DATA COMMUNICATOIN
NETWORKING

Instructor: Ouldooz Baghban Karimi

Course Book & Slides:
Computer Networking, A Top-Down Approach
By: Kurose, Ross

Course Overview

- **Basics of Computer Networks**
  - Internet & Protocol Stack
  - Application Layer
  - Transport Layer
  - Network Layer
  - Data Link Layer

- **Advanced Topics**
  - Case Studies of Computer Networks
  - Internet Applications
  - Network Management
  - Network Security

Random Access Protocols

- **When node has packet to send**
  - transmit at full channel data rate R.
  - no *a priori* coordination among nodes
- **Two or more transmitting nodes**
  - Collision
- **Random access MAC protocol**
  - How to detect collisions
  - How to recover from collisions
- **Examples of random access MAC protocols**
  - slotted ALOHA
  - ALOHA
  - CSMA, CSMA/CD, CSMA/CA

Slotted Aloha

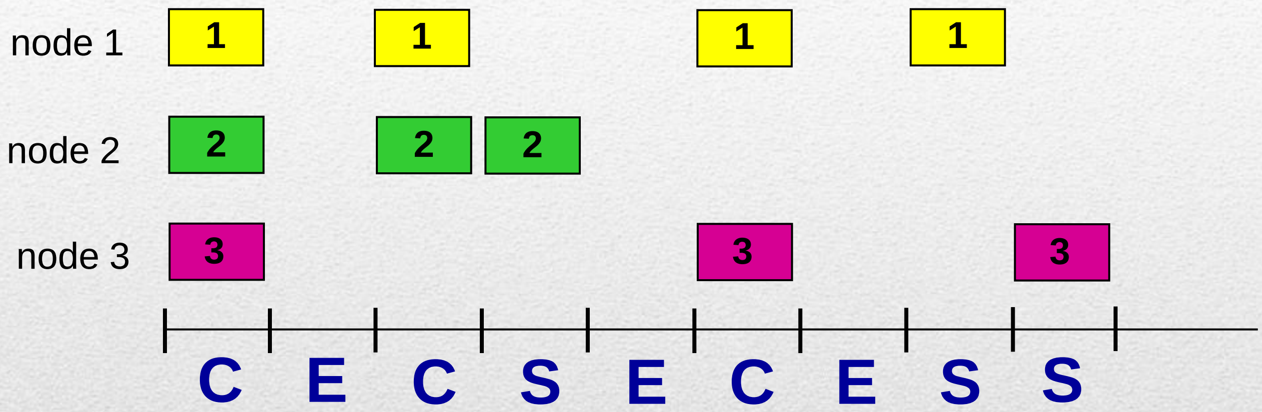
Assumptions:

- All frames same size
- Time divided into equal size slots (time to transmit 1 frame)
- Nodes start to transmit only slot beginning
- Nodes are synchronized
- If 2 or more nodes transmit in slot, all nodes detect collision

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

Slotted Aloha



Pros

- Single active node can continuously transmit at full rate of channel
- Highly decentralized: only slots in nodes need to be in sync
- Simple

Cons

- Collisions, wasting slots
- Idle slots
- Nodes may be able to detect collision in less than time to transmit packet
- Clock synchronization

Slotted Aloha Efficiency

- Suppose: N nodes with many frames to send, each transmits in slot with probability p
- Probability that given node has success in a slot $= p(1-p)^{N-1}$
- Probability that any node has a success $= Np(1-p)^{N-1}$

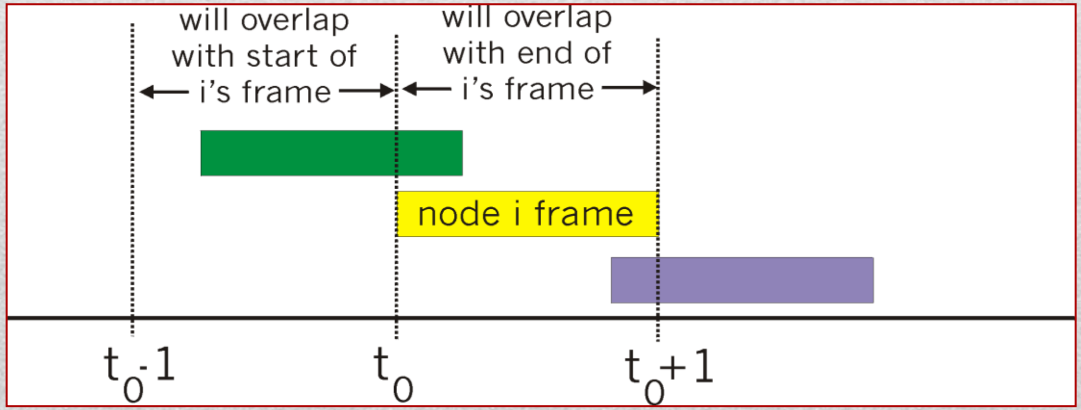
- Max efficiency: find p* that maximizes $Np(1-p)^{N-1}$
- For many nodes, take limit of $Np^*(1-p^*)^{N-1}$ as N goes to infinity, gives:

max efficiency = 1/e = 0.37

efficiency: long-run fraction of successful slots
(many nodes, all with many frames to send)

Pure (Un-Slotted) Aloha

- **Unslotted Aloha**
  - Simpler
  - No synchronization
- **Frame arrival**
  - Transmit immediately
- **Collision probability increases**
  - Frame sent at t_0 collides with other frames sent in $[t_0-1, t_0+1]$



Pure Aloha Efficiency

- P(success by given node) = P(node transmits) ·
- P(no other node transmits in $[t_0-1,t_0]$ ·
- P(no other node transmits in $[t_0-1,t_0]$

$$= p \cdot (1-p)^{N-1} \cdot (1-p)^{N-1}$$

$$= p \cdot (1-p)^{2(N-1)}$$

… choosing optimum p and then letting n $\rightarrow \infty$

$$= 1/(2e) = .18$$

even *worse* than slotted Aloha!

CSMA

Carrier Sense Multiple Access

- Listen before transmit
  - If channel sensed idle
    - Transmit entire frame
  - If channel sensed busy
    - Defer transmission
- Human analogy: don’t interrupt others!

CSMA Collisions

- **Collisions can still occur**
  - Propagation delay means two nodes may not hear each other's transmission
- **When Collision Happens**
  - Entire packet transmission time wasted
  - Distance & propagation delay play role in in determining collision probability

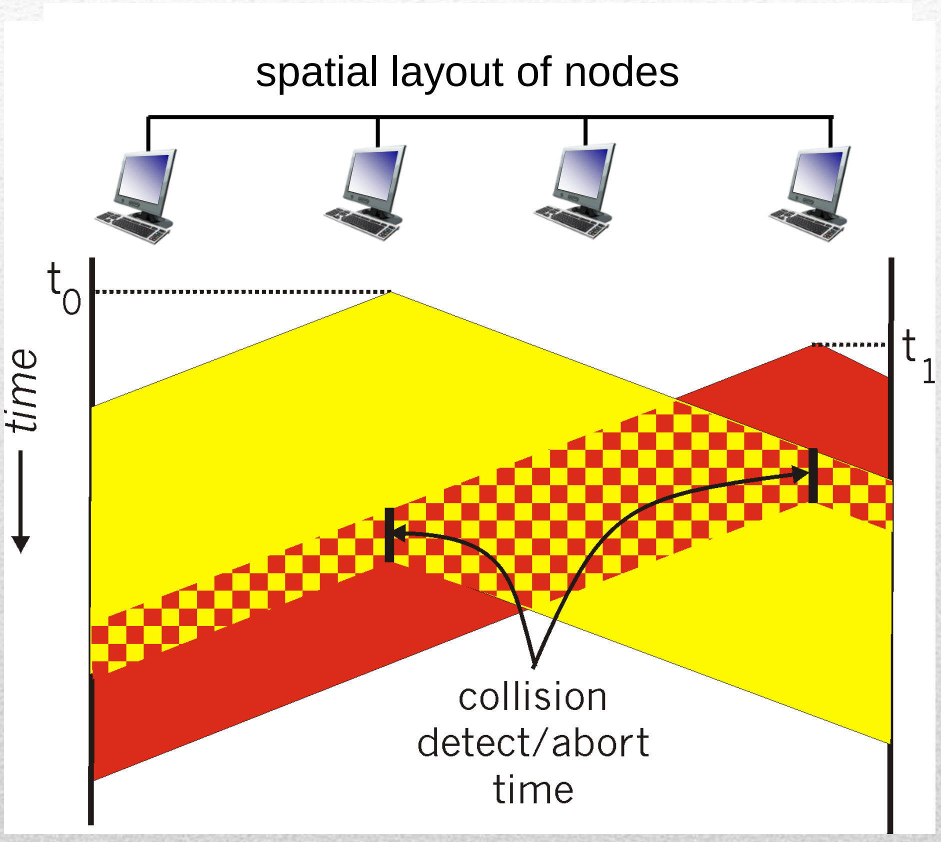
CSMA / CD

- Carrier sensing, deferral as in CSMA
- Collisions *detected* within short time
- Colliding transmissions aborted, reducing channel wastage

- **CD: 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


spatial layout of nodes
t_0
t_1
time
collision detect/abort time

Ehternet CSMA / CD Algorithm

- NIC receives datagram from network layer, creates frame
- If NIC senses channel idle, starts frame transmission. If NIC senses channel busy, waits until channel idle, then transmits.
- If NIC transmits entire frame without detecting another transmission, NIC is done with frame !
- 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 $\{0,1,2, \ldots, 2^m-1\}$. NIC waits K·512 bit times, returns to Step 2
  - longer backoff interval with more collisions

CSMA / CD Efficiency

- T_{prop} = max prop delay between 2 nodes in LAN
- t_{trans} = time to transmit max-size frame

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

- Efficiency goes to 1
  - As t_{prop} goes to 0
  - As t_{trans} goes to infinity
- Better performance than ALOHA
  - Simple
  - Cheap
  - Decentralized!

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

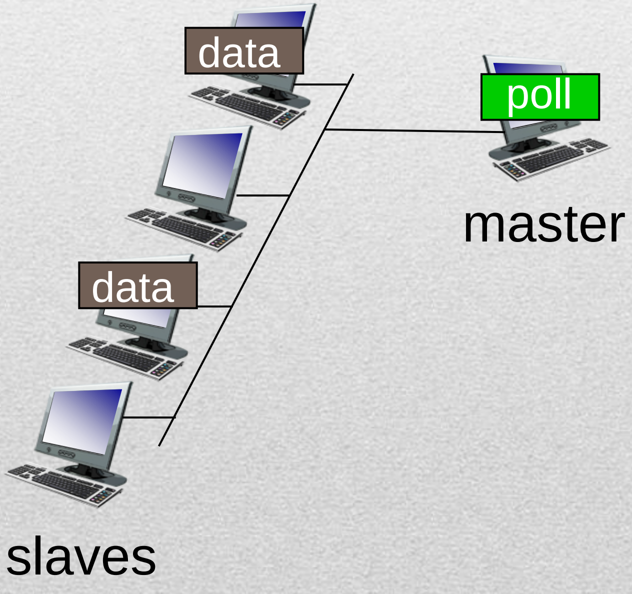
Taking turns protocols

- Look for best of both worlds!

Taking Turns MAC Protocols

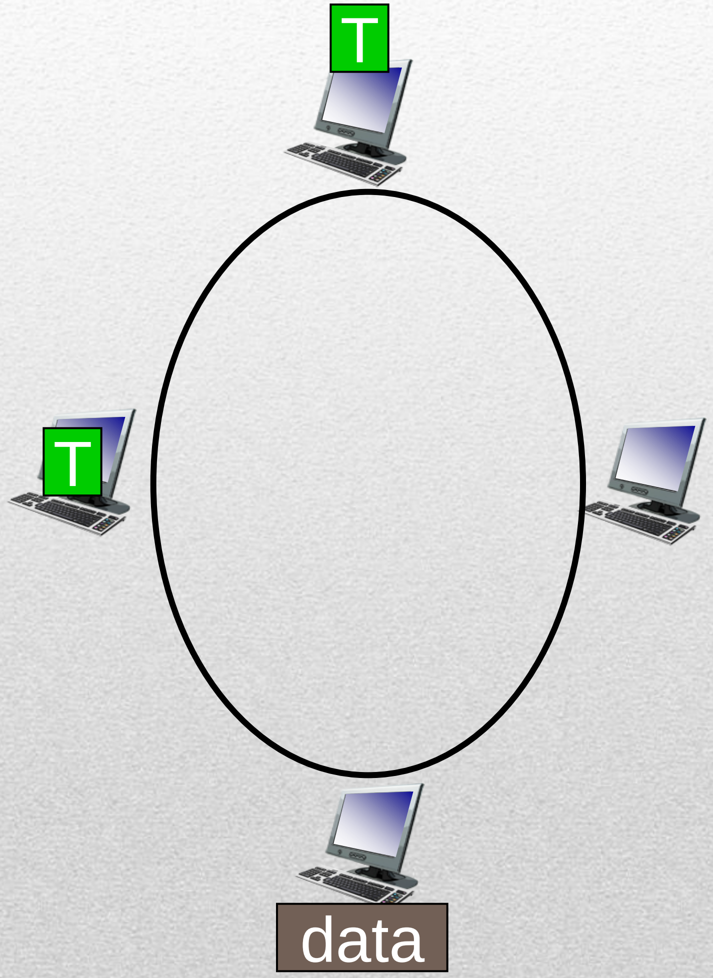
Polling:

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

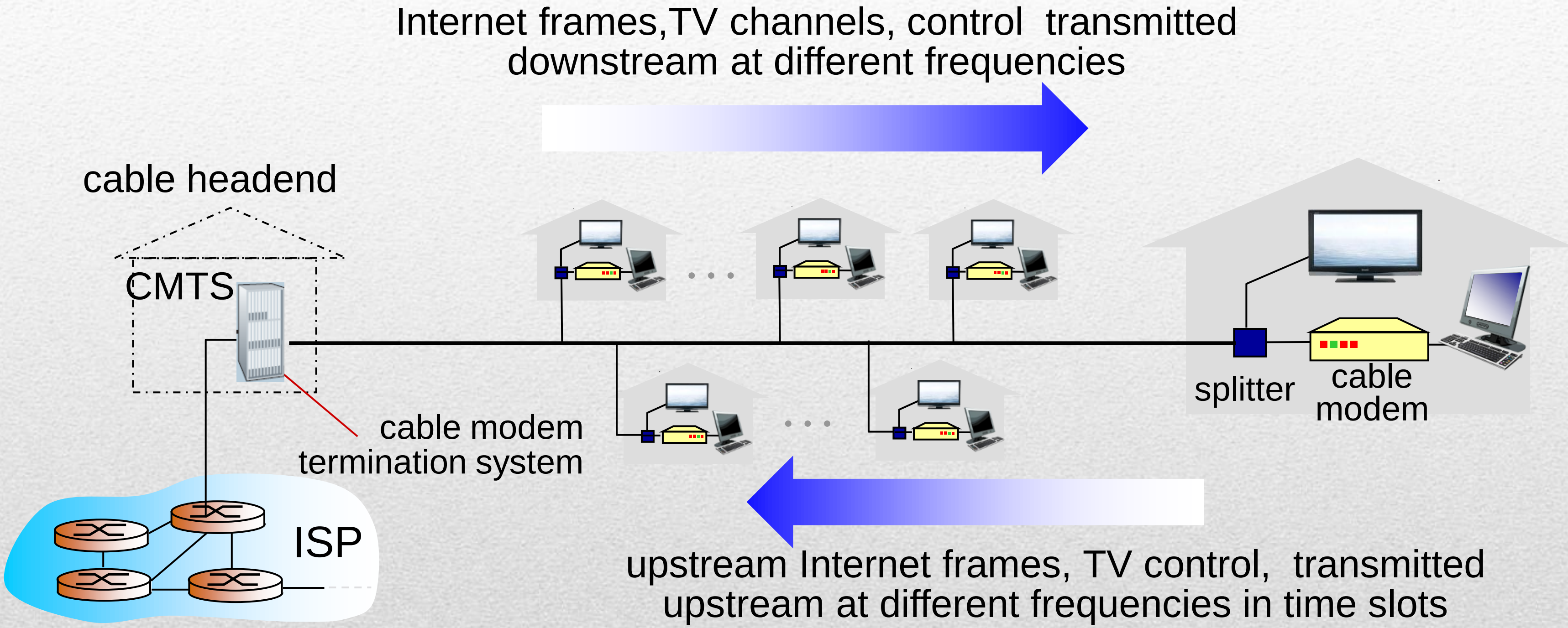


Taking Turns MAC Protocols

- **Token passing**
  - Control token passed from one node to next sequentially.
  - Token message
  - Concerns:
    - Token overhead
    - Latency
    - Single point of failure (token)

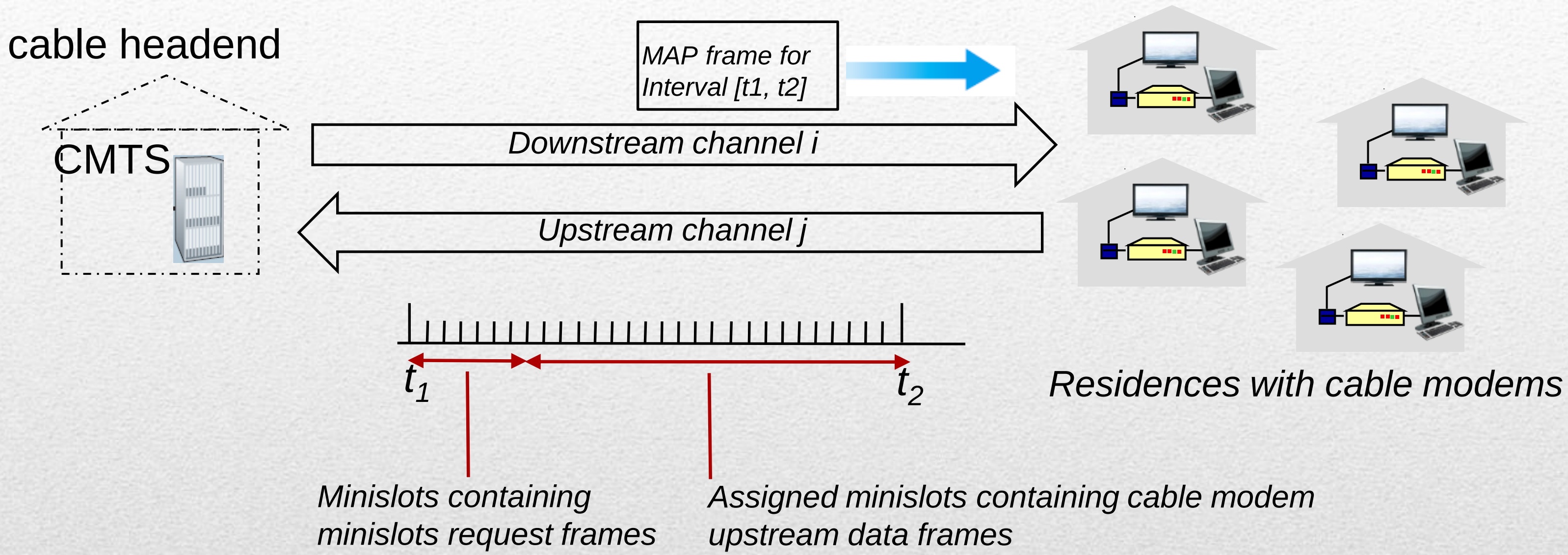


Cable Access Network



- Multiple 40Mbps downstream (broadcast) channels
  - Single CMTS transmits into channels
- Multiple 30 Mbps upstream channels
  - Multiple access: all users contend for certain upstream channel time slots (others assigned)

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

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

MAC Address & ARP

- 32-bit IP address
  - Network-layer address for interface
  - Used for layer 3 (network layer) forwarding

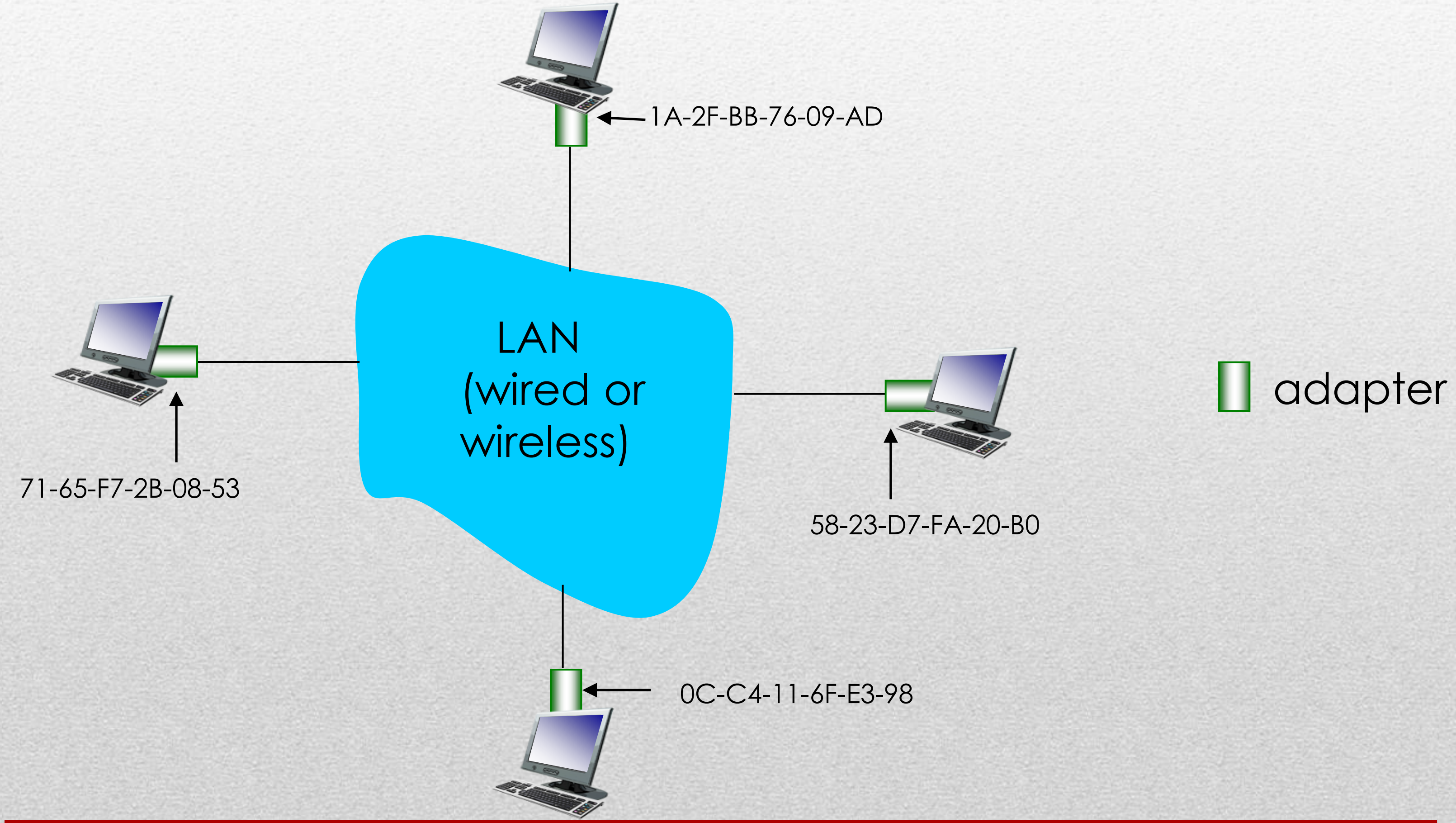
- MAC (or LAN or physical or Ethernet) address:
  - Function: used ‘locally” to get frame from one interface to another physically-connected interface (same network, in IP-addressing sense)

  - 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 & ARP

Each adapter on LAN has unique *LAN* address



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

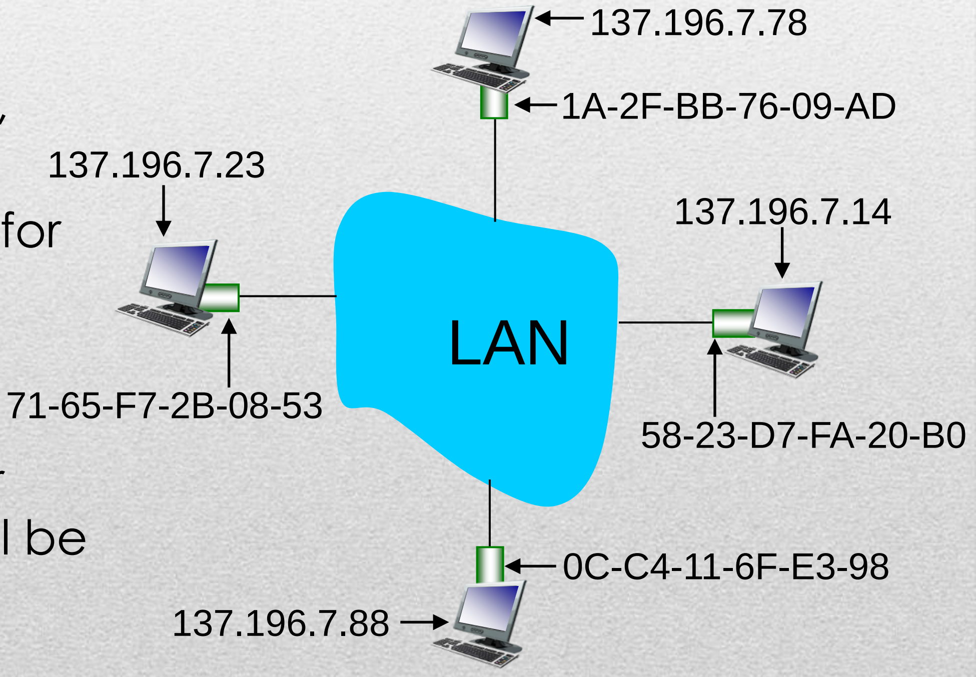
ARP: Address Resolution Protocol

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)



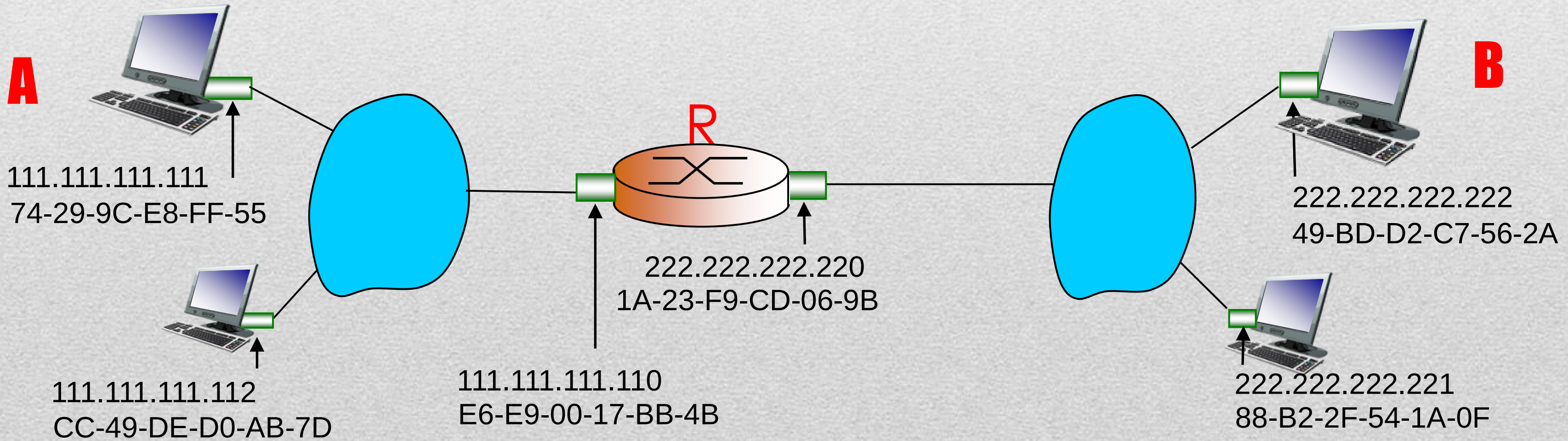
ARP Protocol: Same LAN

- 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
  - 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

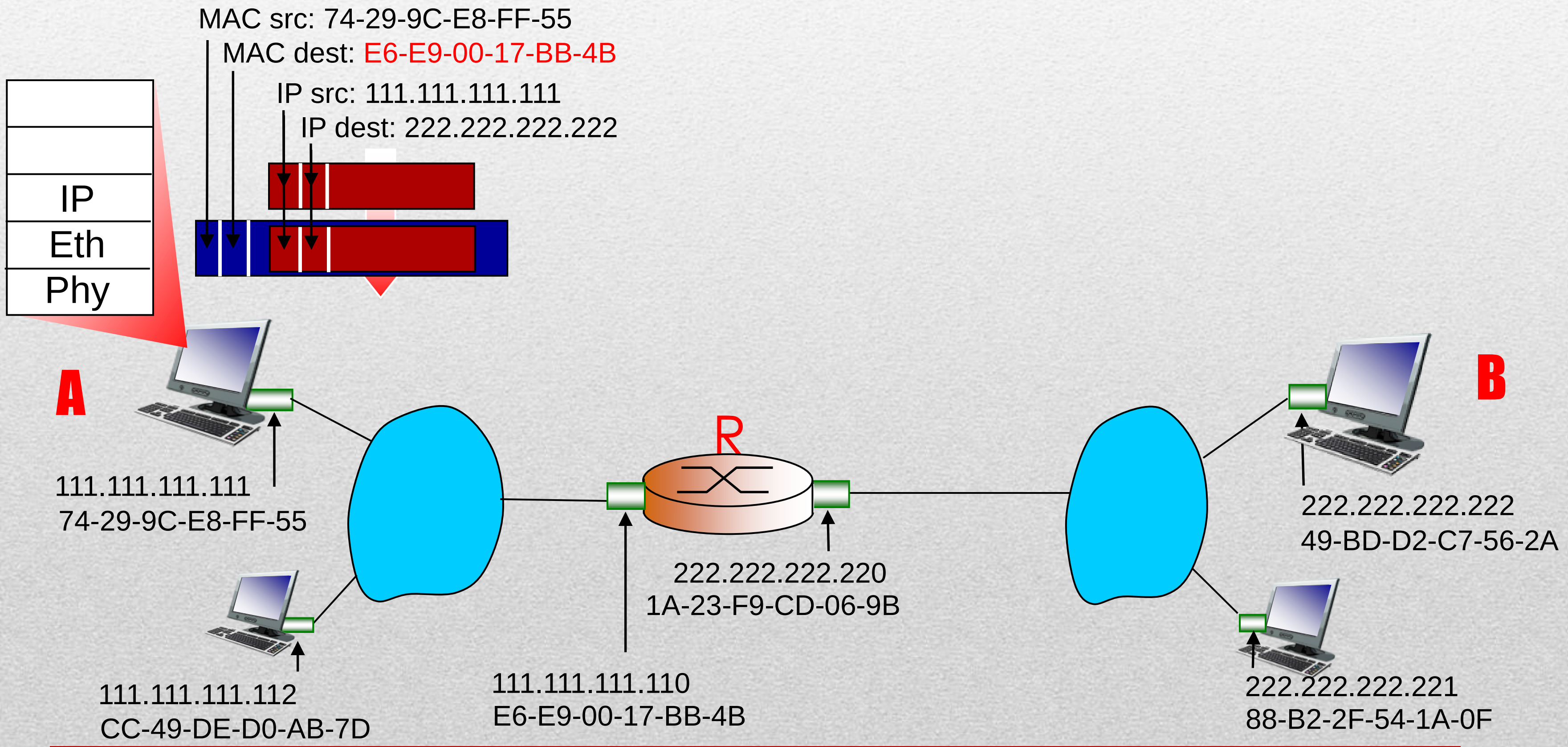
Walkthrough: send datagram from A to B via R

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



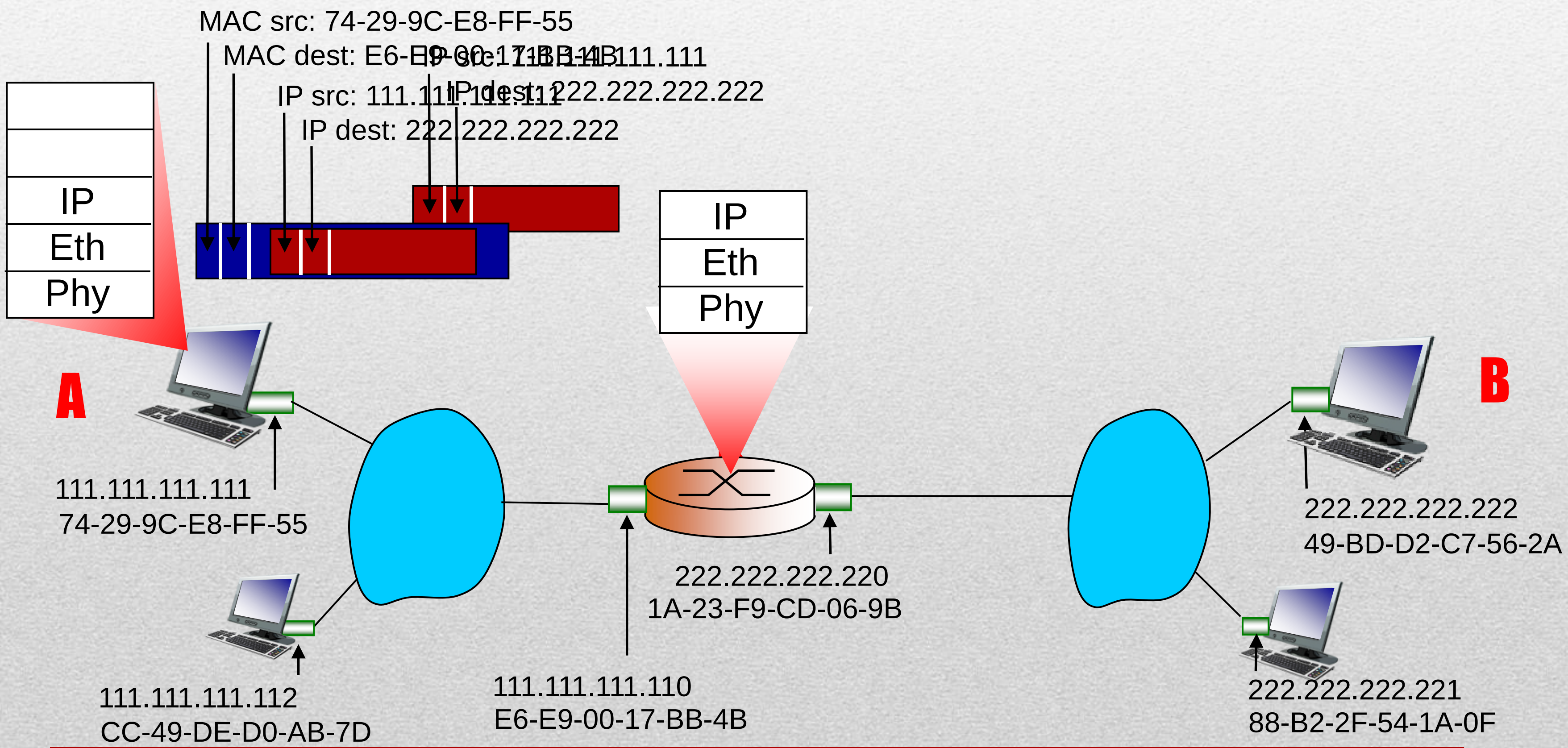
Addressing: Routing to another LAN

- 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



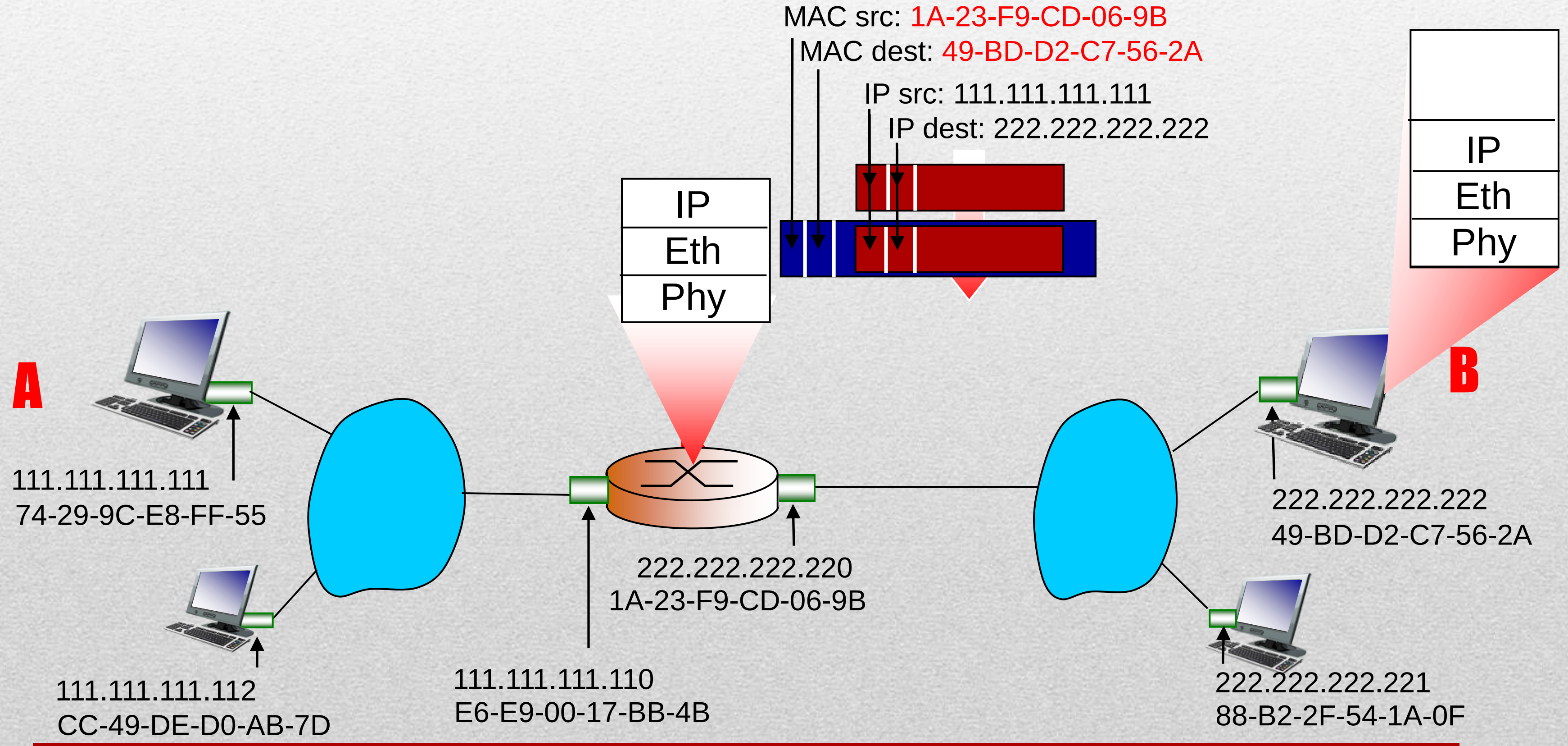
Addressing: Routing to another LAN

- 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