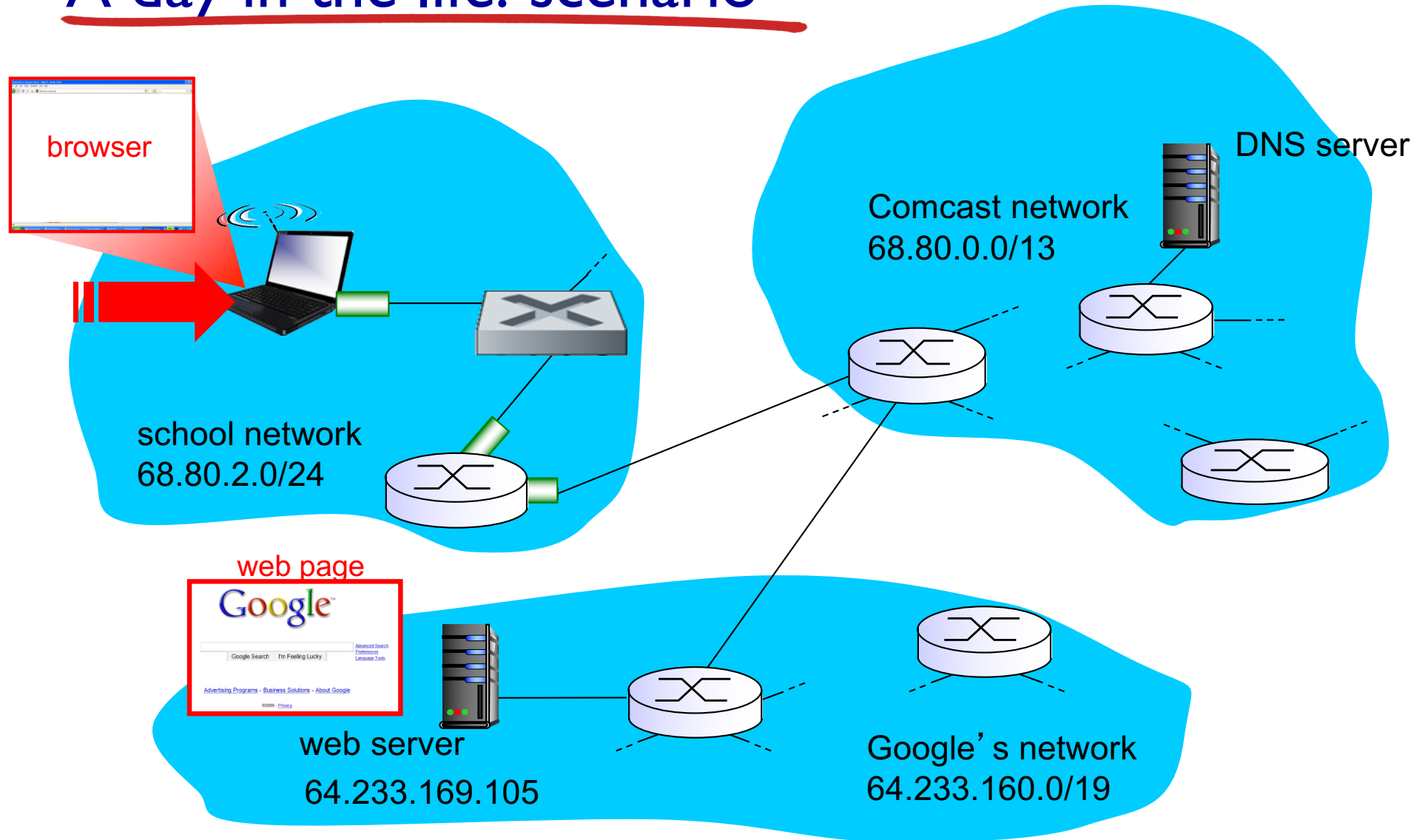# Link layer, LANs

6.7 a day in the life of a web request
     - a wrapup of what we learnt in this course
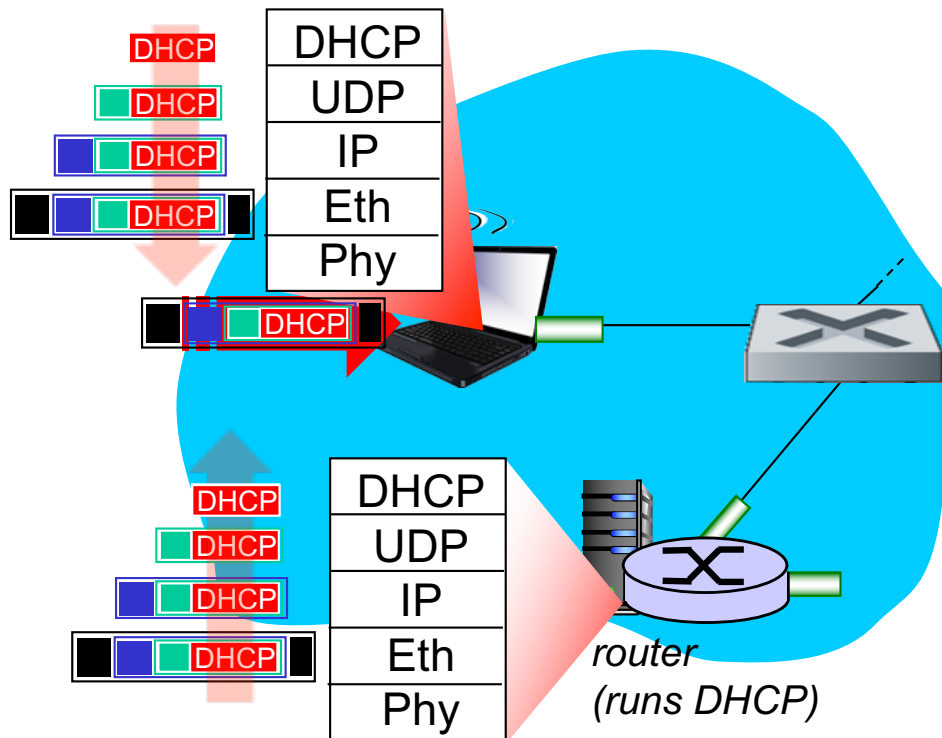
Synthesis: a day in the life of a web request

❖ **journey down protocol stack complete!**
  ▪ application, transport, network, link

❖ **putting-it-all-together: synthesis!**
  ▪ *goal:* identify, review, understand protocols (at all layers) involved in seemingly simple scenario: requesting www page
  ▪ *scenario:* student attaches laptop to campus network, requests/receives www.google.com

A day in the life: scenario

browser

school network
68.80.2.0/24

web page

Google

web server
64.233.169.105

Comcast network
68.80.0.0/13

DNS server

Google's network
64.233.160.0/19

A day in the life… connecting to the Internet

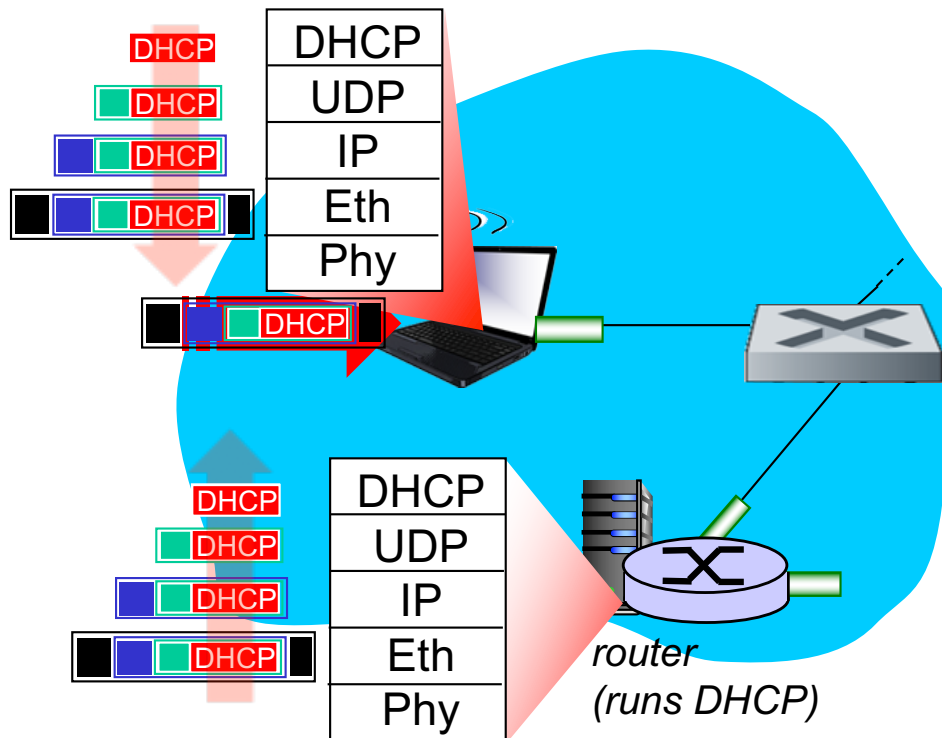DHCP
UDP
IP
Eth
Phy

DHCP
UDP
IP
Eth
Phy

*router
(runs DHCP)*

- ❖ connecting laptop needs to get its own IP address, addr of first-hop router, addr of DNS server: use *DHCP*

- ❖ DHCP Discover Message *encapsulated* in *UDP*, encapsulated in *IP*, encapsulated in *802.3* Ethernet

- ❖ Ethernet frame *broadcast* (dest: FFFFFFFFFFFF) on LAN, received at router running *DHCP* server

- ❖ Ethernet *demuxed* to IP demuxed, UDP demuxed to DHCP

A day in the life... connecting to the Internet

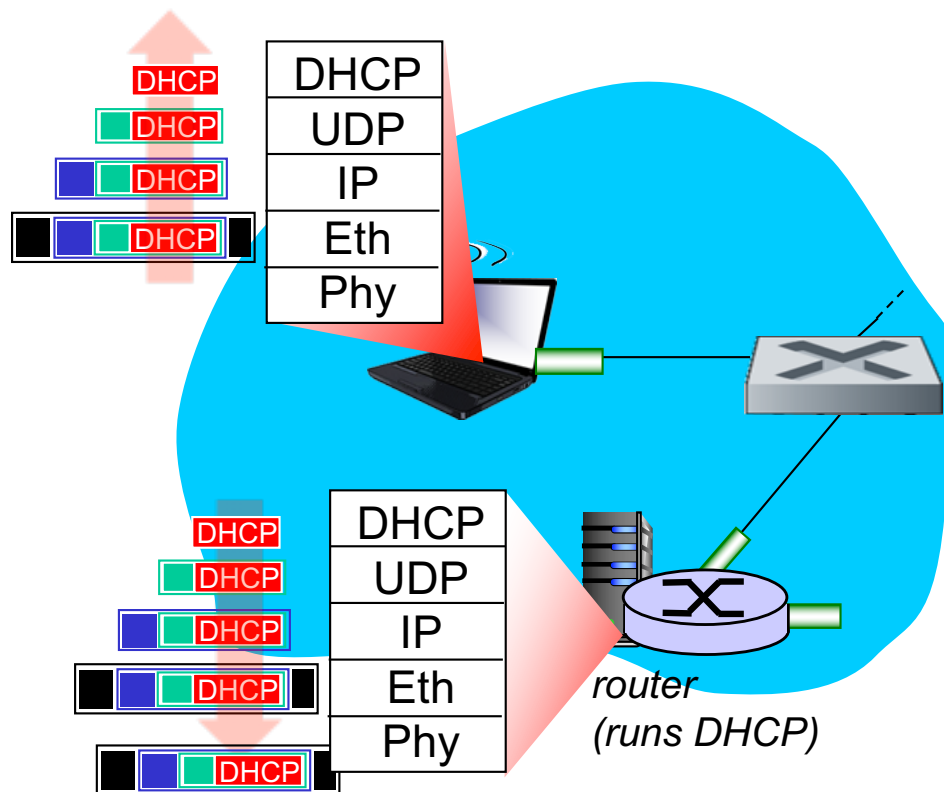DHCP
UDP
IP
Eth
Phy

DHCP
UDP
IP
Eth
Phy

*router
(runs DHCP)*
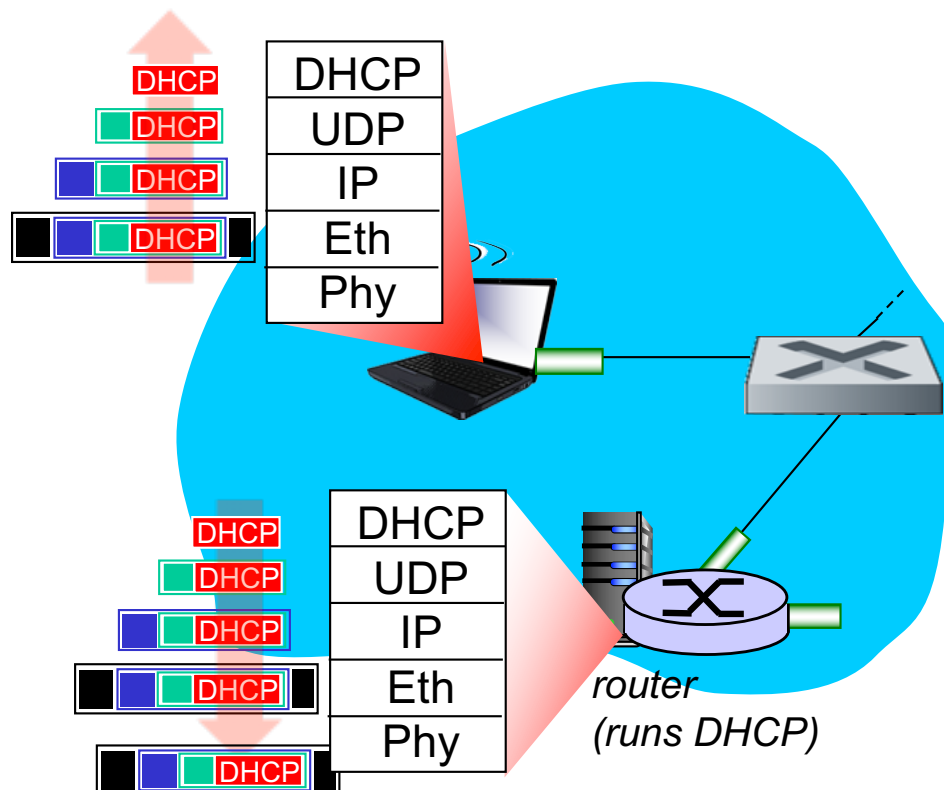
❖ DHCP server formulates *DHCP Offer* message containing client's IP address

❖ encapsulation at DHCP server, frame again broadcasted on LAN

❖ DHCP client receives DHCP Offer message

A day in the life… connecting to the Internet

- ❖ The client initiates DHCP **Request** message

- ❖ DHCP Request *encapsulated* in *UDP*, encapsulated in *IP*, encapsulated in *802.3* Ethernet

- ❖ Ethernet frame *broadcast* (dest: FFFFFFFFFFFF) on LAN, received at router running *DHCP* server

- ❖ Ethernet *demuxed* to IP demuxed, UDP demuxed to DHCP

A day in the life… connecting to the Internet

DHCP
DHCP
DHCP
DHCP

DHCP
UDP
IP
Eth
Phy

DHCP
DHCP
DHCP
DHCP
DHCP

DHCP
UDP
IP
Eth
Phy

router
(runs DHCP)

❖ DHCP server formulates *DHCP ACK* containing client's IP address, IP address of first-hop router for client, name & IP address of DNS server

❖ encapsulation at DHCP server, frame broadcasted through LAN,

❖ DHCP client receives DHCP ACK reply

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

A day in the life… ARP (before DNS, before HTTP)
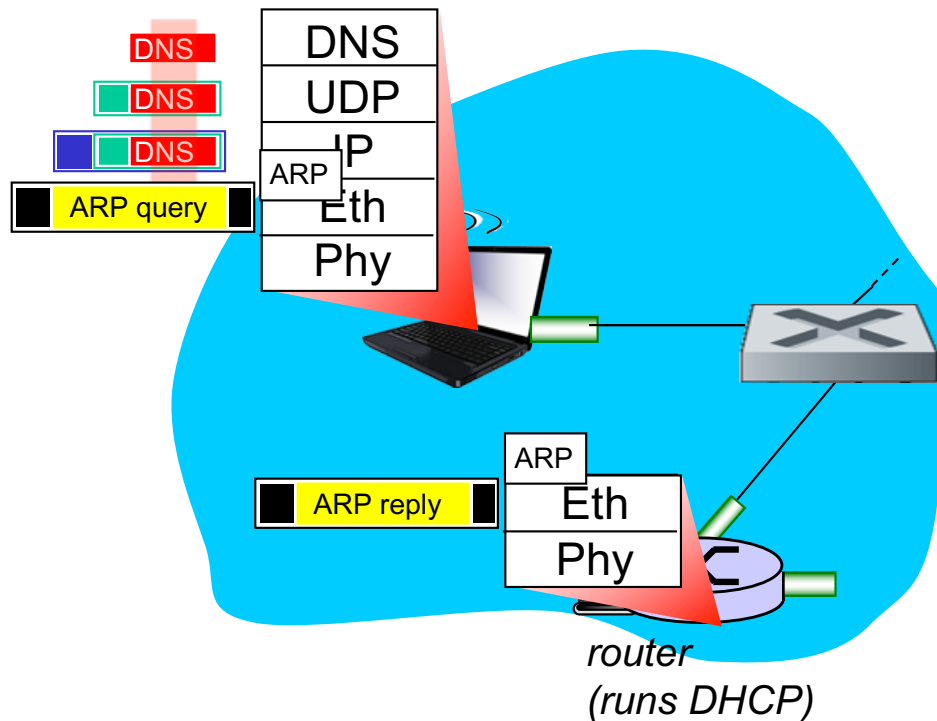
router
(runs DHCP)

- ❖ before sending *HTTP* request, need IP address of www.google.com: *DNS*

- ❖ DNS query created, encapsulated in UDP, encapsulated in IP, encapsulated in Eth. To send frame to DNS server, need MAC address of first hop router: *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

router
(runs DHCP)

Comcast network
68.80.0.0/13
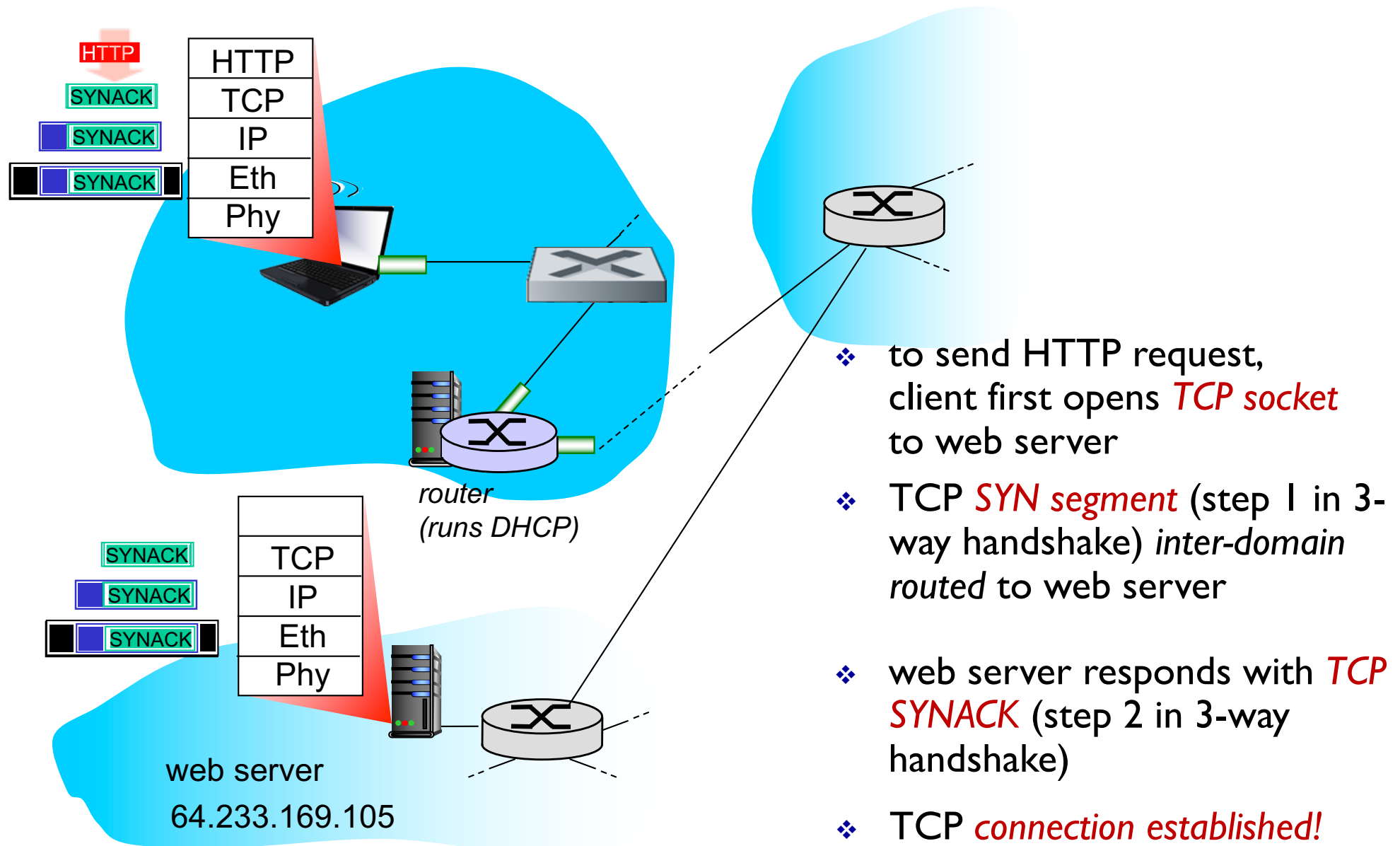
DNS server
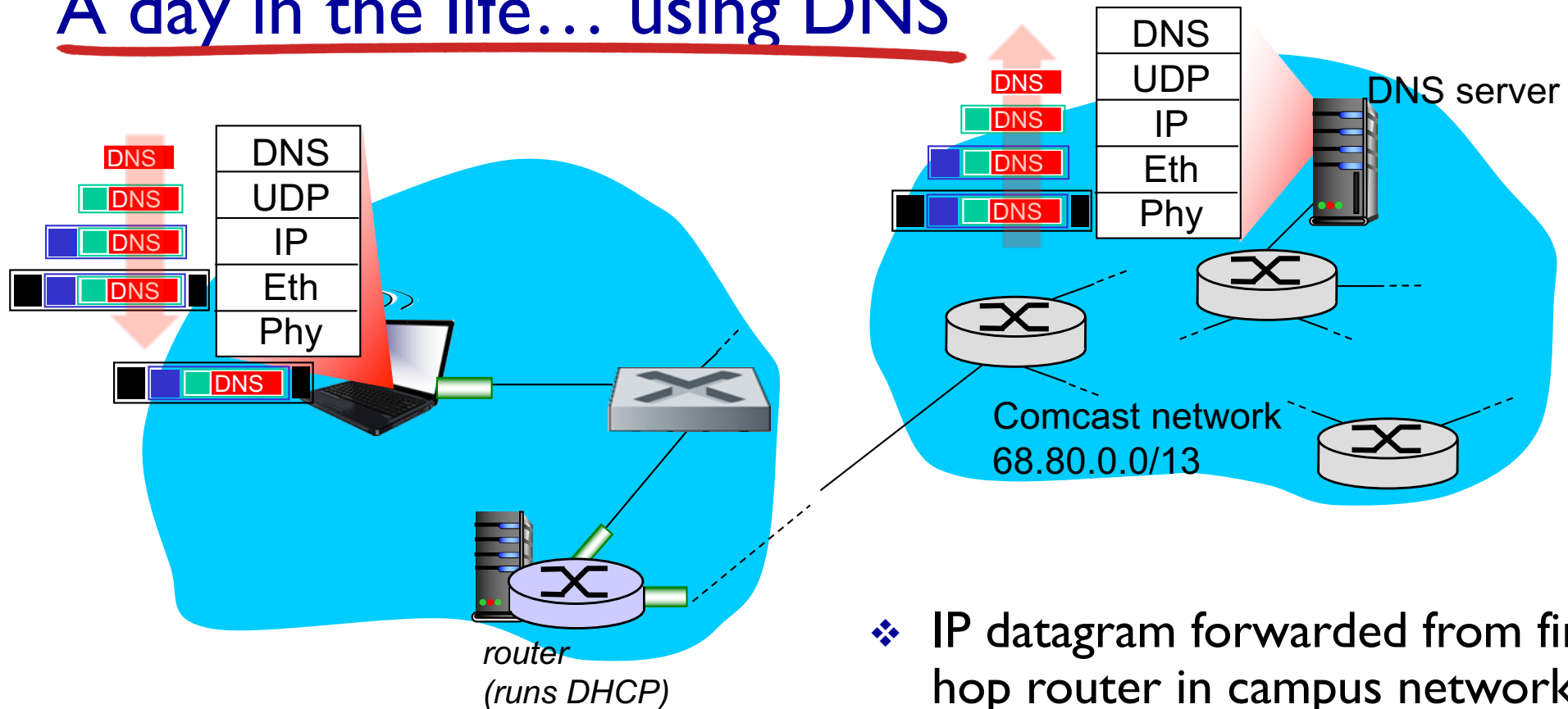
❖ IP datagram containing DNS query forwarded via LAN switch from client to first hop router

❖ IP datagram forwarded from first hop router in campus network into comcast network, routed (tables created by *RIP, OSPF, IS-IS* and/or *BGP* routing protocols) to DNS server

❖ demux'ed to DNS server

❖ DNS server replies to client with IP address of www.google.com

A day in the life…TCP connection carrying HTTP

HTTP

SYNACK

	SYNACK

		SYNACK

HTTP
TCP
IP
Eth
Phy

*router
(runs DHCP)*

SYNACK

	SYNACK

		SYNACK

TCP
IP
Eth
Phy

web server
64.233.169.105

- ❖ to send HTTP request, client first opens *TCP socket* to web server

- ❖ TCP *SYN segment* (step 1 in 3-way handshake) *inter-domain routed* to web server

- ❖ web server responds with *TCP SYNACK* (step 2 in 3-way handshake)

- ❖ TCP *connection established!*

A day in the life… HTTP request/reply

❖ web page *finally (!!!)* displayed

❖ *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