This lecture will be recorded!!!

Welcome to

CS 3516:
Computer Networks

Prof. Yanhua Li

Time: 9:00am -9:50am M, T, R, and F
Zoom Lecture
Fall 2020 A-term

Chapter 6: Link layer

our goals:

- understand principles behind link layer services:

6.1 introduction, services

6.2 error detection, correction

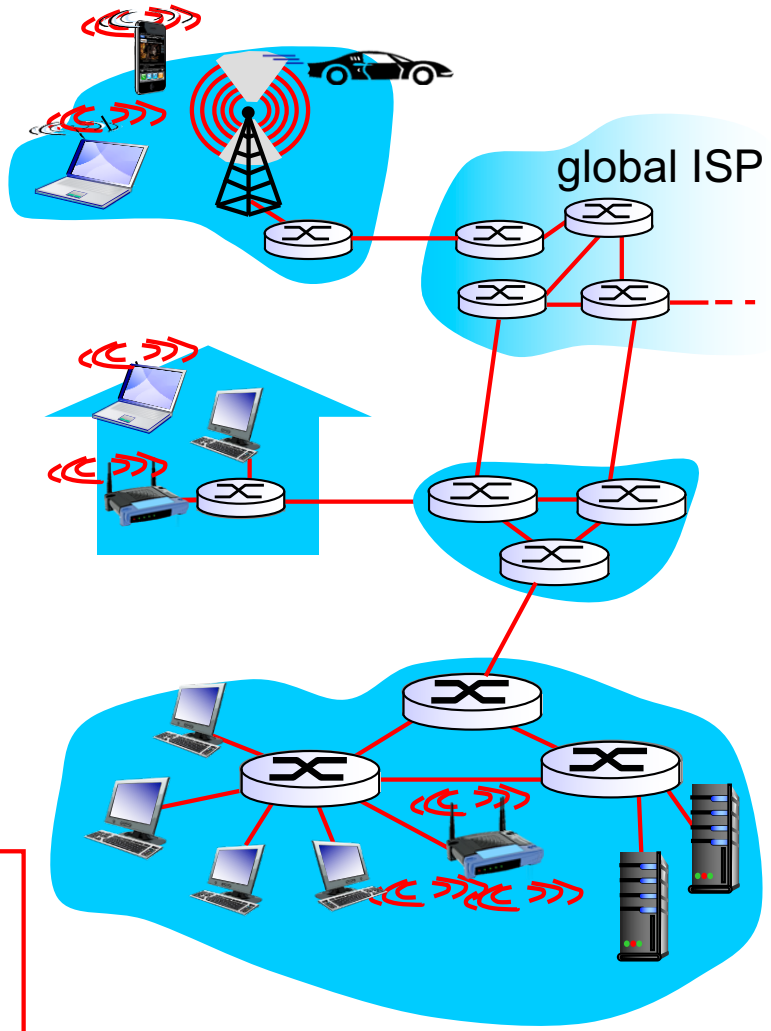
6.4 LANs

- addressing, ARP (address resolution protocol)
- Ethernet

Link layer: introduction

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



Link layer has responsibility of transferring datagram from one node to *physically adjacent* node over a link

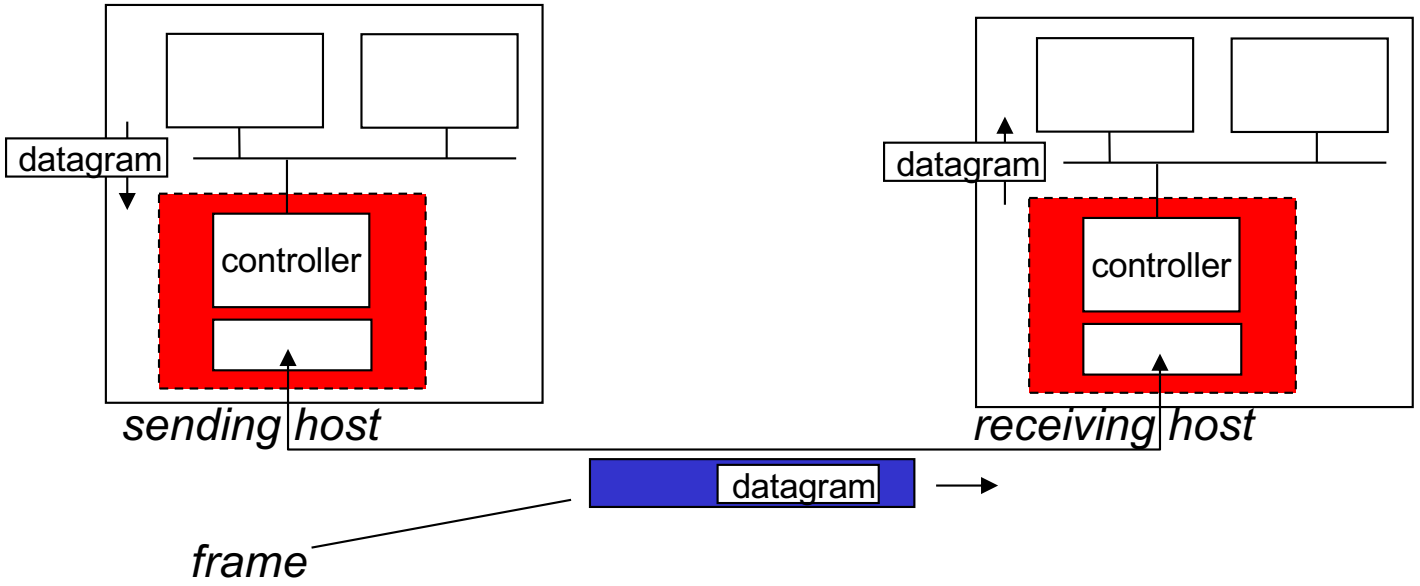
Link layer: context

- 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 Worcester to Minneapolis
  - limo: Worcester to BOS
  - airplane: BOS to MSP
  - train: MSP to Minneapolis
- tourist = datagram
- transport segment = communication link
- transportation mode = link layer protocol
- travel agent = routing algorithm

Adaptors communicating



- sending side:
  - encapsulates datagram in frame
  - adds error checking bits, rdt, etc.
- receiving side
  - looks for errors, rdt, etc
  - extracts datagram, passes to upper layer at receiving side

Link layer, LANs: outline

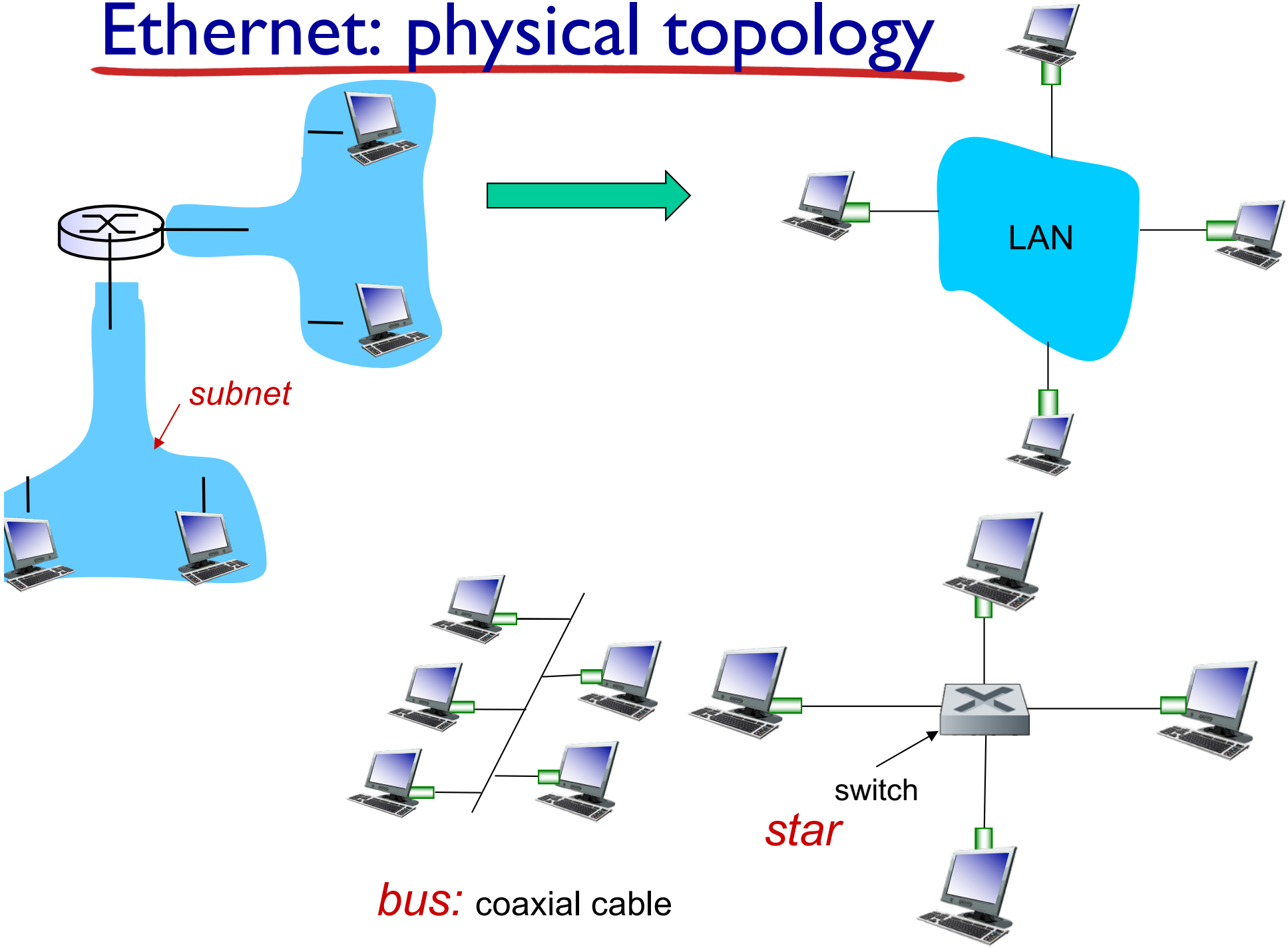
6.1 introduction, services

6.2 error detection, correction

6.4 LANs

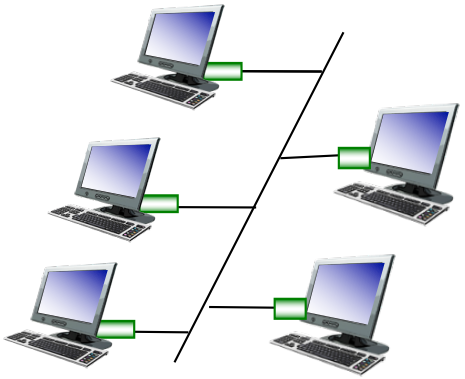
- addressing, ARP
- Ethernet

Ethernet: physical topology

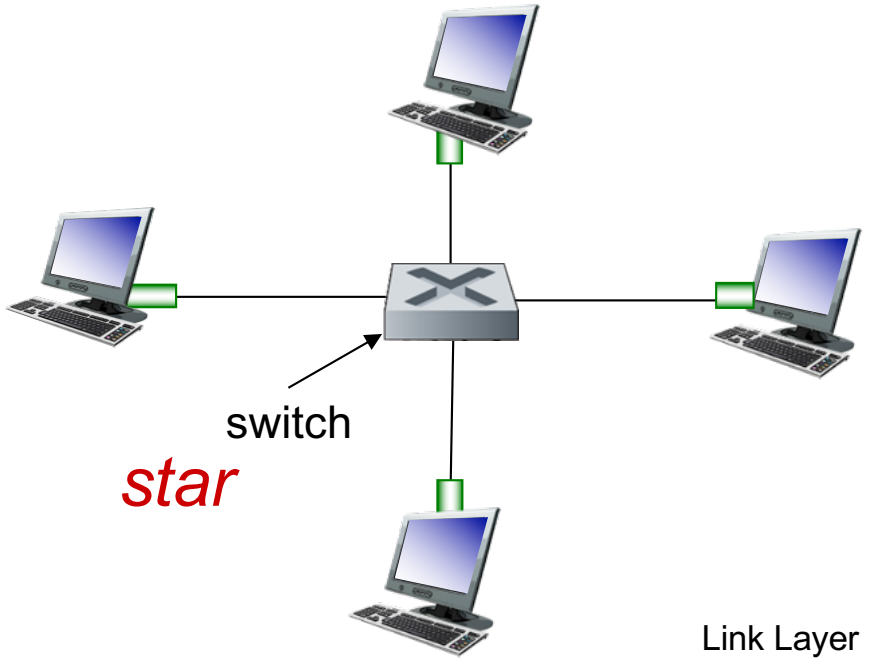


Ethernet: physical topology

- *bus:* popular through mid 90s
  - all nodes in same collision domain (can collide with each other)
- *star:* prevails today
  - active *switch* in center
  - each "spoke" runs a (separate) Ethernet protocol (nodes do not collide with each other)



bus: coaxial cable



LAN: Local area network

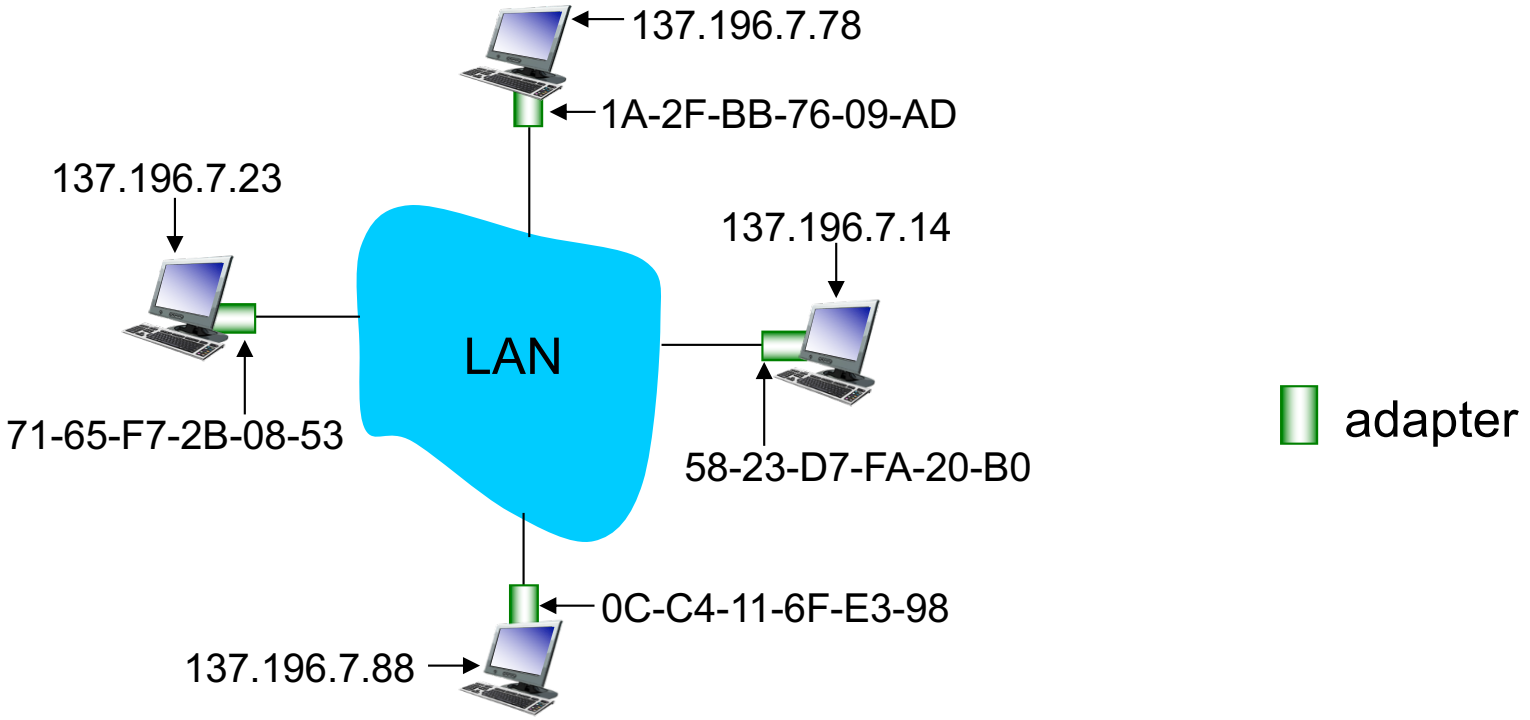
MAC addresses and ARP

- 32-bit IP address:
  - *network-layer* address for interface
  - used for layer 3 (network layer) forwarding
- Media access control (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 (dashed-hexadecimal)

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

LAN, MAC addresses

each adapter on LAN has unique *MAC* address



LAN: Local area network

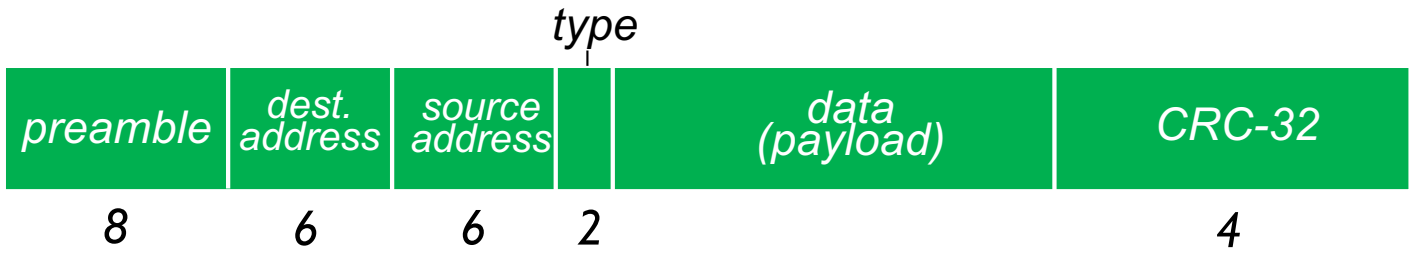
LAN addresses (more)

- MAC address allocation administered by IEEE
- manufacturer buys portion (2^{24}) of MAC address space (to assure uniqueness)
- analogy:
  - MAC address: like Social Security Number
  - IP address: like postal address
  - Domain Name: Person name
- 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

IEEE: Institute of Electrical and Electronics Engineers

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

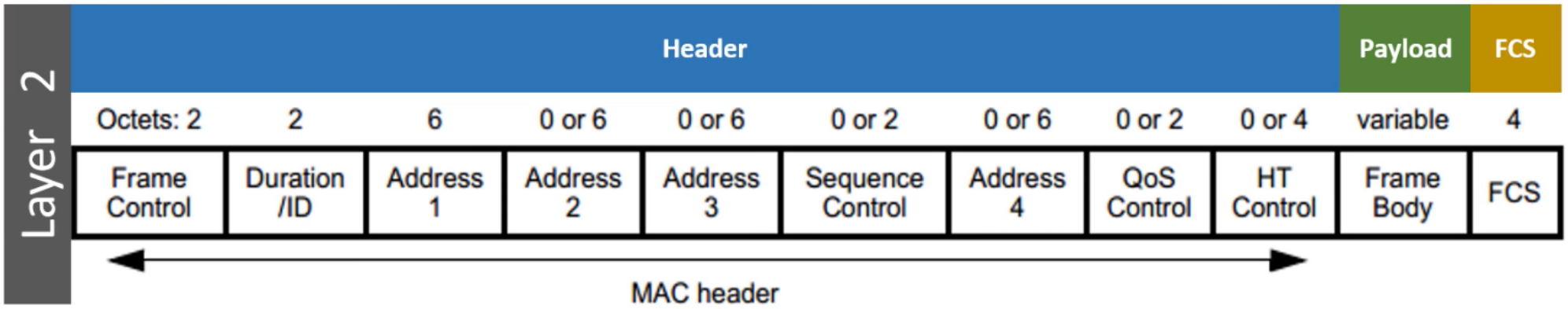
UDP Header?
TCP header?
IPv4 header?

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: (2 bytes)* indicates higher layer protocol (mostly IP but others possible, e.g., Novell IPX, AppleTalk)
- *CRC-32: (4 bytes)*cyclic redundancy check at receiver
  - error detected: frame is dropped

Wireless 802.11 vs Ethernet Frame

- *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



Final exam: this Friday 10/16
Optional Q&A session: Thursday 10/15
We will upload a video for final exam review today.

Quiz 9: 8 points + 1 bonus points
Q1: CRC
Q2: Ethernet frame structure
Q3 (bonus): ARP & MAC address

Link layer, LANs: outline

6.1 introduction, services

6.2 error detection, correction

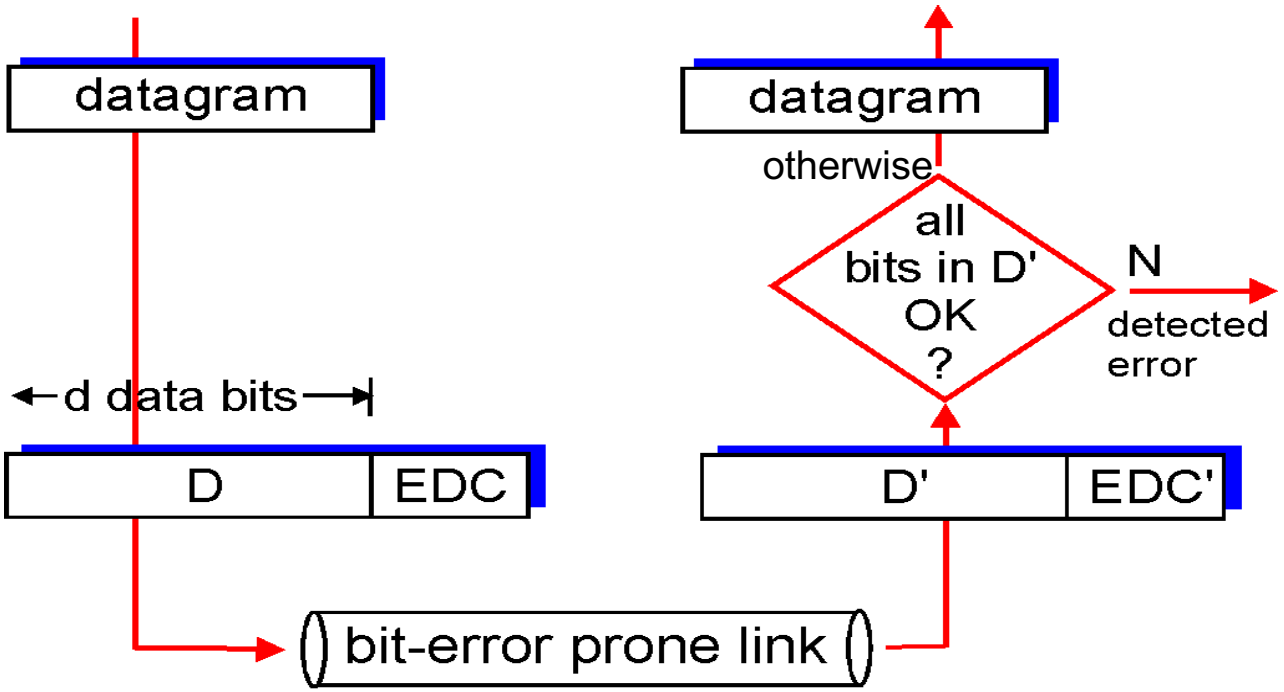
6.4 LANs

- addressing, ARP
- Ethernet

Error detection

EDC= Error Detection and Correction bits
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

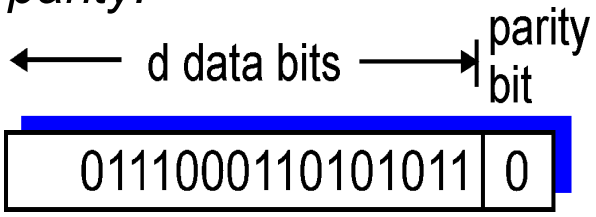


Parity checking

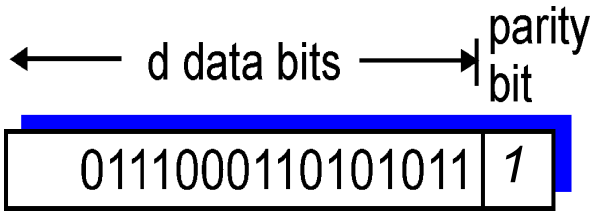
single bit parity:

- *d*etect single bit errors

Odd parity:

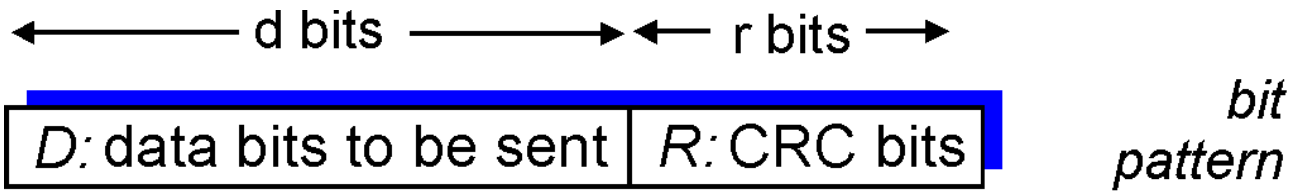


Even parity:



Cyclic redundancy check

- more powerful error-detection coding
- view data bits, D, as a binary number
- choose r+1 bit pattern (generator), G
- goal: choose r CRC bits, R, such that
  - <D,R> exactly divisible by G (modulo 2)
  - receiver knows G, divides <D,R> by G. If non-zero remainder: error detected!
  - can detect all burst errors less than r+1 bits
- widely used in practice (Ethernet, 802.11 WiFi)



$D * 2^r$ XOR R *mathematical formula*

CRC example (binary division, XOR)

Dividing $D \cdot 2^r$ by G yields R

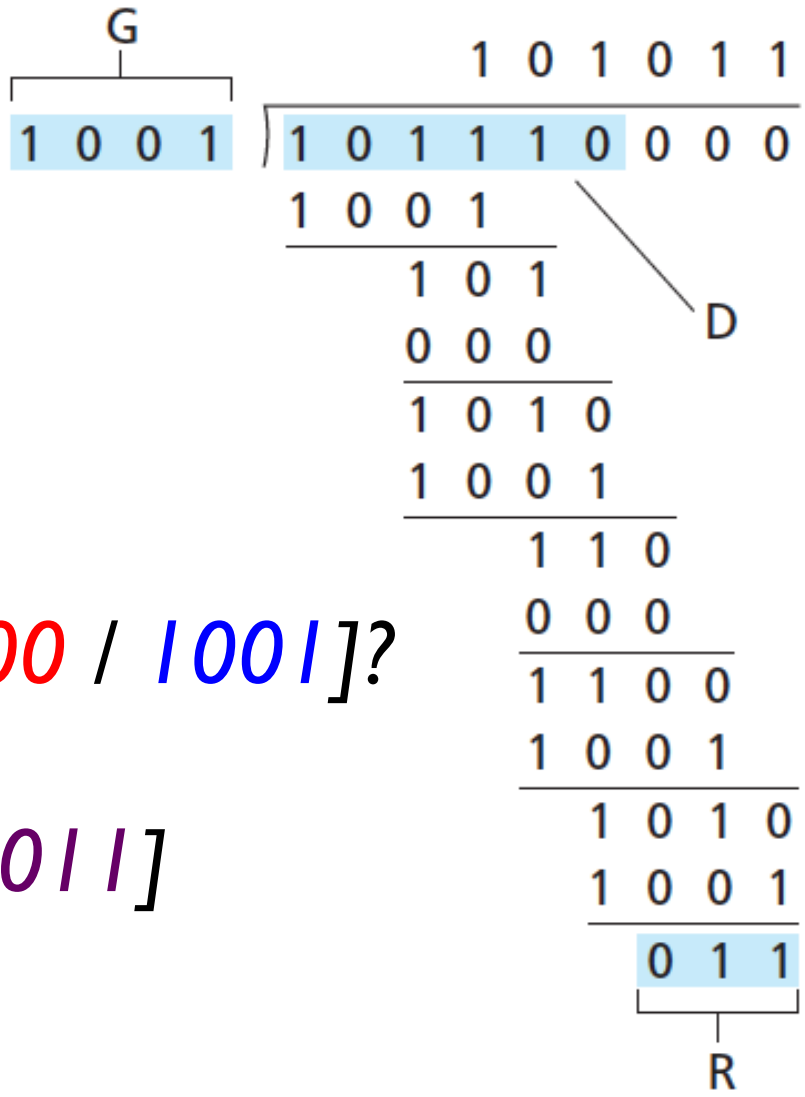
Let D=101110, d=6
Let G=1001, r=3

R= remainder[101110 000 / 1001]?

$$R = remainder\left[\frac{D \cdot 2^r}{G}\right]$$

CRC example

Dividing $D \cdot 2^r$ by G yields R



Let D=101110, d=6
Let G=1001, r=3

R= remainder[101110 000 / 1001]?

R= 011, [D,G]=[101110 011]

$$R = \text{remainder}\left[\frac{D \cdot 2^r}{G}\right]$$

Offline practice: CRC example

Dividing $D \cdot 2^r$ by G yields R

Let D=1011, d=4
Let G=1001, r=3

R= remainder[1011 000 / 1001]?

R= 010, [D,G]=[1011 010]

$$R = remainder[\frac{D \cdot 2^r}{G}]$$

Link layer, LANs: outline

6.1 introduction, services

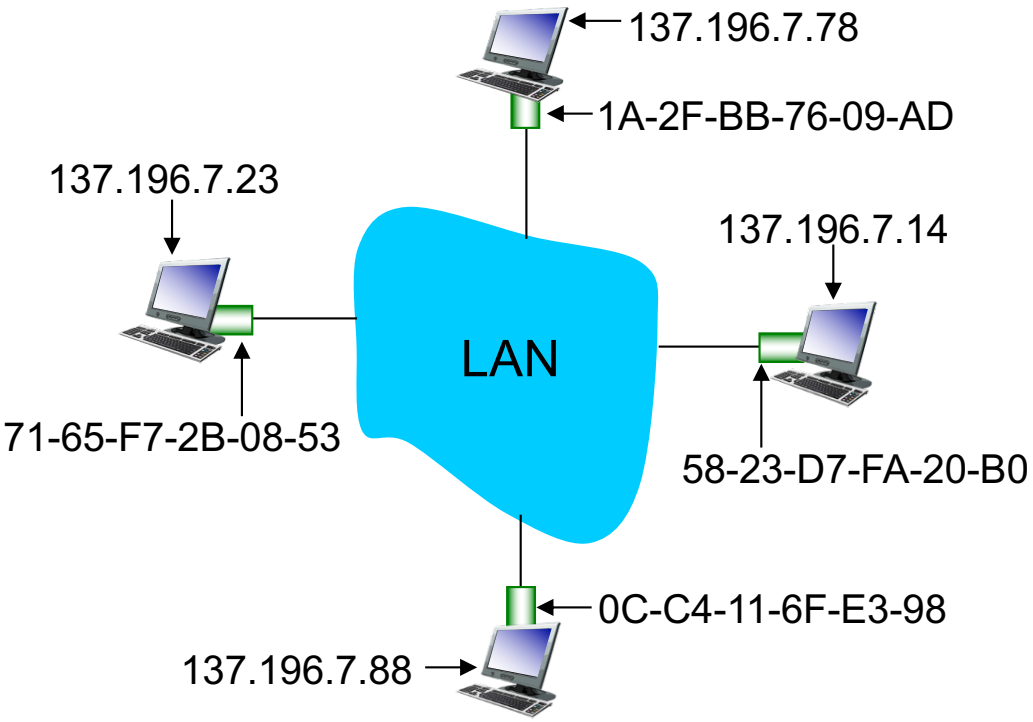
6.2 error detection, correction

6.4 LANs

- addressing, ARP
- Ethernet

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

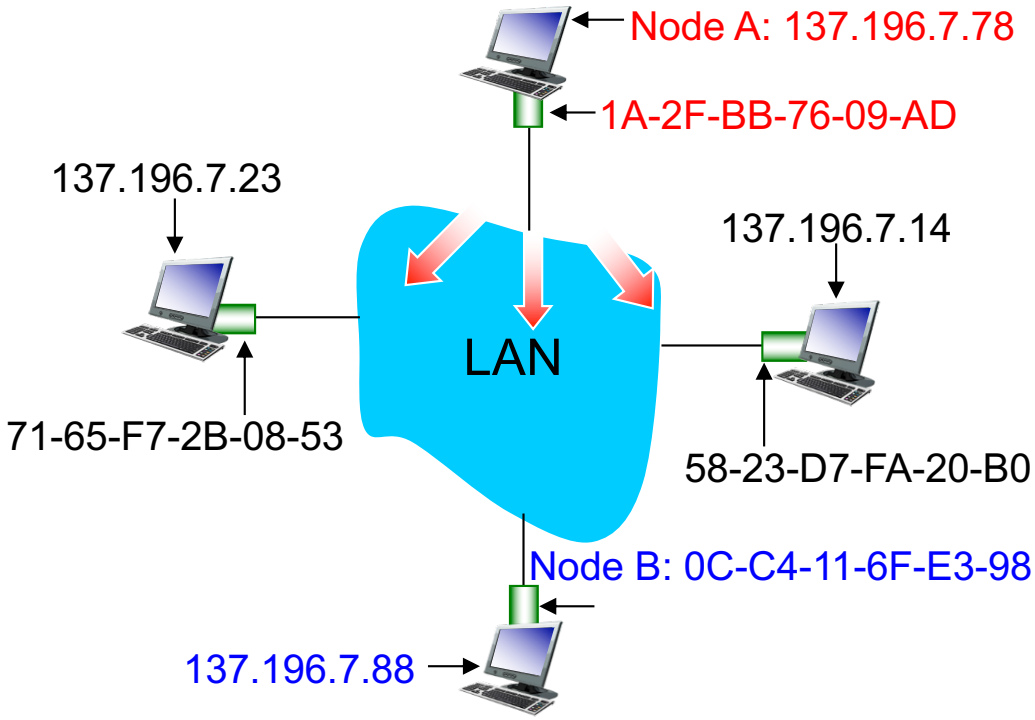
- IP/MAC address mappings for some LAN nodes:

  < IP address; MAC address; TTL>

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

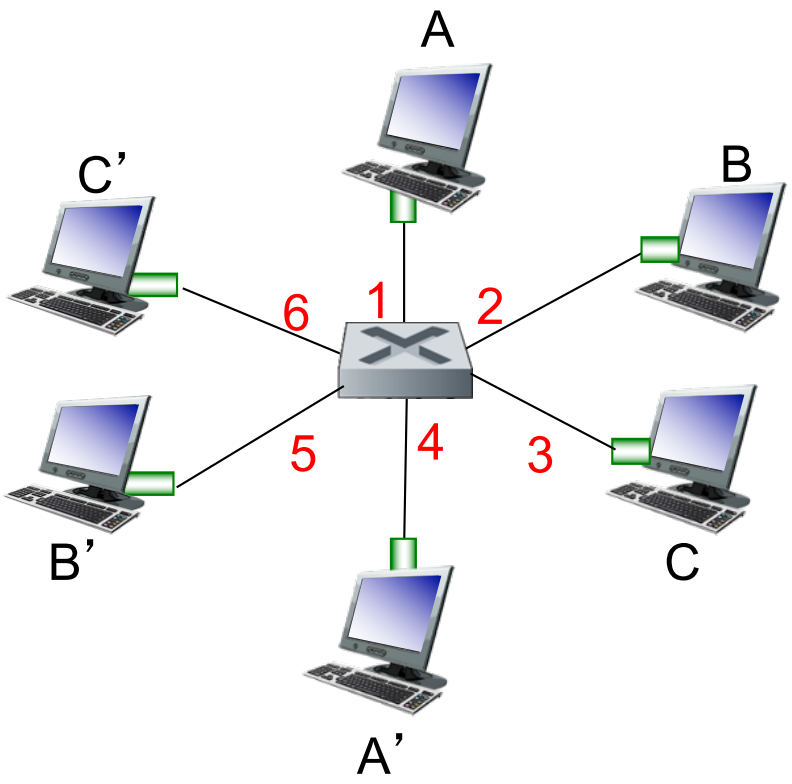
ARP protocol: same LAN

- 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
  - dest 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 times out

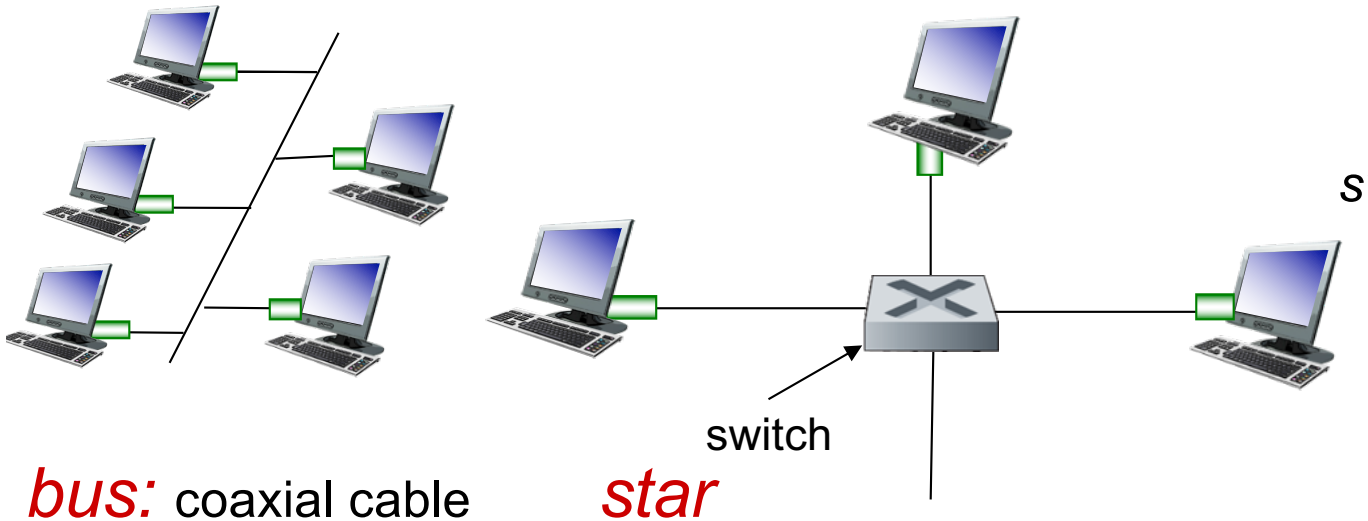


Switch: *multiple* simultaneous transmissions

- switches buffer packets
- 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)



bus: coaxial cable *star*

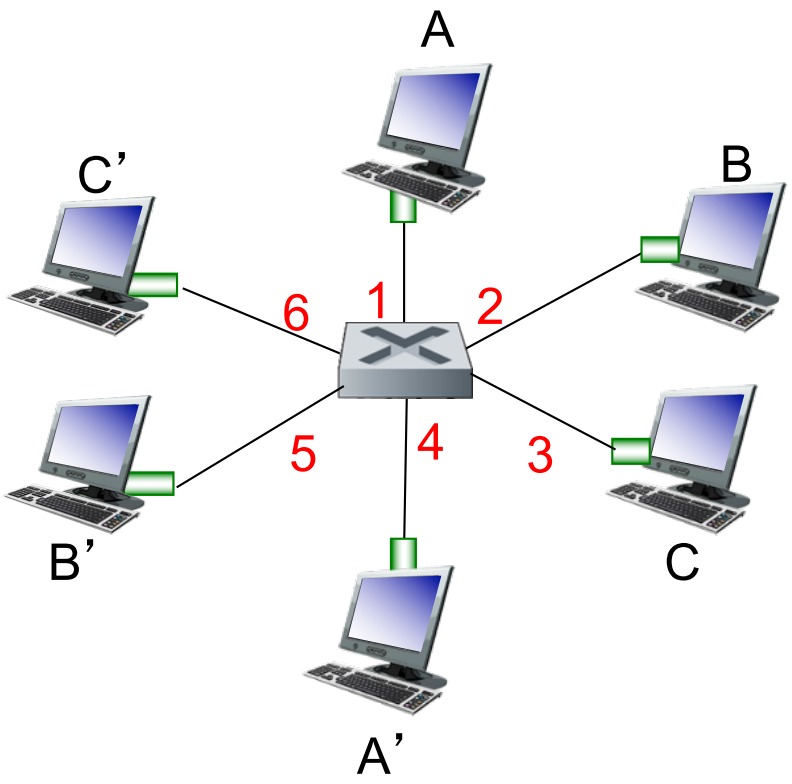
Switch forwarding table

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?*

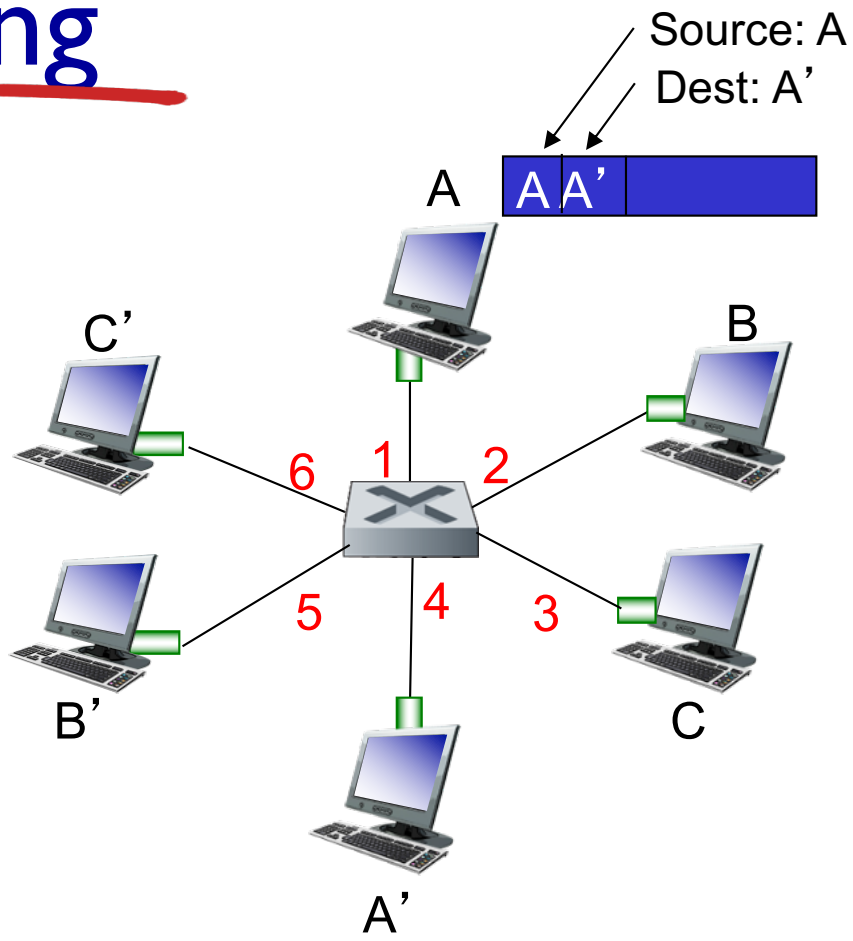
- *something like a routing protocol?*



switch 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



MAC addr	interface	TTL
A	*1*	*60 min*

Switch table (initially empty)

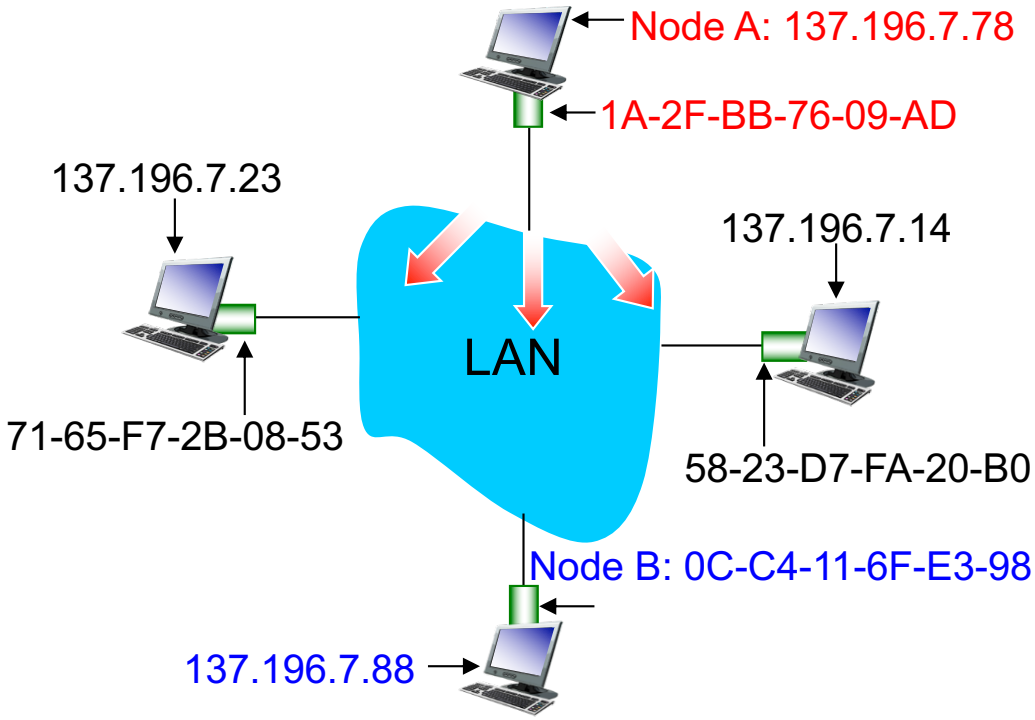
Switch: frame filtering/forwarding

when frame received at switch:

1. record incoming link, MAC address of sending host
2. check 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 */

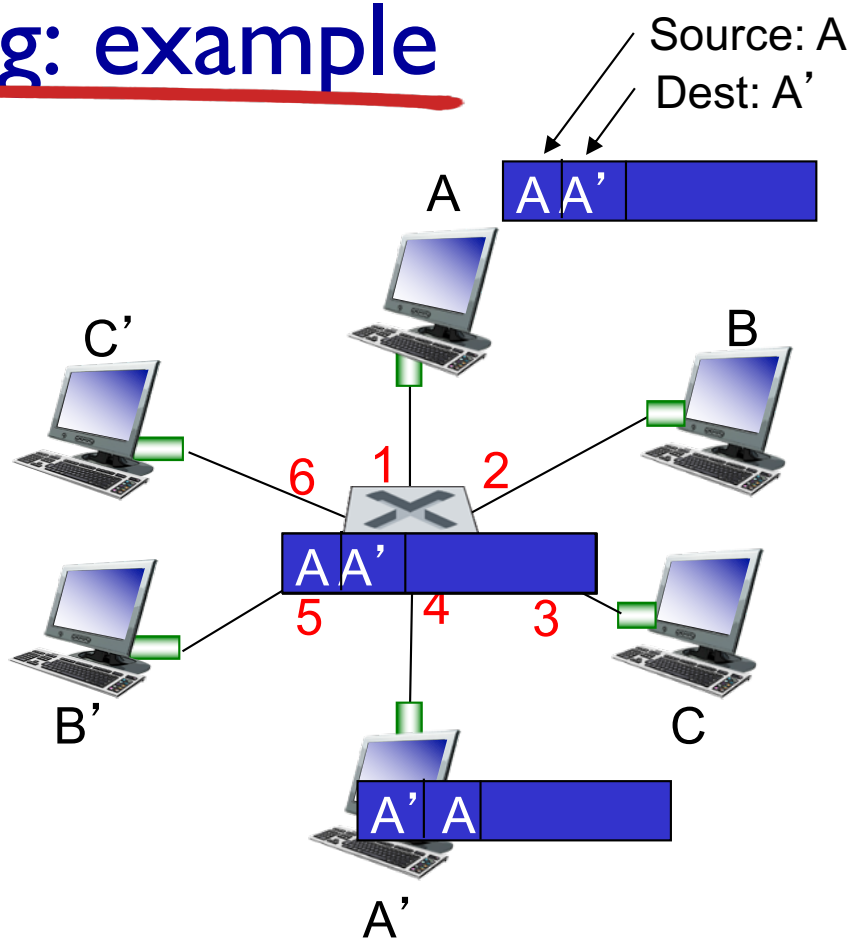
ARP protocol: same LAN

- 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
  - dest 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 times out



Self-learning, forwarding: example

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

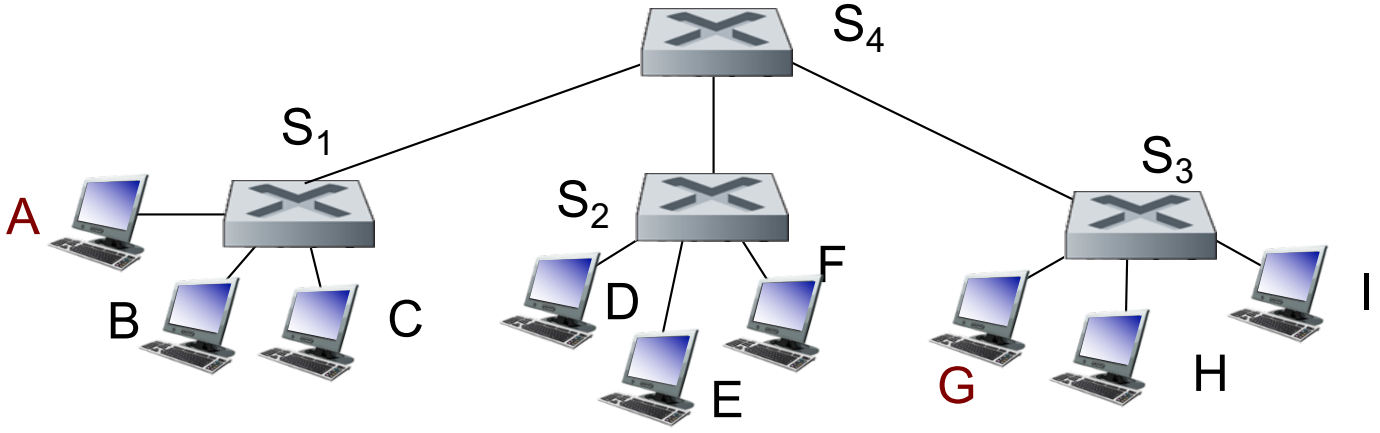


MAC addr	interface	TTL
A	*1*	*60min*
A'	*4*	*60*

switch table (initially empty)

Interconnecting switches

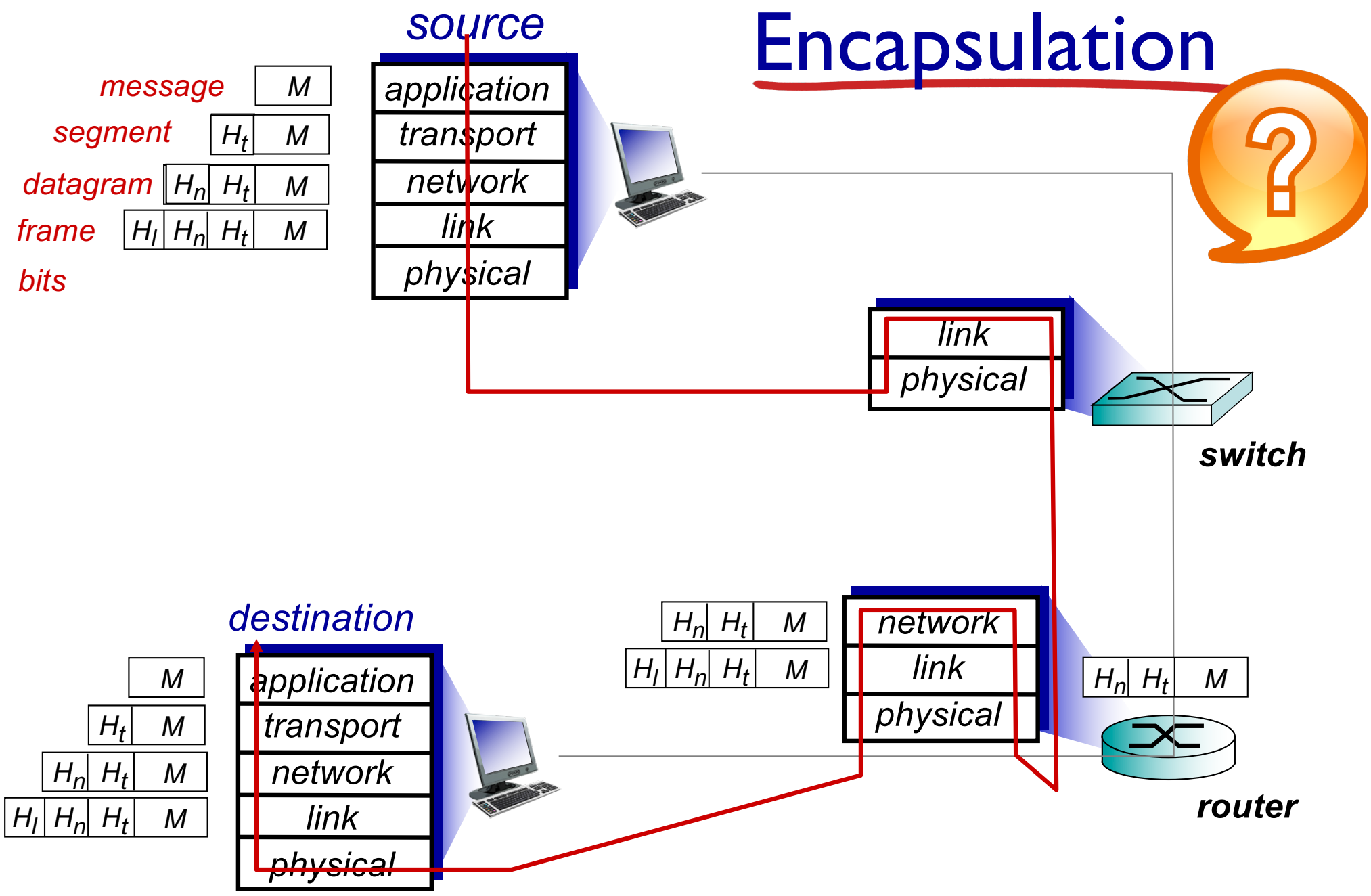
- switches can be connected together



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!)

Encapsulation



To ALL Majors
Opportunity to Pursue a BS/MS in Data Science

Information & Discussion Session

Session Times: Wednesday Oct 14th, 2020 @11:00am

Zoom Link: https://wpi.zoom.us/j/98511661158

Build stronger Resume & Technical skills

Leverage credit double counting

Cost reduction for MS Degree

Complete the combined degree in 4 to 5 years

Program overview: https://web.cs.wpi.edu/~meltabakh/DS-BSMS-Fall2020.pdf

Contact: Prof. Mohamed Eltabakh (meltabakh@wpi.edu)

Questions?

Fille out Class survey on Canvas