IOWA

CS3640

The finals: format and choice

Prof. Supreeth Shastri
Computer Science
The University of Iowa

Based on your feedback, **I'm letting you choose your own path**

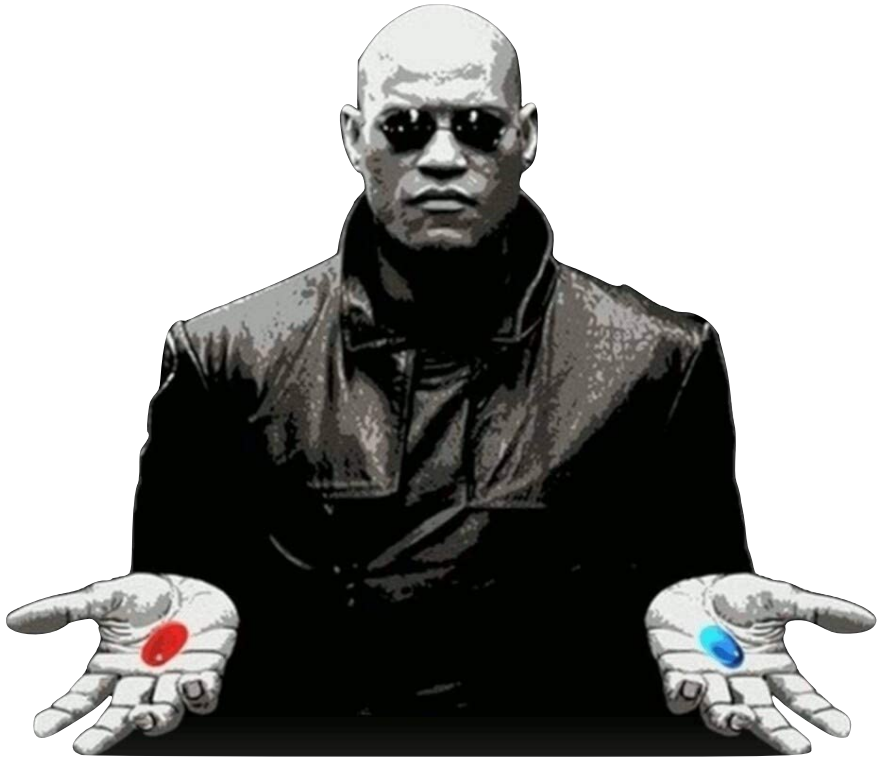
Attempts: 38 out of 38

What is your most preferred format for the final exam?

In-class, pen-paper exam (similar to midterm)	25 respondents	66 %
In-person technical interview	8 respondents	21 %
I'm ok with either format	5 respondents	13 %

Comments worth noting

- Would have loved a take-home, but I understand why this had to change
- No multiple choice exam please
- Make it 50% multiple-choice and 50% free-style
- Very unprofessional



I will offer both **in-class written** exam and **in-person interview**

Structure of the Tech Interview

1-on-1

conducted by your instructor

20%

your interview determines a fifth of your final grade

May 1 - 5

pick a slot between 10am - 2pm that works best for you

15 mins

in which you will be asked 4 questions, all carrying equal weights

Q banks

Each question comes from a distinct category (or bank); more in the next slide

Structure of the Written Exam

1 hour

you will answer 4 questions, all carrying equal weights

20%

your interview determines a fifth of your final grade

May 2

12:30 - 1:30pm (in-class); alternate slots for SDS

Question Choice

Similar in difficulty level to the midterm; a well prepared student should expect to score the same irrespective of the format choice

Question Categories

Category	Example questions and topics	Weight
Networking **Principles**	*Internet's hour-glass model; Middleboxes*	25%
Networking **Protocols**	*A day in the life of a packet*	25%
Networking **Problems**	*Construct Dijkstra's LS table*	25%
Networking **Practice**	*SDN; Cloud computing; Solar superstorms*	25%

Each category has a bank of 5-8 question; for each interviewee, I will generate a sequence of four random numbers that will determine the specific questions picked from each bank.

*There will be an optional **bonus question** carrying 10% extra points (only for the written exam)*

What would I advice?

A well prepared student should expect to score the same irrespective of the format choice

> “In theory, theory and practice are the same.
> In practice, they are not.” — Albert Einstein

IOWA

CS3640

Link Layer (1): Services and Protocols

Prof. Supreeth Shastri
Computer Science
The University of Iowa

Multiple Access Protocols

sharing a single broadcast channel amongst multiple nodes

1. Channel Partitioning Protocols

divide channel into small pieces (e.g., time slots, frequency), and allocate each piece to one node

2. Random Access Protocols

do not divide the channel, and allow nodes to transmit at any time, but detect/ recover from collisions

3. Taking-turns Protocols

nodes take turn to send frames; this dynamism achieves balance between the first two class of protocols

Random Access Protocols

Random Access Protocols

- When node has packet to send,
  - ➡ it transmits at full channel data rate R
  - ➡ no *a priori* coordination among nodes
- If two or more nodes transmit simultaneously: collision
- Random Access Protocols specify
  - ➡ how to detect collisions
  - ➡ how to recover from collisions
- Examples of random access protocols
  - ➡ ALOHA, slotted ALOHA (impolite speakers)
  - ➡ CSMA, CSMA/CD (more polite speakers)

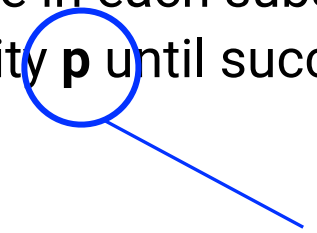
Slotted ALOHA

Channel assumptions

- all frames are of same size
- time is divided into equal length slots *(e.g., time to transmit 1 frame)*
- clocks at all nodes are synchronized
- nodes start to transmit only at the beginning of a slot
- if two or more nodes transmit in slot, all nodes detect collision

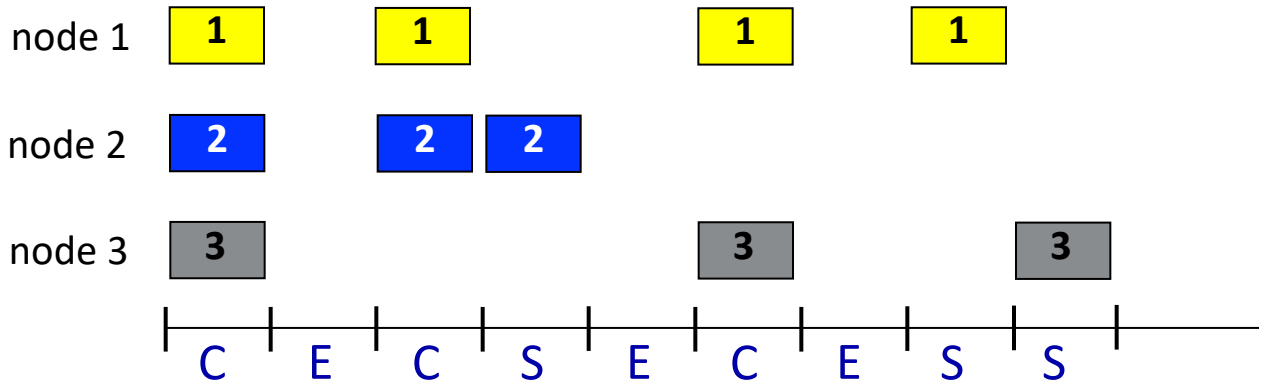
Protocol operation

- When a node has a new frame, it transmits it in next slot
- If no collision is detected: node can send any new frame in next slot
- If collision is detected: node retransmits the frame in each subsequent slot with probability **p** until success



randomization – *why*?

Slotted ALOHA



C: collision
S: success
E: empty

Pros

- single active node can continuously transmit at full rate of channel
- highly decentralized
- simple

Cons

- a collision wastes the full slot
- leaves idle slots even when nodes have data to transmit
- clock synchronization

Carrier Sense Multiple Access (CSMA)

Simple CSMA: listen before transmit

- if channel sensed to be idle: *transmit entire frame*
- if channel sensed to be busy: *defer transmission*

human analogy: don't interrupt while others are talking!

CSMA/CD: CSMA with collision detection

- collisions detected within short time
- colliding transmissions are immediately aborted, thus reducing channel wastage

human analogy: a polite conversationalist

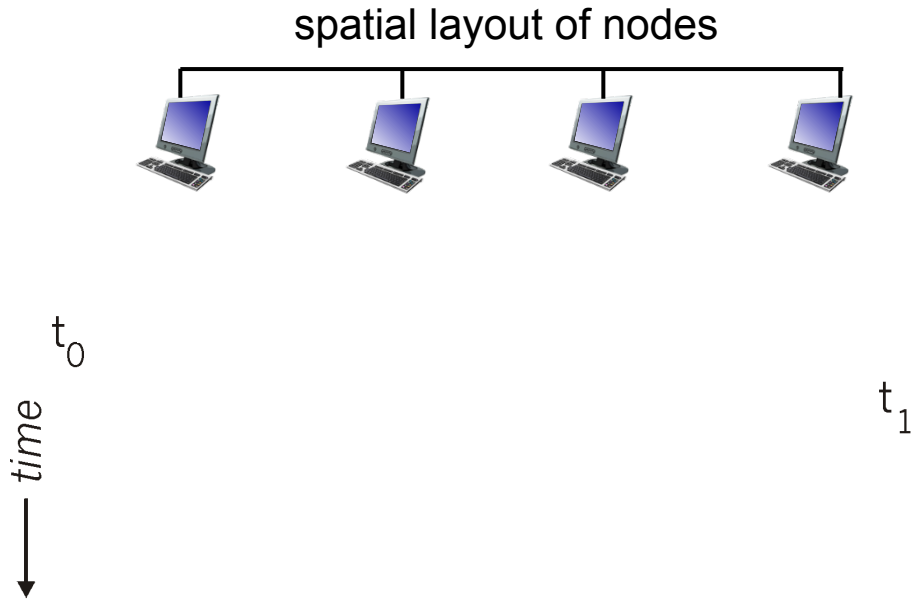
Simple CSMA

Collisions can still occur with carrier sensing:

- propagation delay means two nodes may not hear each other's just-started transmission

When there is a collision: entire packet transmission time is wasted

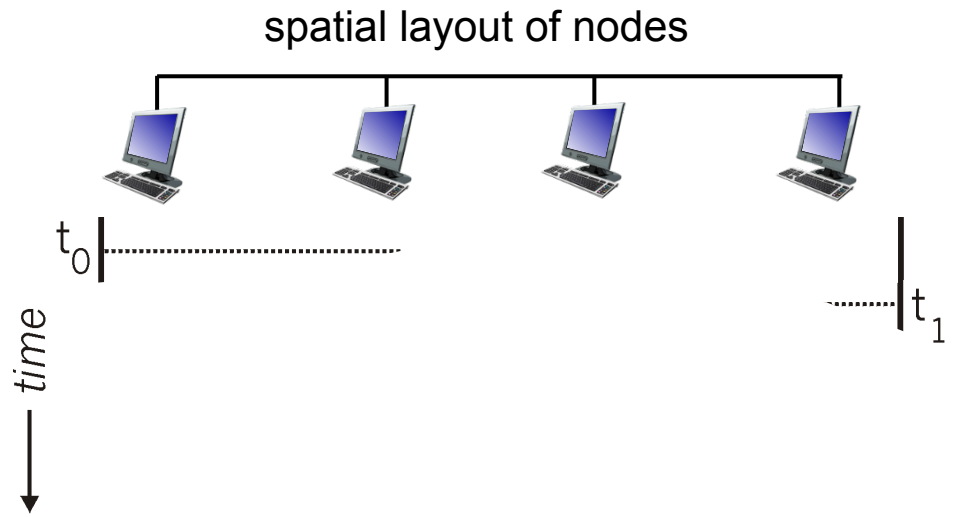
- distance and propagation delay play role in in determining collision probability



CSMA/CD

CSMA/CD reduces the amount of time wasted in collisions

- transmission aborted on collision detection



Ethernet CSMA/CD algorithm

1. Ethernet receives datagram from network layer, creates a frame
2. If Ethernet senses channel:

   if idle: start frame transmission.

   if busy: wait until channel idle, then transmit
3. If entire frame transmitted without collision - done!
4. If another transmission detected while sending: abort, send jam signal
5. After aborting, enter binary exponential backoff

   - after m^{th} collision, chooses K at random from $\{0,1,2, \ldots, 2^m - 1\}$. Ethernet waits K slots, returns to Step 2
   - more collisions for the same frame: longer backoff interval

Taking-Turns Protocols

Taking Turns MAC Protocols

Channel Partitioning MAC Protocols

- inefficient at low load: if only one node is active, 1/N bandwidth is allocated to it!
- allows sharing the channel *efficiently and fairly* at high load

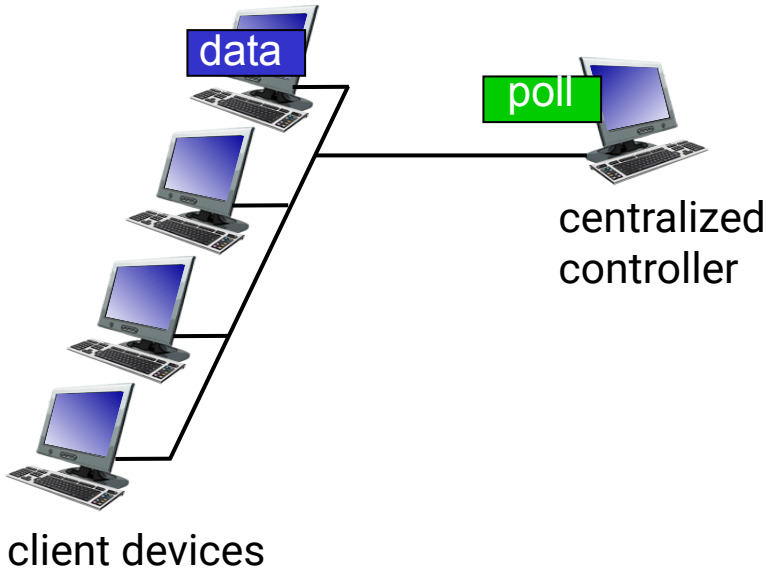
Random Access MAC Protocols

- *efficient at low load*: a single node can fully utilize channel
- experiences overhead at high load due to collisions

Taking Turns Protocols *look for best of both worlds!*

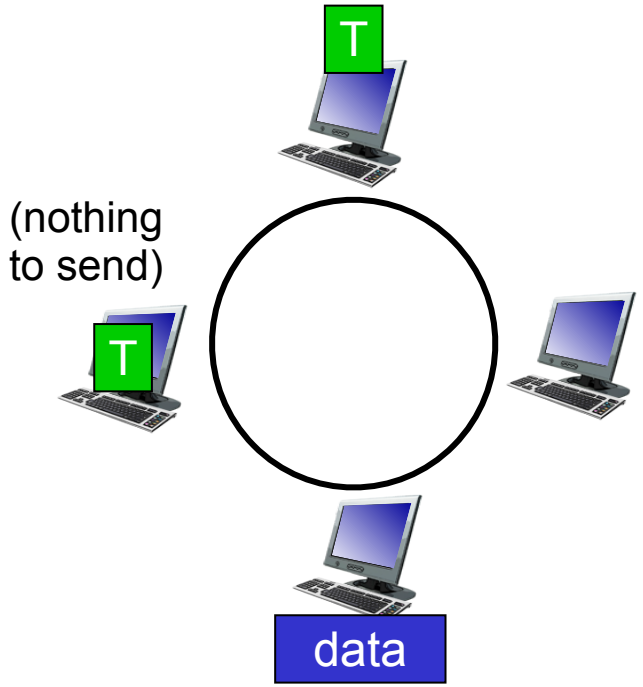
Polling Protocols

- a controller “invites” other nodes to transmit in turn
- typically used with “dumb” devices
- suffers from polling overhead, latency, and single point of failure (controller)



Token Passing Protocols

- control token message explicitly passed from one node to next, sequentially
- transmit while holding token
- suffers from token overhead, latency, and single point of failure (token)



IOWA

CS3640

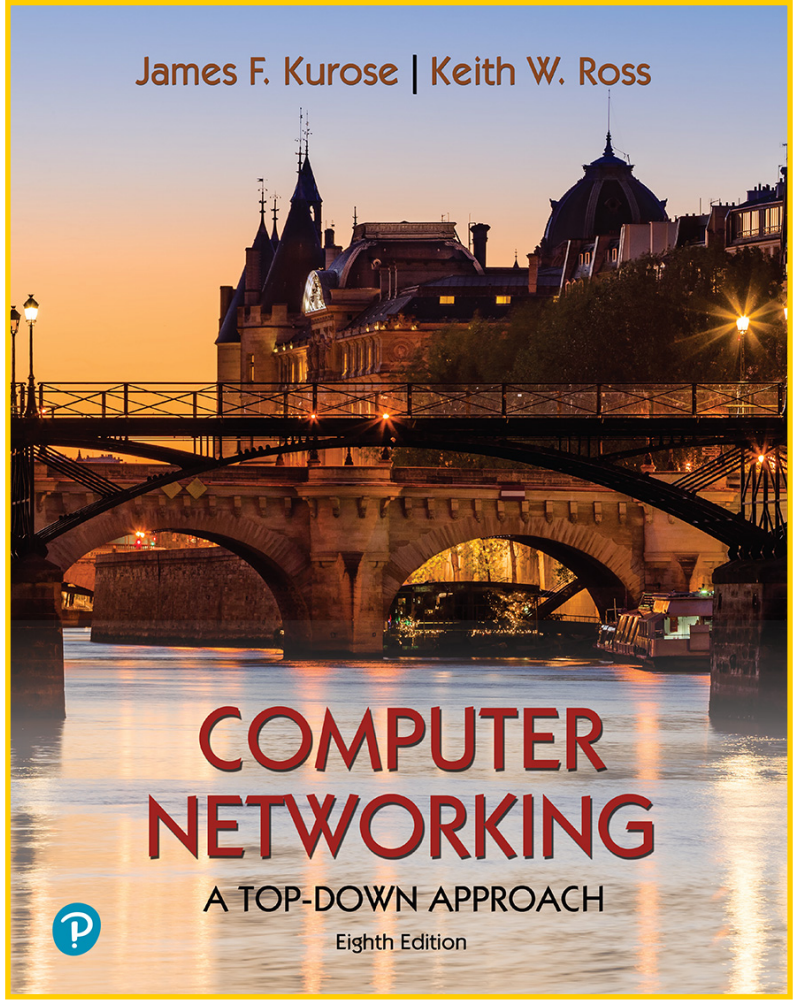
Link Layer (2): Addressing and Ethernet

Prof. Supreeth Shastri
Computer Science
The University of Iowa

Lecture goals

second chapter on link layer covering principles, practices, and protocols

- *Link layer addressing*
- *Ethernet*



Chapter 6.4

IOWA

Addressing in the Link Layer

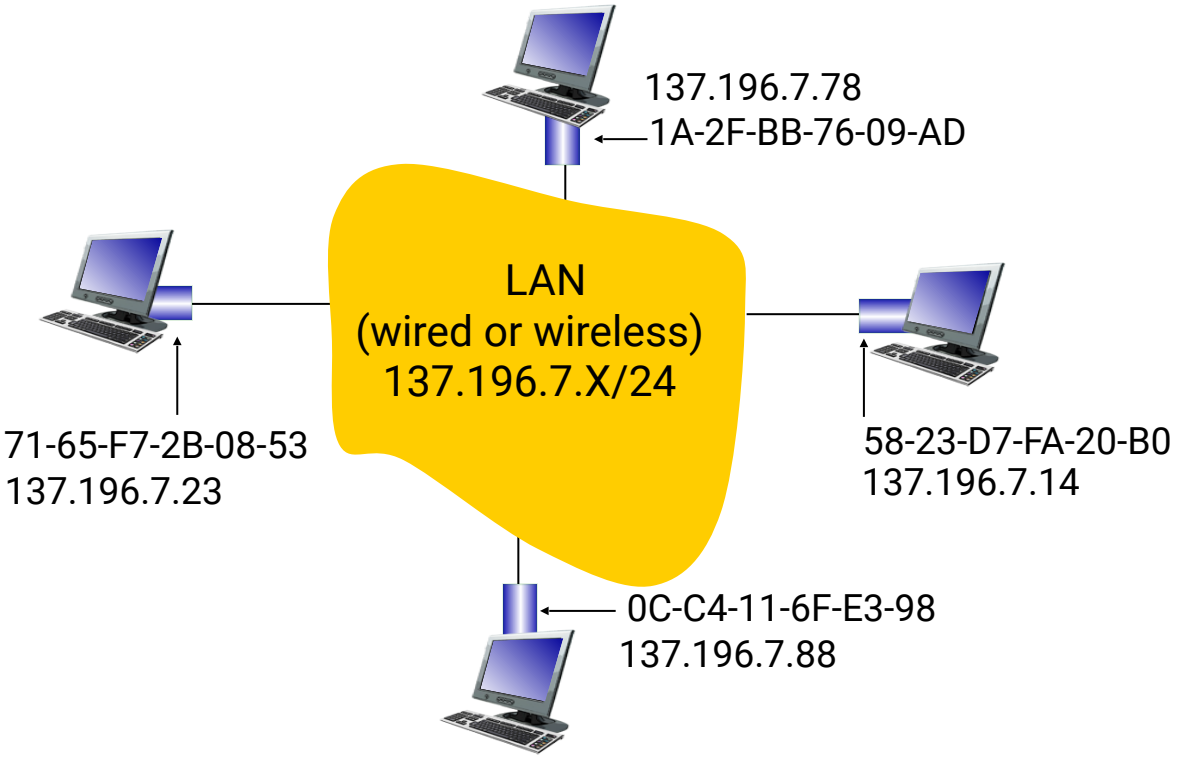
every network interface will have a network layer address and a link layer address

IP Address

- 32 bits (IPv4) or 128 bits (IPv6)
- E.g., 142.250.191.206 *(usually in decimal)*
- used in layer-3 routing and forwarding

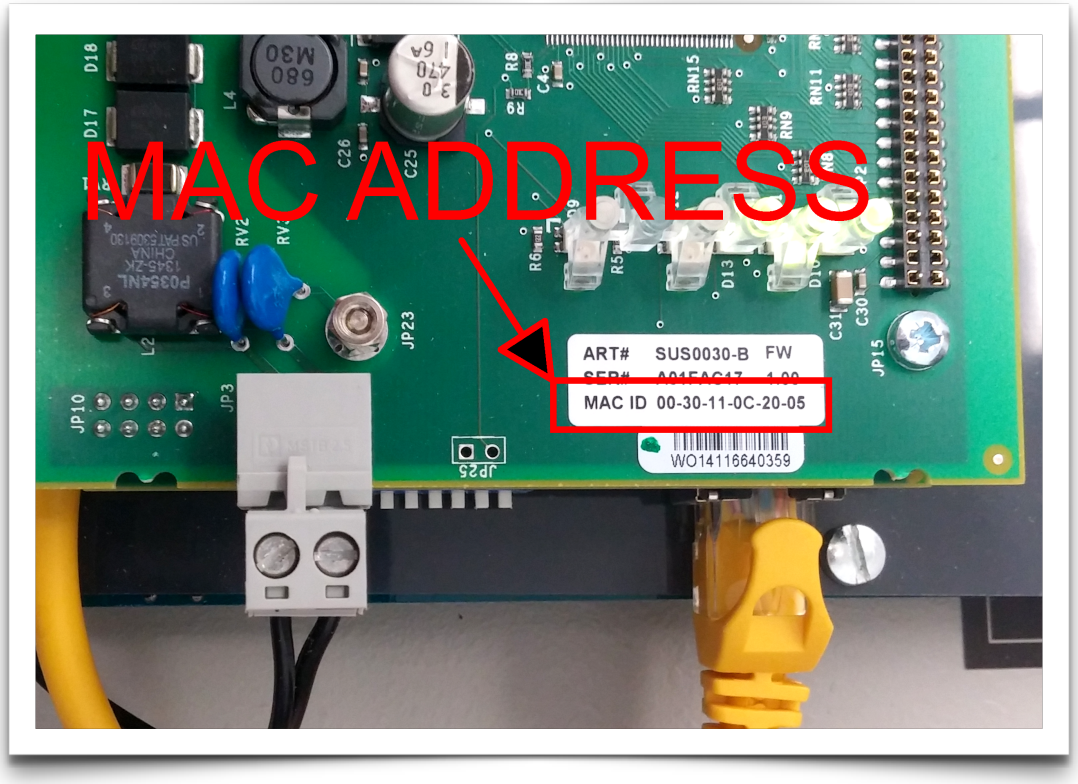
MAC/Ethernet/Physical Address

- 48-bit addressed burned in NIC
- E.g., 14:7d:da:69:d4:55 *(usually in hex)*
- used in layer-2 switching



Addressing in the Link Layer

- MAC address allocation is administered by IEEE
- Every manufacturer buys a portion of MAC address space (to assure uniqueness)
- Portability
  - ➡ *NICs retain their MAC address when they move from one LAN to another*
  - ➡ *IP address is not portable: depends on IP subnet to which node is attached*



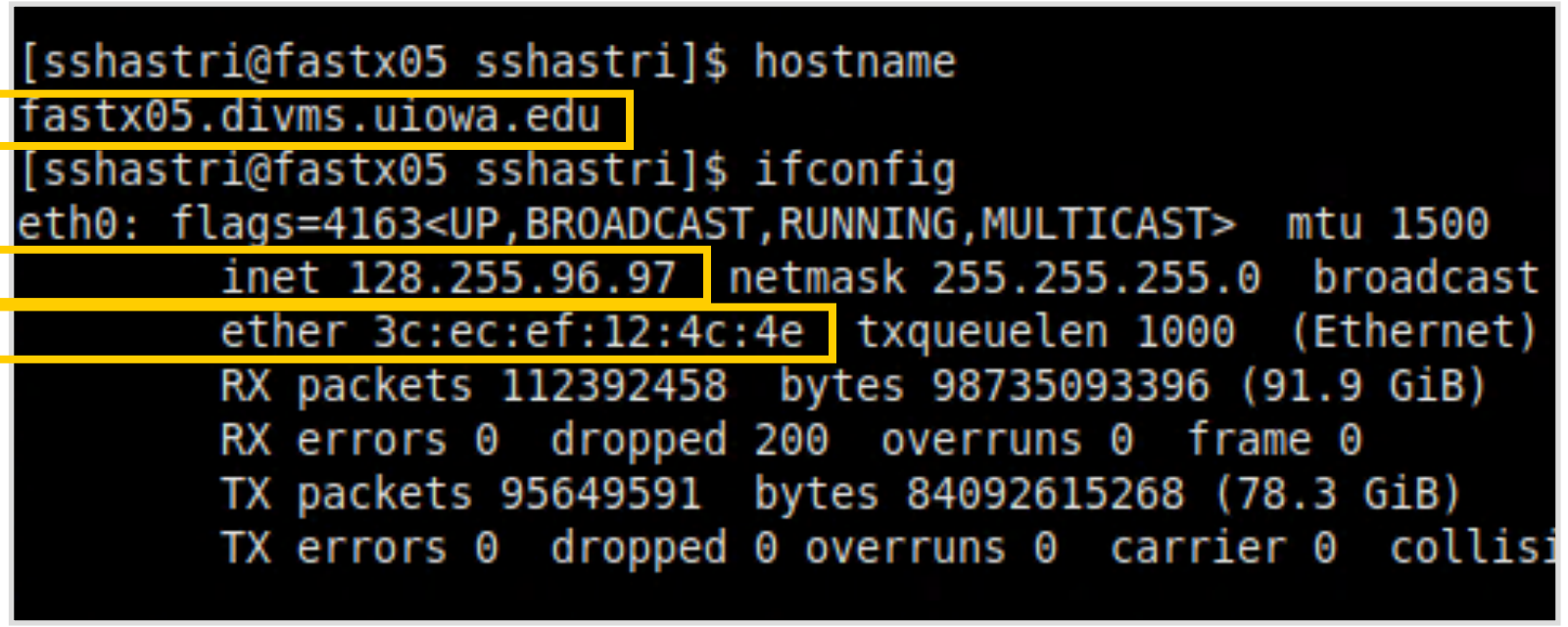
MAC address : IP address :: Social Security Number : Postal address

The Three Network Identities

Host name

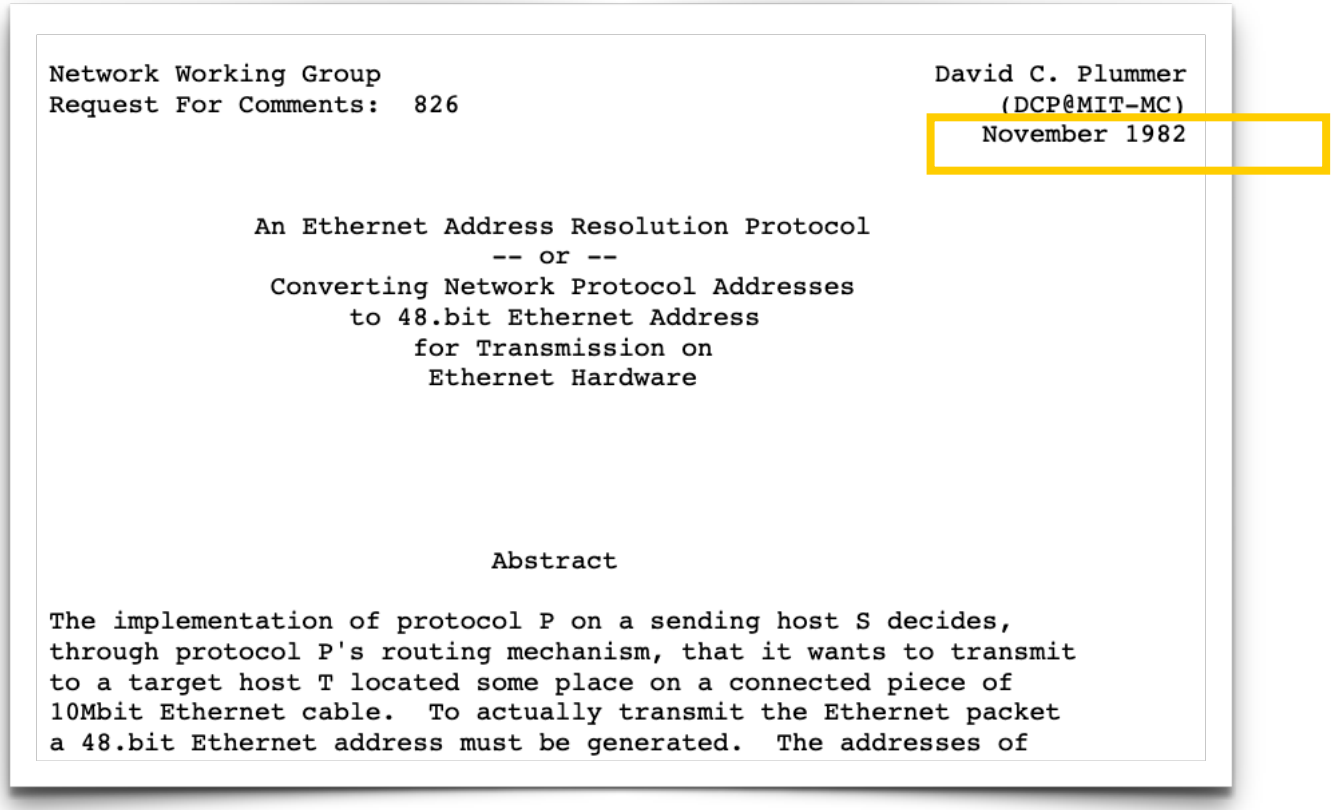
IP Address

Phy Address



Q: Is having multiple network identities redundant? Are they useful?

Address Resolution Protocol (ARP)



Network Working Group
Request For Comments: 826

David C. Plummer
(DCP@MIT-MC)
November 1982

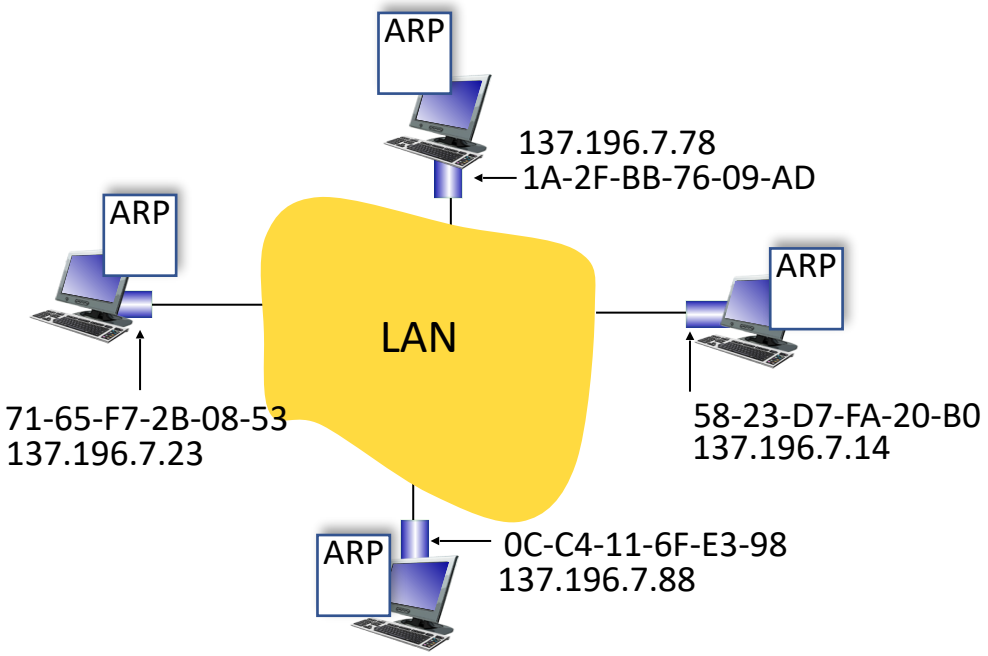
An Ethernet Address Resolution Protocol
-- or --
Converting Network Protocol Addresses
to 48.bit Ethernet Address
for Transmission on
Ethernet Hardware

Abstract

The implementation of protocol P on a sending host S decides, through protocol P's routing mechanism, that it wants to transmit to a target host T located some place on a connected piece of 10Mbit Ethernet cable. To actually transmit the Ethernet packet a 48.bit Ethernet address must be generated. The addresses of

Address Resolution Protocol (ARP)

Given the IP address of an interface, how to determine its MAC address?



Every interface (host, router) has an ARP table

- ARP table contains IP to MAC address mappings for nodes on the same LAN
- Three tuple: <IP address; MAC address; TTL>
- TTL (Time To Live): time after which address mapping will be forgotten (typically 20 min)

ARP in Action

E.g., A wants to send datagram to B

B's MAC address not in A's ARP table, so A uses ARP to find B's MAC address

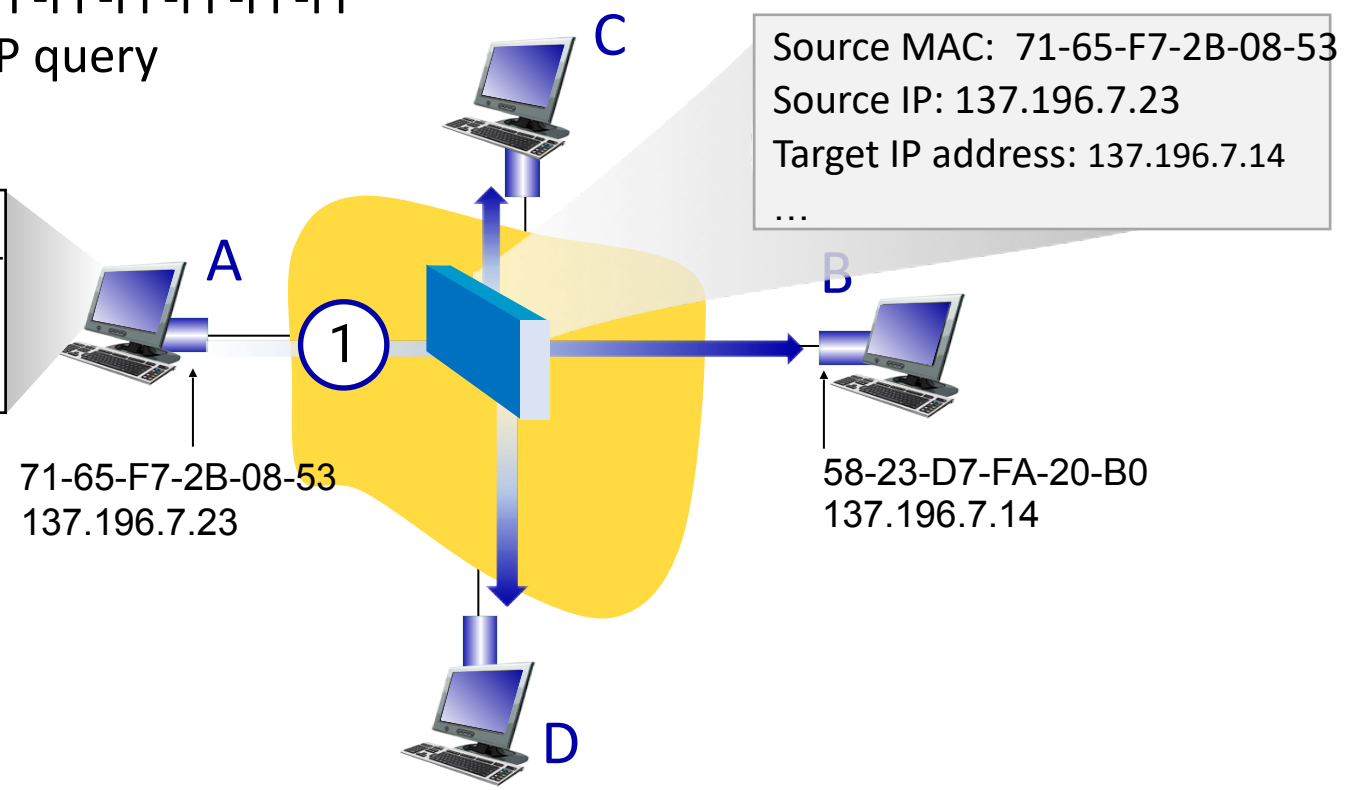
A broadcasts ARP query, containing B's IP addr

1

- destination MAC address = FF-FF-FF-FF-FF-FF
- all nodes on LAN receive ARP query

ARP table in A

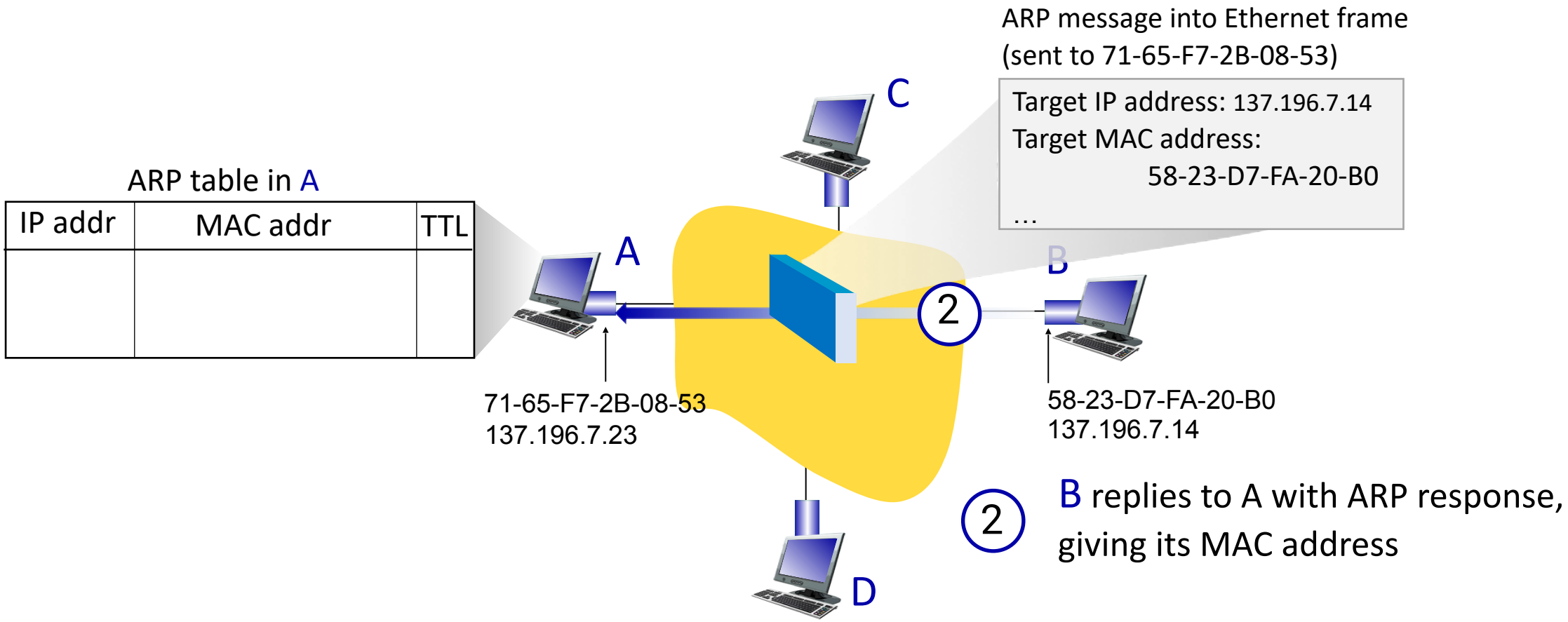
IP addr	MAC addr	TTL



ARP in Action

E.g., A wants to send datagram to B

B's MAC address not in A's ARP table, so A uses ARP to find B's MAC address



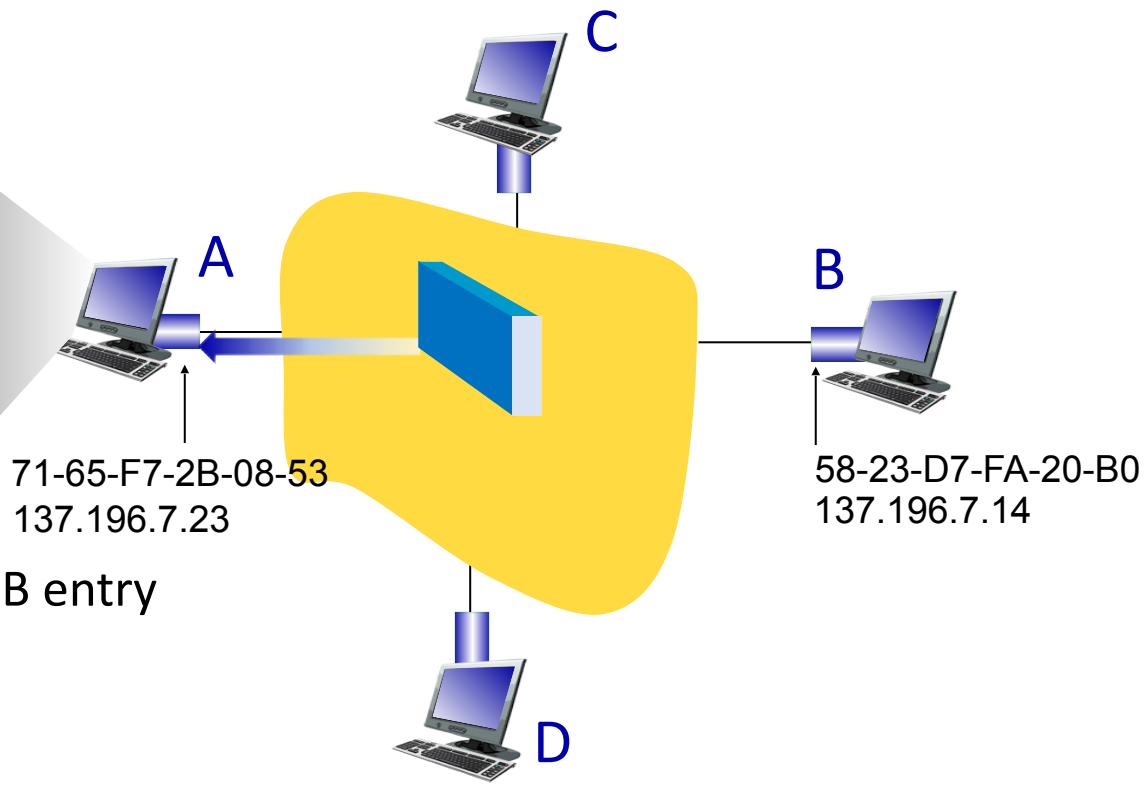
ARP in Action

E.g., A wants to send datagram to B

B's MAC address not in A's ARP table, so A uses ARP to find B's MAC address

ARP table in A

IP addr	MAC addr	TTL
137.196.7.14	58-23-D7-FA-20-B0	500

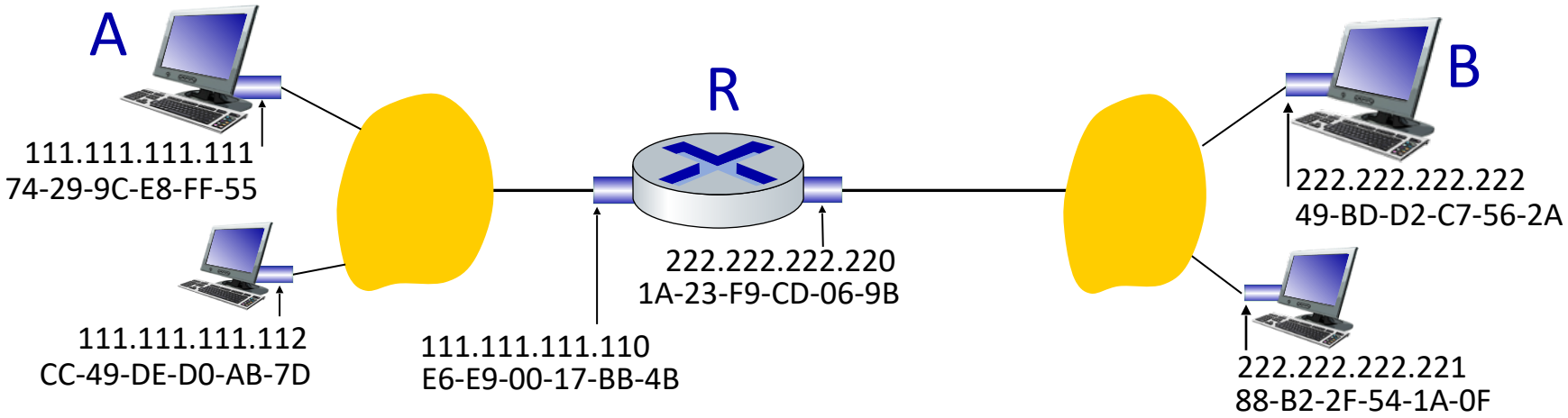


3 A receives B's reply, adds B entry into its local ARP table

Routing to Another Subnet

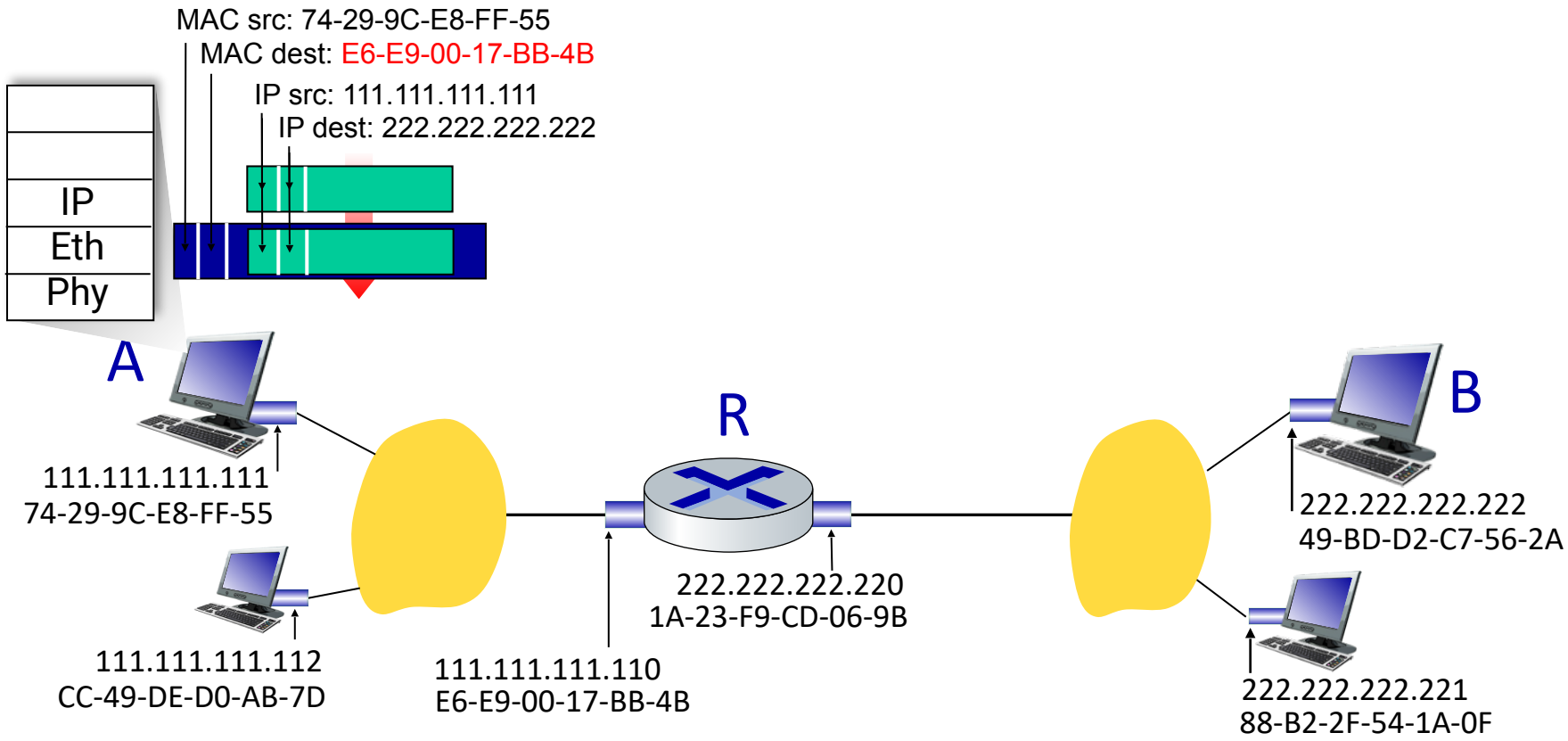
E.g., A wants to send datagram to B **via R**

A knows B's IP address; A also knows R's IP address and MAC address (how?)



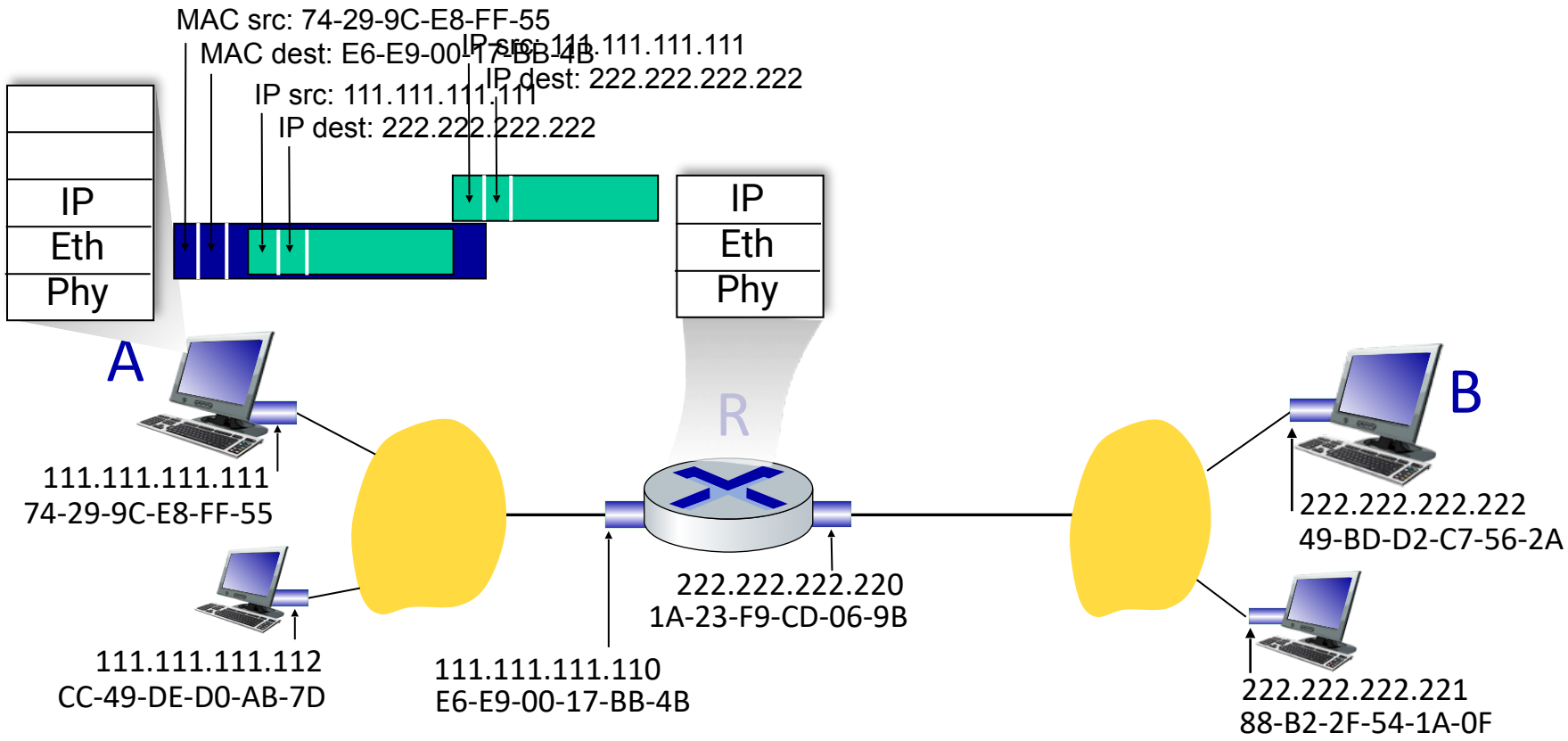
Routing to Another Subnet

- A creates IP datagram with IP source A, destination B
- A creates link-layer frame containing A-to-B IP datagram, then puts R's MAC address is the frame's destination



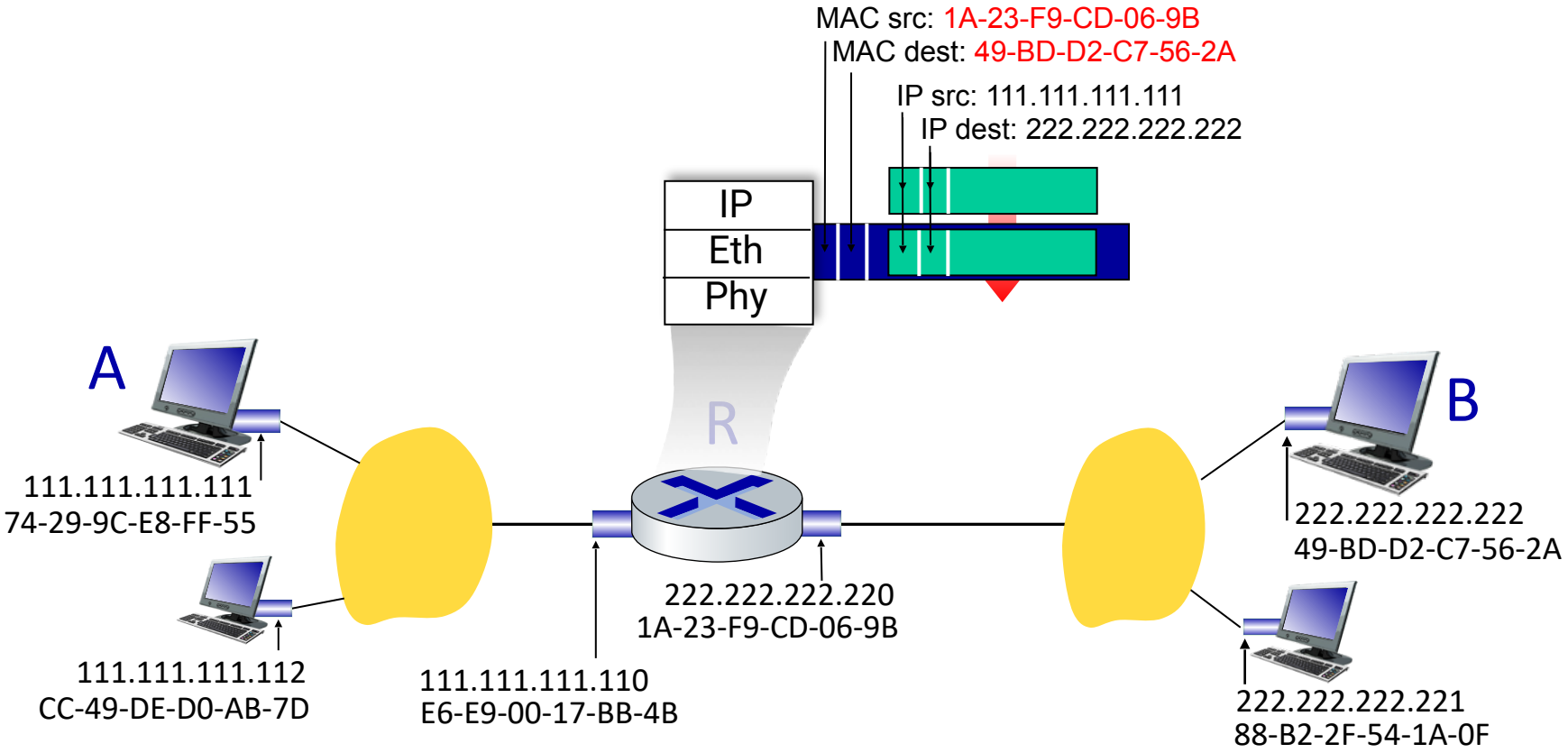
Routing to Another Subnet

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



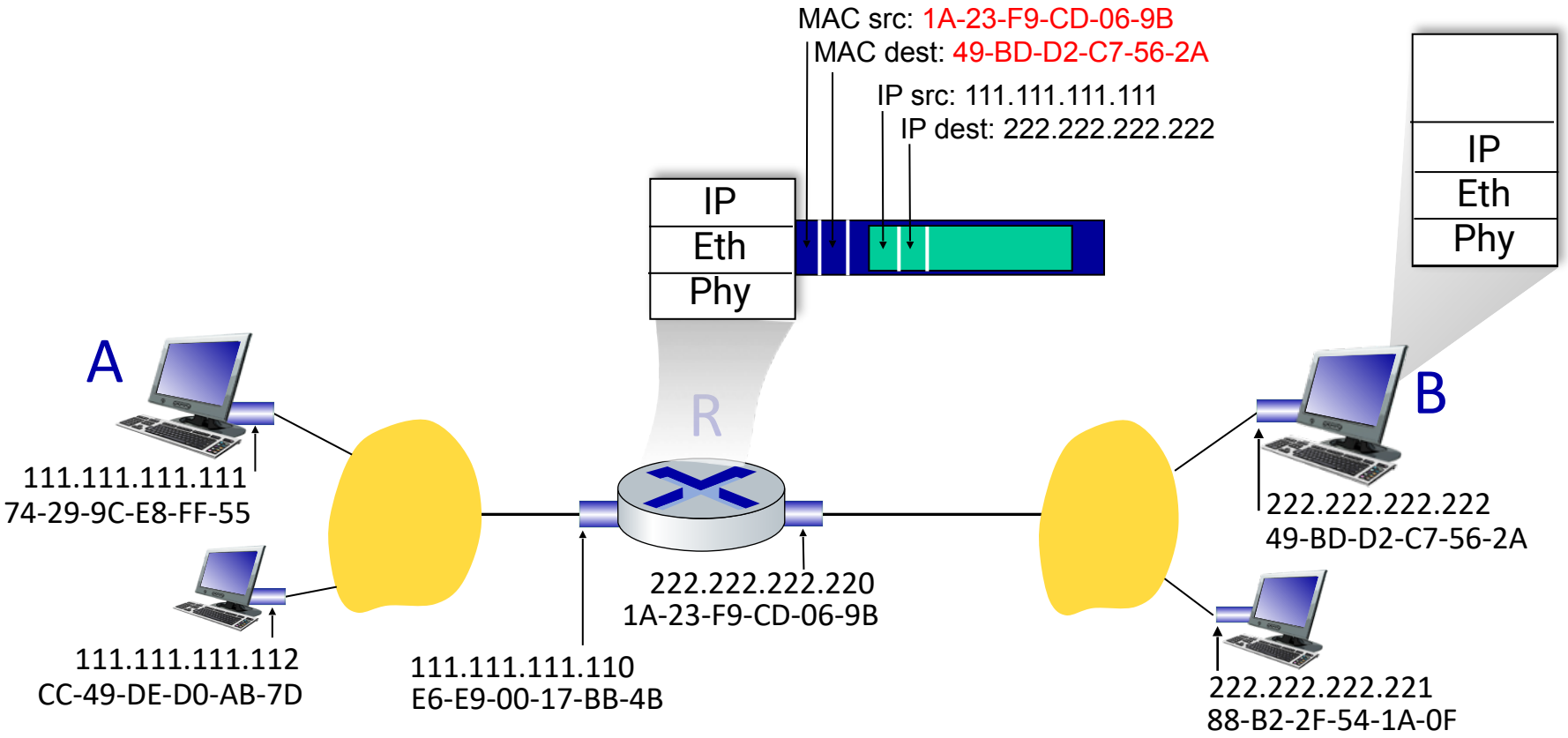
Routing to Another Subnet

- R determines outgoing interface, passes datagram with IP source A, destination B to link layer
- R creates link-layer frame containing A-to-B IP datagram. Frame destination address: B's MAC address



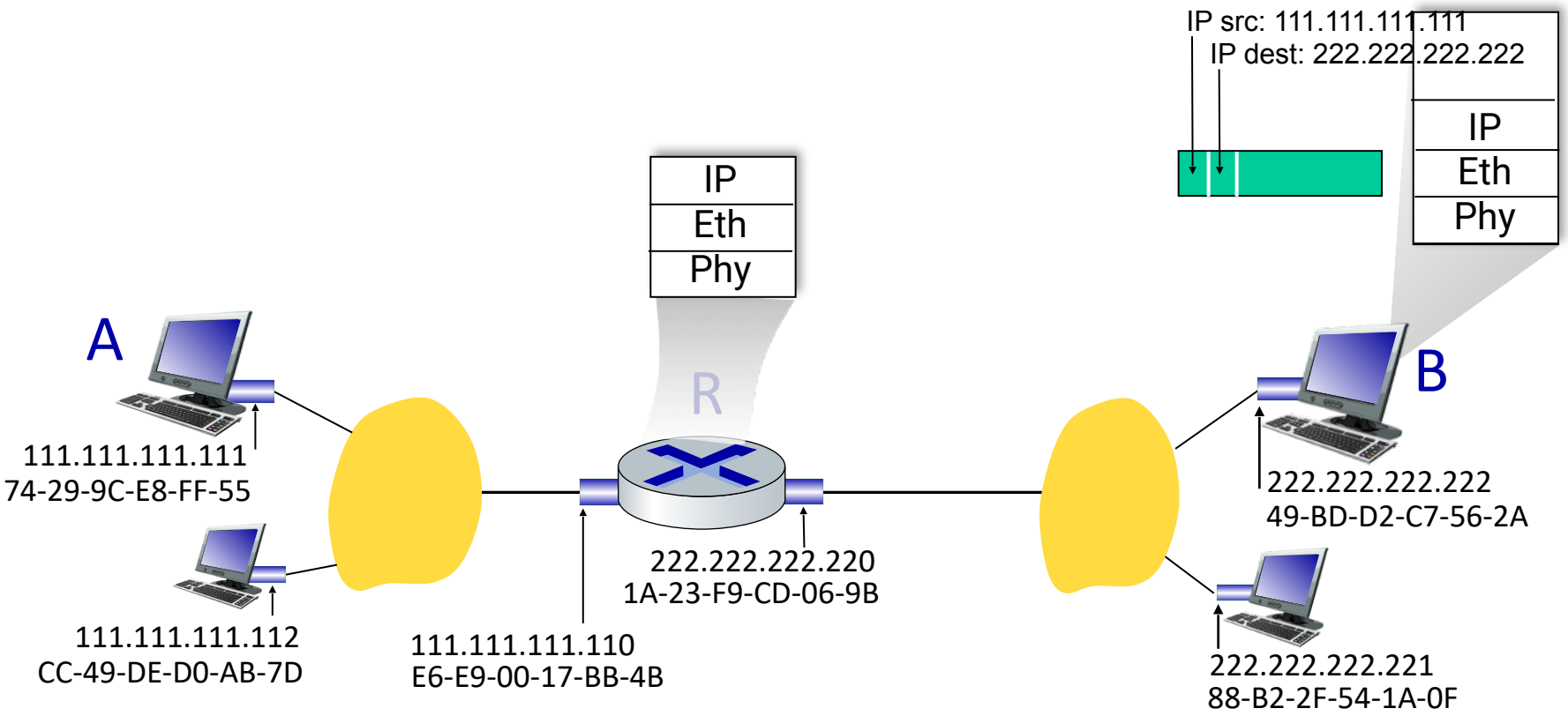
Routing to Another Subnet

- R determines outgoing interface, passes datagram with IP source A, destination B to link layer
- R creates link-layer frame containing A-to-B IP datagram. Frame destination address: B's MAC address
- transmits link-layer frame

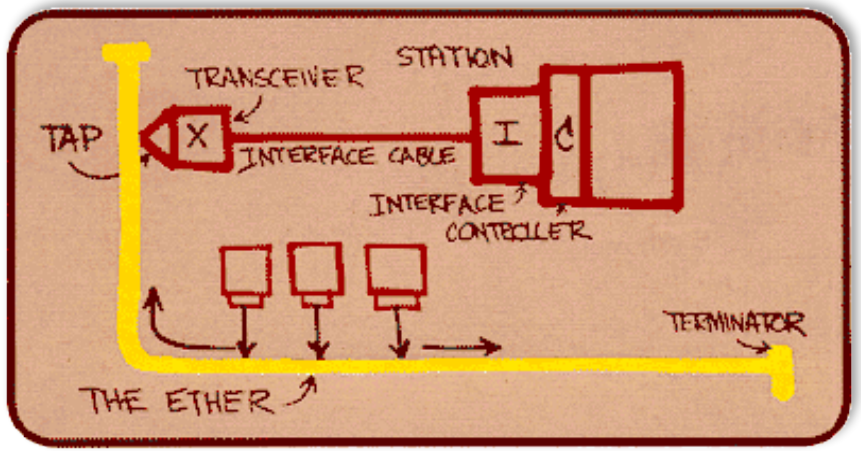


Routing to Another Subnet

- B receives frame, extracts IP datagram destination B
- B passes datagram up protocol stack to IP



Ethernet



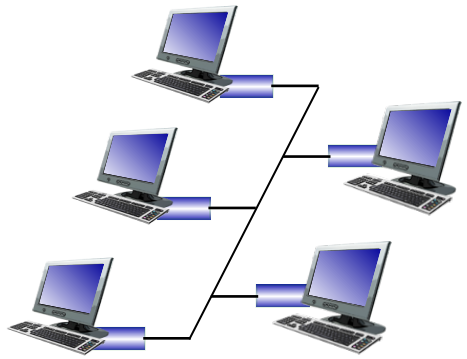
Bob Metcalfe's Ethernet sketch, 1976

Ethernet: Overview and Topologies

- first widely used, and today's dominant wired LAN tech
- simple to build, and cheap to maintain
- has kept up with speed race: 10 Mbps – 400 Gbps

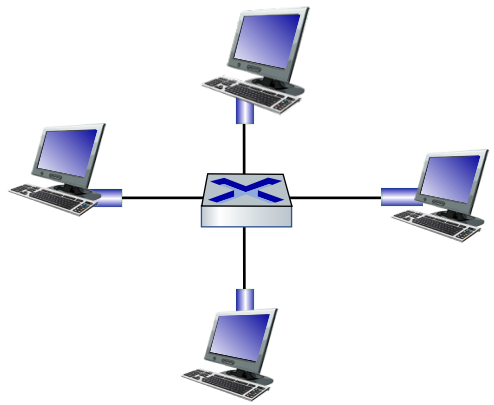
Bus: popular through mid 90s

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



Switched: most prevalent today

active link-layer 2 switch in center with each "spoke" running a separate Ethernet protocol (no collisions)



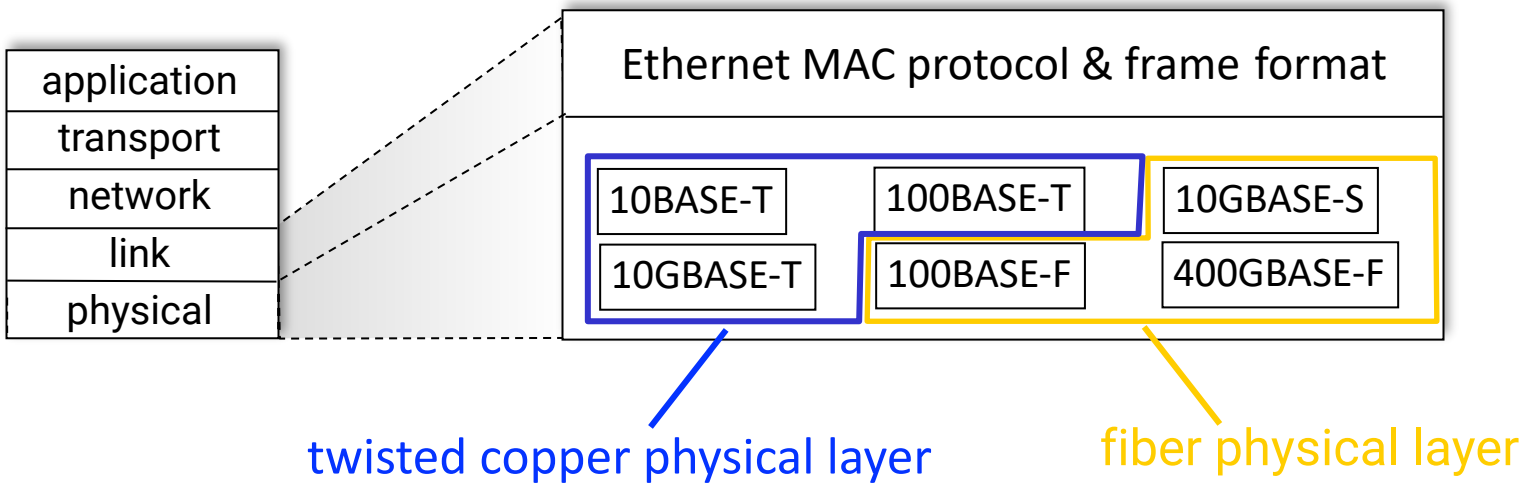
Ethernet Frame Format

- **Preamble**
  - ➡ used to synchronize receiver, sender clock rates
  - ➡ 7 bytes of 10101010 followed by one byte of 10101011
- **Addresses:** 6 byte source and destination MAC addresses
  - ➡ if adapter receives frame with matching destination address, or with broadcast address, pass payload to the network layer protocol
  - ➡ otherwise, adapter discards frame
- **Type**: indicates higher layer protocol
  - ➡ mostly IP but others possible, e.g., Novell IPX, AppleTalk
  - ➡ used to demultiplex up at receiver
- **CRC**: cyclic redundancy check at receiver
  - ➡ if error is detected, drop the frame

Ethernet: Characteristics and Standards

- **Connectionless**: no handshaking between sending and receiving NICs
- **Unreliable**: receiving NIC doesn't send ACKs or NAKs to sending NIC
  - ➡ *data in dropped frames is recovered only if initial sender uses higher layer RDT (e.g., TCP)*

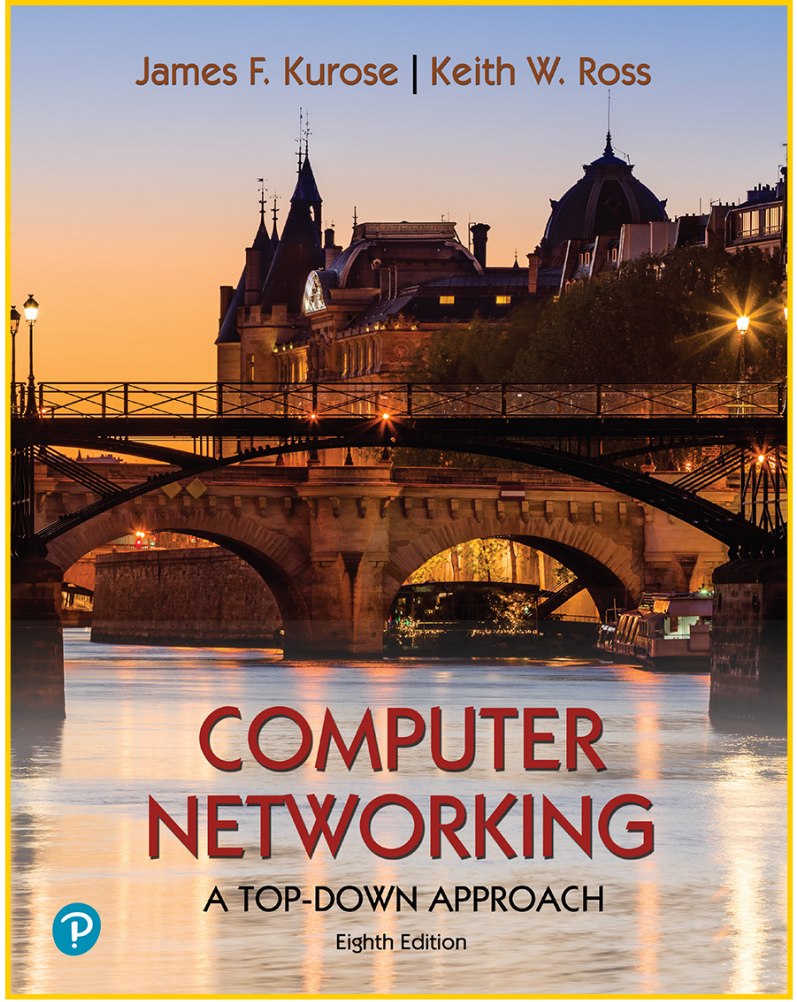
- **Ethernet's MAC protocol**: Unslotted CSMA/CD with binary exponential backoff
- **Ethernet standards**: IEEE 802.3 family



Next Lecture

***Retrospective**: a day in the life of a web page request*

- *touches all five layers of the stack*
- *utilizes more than a dozen protocols*
- *end-to-end flow of control and data*



Chapter 6.7

IOWA

Spot Quiz (ICON)