CS 3205 COMPUTER NETWORKS

JAN-MAY 2020

LECTURE 8: 4TH FEB 2020

Text book and section(s) covered in this lecture:
Book Kurose and Ross – Sections 5.3, 5.4

P10. Consider two nodes, A and B, that use the slotted ALOHA protocol to contend for a channel. Suppose node A has more data to transmit than node B, and node A's retransmission probability p_A is greater than node B's retransmission probability, p_B.

a. Provide a formula for node A's average throughput. What is the total efficiency of the protocol with these two nodes?

b. If $p_A = 2p_B$, is node A's average throughput twice as large as that of node B? Why or why not? If not, how can you choose p_A and p_B to make that happen?

c. In general, suppose there are N nodes, among which node A has retransmission probability $2p$ and all other nodes have retransmission probability p. Provide expressions to compute the average throughputs of node A and of any other node.

P11. Suppose four active nodes—nodes A, B, C and D—are competing for access to a channel using slotted ALOHA. Assume each node has an infinite number of packets to send. Each node attempts to transmit in each slot with probability p. The first slot is numbered slot 1, the second slot is numbered slot 2, and so on.

a. What is the probability that node A succeeds for the first time in slot 5?

b. What is the probability that some node (either A, B, C or D) succeeds in slot 4?

c. What is the probability that the first success occurs in slot 3?

d. What is the efficiency of this four-node system?

Random Access Protocols

Section 5.3.2

CSMA (carrier sense multiple access)

Basically we apply the two rules of polite human conversation

Listen before speaking:

- *Networking domain - Carrier Sense, Listen before transmit*
- if channel sensed idle: transmit entire frame

If someone else begins to talk at the same time, stop talking:

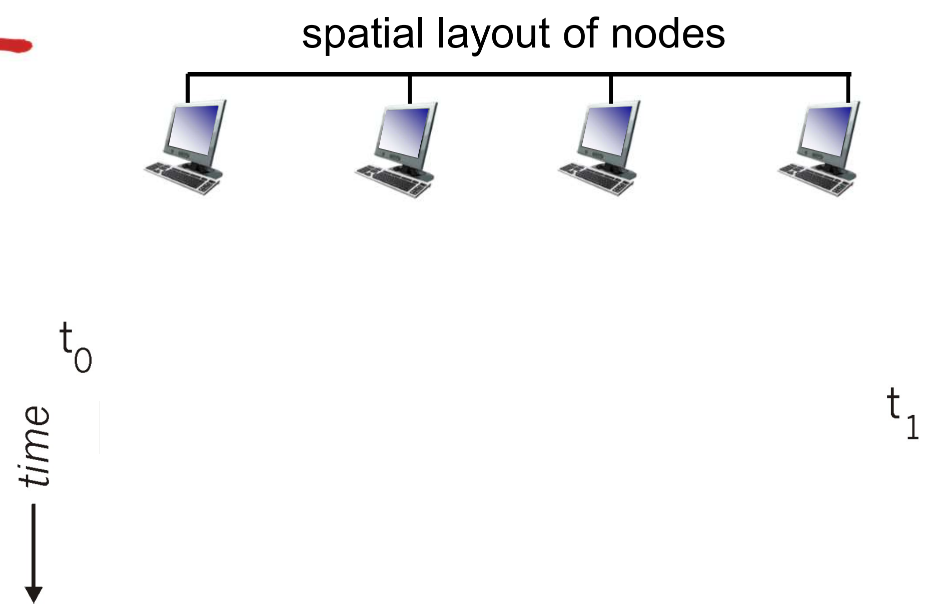
Networking domain – Collision detection

- if channel sensed busy, defer transmission

- human analogy: don't interrupt others!
- Qn: Wont there be any collisions? If we detect and follow the principles?

CSMA collisions

- collisions *can* still occur: propagation delay means two nodes may not hear each other's transmission
- collision: entire packet transmission time wasted
  - distance & propagation delay play role in in determining collision probability

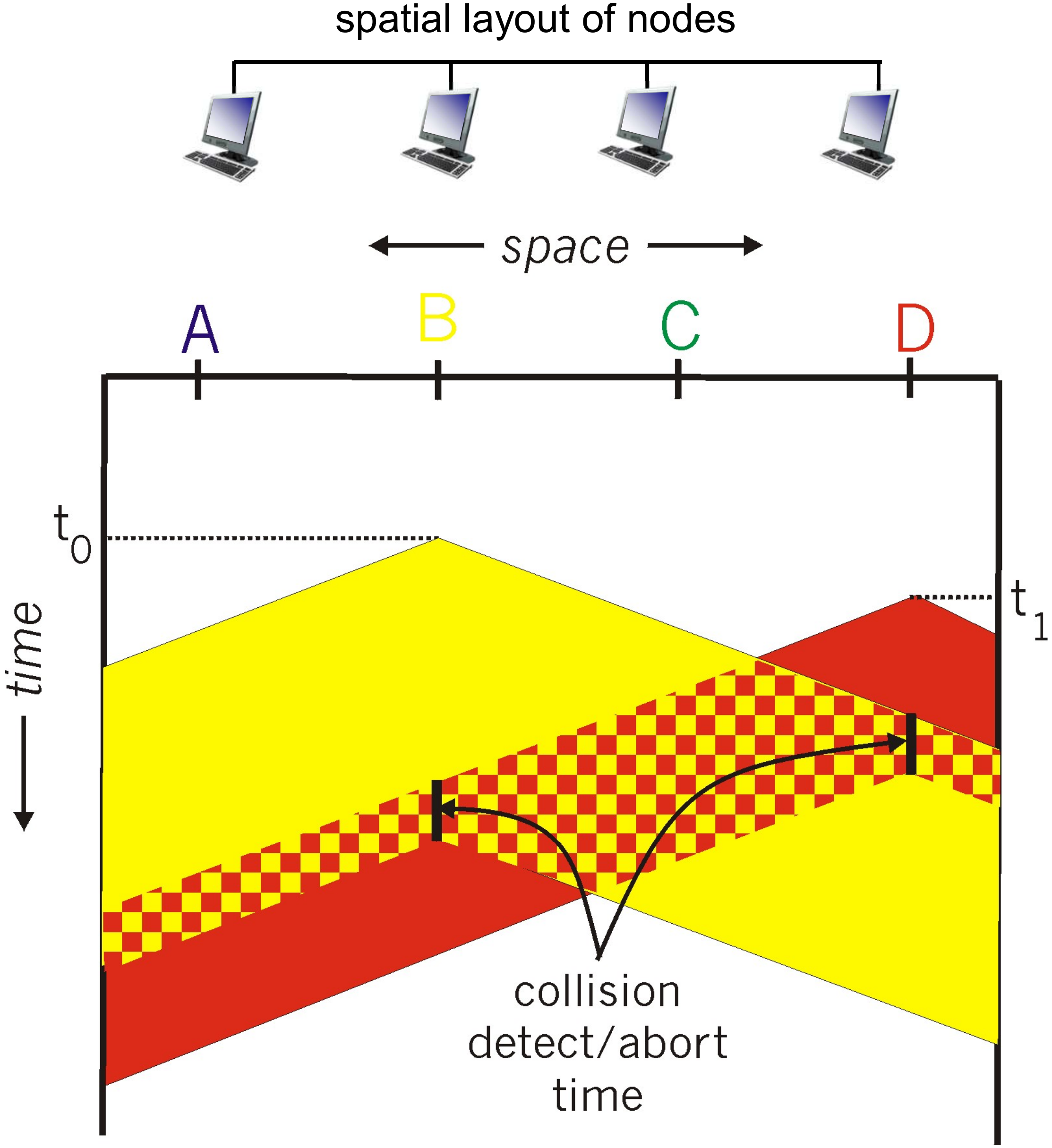


CSMA/CD (collision detection)

CSMA/CD: carrier sensing, deferral as in CSMA
- collisions *detected* within short time
- colliding transmissions aborted, reducing channel wastage

- collision detection:
  - easy in wired LANs: measure signal strengths, compare transmitted, received signals
  - difficult in wireless LANs: received signal strength overwhelmed by local transmission strength
- human analogy: the polite conversationalist

CSMA/CD (collision detection)



Ethernet CSMA/CD algorithm

1. NIC receives datagram from network layer, creates frame
2. If NIC senses channel idle, starts frame transmission. If NIC senses channel busy, waits until channel idle, then transmits.
3. If NIC transmits entire frame without detecting another transmission, NIC is done with frame !
4. If NIC detects another transmission while transmitting, aborts and sends jam signal
5. After aborting, NIC enters *binary (exponential) backoff:*
   - after m^{th} collision, NIC chooses K at random from $\{0,1,2, \ldots, 2^{(m-1)}\}$. NIC waits K*512 bit times, returns to Step 2
   - longer backoff interval with more collisions

NIC – Network Interface Card

Ethernet CSMA/CD algorithm – expanded

- Ethernet – backoff time is **512 bit times** the chosen random **K**
- Node attempts to transmit fist time, collision happens.
  - Backoff is chosen from {0,1}.
  - If 0 is chosen, then immediately the channel is sensed again, if channel free, transmit.
  - If 1 is chosen, wait for 512 bit times, before sensing the channel.
- If on second transmission attempt, there is a collision, then backoff is chosen from {0,1,2,3}. K is equally random among the possible values.
- If on third transmission, collision, K is chosen from {0,1,…7}
- The possible values, grows up to {0,1, …, 1023} or 2^{10}

Ethernet CSMA/CD algorithm – A numerical

- Suppose, a node made fourth attempt. There is a collision. What are the possible values of K?
- Ans: K is chosen from {0, 1, 2, … 13, 14, 15}

- Assume it chose a value k say 10, how many bit times will it wait?
- Ans: 5120 bit times

- What is the probability to choose 10?
- Ans: 1/16

- What is the actual back off time for a 10Mbps link, if 10 was chosen?
- Ans: 1 bit time = $1/(10^* 10^6)$ = 0.1 micro sec. 5120 bit times = 512 micro seconds.

CSMA/CD efficiency

- Efficiency of CSMA/CD – long-run fraction of time during which there is a frames being transmitted on the channel without collisions when there is a large number of active nodes, with each node having a large number of frames to send.
- T_{prop} = max prop delay between 2 nodes in LAN
- t_{trans} = time to transmit max-size frame
- efficiency goes to 1
  - as t_{prop} goes to 0
  - as t_{trans} goes to infinity

$$\boxed{efficiency = \frac{1}{1 + 5t_{prop}/t_{trans}}}$$

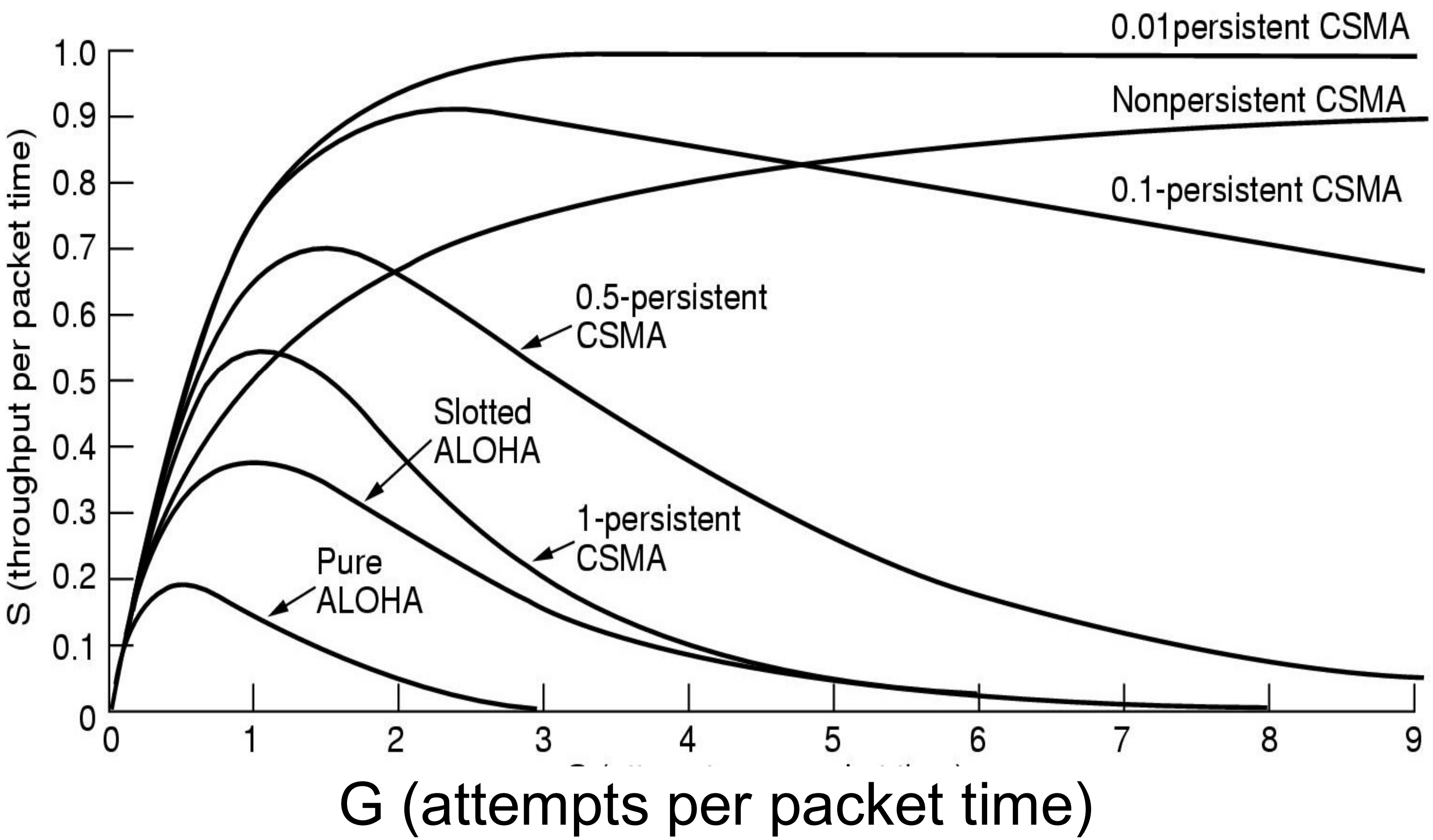
- better performance than ALOHA: and simple, cheap, decentralized!
- Qn: A 10 Mbps Link of 10 mts, handing 1500 byte frames, efficiency is?
- $t_{trans} = 1.2 \times 10^{(-3)}$ $t_{prob} = 50 \times 10^{(-9)}$ $_{\text{Efficiency}} = 99.9979\%$

CSMA Protocols

- There are different variations of the CSMA protocols:
  - 1-persistent CSMA : Sense Channel free, transmit with probability 1, Channel busy, wait. Sense again, immediately to see when channel free.
  - Non-persistent CSMA: Sensing is delayed, random wait, before the sensing is done. Better channel utilization than 1-persistent, but delayed transmission.
  - *p*-persistent CSMA: Applies to slotted channels. If channel idle, transmit with probability *p*. With probability q = 1-p, defer transmission to next available slot.

Sec 2.2 in Tanenbaum

A Comparison



Sec 2.2 in Tanenbaum

Taking Turns Protocols

Section 5.3.3

"Taking turns" MAC protocols

channel partitioning MAC protocols:

- share channel *efficiently* and *fairly* at high load
- inefficient at low load: delay in channel access, 1/N bandwidth allocated even if only 1 active node!

random access MAC protocols

- efficient at low load: single node can fully utilize channel
- high load: collision overhead

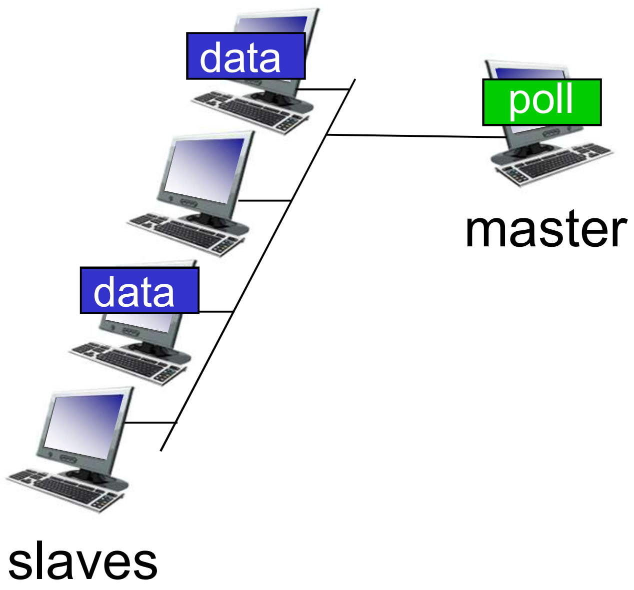
"taking turns" protocols

look for best of both worlds!

"Taking turns" MAC protocols

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)

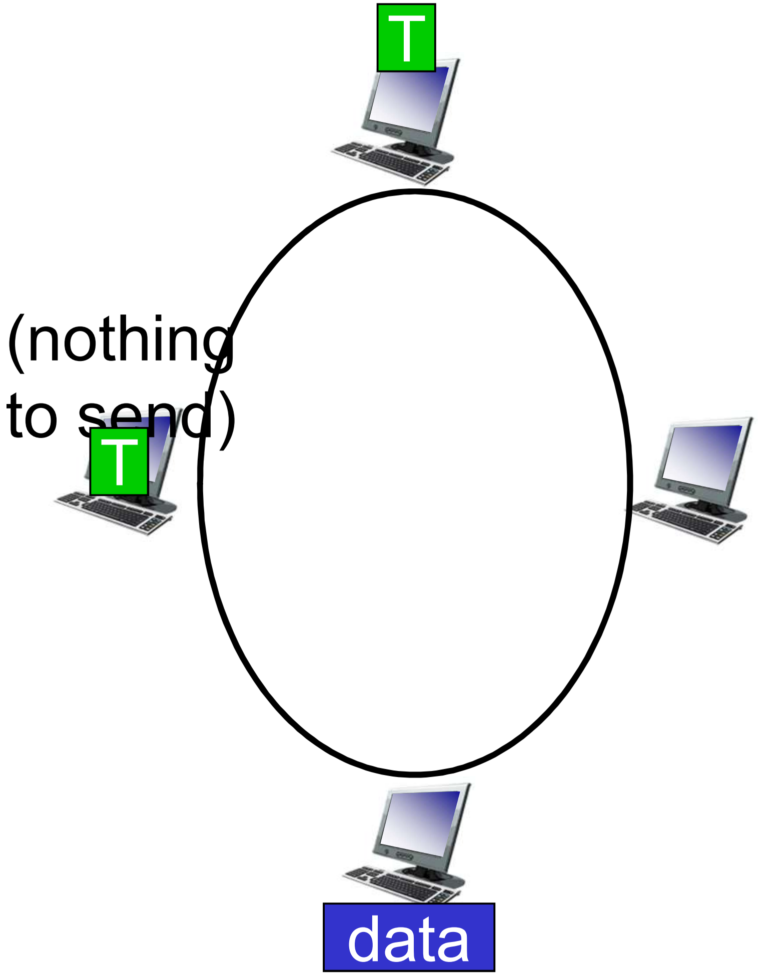


802.15 and Bluetooth are examples for polling protocols

"Taking turns" MAC protocols

token passing:

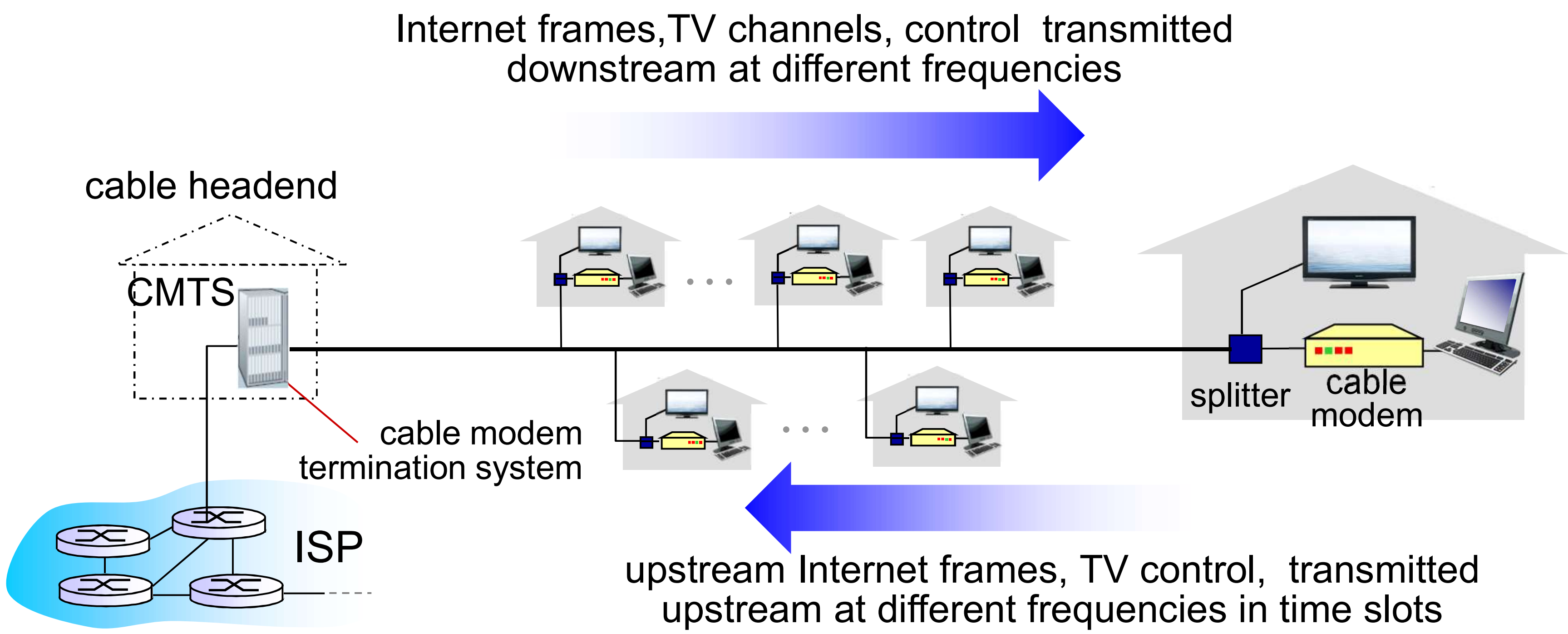
- No master node
- control token passed from one node to next sequentially.
- token message
- concerns:
  - token overhead
  - latency
  - single point of failure (token)



FDDI – (fiber distributed data interface) and IEE 802.5 token ring protocol are examples for token passing protocols

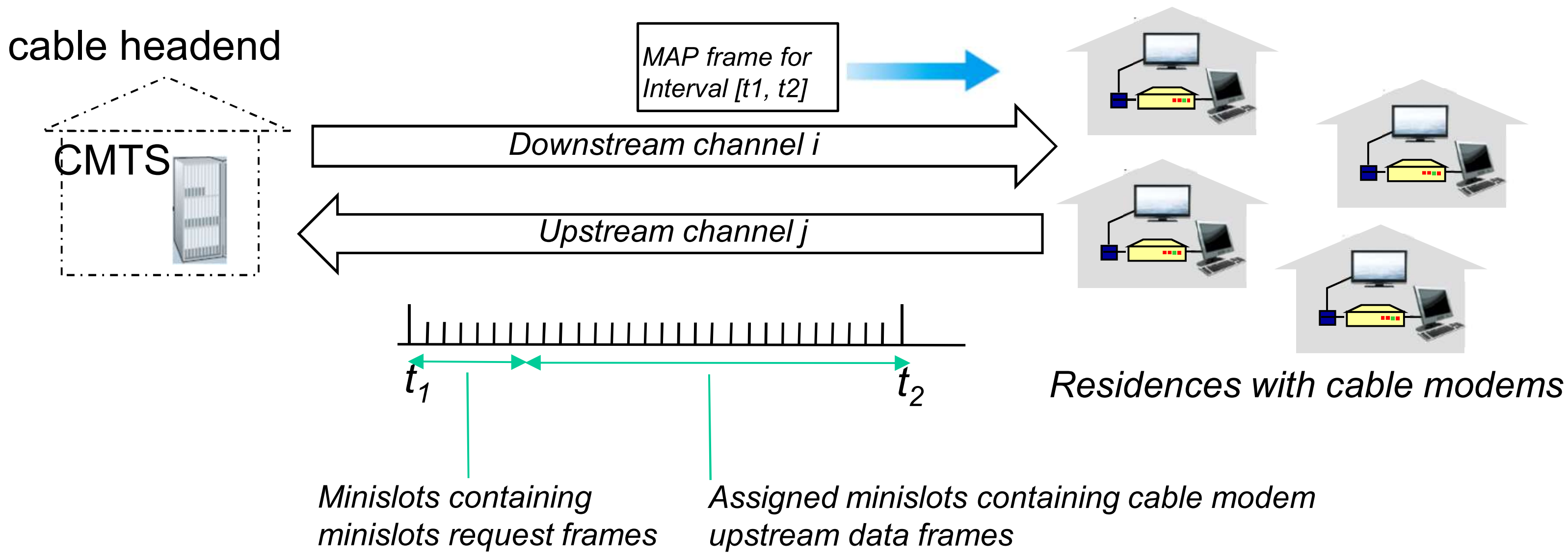
Cable access network

Data-Over-Cable Service Interface Specification (DOCSIS)



- *multiple* 40Mbps downstream (broadcast) channels – FDM used
  - single CMTS transmits into channels – no multi access problem
- *multiple* 30 Mbps upstream channels – FDM
  - *multiple access: all* users contend for certain upstream channel time slots (others assigned) – TDM within a channel.

Cable access network



DOCSIS: data over cable service interface spec

- FDM over upstream, downstream frequency channels
- TDM upstream: some slots assigned, some have contention
  - downstream MAP frame: assigns upstream slots
  - request for upstream slots (and data) transmitted random access (binary backoff) in selected slots

Summary of MAC protocols

- *channel partitioning,* by time, frequency or code
  - Time Division, Frequency Division
- *random access* (dynamic),
  - ALOHA, S-ALOHA, CSMA, CSMA/CD
  - carrier sensing: easy in some technologies (wire), hard in others (wireless)
  - CSMA/CD used in Ethernet
  - CSMA/CA used in 802.11
- *taking turns*
  - polling from central site, token passing
  - bluetooth, FDDI, token ring

Link Layer Addressing and ARP

Section 5.4.1

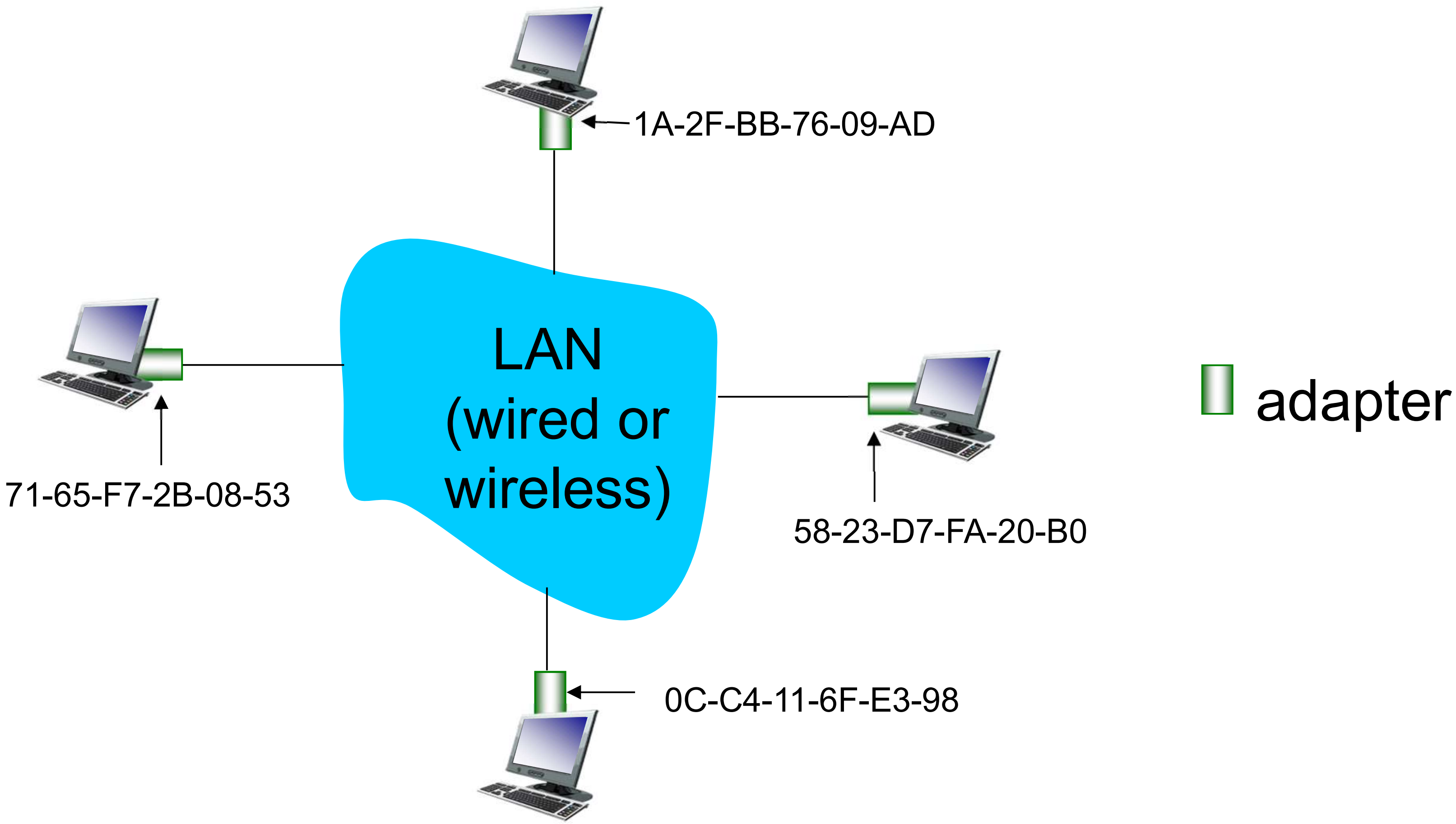
MAC addresses and ARP

- 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 "number" represents 4 bits)

LAN addresses and ARP

each adapter on LAN has unique *LAN* address

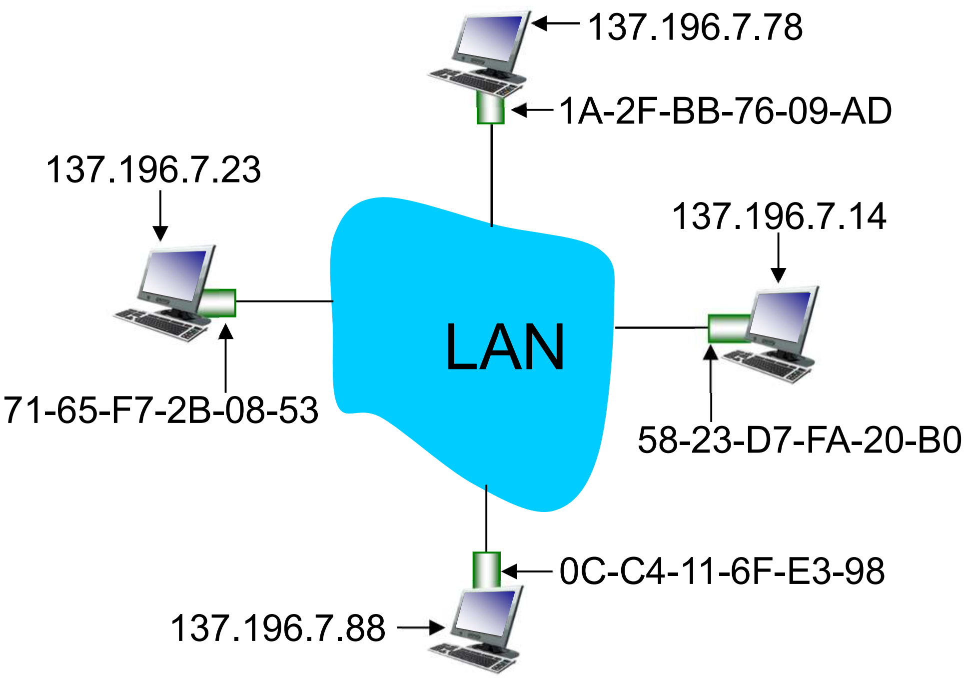


LAN addresses (more)

- 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
- IP hierarchical address *not* portable
  - address depends on IP subnet to which node is attached

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)

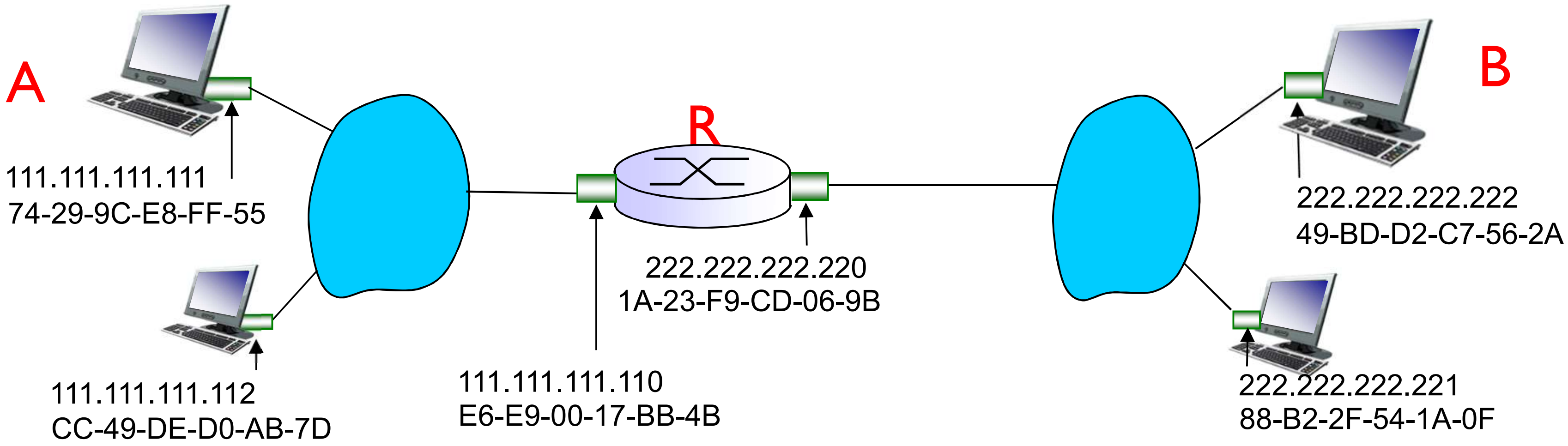
ARP protocol: same LAN

- 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 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

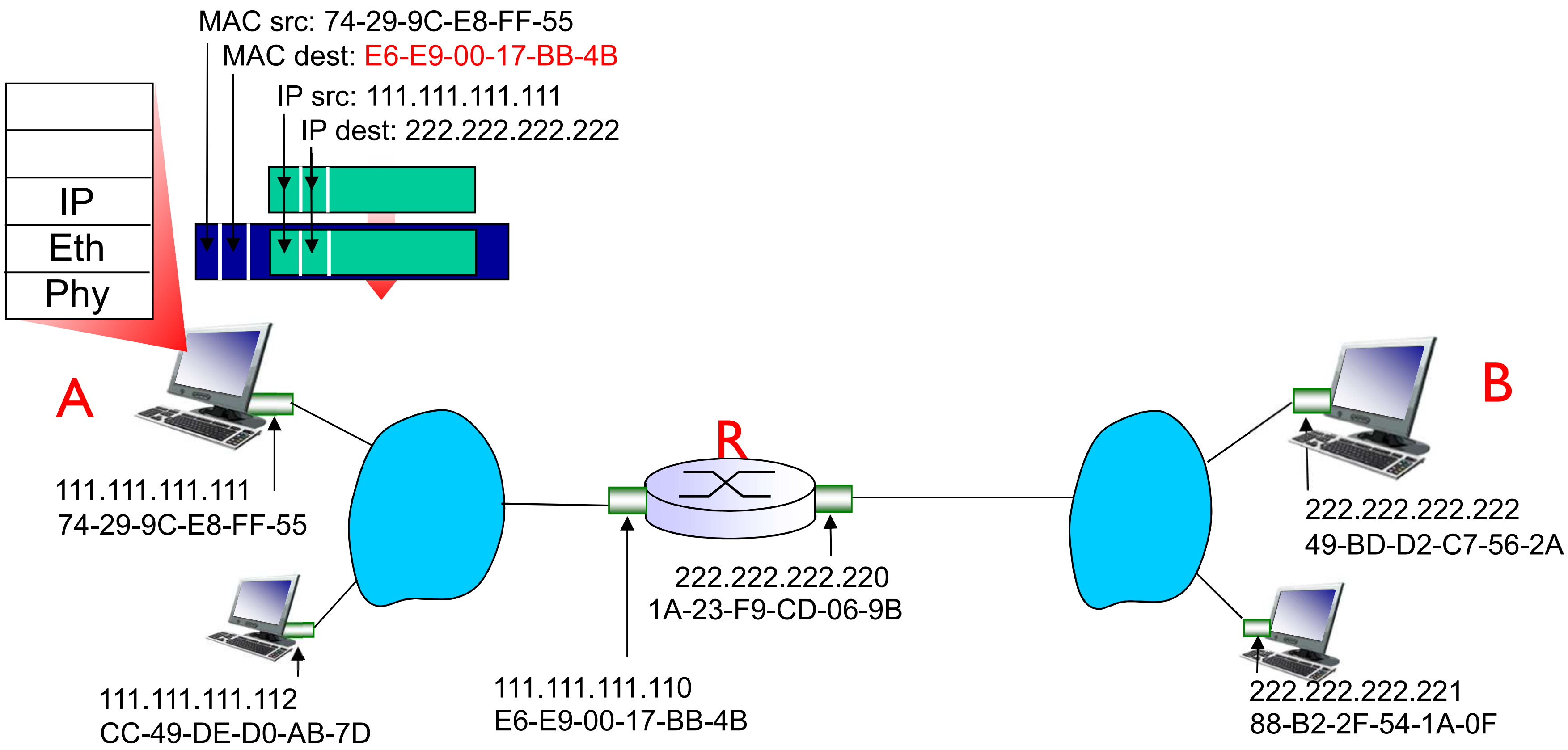
walkthrough: send datagram from A to B via R

- 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 (how?)
- assume A knows R's MAC address (how?)



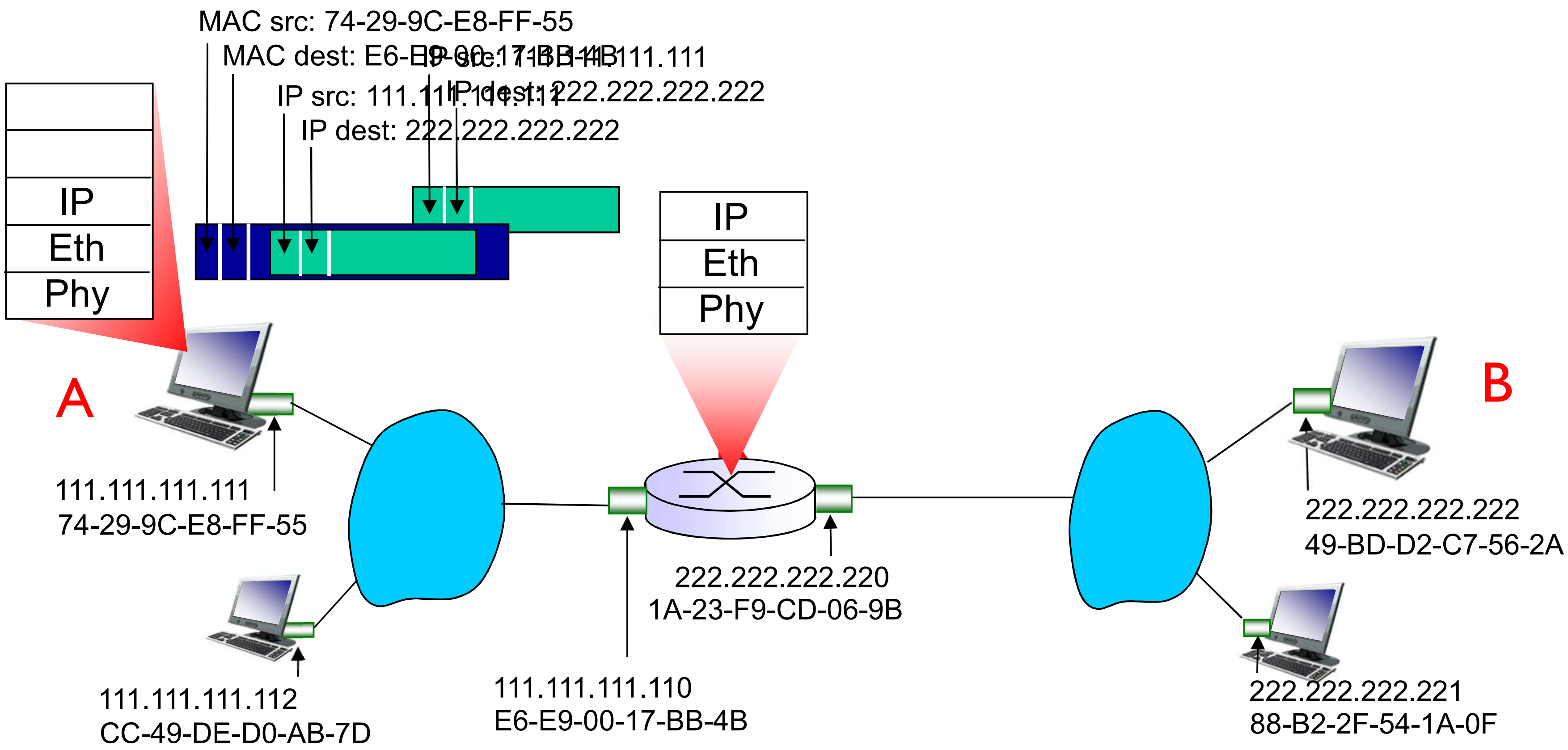
Addressing: routing to another LAN

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



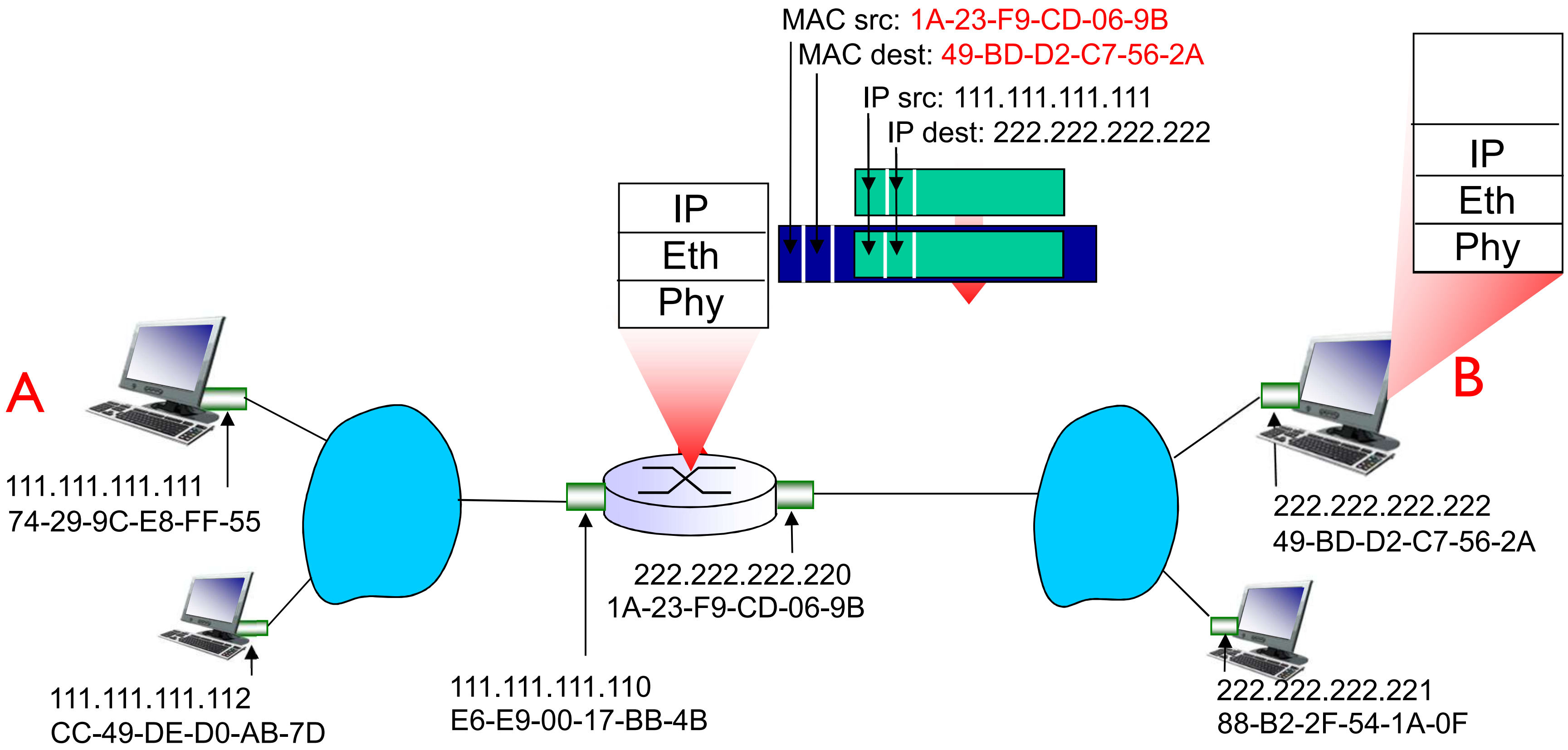
Addressing: routing to another LAN

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



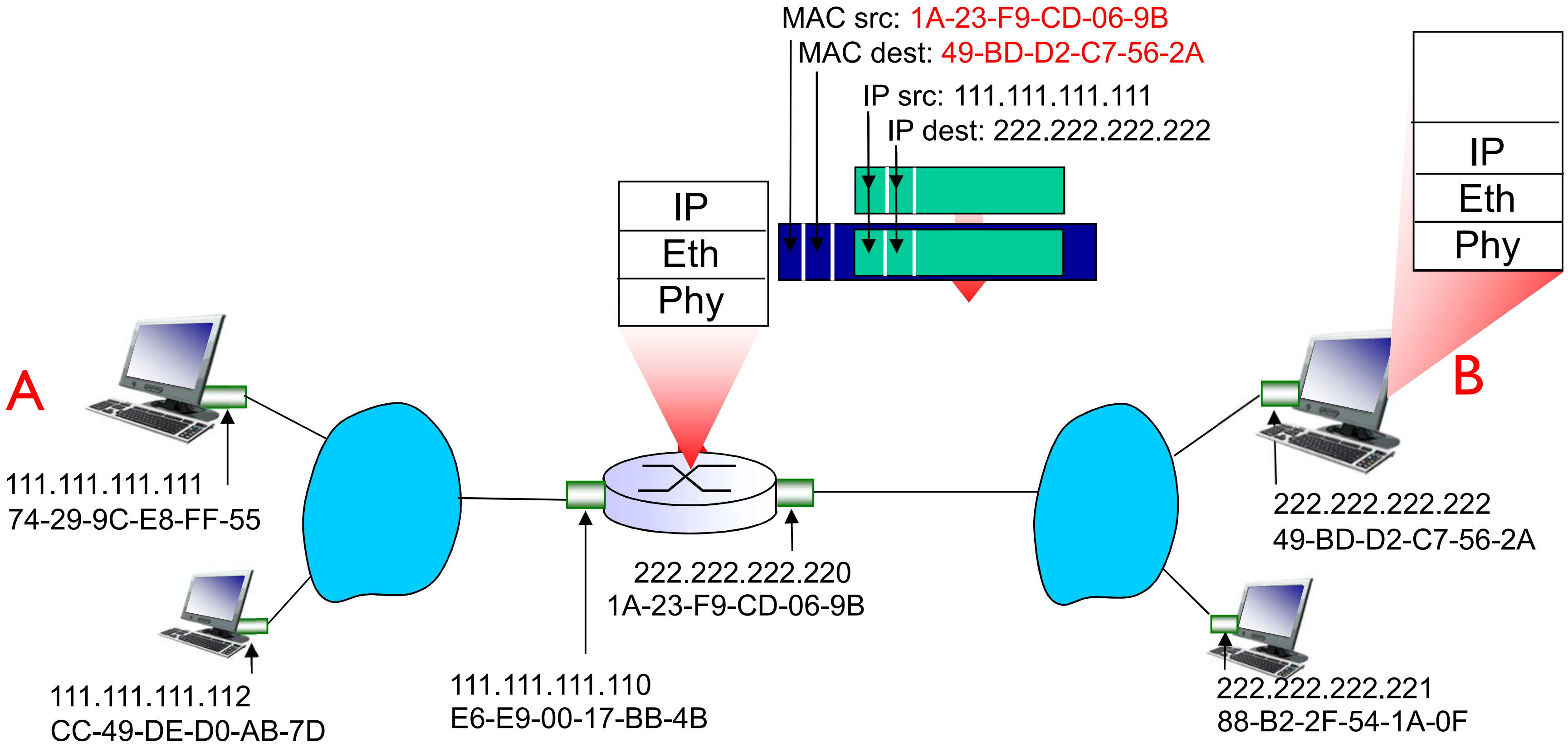
Addressing: routing to another LAN

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



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