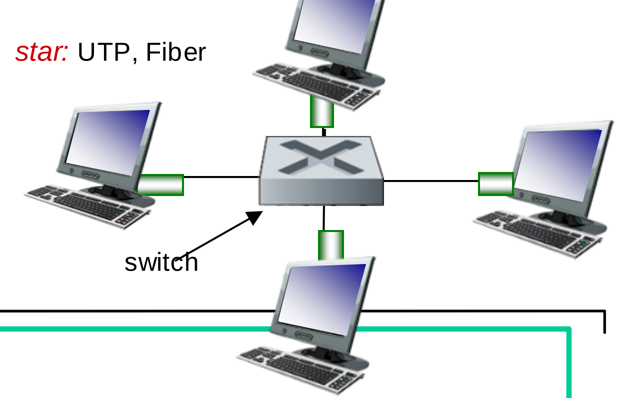
MAC address, ARP, and the Ethernet

CE 352, Computer Networks

Salem Al-Agtash

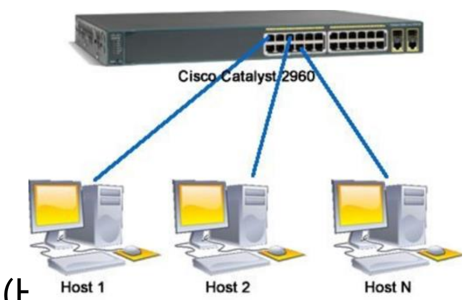
Lecture 21

Recap (Link types)



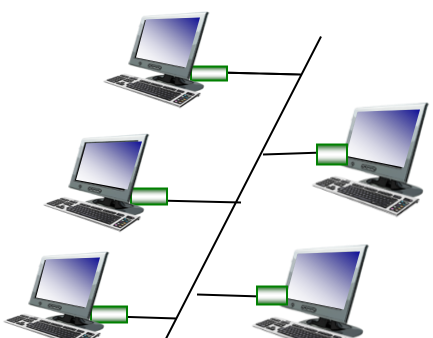
1. Point-to-point

- Single sender and single receiver
  - PPP for dial-up access
  - point-to-point link between Ethernet switch, host
- Protocols
  - Point-to-point protocol (PPP) and High-level data link control (HDLC),



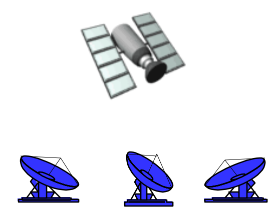
2. Broadcast (shared wire or medium) → what is a proper protocol?

- Multiple senders and receivers
  - Ethernet and Wireless LANs



bus: coaxial cable

shared RF
(e.g., 802.11 WiFi)



shared RF
(satellite)

Recap (MAC protocols):

1. channel partitioning

- divide channel into smaller "pieces"
- allocate piece to node for exclusive use

(1.1 time slots, 1.2 frequency, 1.3 code)

Efficiency → R/N *Widely used in 3G, 4G, … (Ch7)*

2. random access

- channel not divided, allow collisions
- "recover" from collisions

(2.1 Slotted ALOHA, 2.2 Pure ALOHA, 2.3 CSMA, CSMA/CD, CSMA/CA)

Efficiency → 1/e (37%), *1/2e (18%),* *$1/(1+5t_{prop}/t_{trans})$ [can be 100%]*

widely used in Wifi, Ethernet, WSN

3. "taking turns"

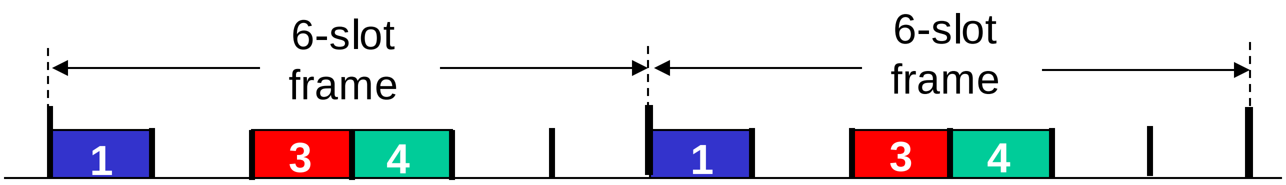
- nodes take turns, but nodes with more to send can take longer turns

(3.1 polling, 3.2 token passing)

Recap (1.1 TDMA)

Channel partitioning MAC protocol - time division multiple access

- access to channel in "rounds"
- each station gets fixed length slot (length = packet transmission time) in each round
- unused slots go idle
- example: 6-station LAN, 1,3,4 have packets to send, slots 2,5,6 idle

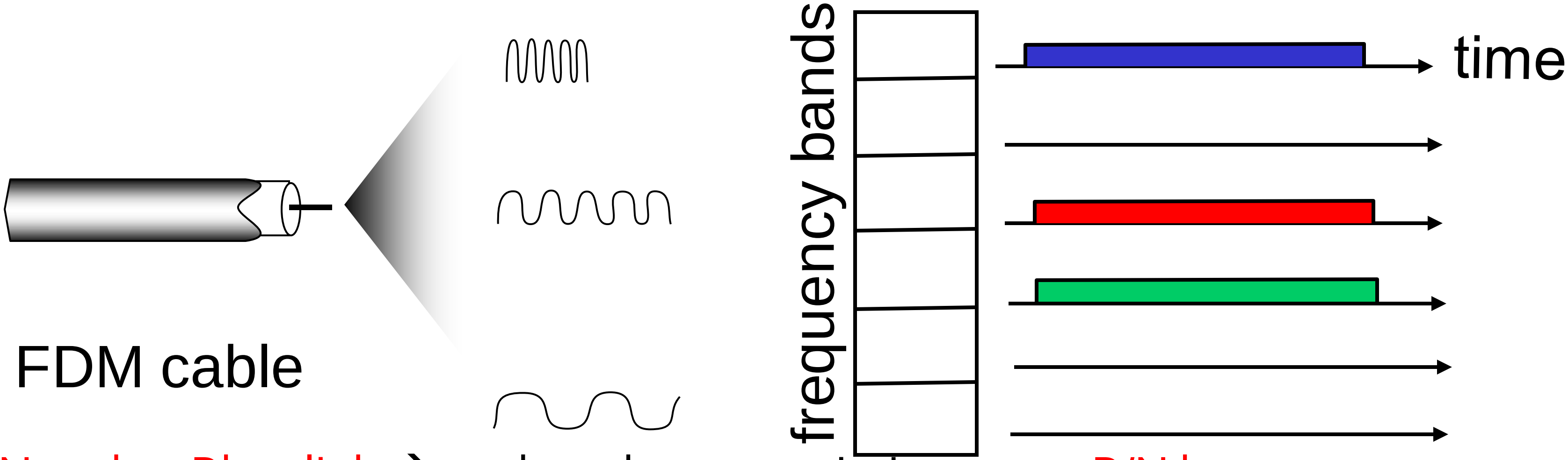


- N nodes, Rbps link → each nodes transmission rate = R/N bps
- Pros: no collision and fair
- Cons: R/N bps at maximum and must wait N-1 time slots, even if no other node is transmitting

Recap (1.2 FDMA)

Channel partitioning MAC protocol: frequency division multiple access

- channel spectrum divided into frequency bands
- each station assigned fixed frequency band
- unused transmission time in frequency bands go idle
- example: 6-station LAN, 1,3,4 have packet to send, frequency bands 2,5,6 idle



- N nodes, Rbps link → each nodes transmission rate = R/N bps
- Pros: no collision and fair
- Cons: R/N bps at maximum even if no other node is transmitting

Recap (2.1 Slotted ALOHA)

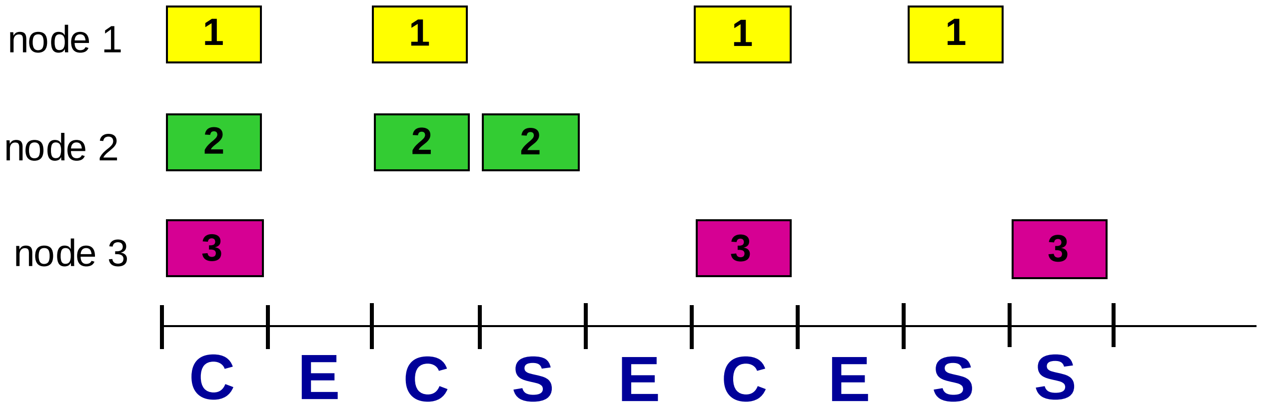
assumptions:

- all frames same size – L bits
- time divided into equal size slots (time to transmit 1 frame) – L/R
- nodes start to transmit only slot beginning
- nodes are synchronized with slot timing
- if 2 or more nodes transmit in slot, all nodes detect collision before slot time ends

operation:

when node obtains fresh frame, transmits in next slot

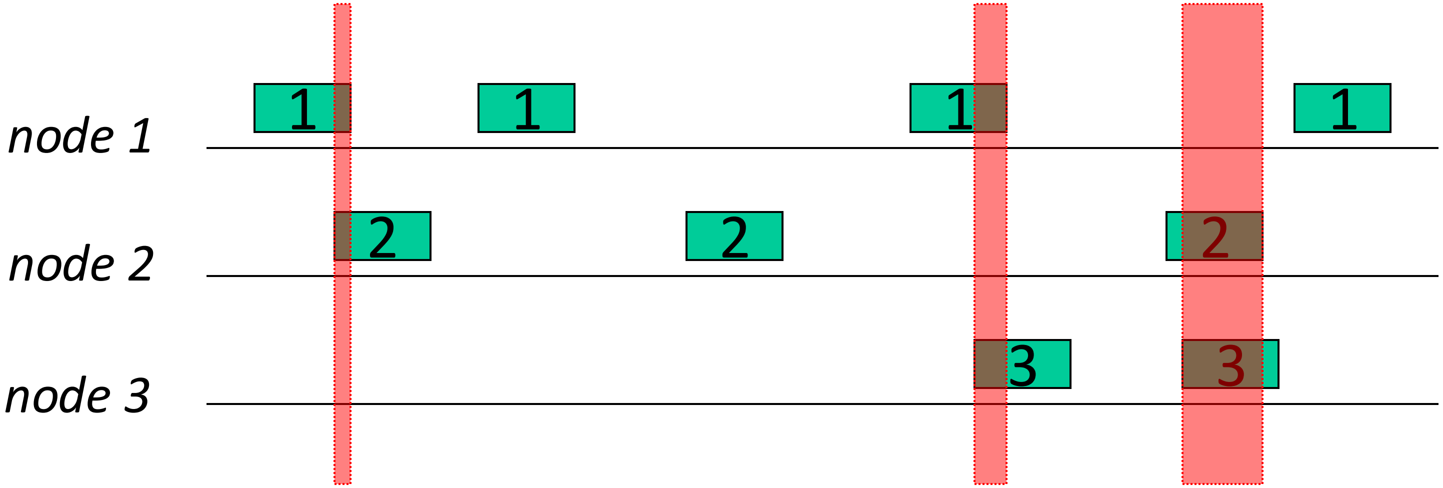
- *if no collision:* node can send new frame in next slot
- *if collision:* node retransmits frame in each subsequent slot with *probability p* until success



C: Collision slot (waste)
E: Empty slot (no use)
S: Successful slot

Recap (2.2 Pure (unslotted) ALOHA)

- Pure Aloha: simpler, no synchronization, when a frame first arrives
  - transmit immediately
- If collision is detected, the node will then retransmit with probability p
- Else, the node waits, then not retransmit with probability $1 - p$
- Collision probability increases:
  - frame sent may collide with other frames sent



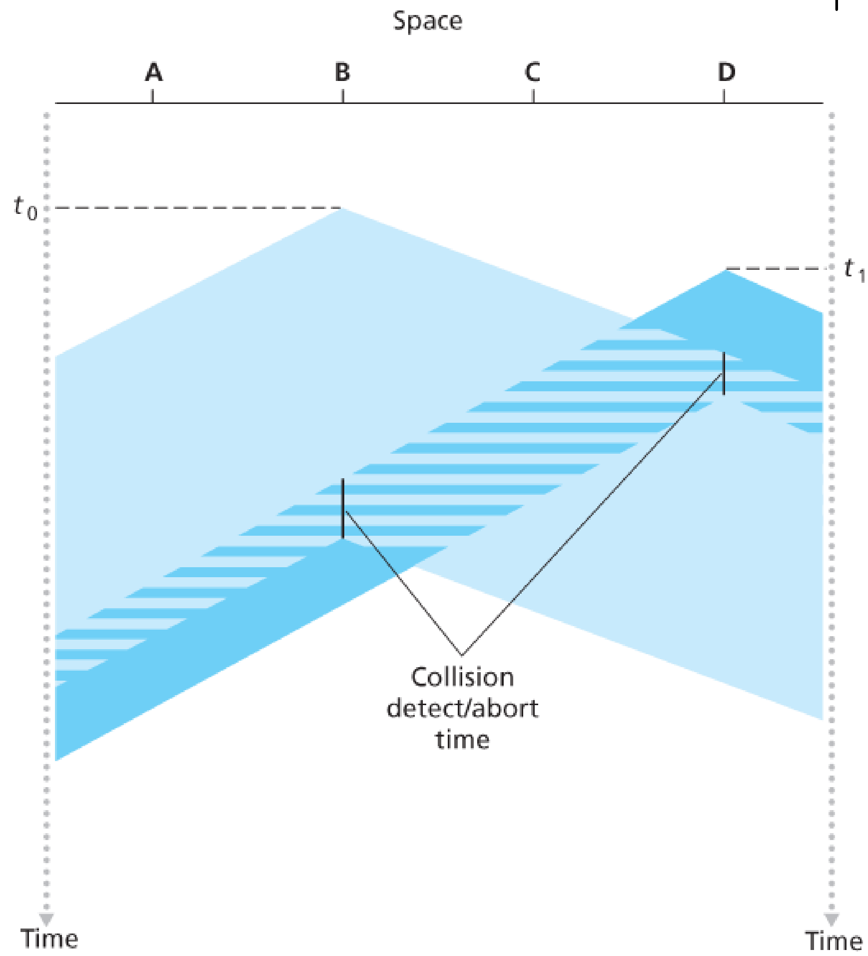
Recap (CSMA/CD (collision detection))

CSMA: carrier sensing → listen before transmit:

- if channel sensed idle: transmit entire frame
- if channel sensed busy, defer transmission

/CD collision detection:

- collisions *detected* within short time
- colliding transmissions aborted, reducing channel wastage
- waits a random amount of time, then retransmits

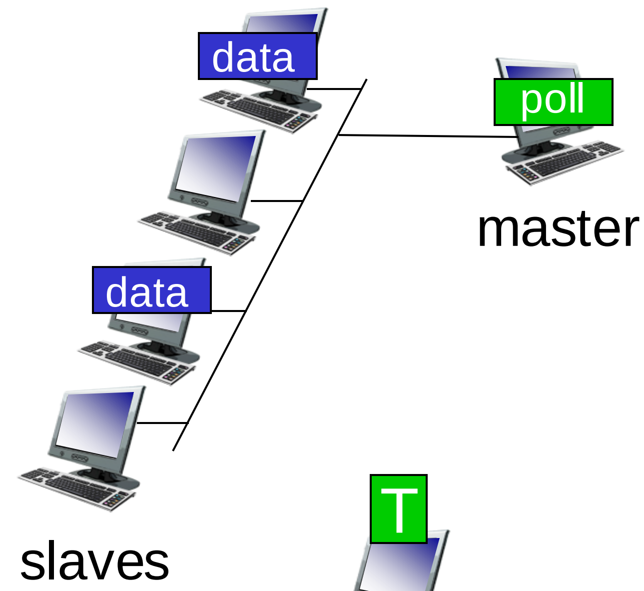


Recap (3.Taking turns)

3.1 polling (Bluetooth):

Master node "invites" slave nodes to transmit in turn → concerns:

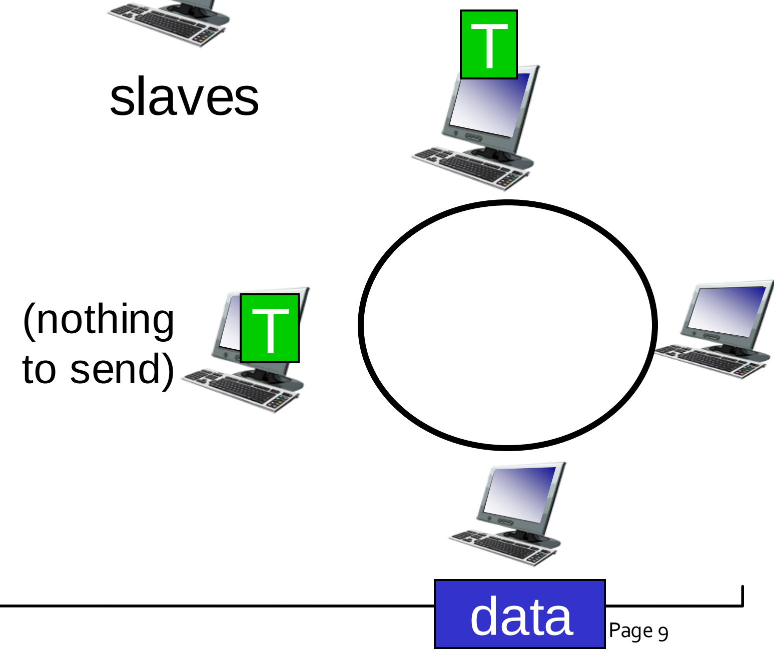
- polling overhead
- latency
- single point of failure (master)



3.2 token passing (fiber distributed data interface (FDDI):

control *token* passed from one node to next sequentially → concerns:

- token overhead
- Latency
- single point of failure (token)



Questions

- Consider the following multiple access protocols that we've studied: (1) TDMA and FDMA (2) CSMA (3) Aloha, and (4) polling. Which of these protocols:
  - are collision-free and (e.g., collisions will never happen)?
  - requires some form of centralized control to mediate channel access?
  - is there a maximum amount of time that a node knows that it will have to wait until it can successfully gain access to the channel?

Questions

- Consider the following multiple access protocols that we've studied: (1) TDMA and FDMA (2) CSMA (3) Aloha, and (4) polling. Which of these protocols:
  - are collision-free and (e.g., collisions will never happen)?

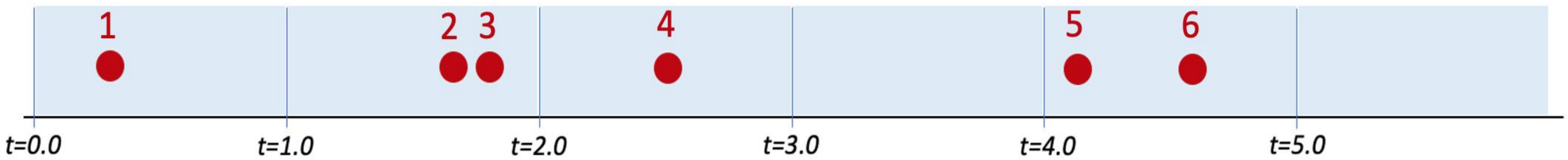
    TDMA and FDMA and Polling
  - requires some form of centralized control to mediate channel access?

    TDMA and FDMA and Polling
  - is there a maximum amount of time that a node knows that it will have to wait until it can successfully gain access to the channel?

    TDMA and FDMA and Polling

Questions

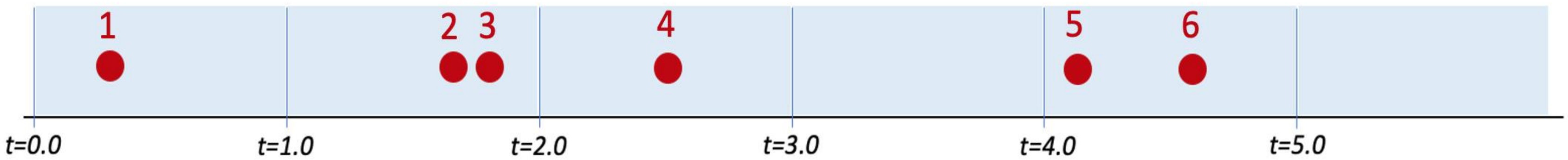
- Consider the figure below, which shows the arrival of 6 messages for transmission at different multiple access wireless nodes at times t = 0.3, 1.7, 1.8, 2.5, 4.2, 4.6. Each transmission requires exactly one time unit



- Indicate which packets are successfully transmitted for each of the following MAC protocols. Assume that if a packet experiences *a collision*, a node will not attempt a retransmission of that packet until sometime after t=5. Assume also that it takes 0.2 time units for a signal to propagate from one node to each of the other nodes
- Slotted ALOHA →
- Pure ALOHA →
- CSMA (without collision detection) →
- CSMA/ CD →

Questions

- Consider the figure below, which shows the arrival of 6 messages for transmission at different multiple access wireless nodes at times t = 0.3, 1.7, 1.8, 2.5, 4.2, 4.6. Each transmission requires exactly one time unit



- Indicate which packets are successfully transmitted for each of the following MAC protocols. Assume that if a packet experiences *a collision*, a node will not attempt a retransmission of that packet until sometime after t=5. Assume also that it takes 0.2 time units for a signal to propagate from one node to each of the other nodes
- Slotted ALOHA → 1, 4
- Pure ALOHA → 1
- CSMA (without collision detection) → 1, 5
- CSMA/ CD → 1, 4, 5

Questions

- Suppose two nodes start to transmit at the same time a packet of length *L* over a broadcast channel of rate *R*. Denote the propagation delay between the two nodes as d_{prop}. Will there be a collision if $d_{prop} < L/R$?

- In CSMA/CD, after the fifth collision, what is the probability that a node chooses K=4? The result K=4 corresponds to a delay of how many seconds on a 10 Mbps Ethernet?

Questions

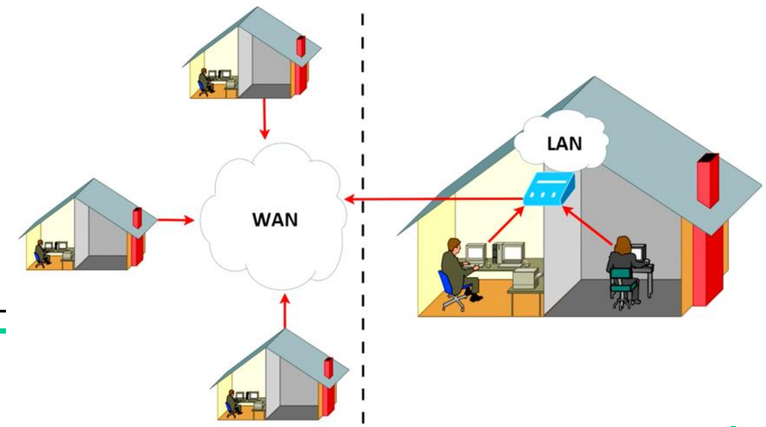
- Suppose two nodes start to transmit at the same time a packet of length *L* over a broadcast channel of rate *R*. Denote the propagation delay between the two nodes as d_{prop}. Will there be a collision if $d_{prop} < L/R$?

There will be a collision in the sense that while a node is transmitting it will start to receive a packet from the other node.

- In CSMA/CD, after the fifth collision, what is the probability that a node chooses K=4? The result K=4 corresponds to a delay of how many seconds on a 10 Mbps Ethernet?

After the 5th collision, the adapter chooses from {0, 1, 2,…, 31}. So the probability that it chooses 4 is 1/32. It waits k*512 bit times = $4*512/10*10^6$= 204.8 microseconds.

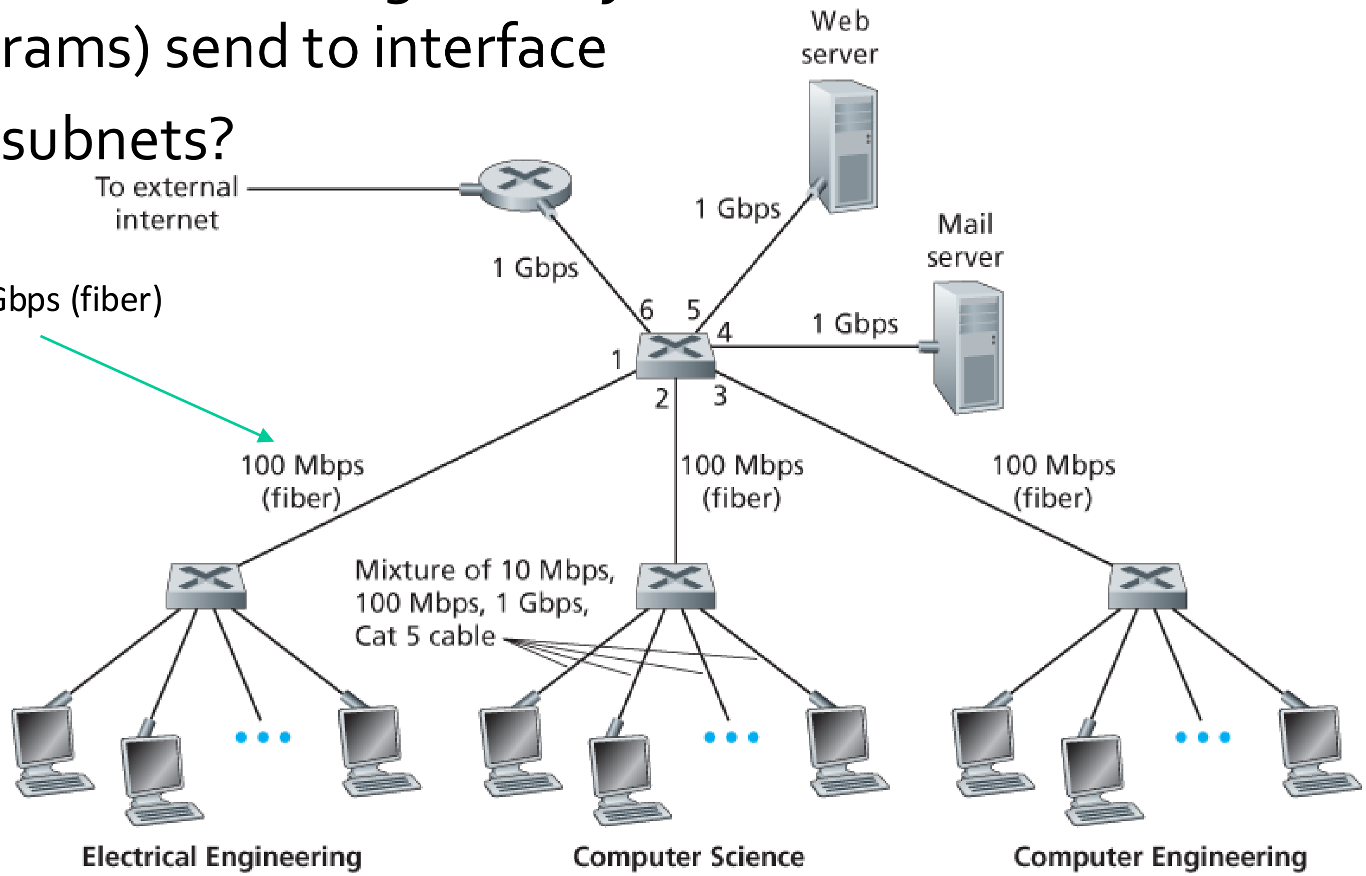
Switched LAN



Institutional network:

- Switches process incoming *Link-layer* frames (not network-layer datagrams) send to interface
- How many subnets?

100 Gbps (fiber)



MAC/LAN addresses

32-bit IP address:

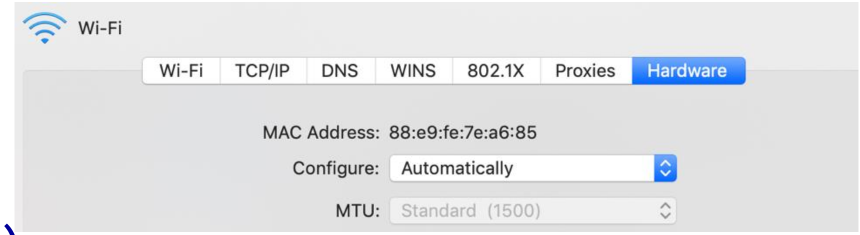
- *network-layer* address for interface
- used for layer 3 (network layer) forwarding

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
- e.g.: 1A:2F:BB:76:09:AD

hexadecimal (base 16) notation

(each "numeral" represents 4 bits)



MAC/LAN addresses

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

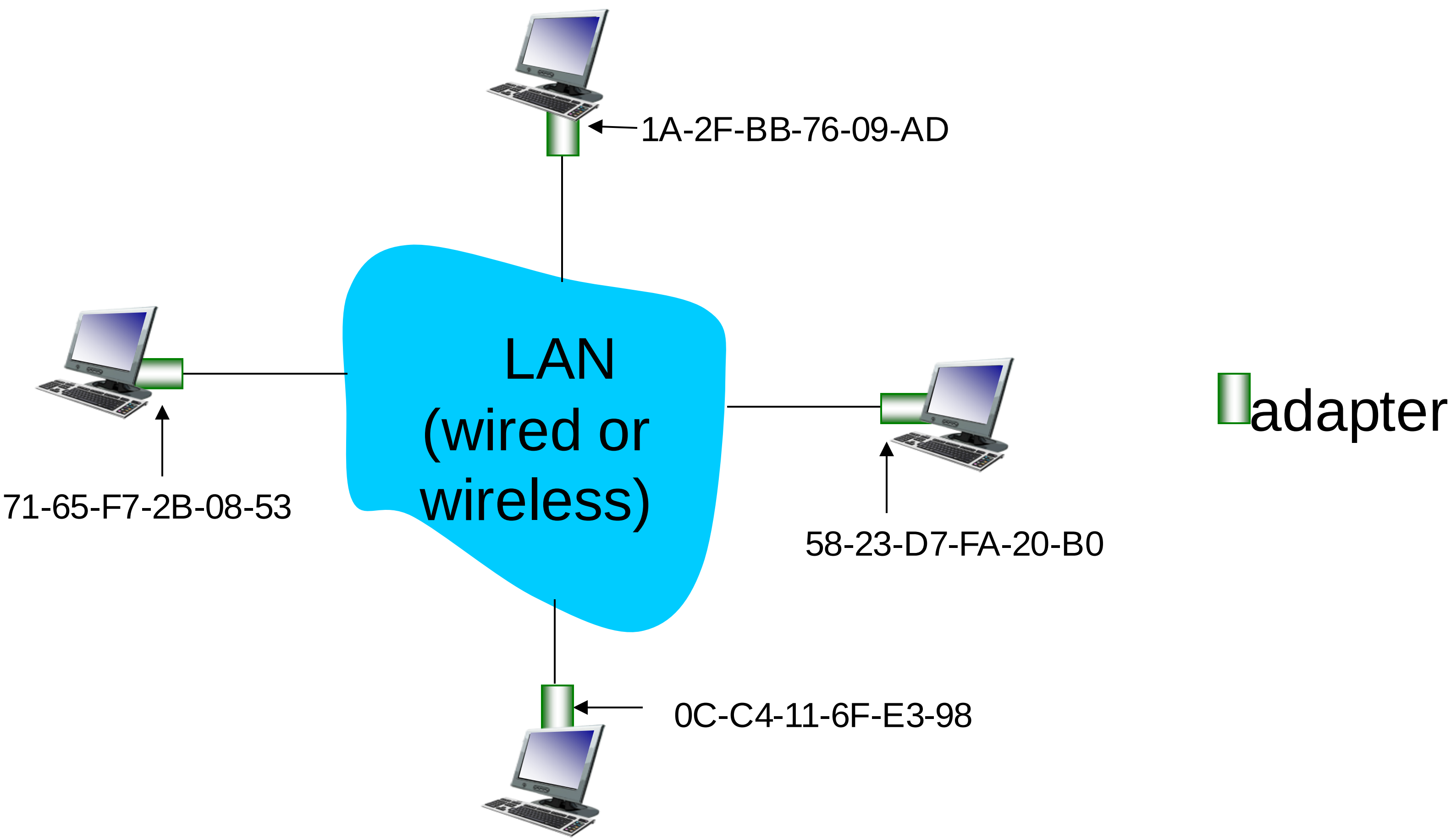
- can move LAN card from one LAN to another

IP hierarchical address *not* portable

- address depends on IP subnet to which node is attached

Example

each adapter on LAN has unique *LAN* address



ARP: address resolution protocol

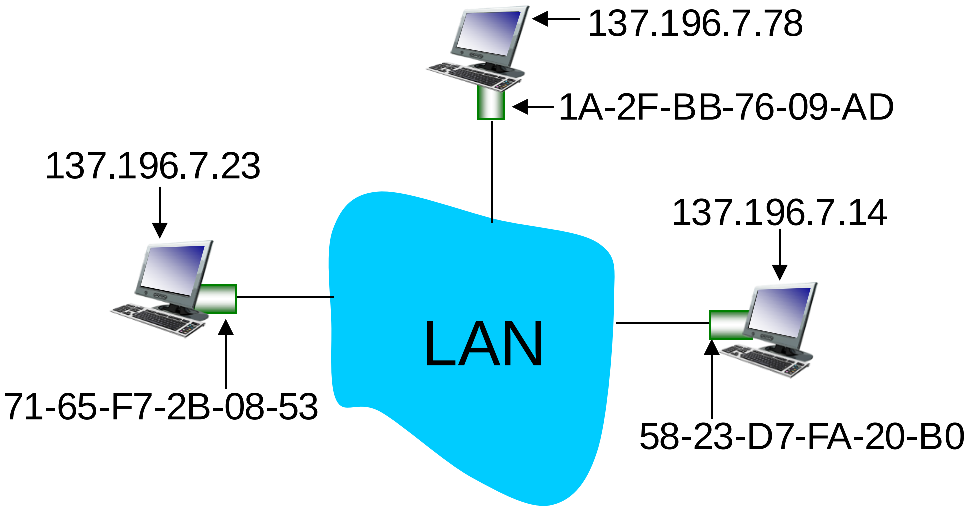
ARP table: each IP node (host, router) on LAN has table

- IP/MAC address mappings for LAN nodes:

  < IP address; MAC address; TTL>

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

- Analogous to DNS in WAN
- But ARP resolves in LAN



https://www.youtube.com/watch?v=cn8Zxh9bPio

ARP: address resolution protocol

ARP protocol: same LAN

A wants to send datagram to B

- B's MAC address not in A's ARP table.

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

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

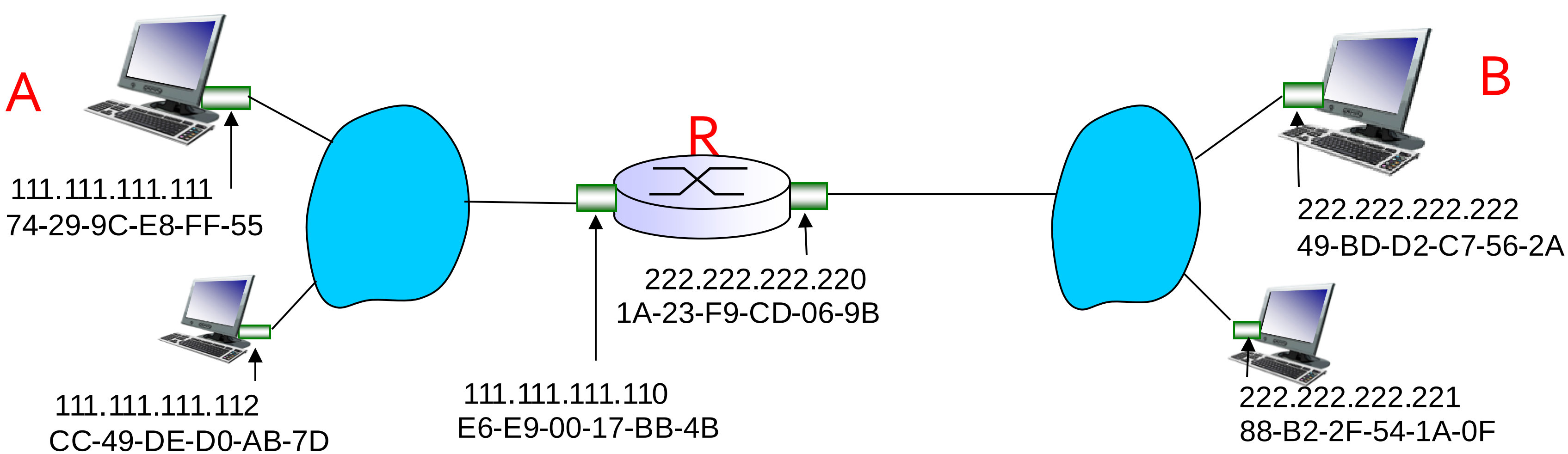
ARP is "plug-and-play":

- nodes create their ARP tables *without intervention from net administrator*

Addressing: routing to another LAN

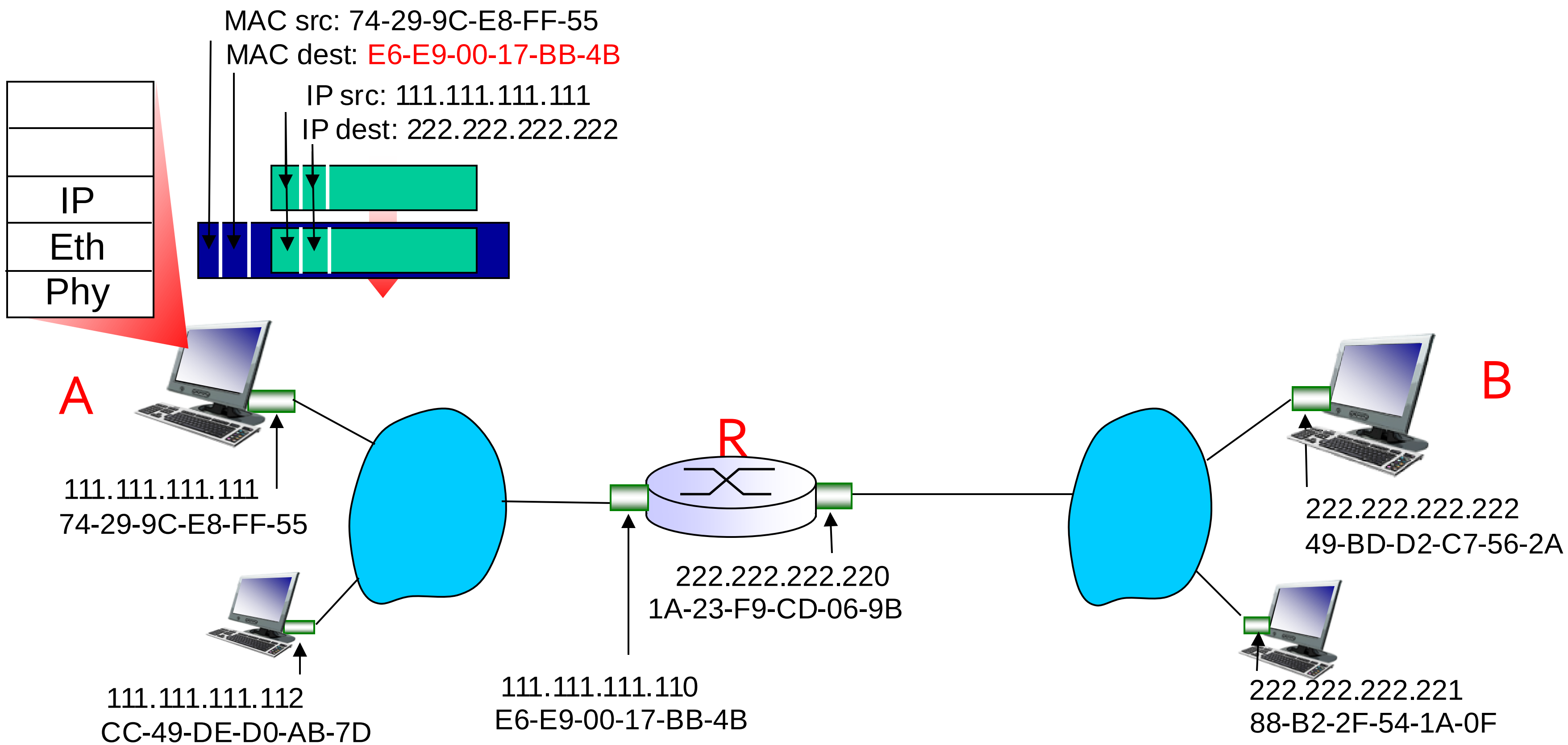
walkthrough: send datagram from A to B via R

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



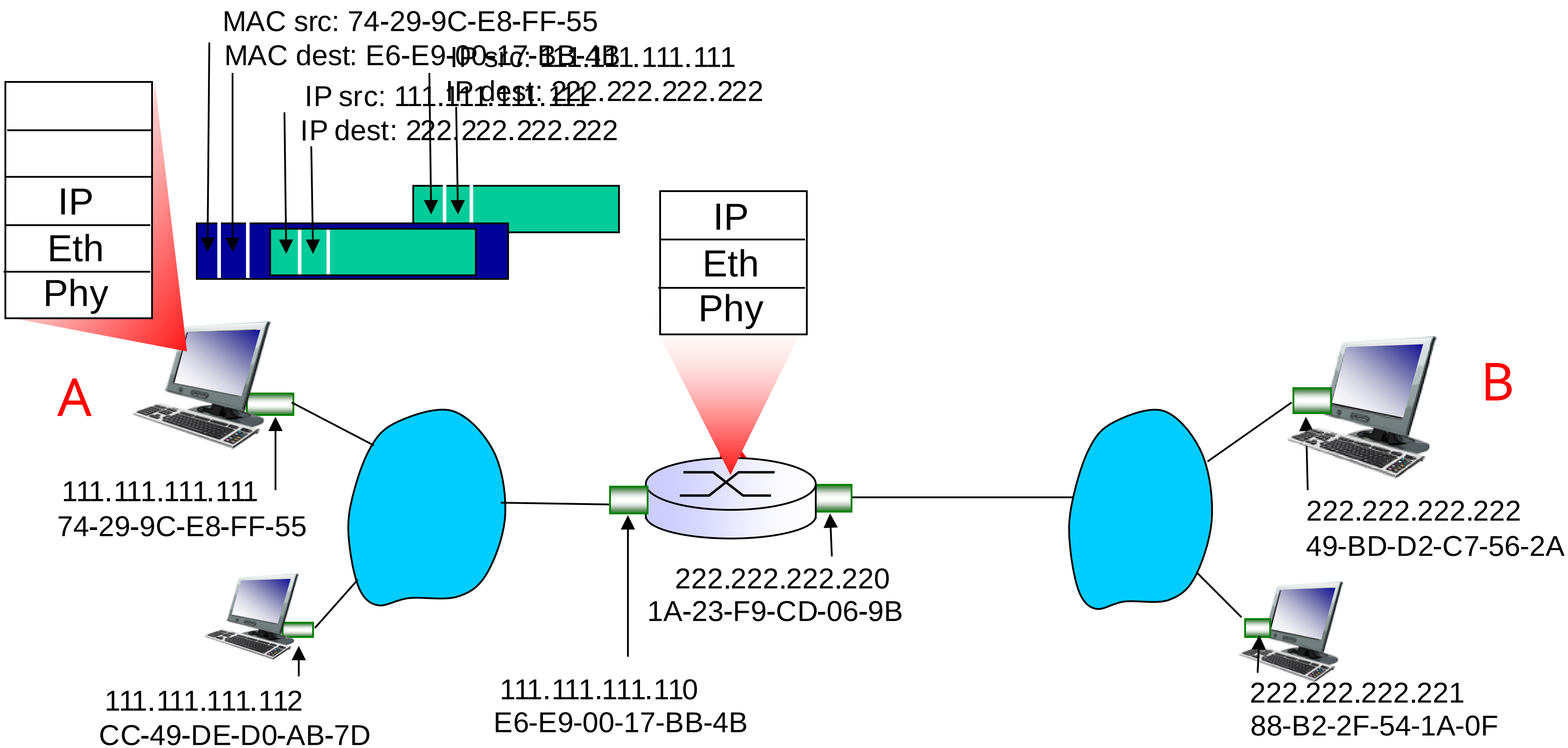
Addressing: routing to another LAN

- 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



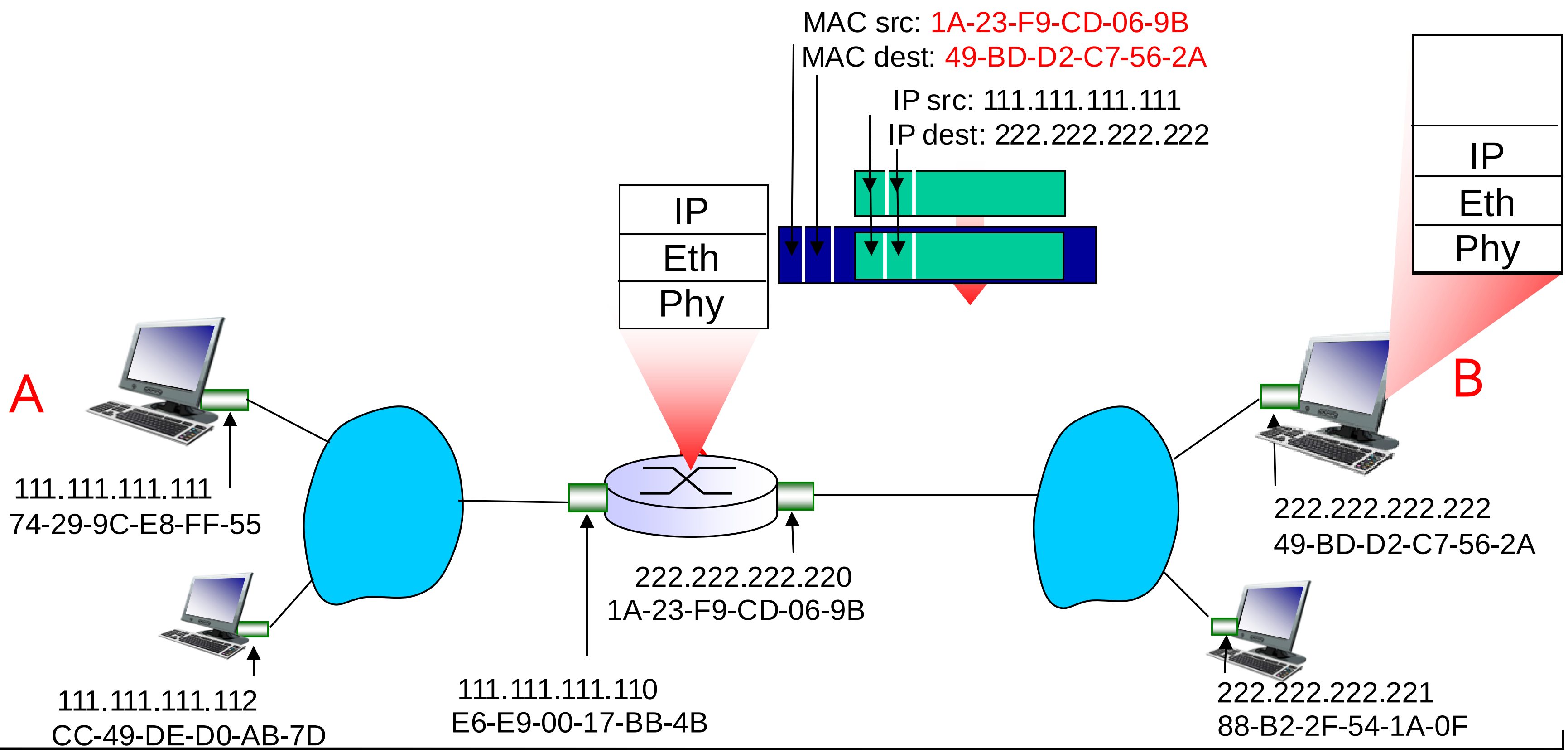
Addressing: routing to another LAN

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



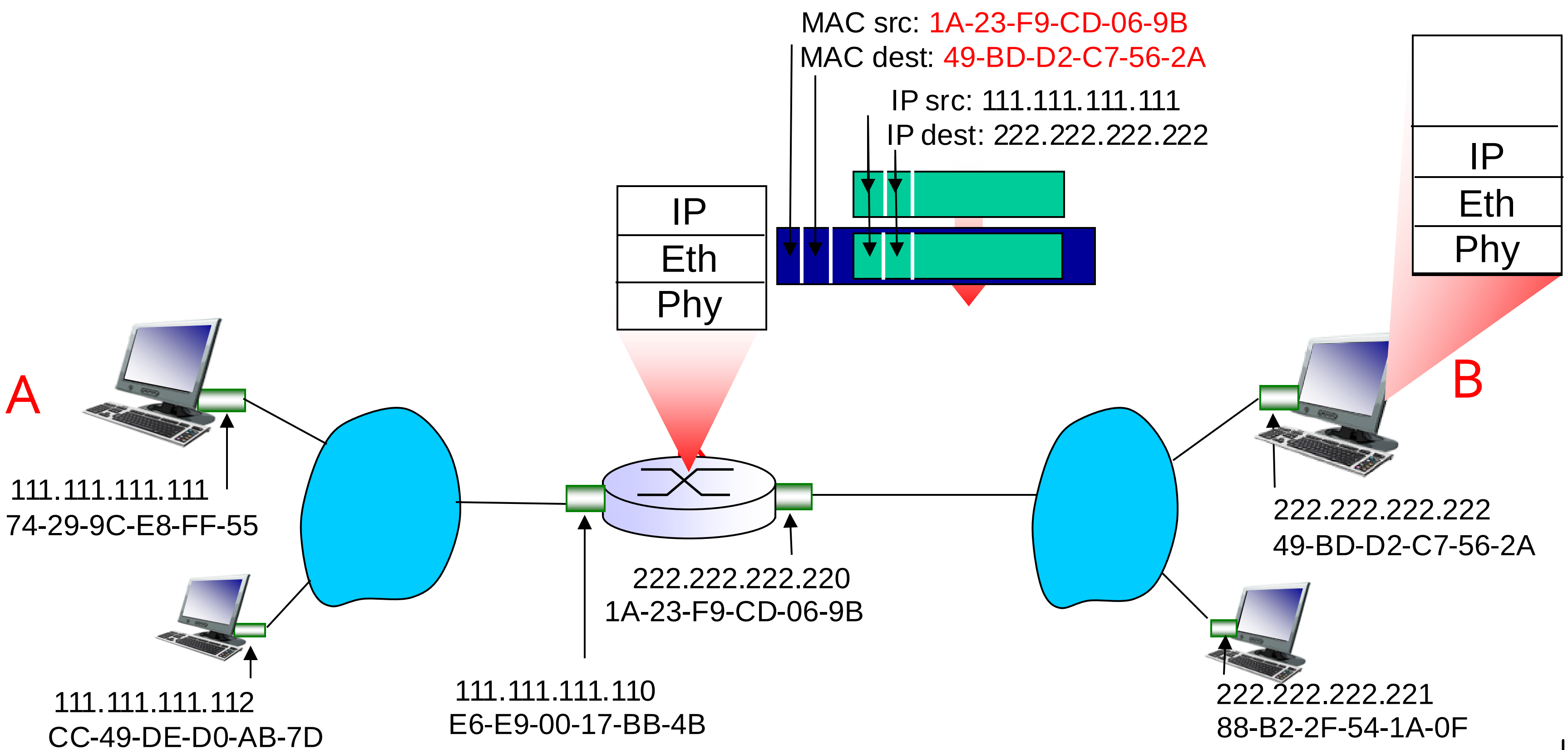
Addressing: routing to another LAN

- 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



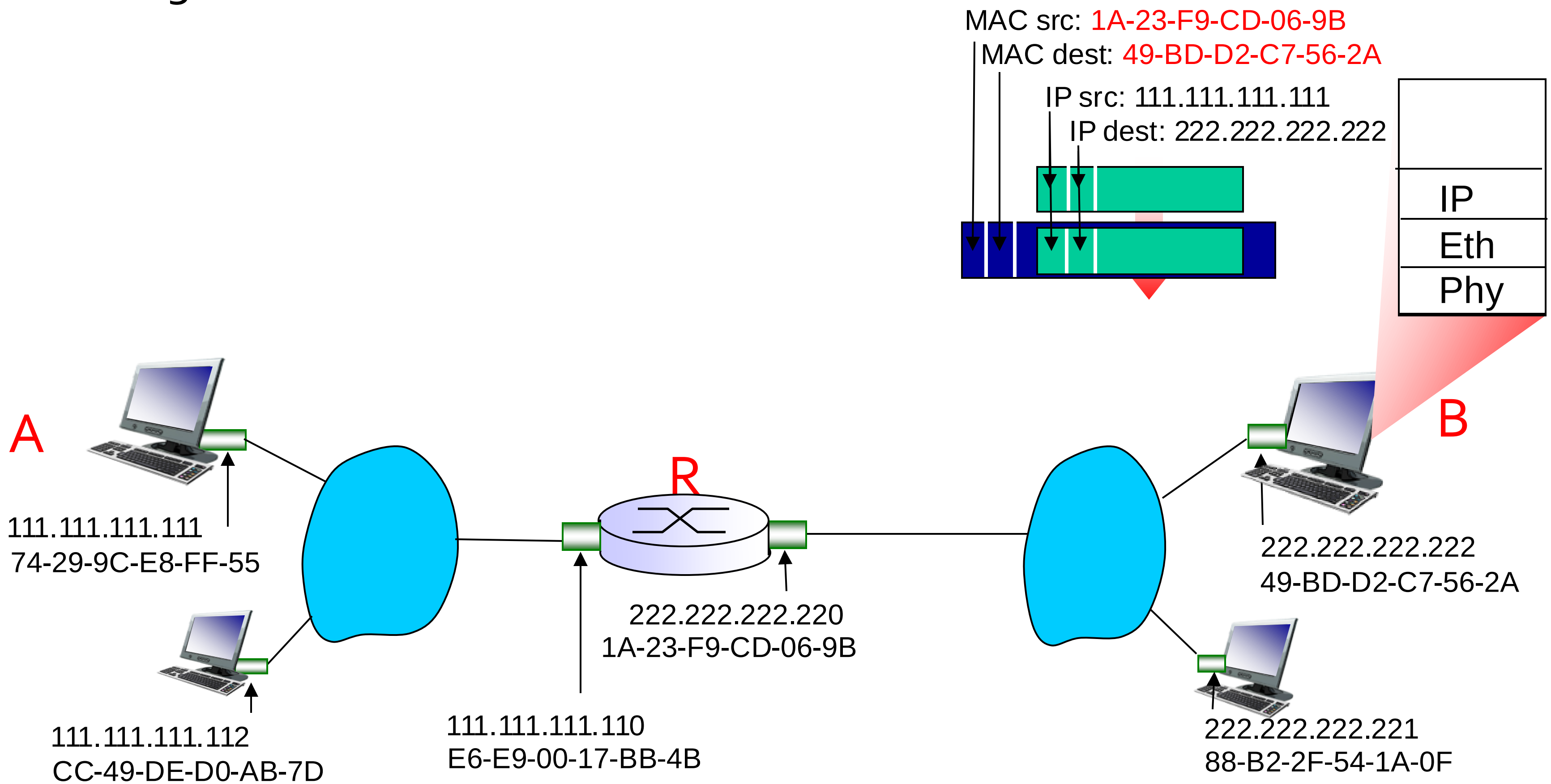
Addressing: routing to another LAN

- 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



Addressing: routing to another LAN

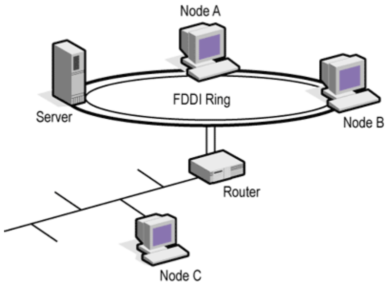
- R forwards datagram with IP source A, destination B
- R creates link-layer frame with B's MAC address as dest, frame contains A-to-B IP datagram

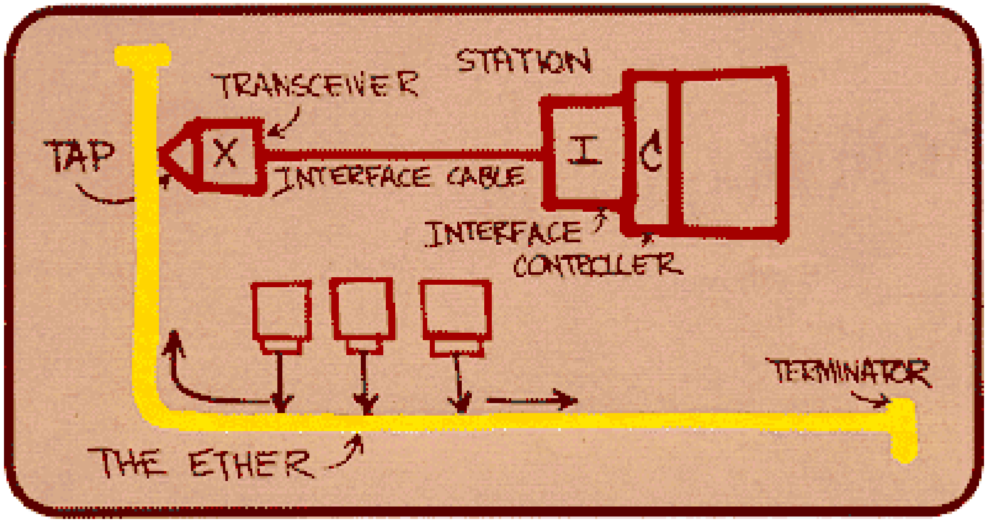


Ethernet

"Dominant" wired LAN technology:

- cheap $20 for NIC
- widely used LAN technology
- simpler, cheaper than token ring, FDDI and ATM LANs
- kept up with speed race: 10 Mbps – 10 Gbps





Bob Metcalfe's Ethernet sketch in 70s

Ethernet: physical topology

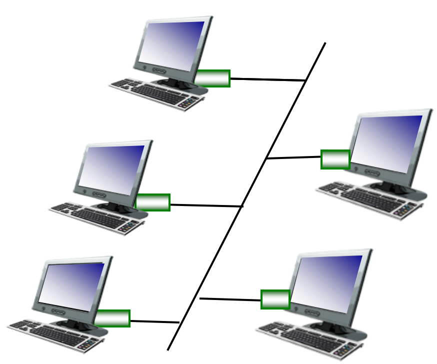
bus: popular through mid 90s

- all nodes in same collision domain (can collide with each other)

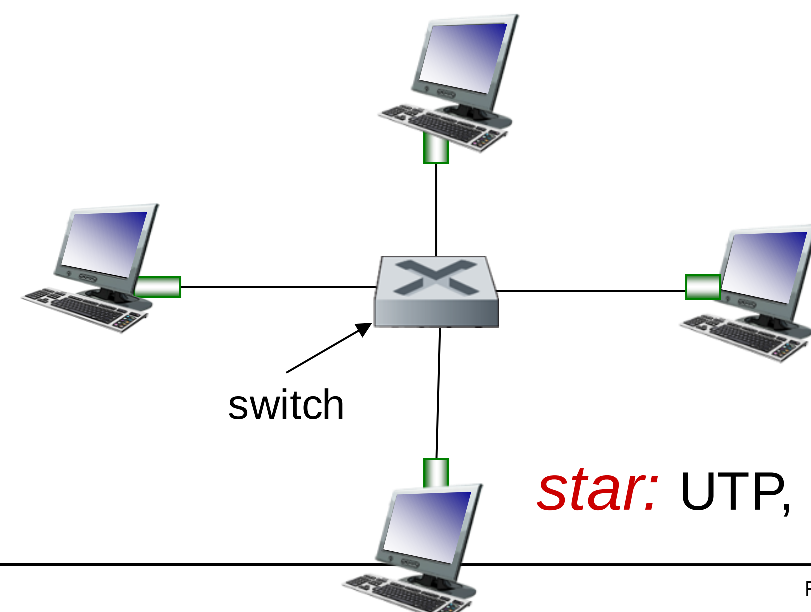
star: prevails today

- active *switch* in center (earlier *hub* (physical layer device that acts on bits)
- nodes do not collide with each other and may connect with a different Ethernet protocol

Think of Ethernet for LAN similar to Internet for WAN



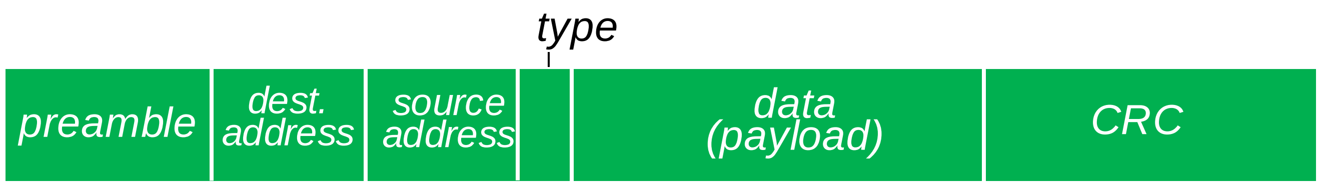
bus: coaxial cable



star: UTP, Fiber

Ethernet frame structure

sending adapter encapsulates IP datagram in Ethernet frame



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

Ethernet frame structure (more)

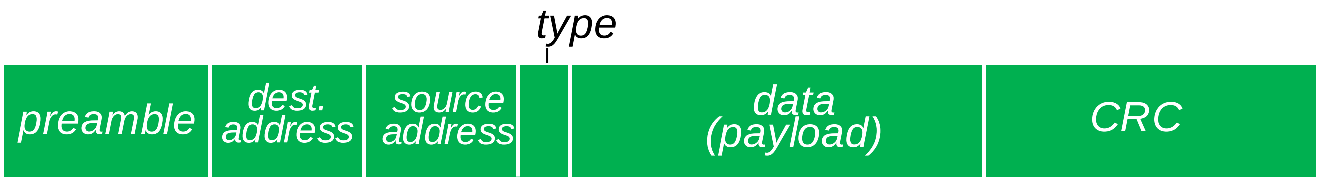
type: indicates higher layer protocol (mostly IP but others possible, e.g., Novell IPX, AppleTalk(discontinued))

CRC: cyclic redundancy check at receiver

- error detected: frame is dropped

Data field (46 to 1,500 bytes): carries the IP datagram

- Maximum transmission unit (MTU) of Ethernet is 1500 bytes (if more bytes, then the datagram is fragmented)
- Minimum size of data field is 46 bytes (if less, padding)



Ethernet: unreliable, connectionless

connectionless: no handshaking between sending and receiving NICs

unreliable: receiving NIC doesn't send acks or nacks to sending NIC

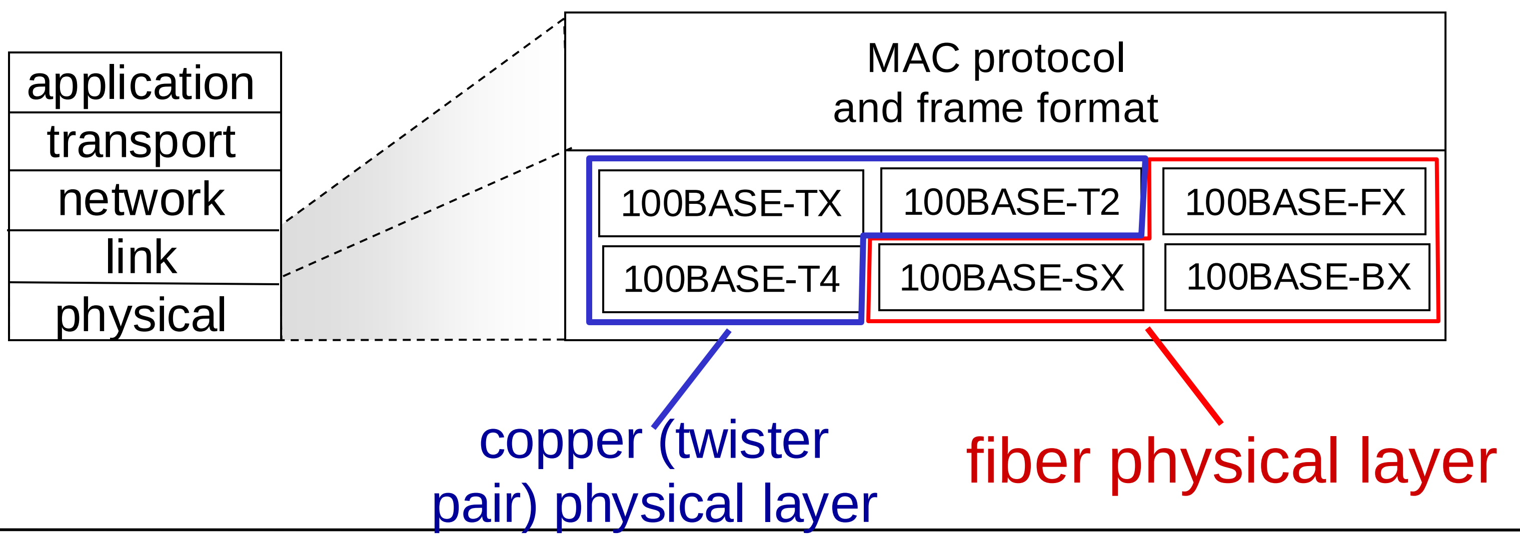
- data in dropped frames recovered only if initial sender uses higher layer rdt (e.g., TCP), otherwise dropped data lost

Ethernet's MAC protocol: unslotted *CSMA/CD with binary backoff* originally on a bus-topology with coaxial cable. Today nodes are connected to a switch via point-to-point segments on UTP

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
- different physical layer media: fiber, UTP cable



Ethernet switch

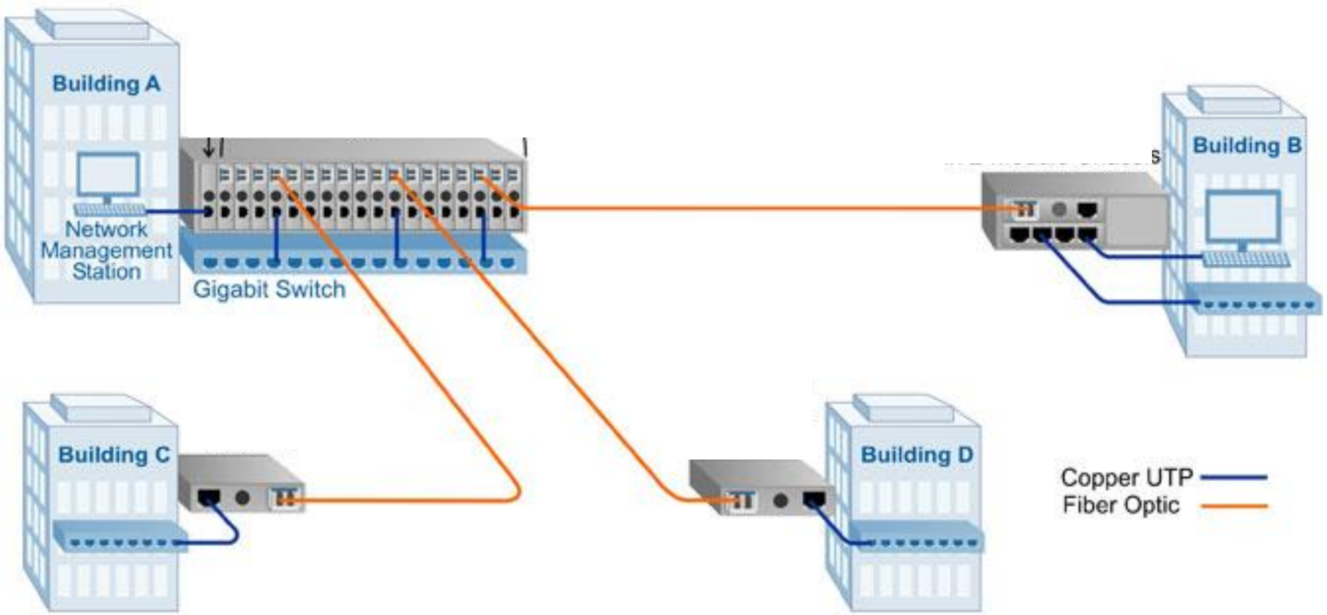
link-layer device: takes an *active* role

- store, forward Ethernet frames
- examine incoming frame's MAC address, selectively forward frame to one-or-more outgoing links

plug-and-play, self-learning

- switches do not need to be configured

Example



Linksys EW5HUB

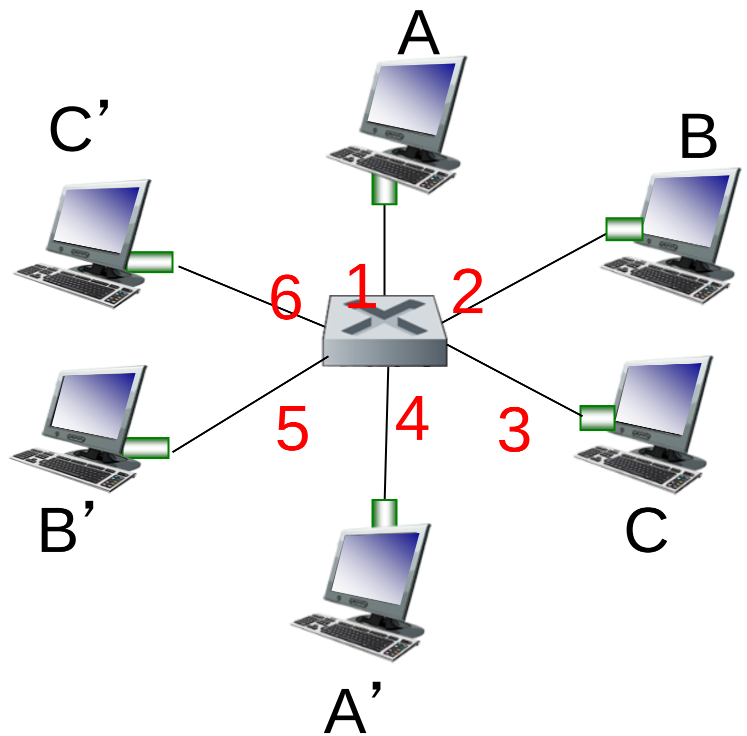
3Com Switch 5500G-EI

Cisco - WS-C4506E-S7L

Source: Google images

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
- *switching:* A-to-A' and B-to-B' can transmit simultaneously, without collisions



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

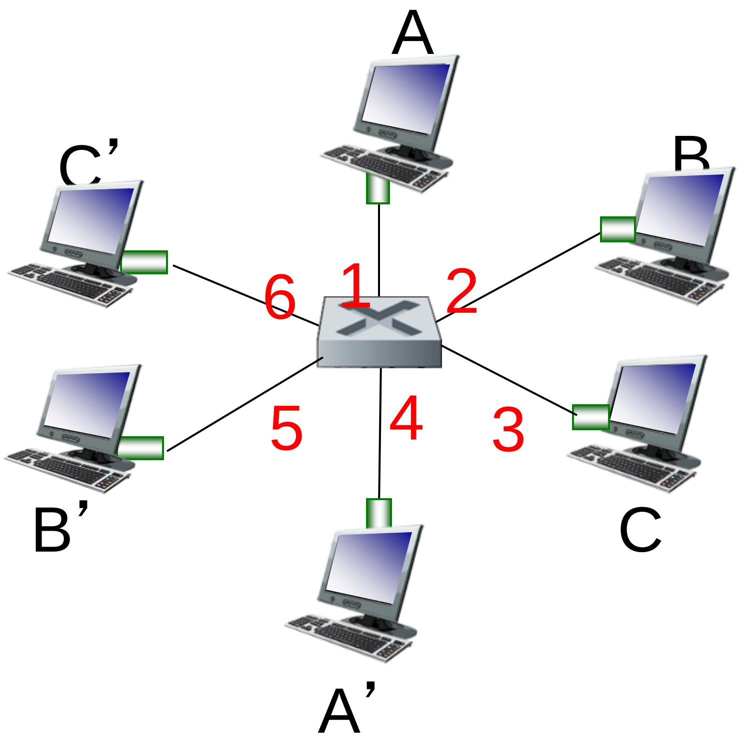
Switch forwarding table

How does switch know A' reachable via interface 4, B' reachable via interface 5?

- each switch has a switch table, each entry:
- (MAC address of host, interface to reach host, time stamp)

MAC addr	interface	TTL
A	*1*	*60*

Switch table (initially empty)

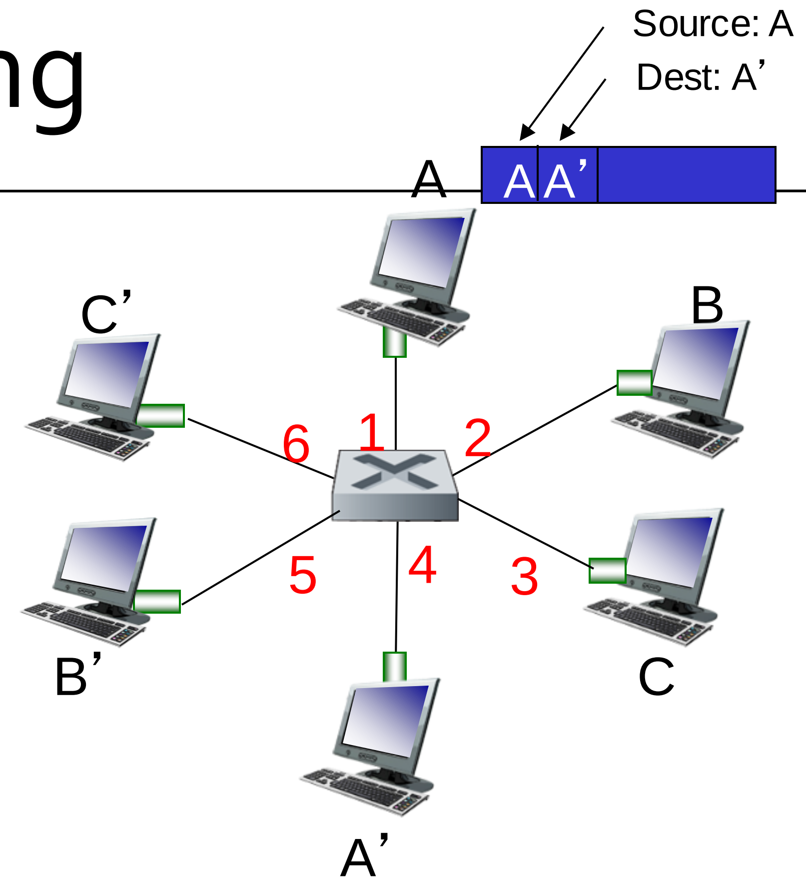


switch with six interfaces (1,2,3,4,5,6)

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



Switch: frame filtering/forwarding

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 {
      forward frame on interface indicated by entry
   }
   else flood /* forward on all interfaces except arriving
      interface */

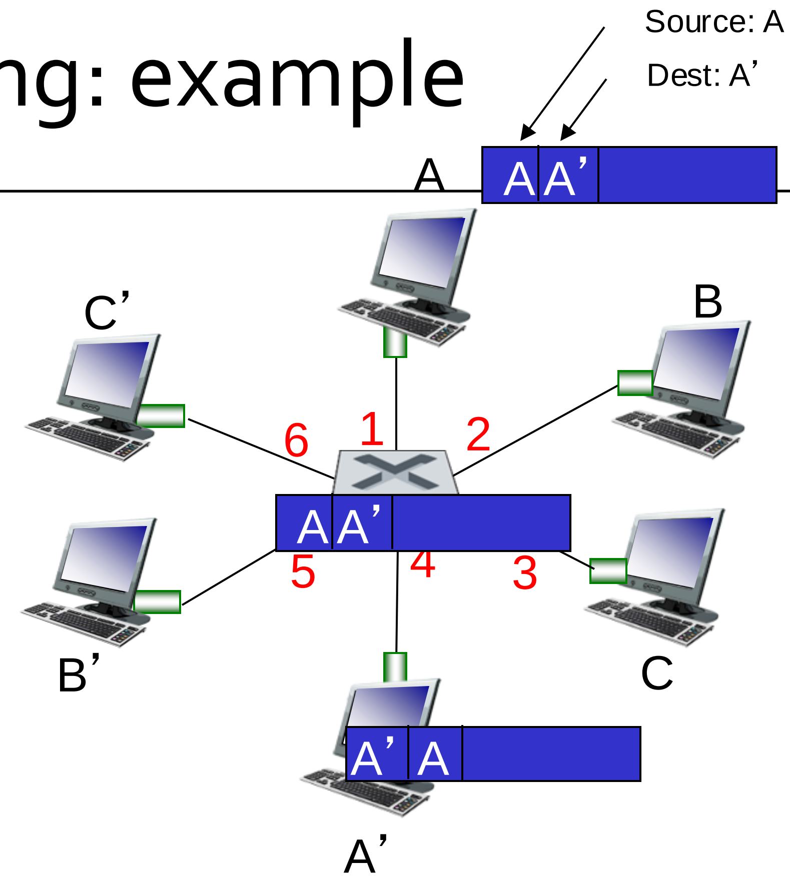
Self-learning, forwarding: example

frame destination, A',
location unknown: *flood*

- destination A location known: selectively send on just one link

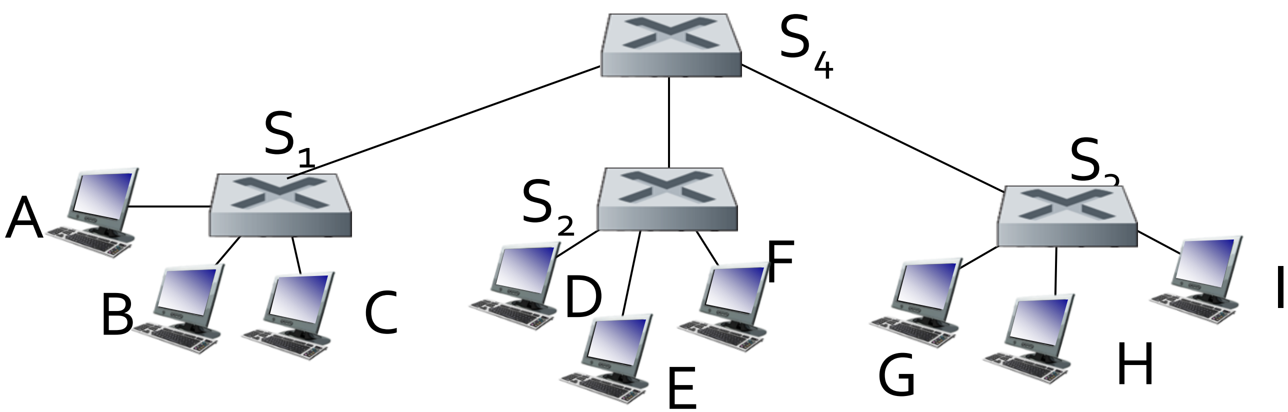
MAC addr	interface	TTL
A	*1*	*60*
A'	*4*	*60*

switch table
(initially empty)



Interconnecting switches

- Self-learning switches can be connected together
- Sending from A to G - how does S_1 know to forward frame destined to G via S_4 and S_3?
  - self learning! (works exactly the same as in single-switch case!)



Summary

Today:

- MAC Address, ARP
- Ethernet

Canvas discussion:

- Reflection
- Exit ticket

Next time:

- read 6.5 and 6.6 of KR (VLAN, MPLS, and Datacenter)
- follow on Canvas! material and announcements

Any questions?