IOWA

CS3640

Link Layer (2): Addressing and Ethernet

Continuing from previous lecture

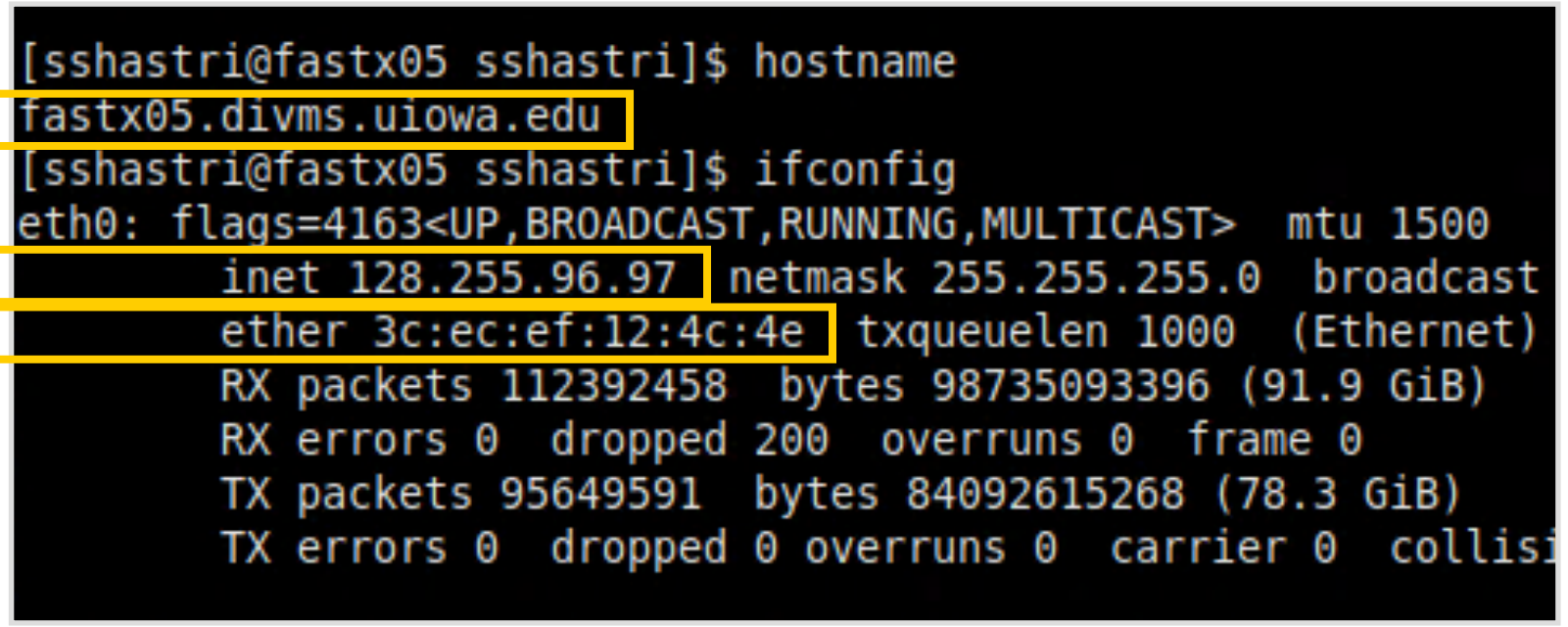
Prof. Supreeth Shastri
Computer Science
The University of Iowa

The Three Network Identities

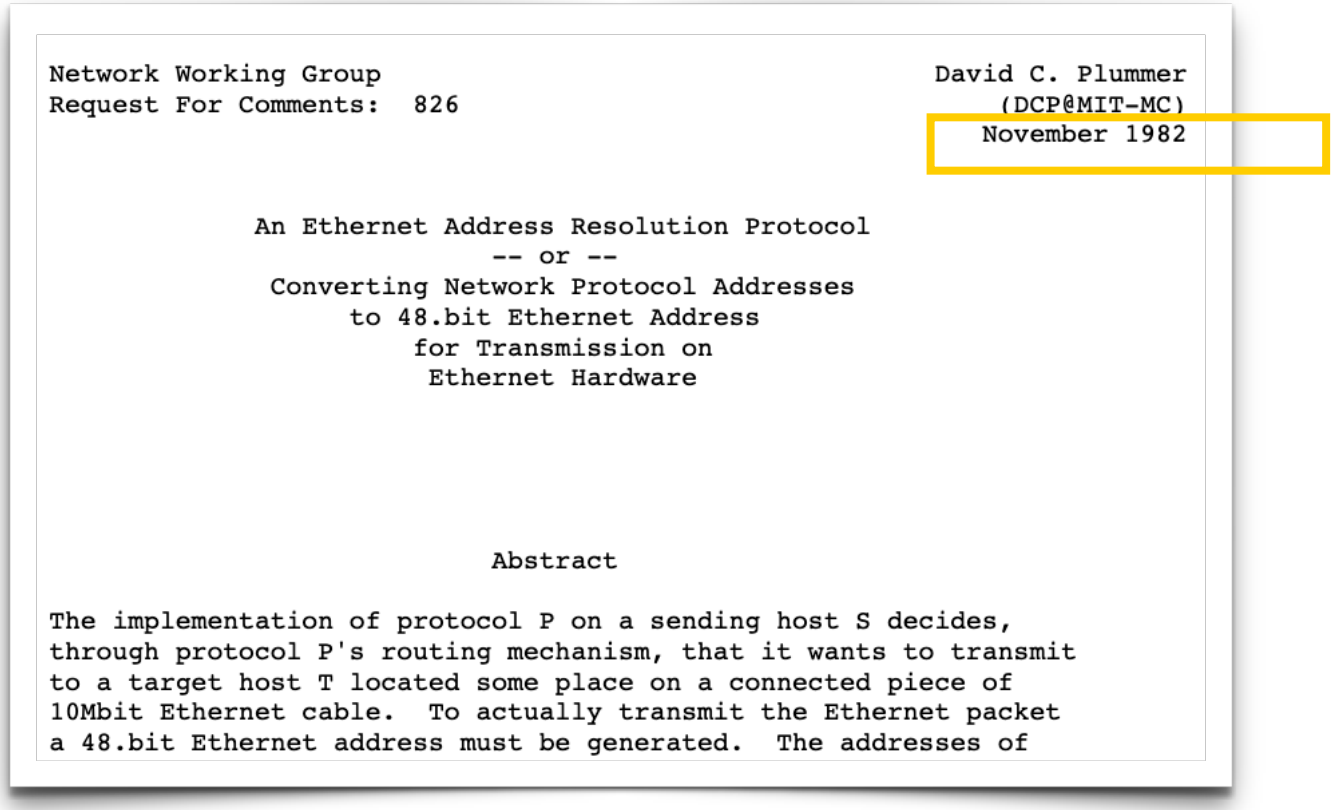
Host name

IP Address

Phy Address



Address Resolution Protocol (ARP)



Network Working Group
Request For Comments: 826

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

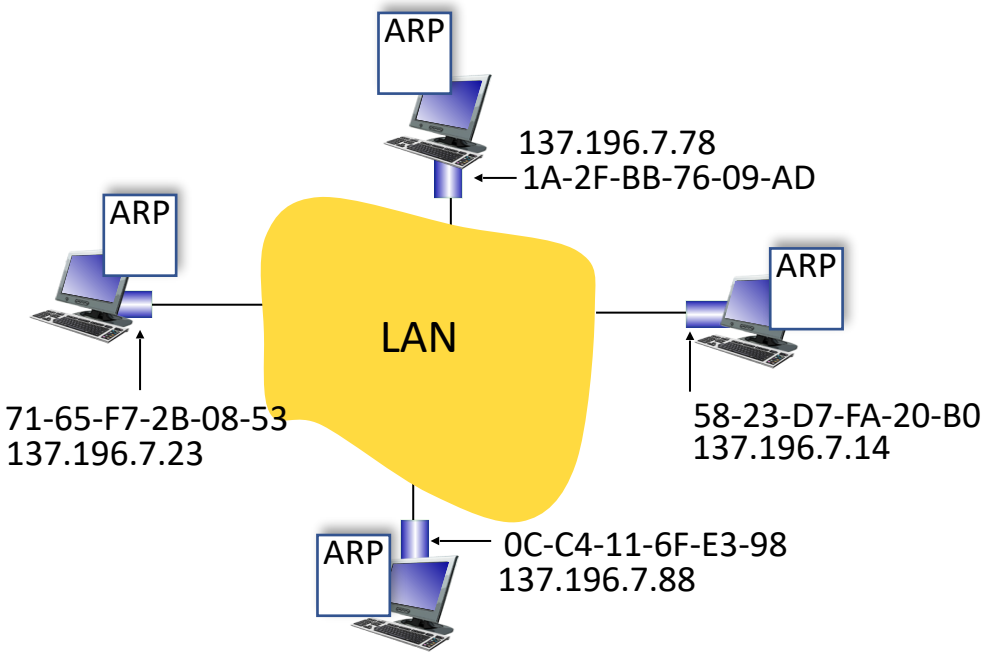
An Ethernet Address Resolution Protocol
-- or --
Converting Network Protocol Addresses
to 48.bit Ethernet Address
for Transmission on
Ethernet Hardware

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 node (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

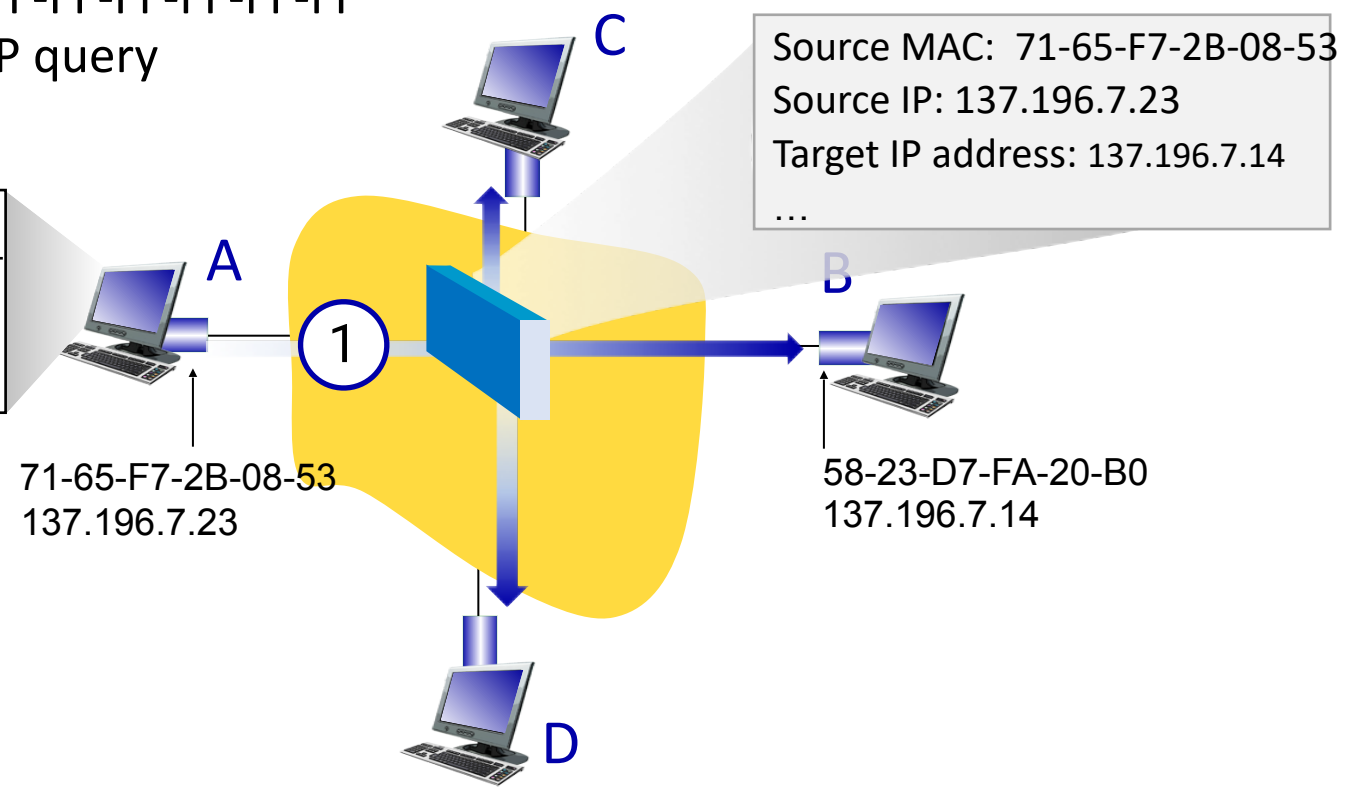
A broadcasts ARP query, containing B's IP addr

1

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

ARP table in A

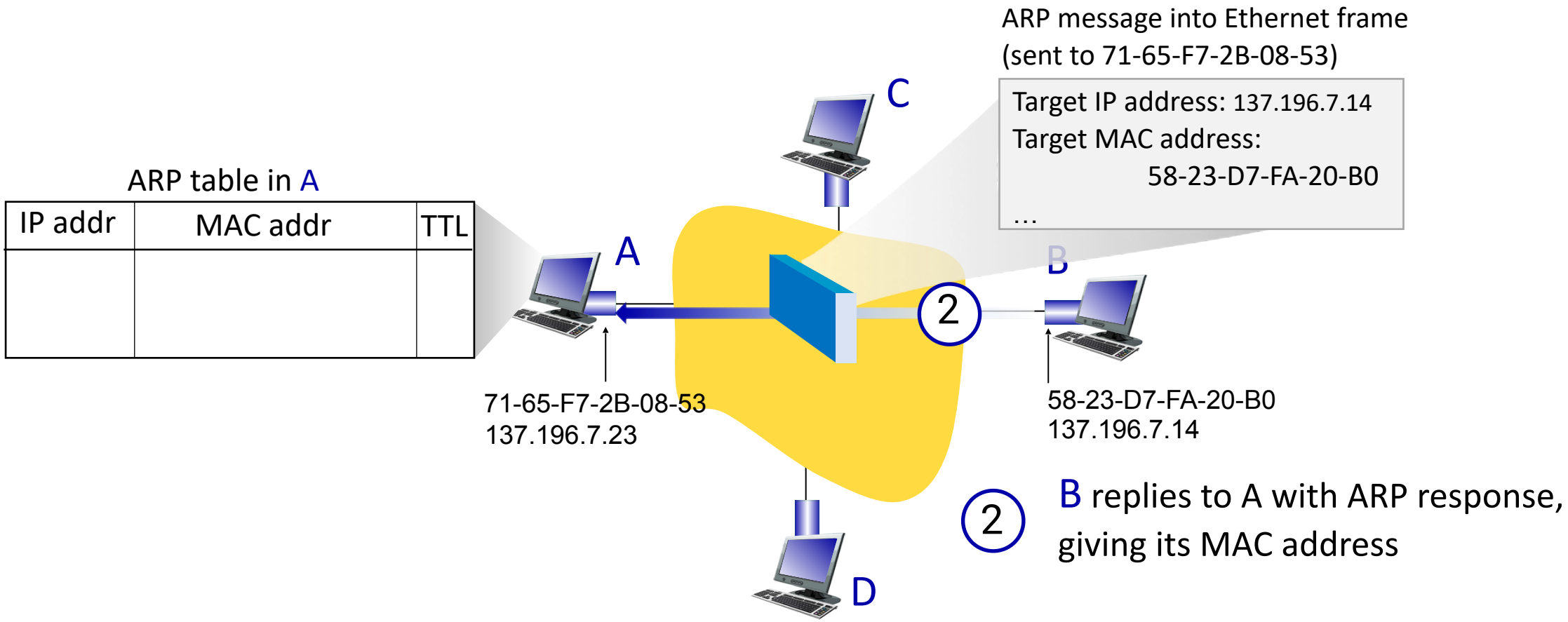
IP addr	MAC addr	TTL



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



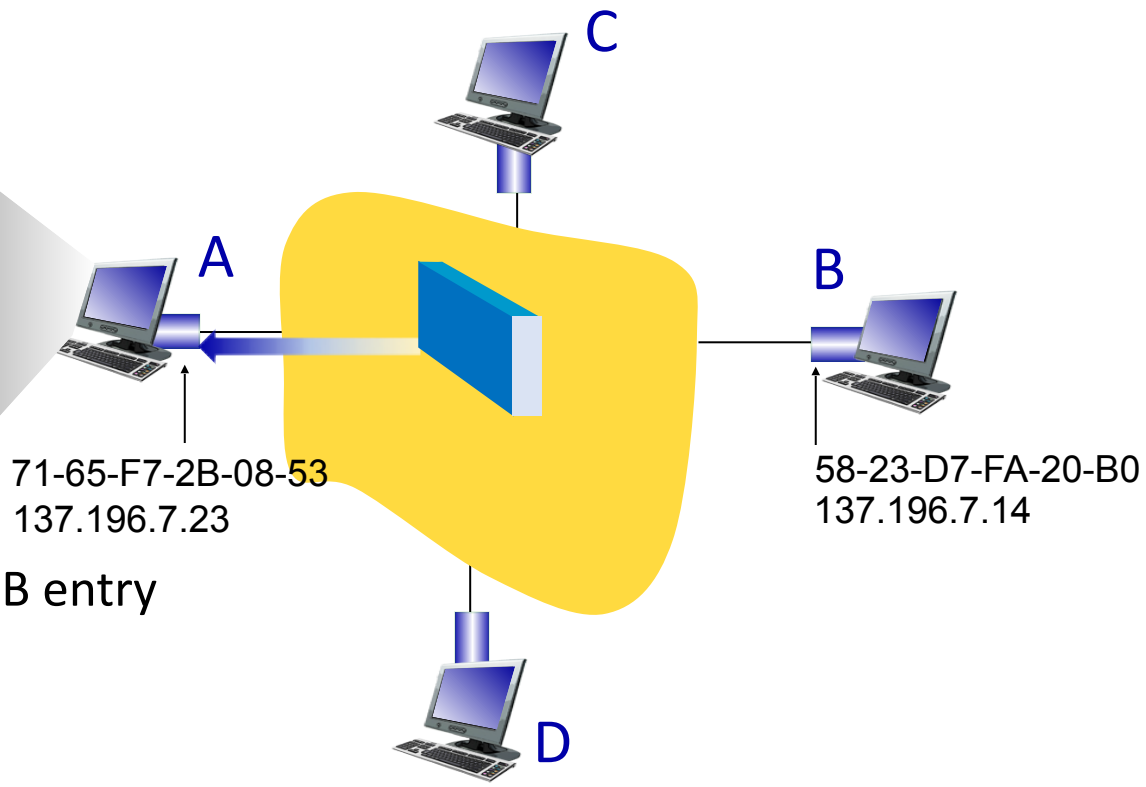
ARP in Action

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

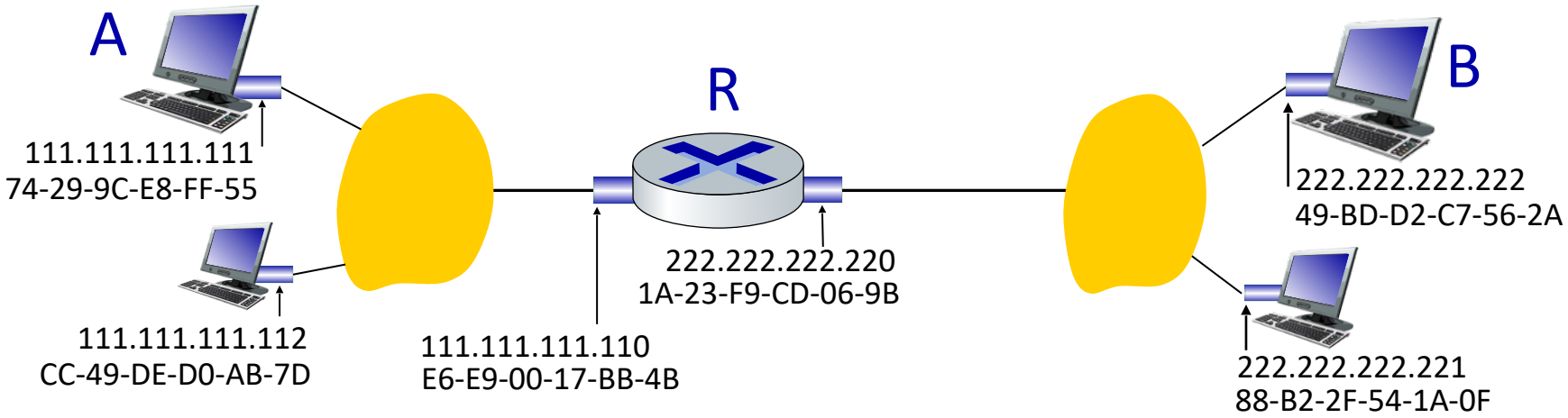


3 A receives B's reply, adds B entry into its local ARP table

Routing to Another Subnet

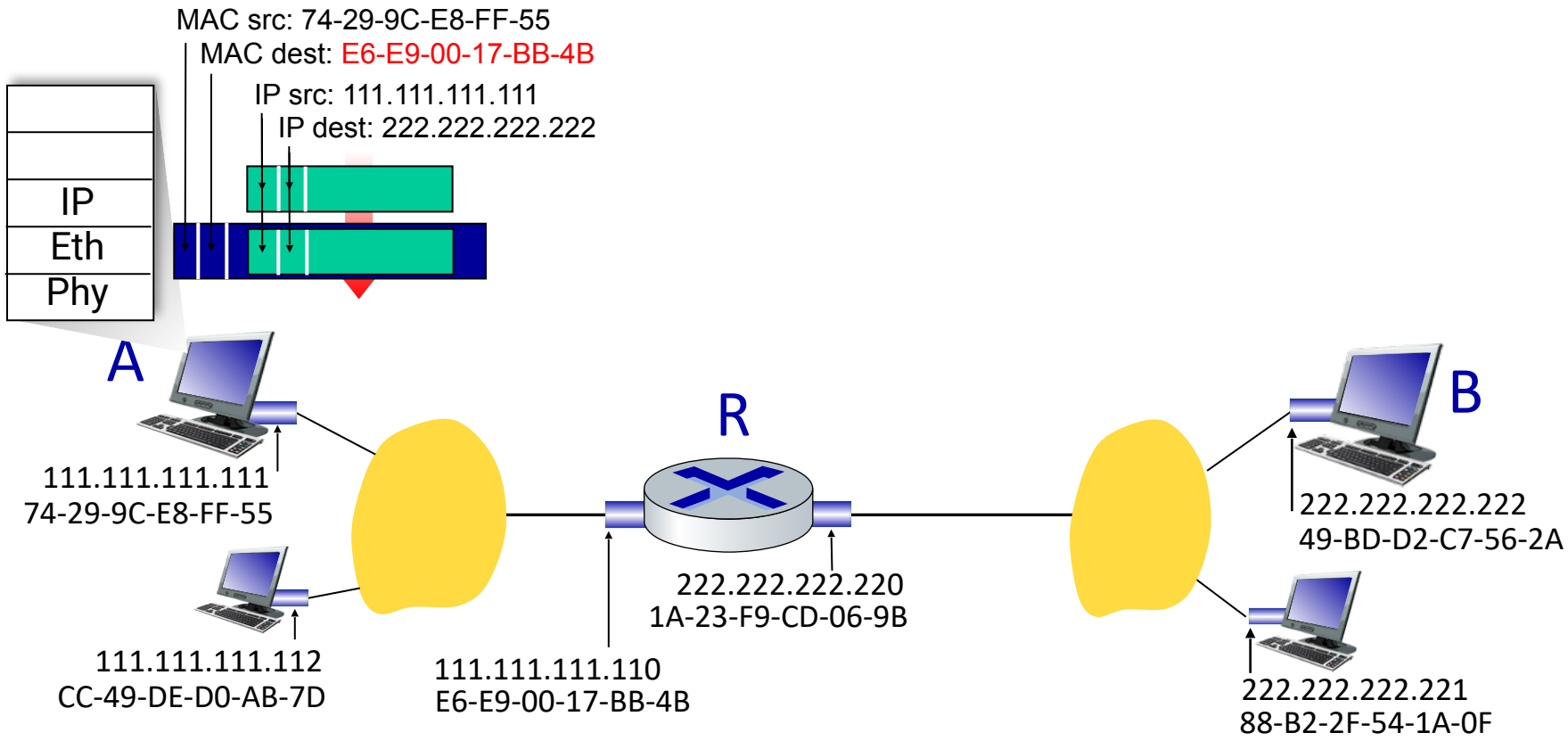
E.g., A wants to send datagram to B **via R**

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



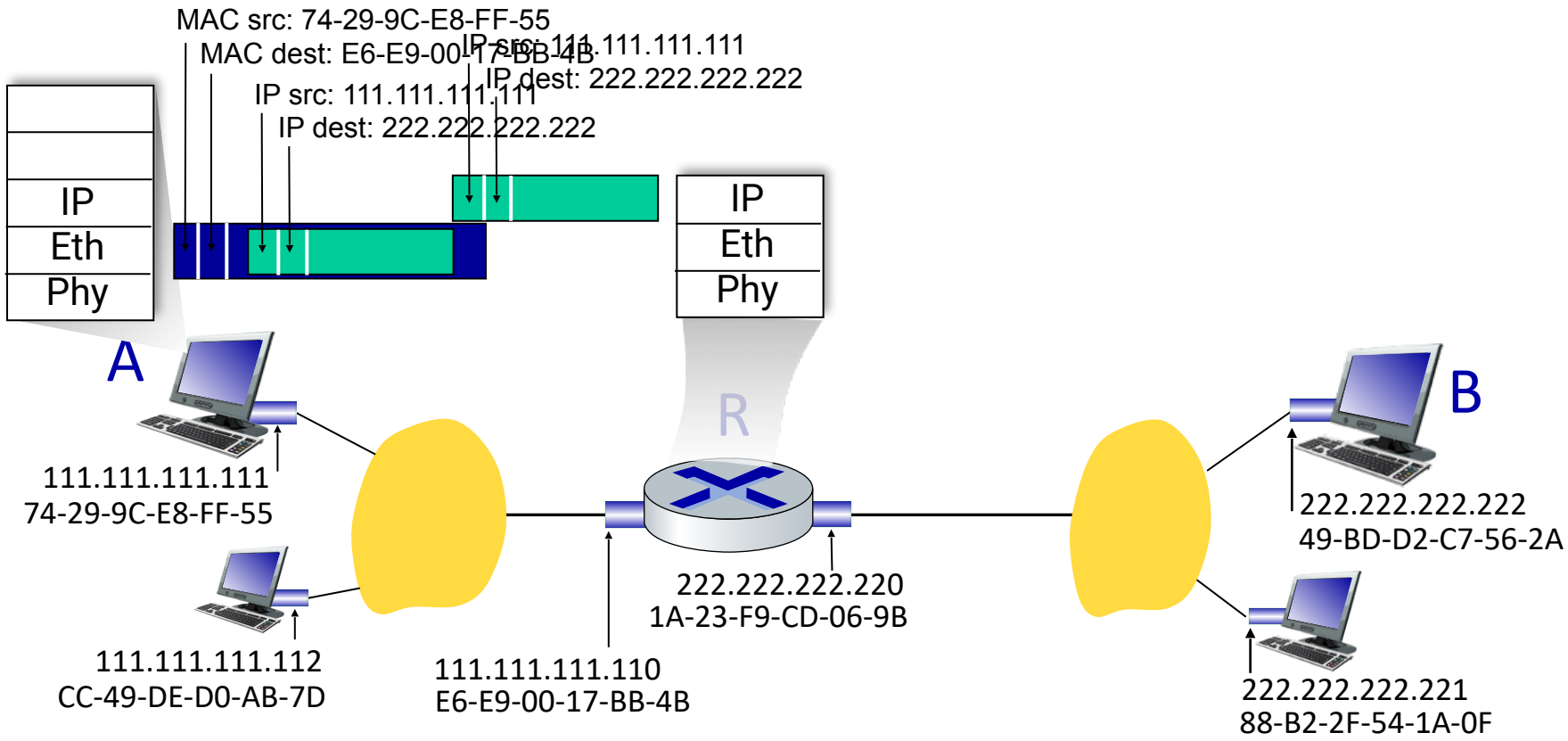
Routing to Another Subnet

- 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



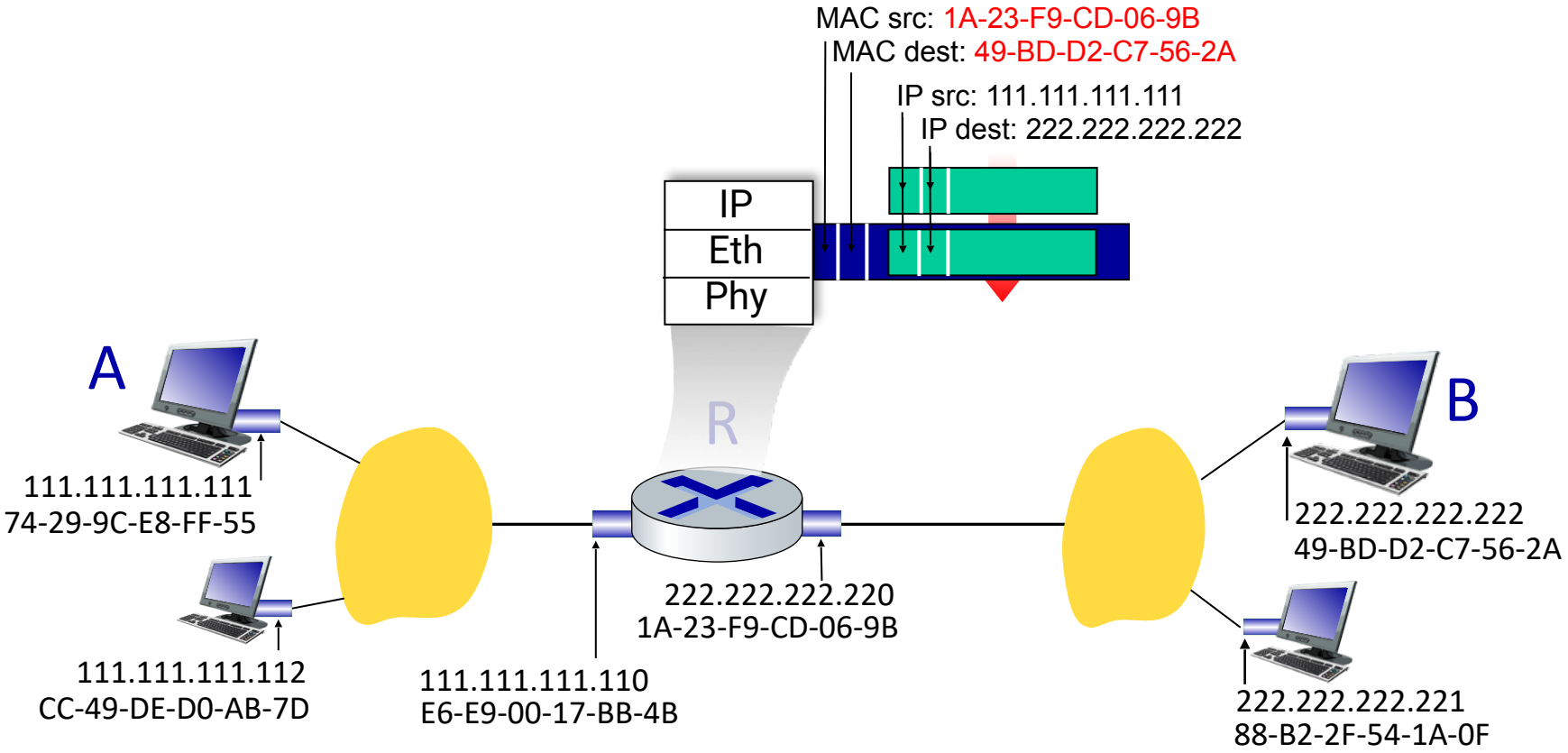
Routing to Another Subnet

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



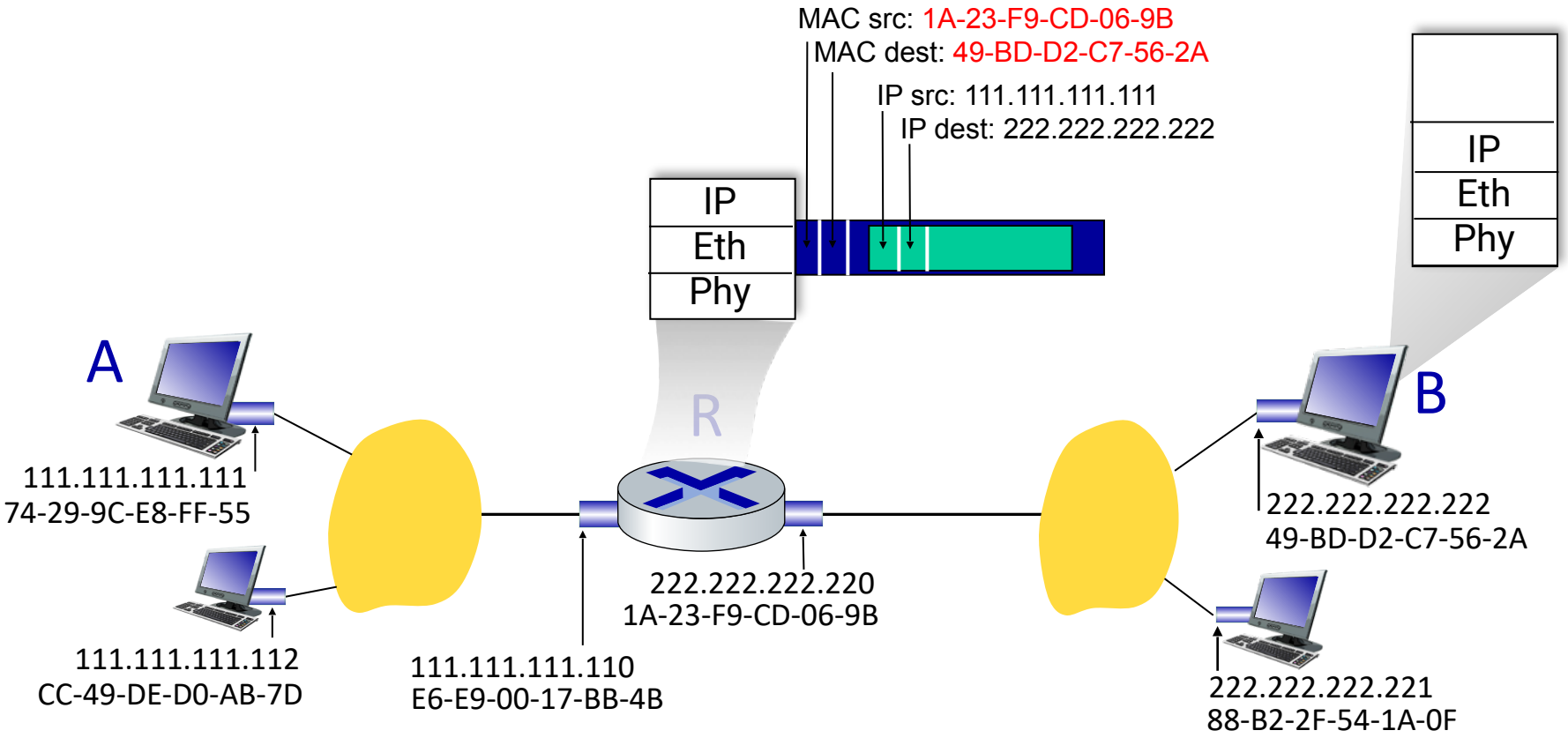
Routing to Another Subnet

- 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



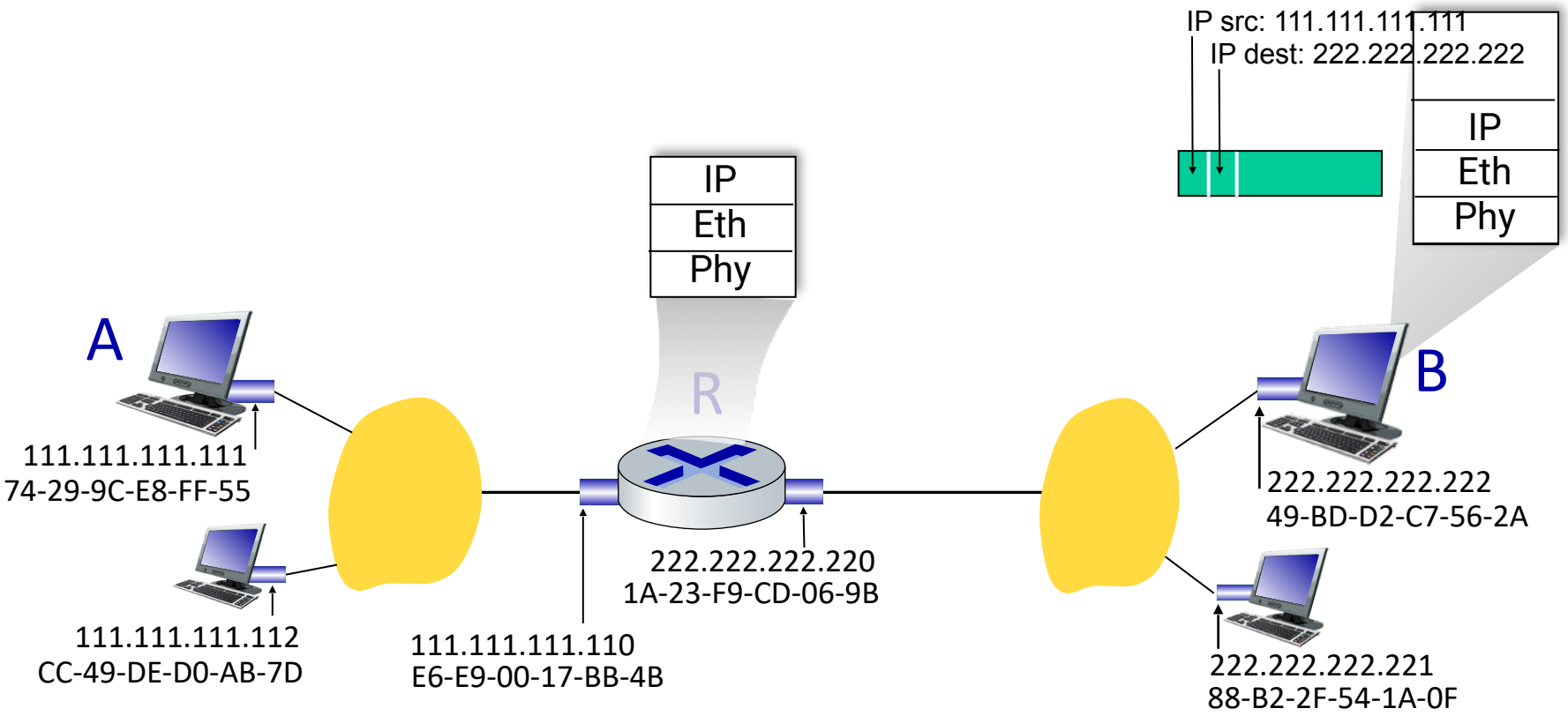
Routing to Another Subnet

- 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



IOWA

CS3640

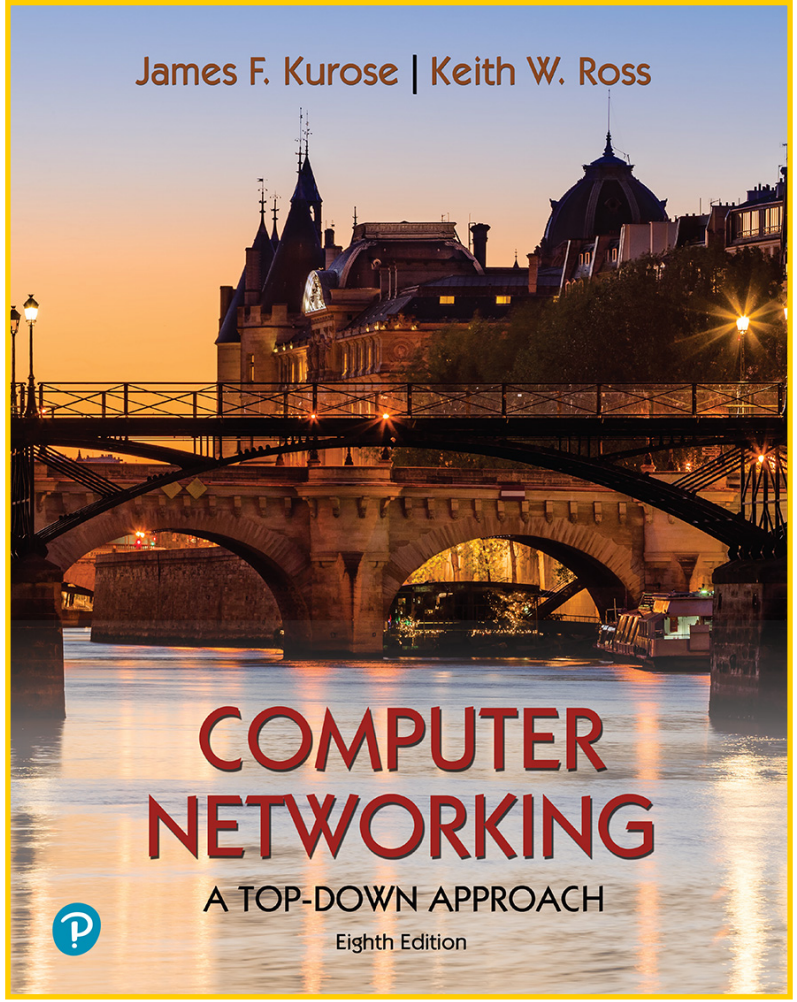
Link Layer (3): A Day in The Life of a Packet

Prof. Supreeth Shastri
Computer Science
The University of Iowa

Lecture Goals

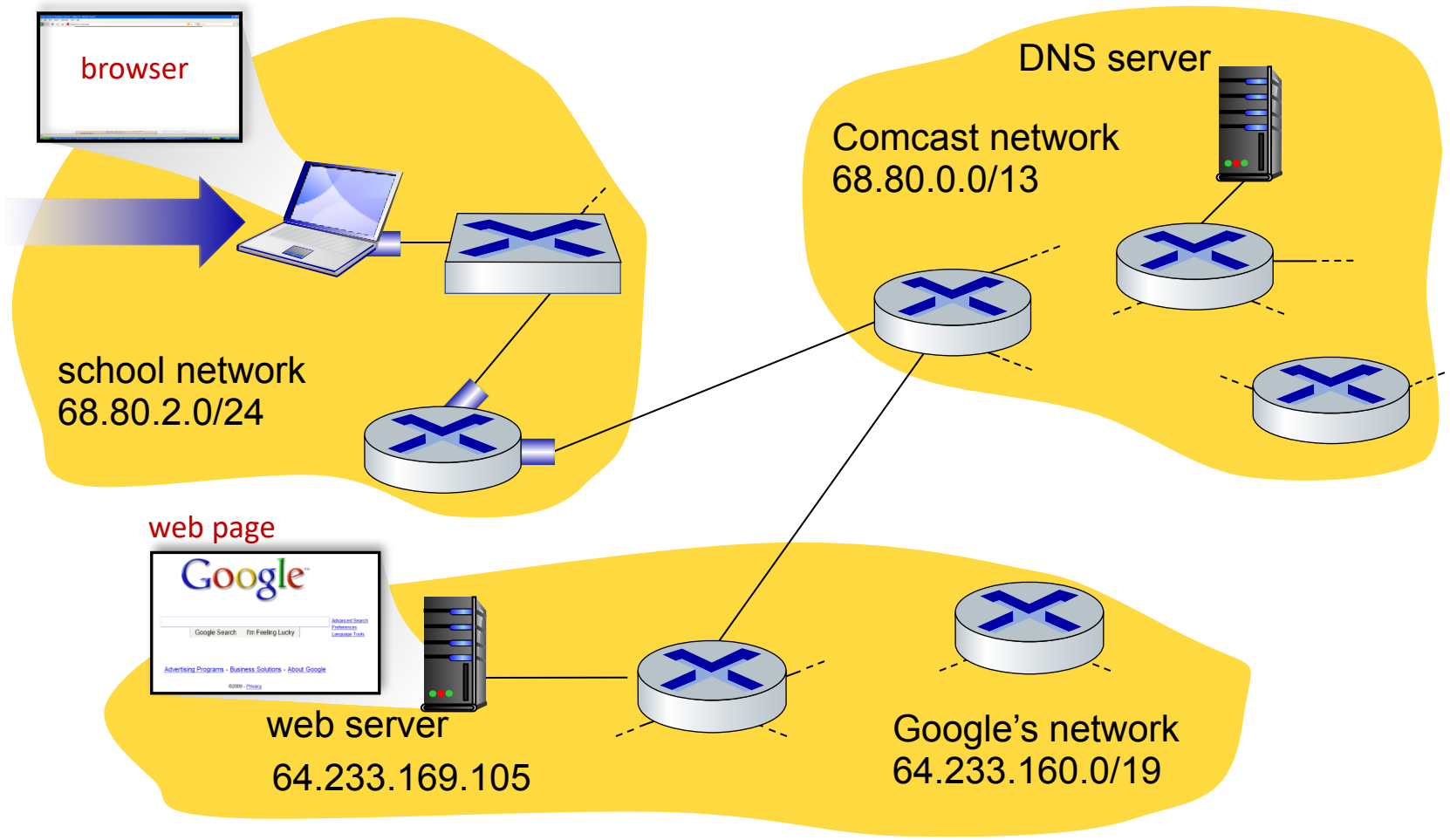
***Retrospective**: a day in the life of a web page request*

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



Chapter 6.7

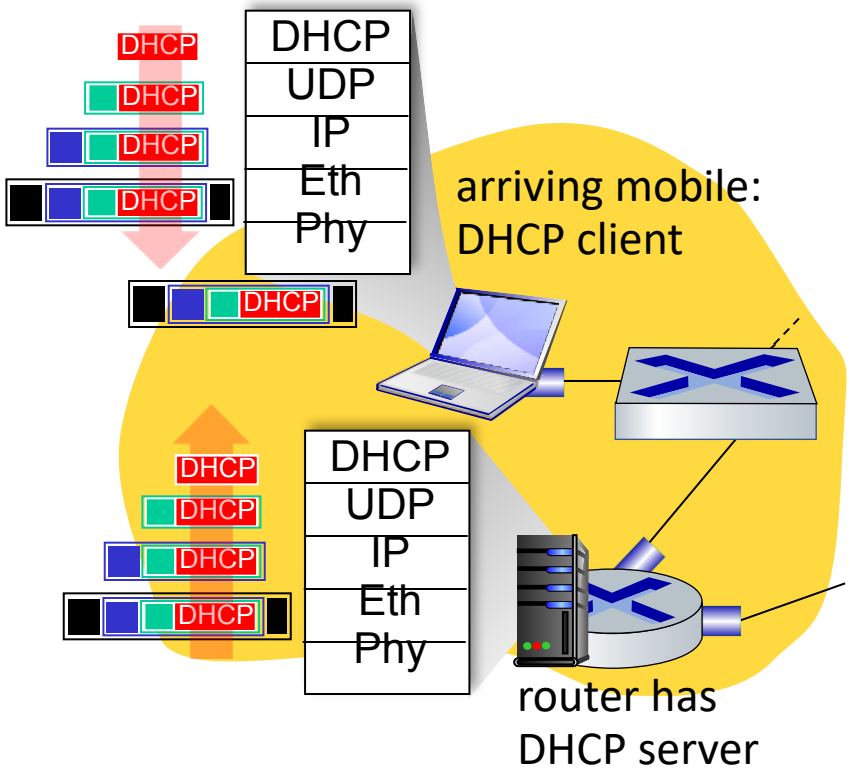
A day in the life: scenario



- *arriving mobile client attaches to network*
- *requests web page: www.google.com*

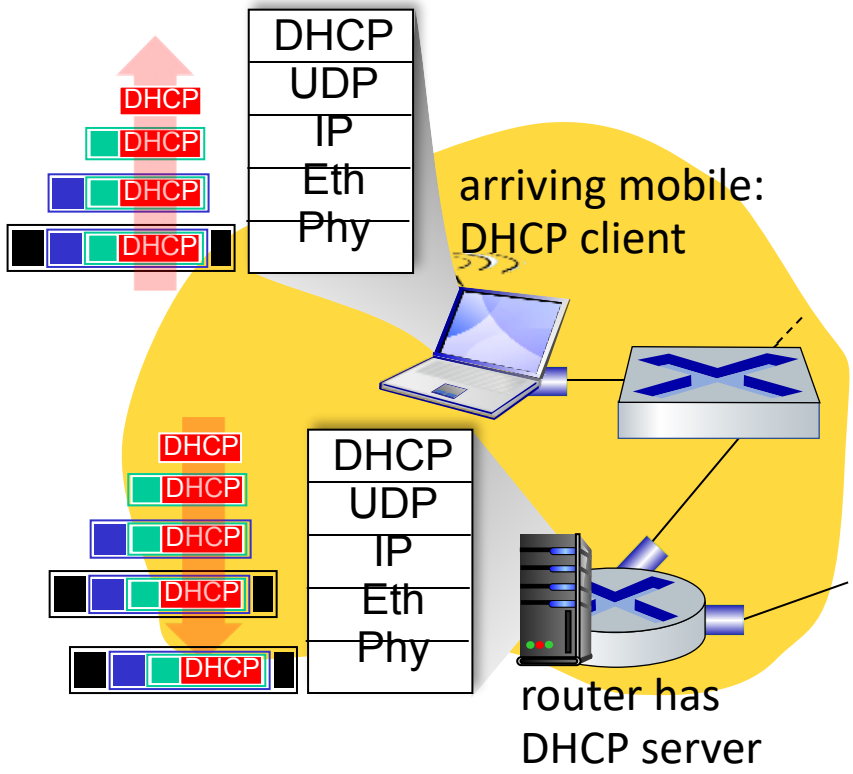
Sounds simple !

A day in the life: connecting to the Internet



- *connecting laptop needs to get its own IP address, address of first-hop router, address of DNS server: use DHCP*
- *DHCP request encapsulated in UDP, encapsulated in IP, encapsulated in 802.3 Ethernet*
- *Ethernet frame broadcast (dest: FF:FF:FF:FF:FF:FF) on LAN, received at the router running DHCP server*
- *Ethernet demuxed to IP demuxed to UDP demuxed to DHCP*

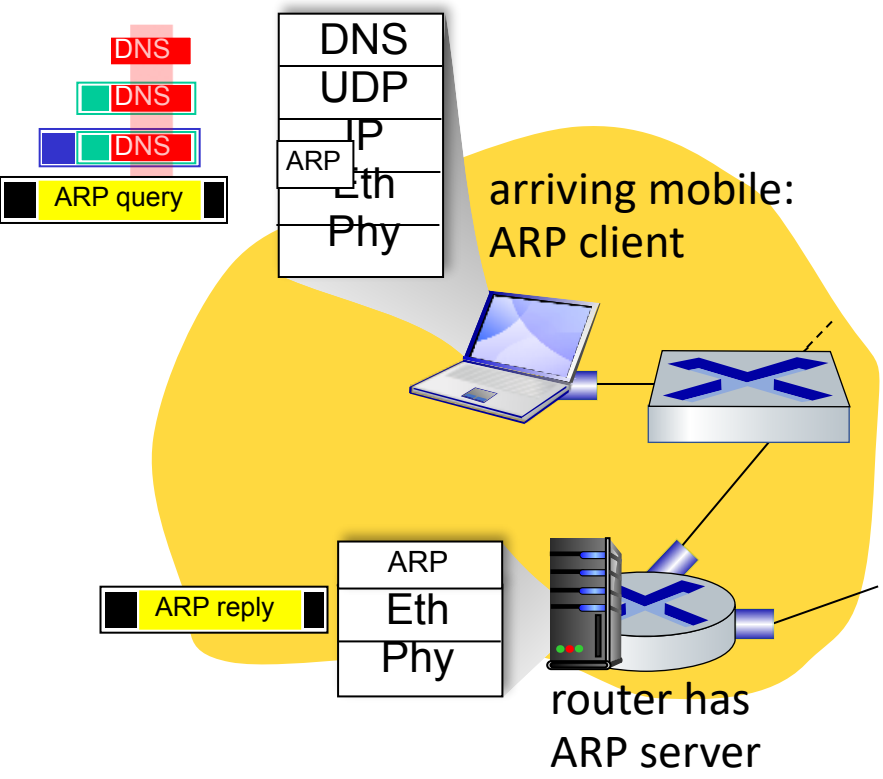
A day in the life: connecting to the Internet



- *DHCP server formulates DHCP ACK containing client's IP address, IP address of first-hop router, name and IP address of DNS server*
- *encapsulation at DHCP server, frame forwarded through LAN, demultiplexing at client*
- *DHCP client receives DHCP ACK reply*

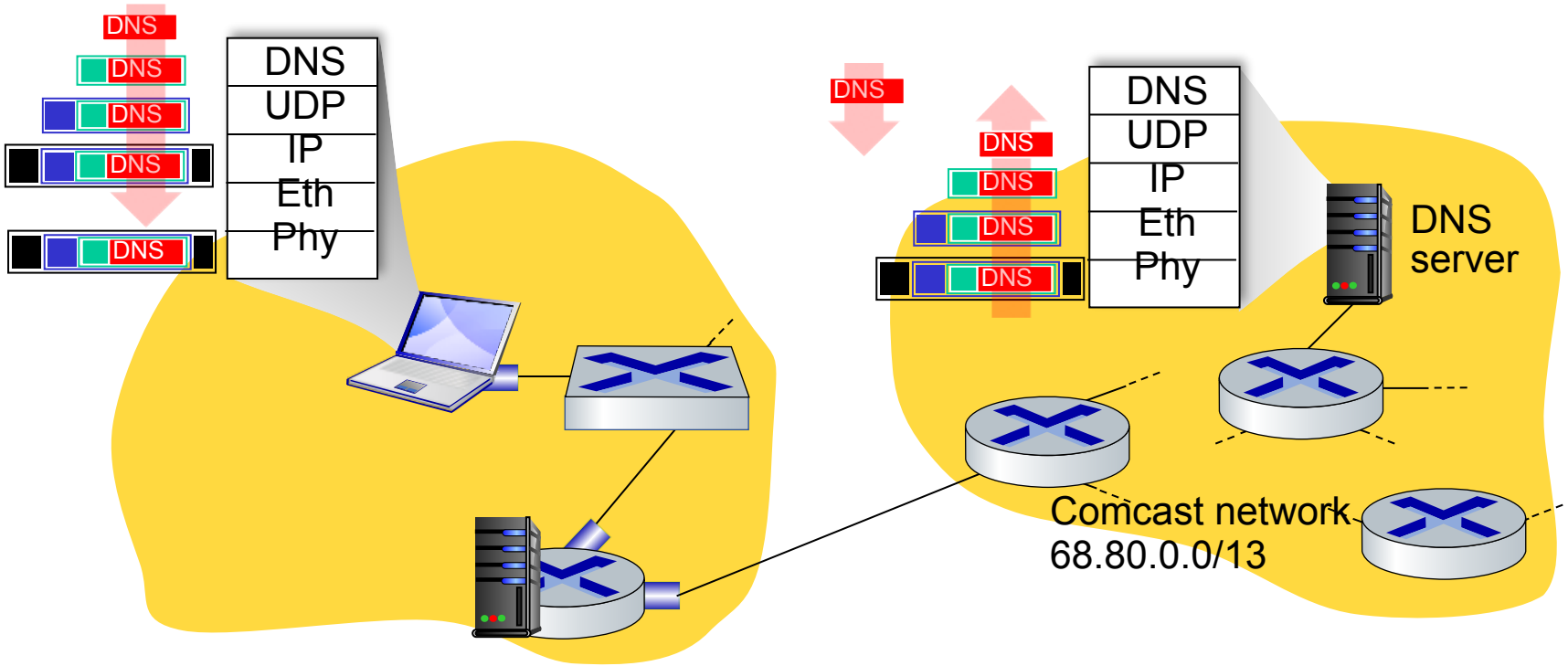
Client now has IP address, knows name and address of ***DNS server*** *and IP address of its* ***first-hop router***

A day in the life: ARP (before DNS, before HTTP)



- *before sending HTTP request, need IP address of www.google.com: use DNS*
- *DNS query created, encapsulated in UDP, encapsulated in IP, encapsulated in Ethernet. However, to send frame to router, need MAC address of router interface: use ARP*
- *ARP query broadcast, received by router, which replies with ARP reply giving MAC address of router interface*
- *client now knows MAC address of first hop router, so can now send frame containing DNS query*

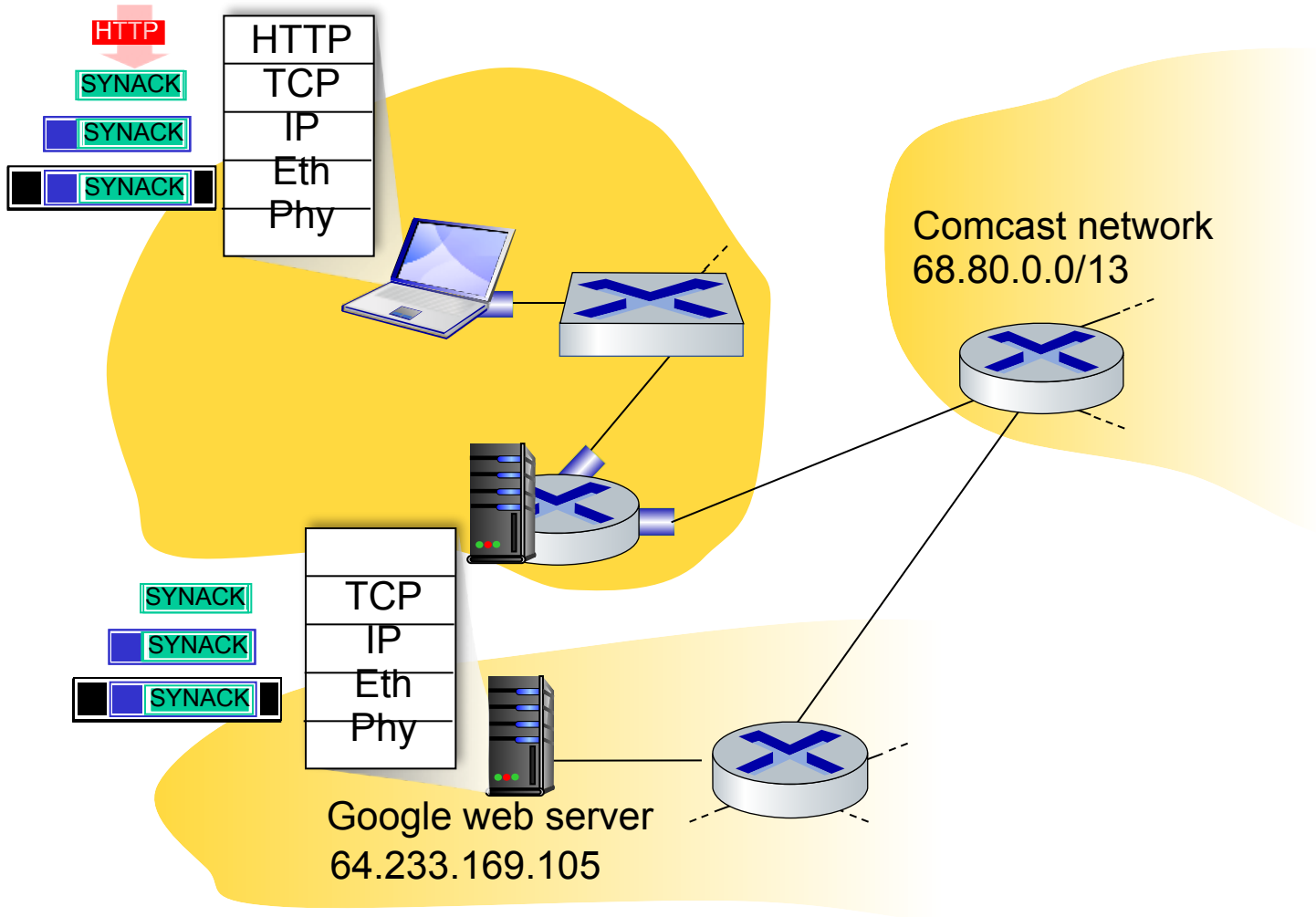
A day in the life: using DNS



- *demuxed to DNS*
- *DNS replies to client with IP address of www.google.com*

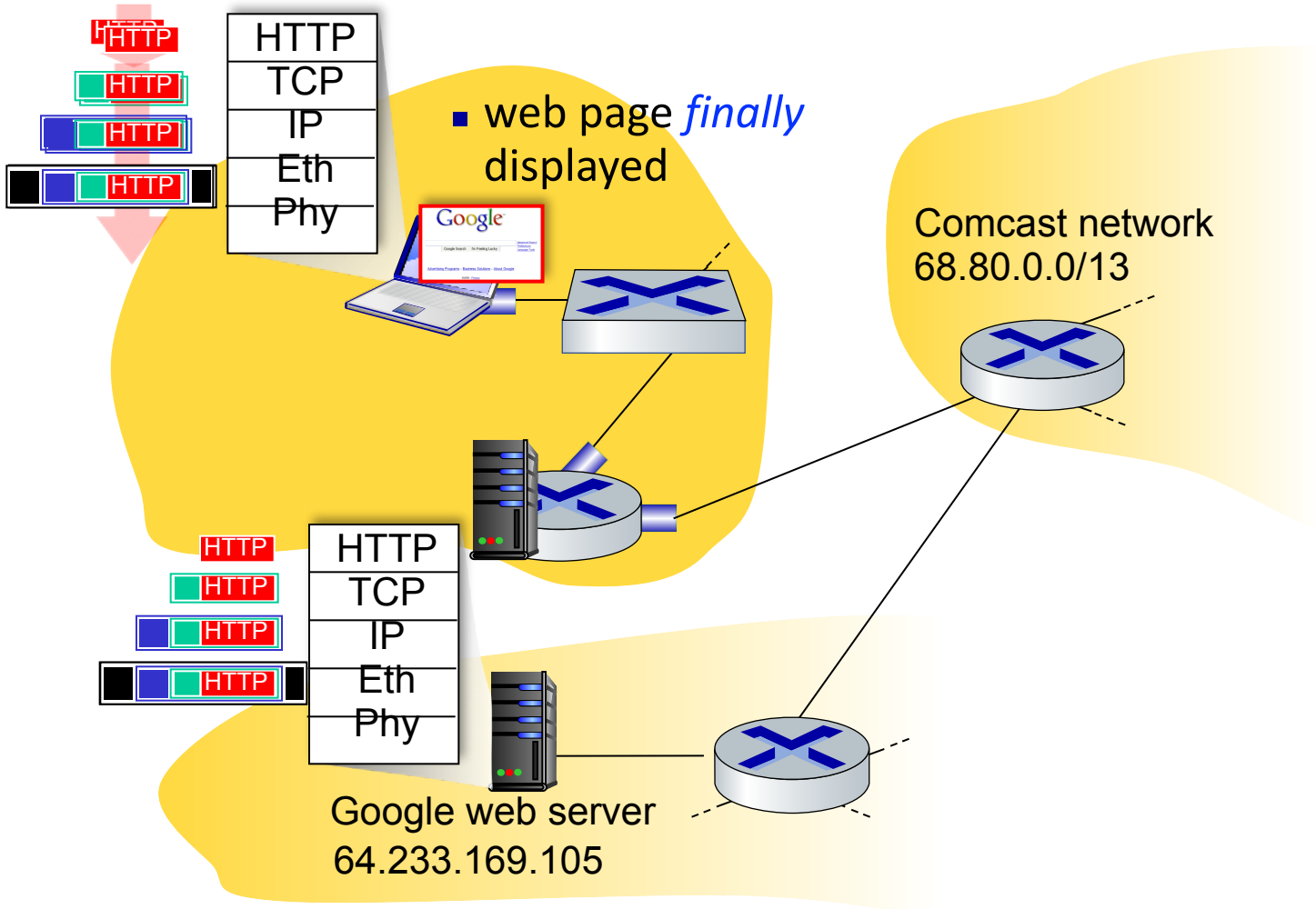
- *IP datagram containing DNS query forwarded via LAN switch from client to first-hop router*
- *IP datagram forwarded from campus network into Comcast network, routed (tables created by RIP, OSPF, and/or BGP routing protocols) to DNS server*

A day in the life: TCP connection carrying HTTP



- *To send HTTP request, client first opens TCP socket to web server*
- *TCP SYN segment (step 1 in TCP 3-way handshake) inter-domain routed to web server*
- *web server responds with TCP SYNACK (step 2 in TCP 3-way handshake)*
- *TCP connection established!*

A day in the life: HTTP request/reply

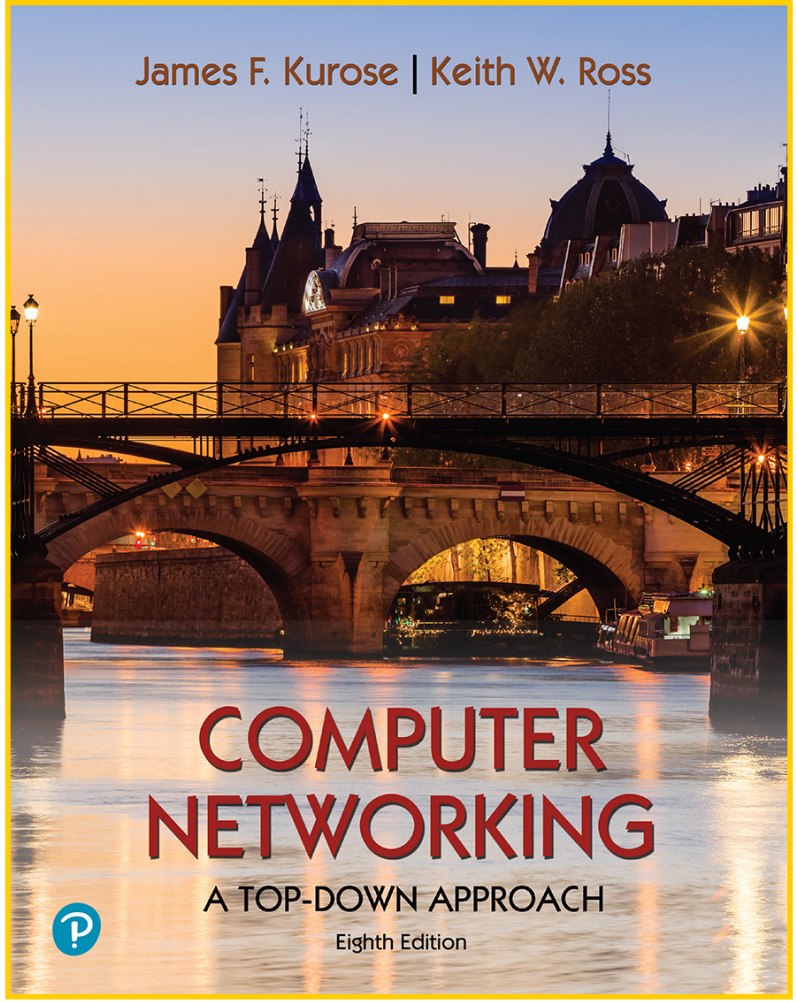


- *HTTP request sent into TCP socket*
- *IP datagram containing HTTP request routed to www.google.com*
- *web server responds with HTTP reply (containing web page)*
- *IP datagram containing HTTP reply routed back to client*

Next two lectures

Research topics to expand our horizon and get a taste of the state-of-the-art in networking

- *Software-Defined Networking*
- *Cloud Computing*



Chapters 4.4, 5.5

Spot Quiz (ICON)