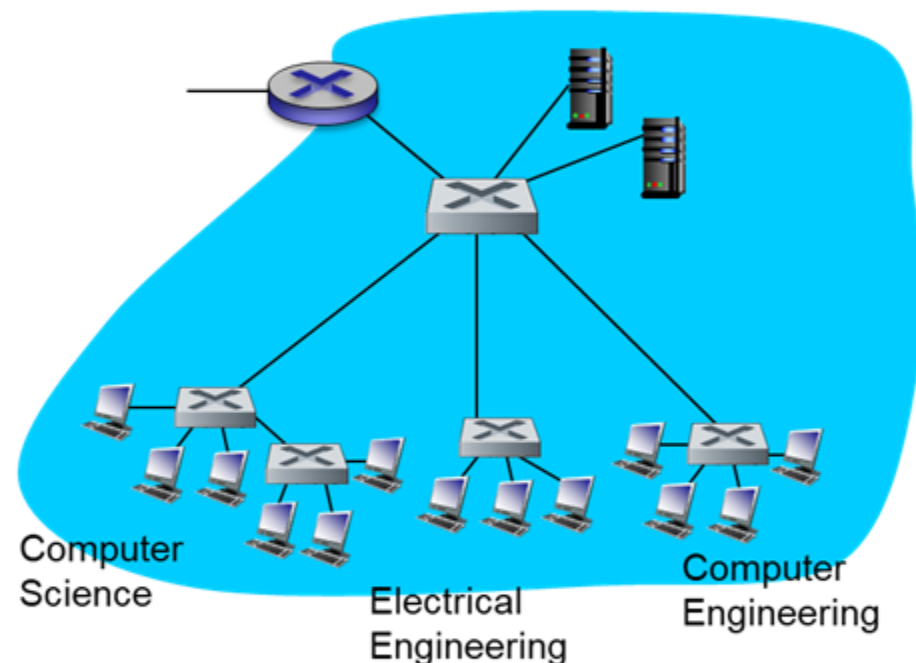
Both have forwarding tables:

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



consider:

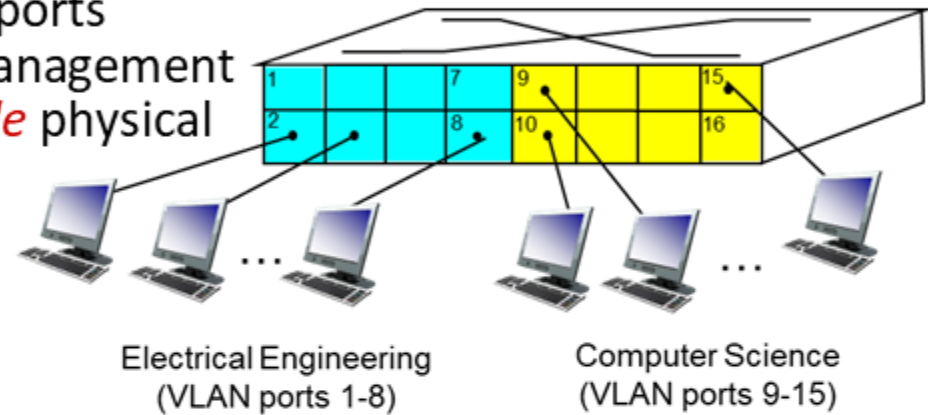
- CS user moves office to EE, but wants connect to CS switch?
- Single broadcast domain:
 - All layer-2 broadcast traffic (ARP, DHCP, unknown location of destination MAC address) must cross entire LAN
 - Security/privacy, efficiency issues.



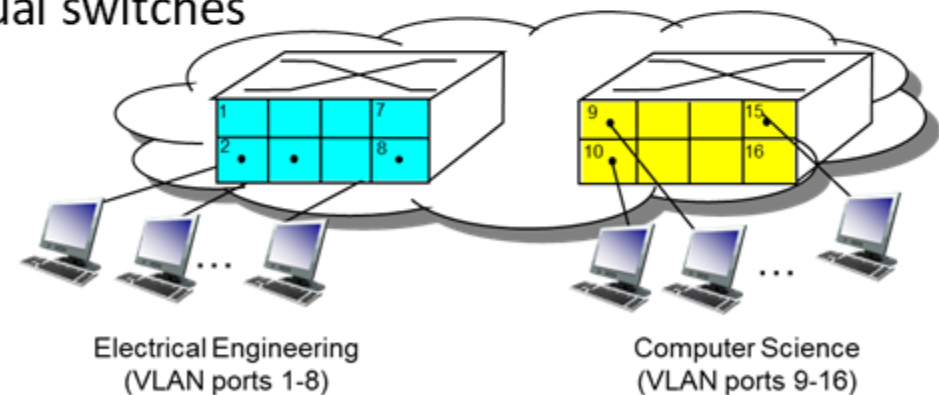
**Virtual Local
Area Network**

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

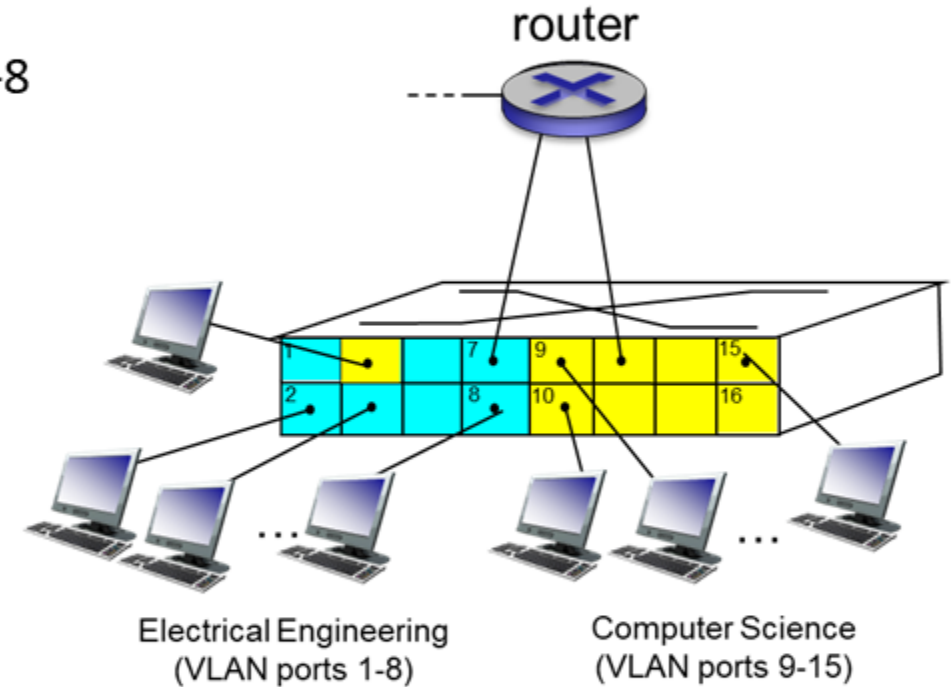
port-based VLAN: switch ports grouped (by switch management software) so that **single** physical switch



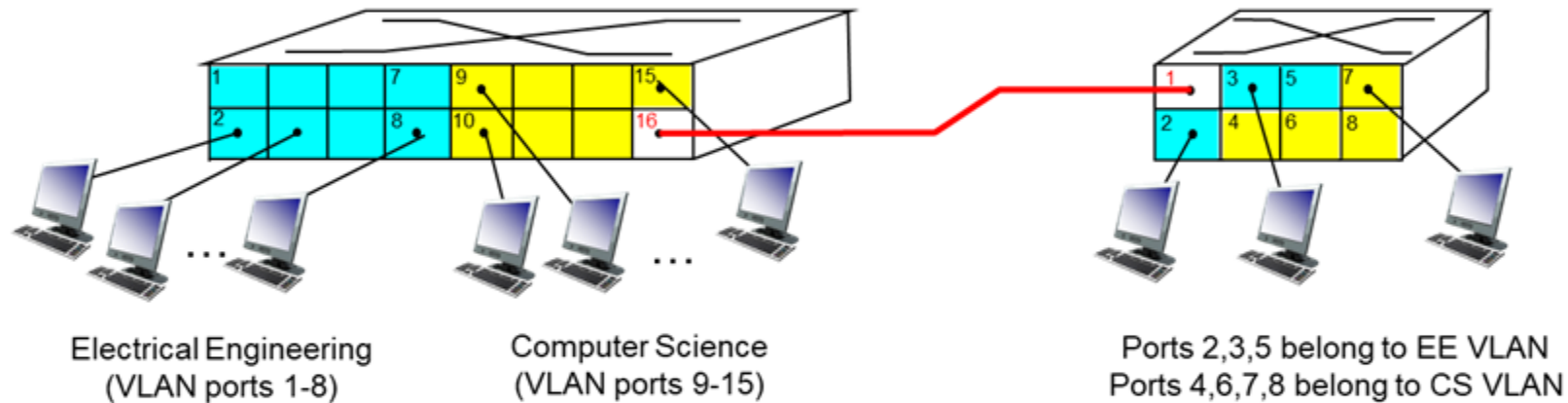
... operates as **multiple** virtual switches



- **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)



VLANs spanning multiple switches



Trunk port:

Carries frames between VLANs defined over multiple physical switches.

VLAN frame format

