50.012 Networks

Lecture 19: Datacenter networking, putting things together

2021 Term 6

Assoc. Prof. CHEN Binbin

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Outline

Data center networking

Synthesis: a day in the life of a web request

Read textbook Section 6.6, 6.7

Data center networks

- 10's to 100's of thousands of servers, often closely coupled, in close proximity:
  - content-servers (e.g., YouTube, Apple, Facebook)
  - search engine, eCommerce
  - XaaS

- challenges:
  - multiple applications, each serving massive numbers of clients
  - managing/balancing load
  - avoiding processing, networking, data bottlenecks

Inside a 40-ft Microsoft container, Chicago data center

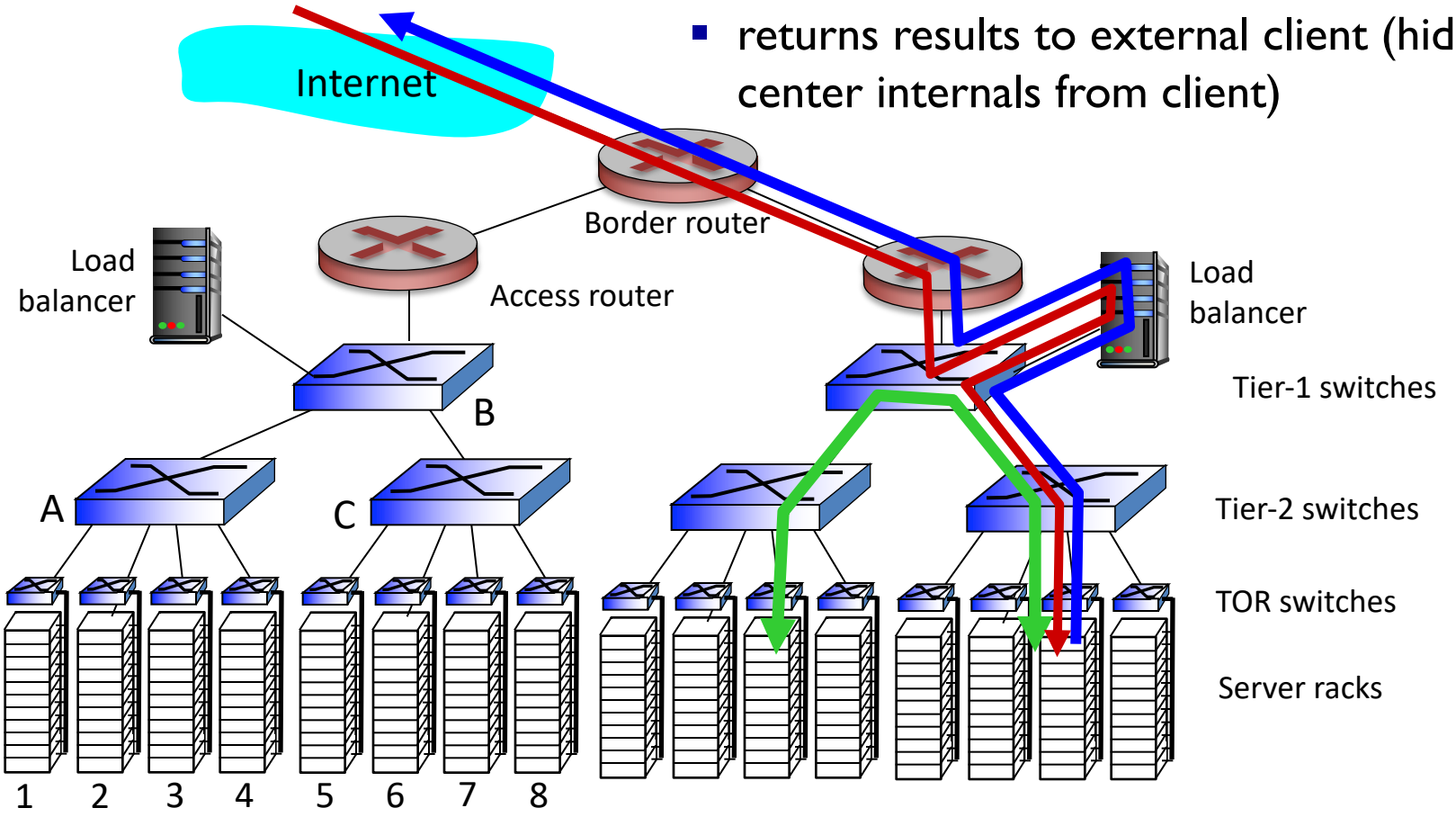
Know the hardware in a DC

- Servers are typically stacked in 19" racks
  - 42U rack is common, containing ~40 1U servers
- Each rack has its own top-of-the-rack (ToR) switch
  - Connects all servers with each other (48 x 10GbE ports is common)
  - also provides uplink
- Data center servers:
  - Compute nodes: high memory (a few hundred GB) + high number of cores (e.g., 24+)
  - Storage nodes (Network-Attached Storage): e.g. SSD for fast access, HDD for cheap storage

Data center networks (early generation)

load balancer: application-layer routing

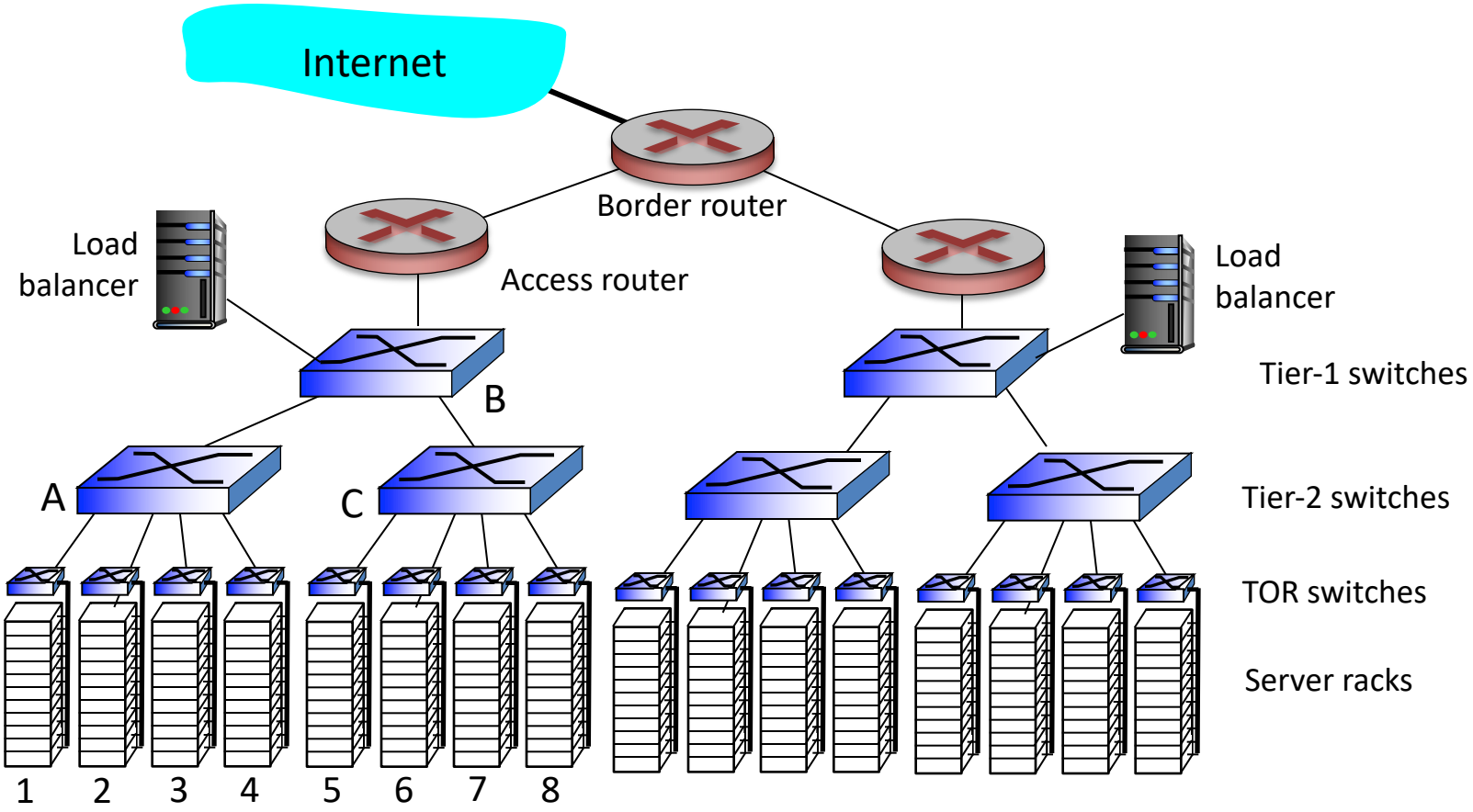
- receives external client requests
- directs workload within data center
- returns results to external client (hiding data center internals from client)



Data center networks (early generation)

Performance questions:

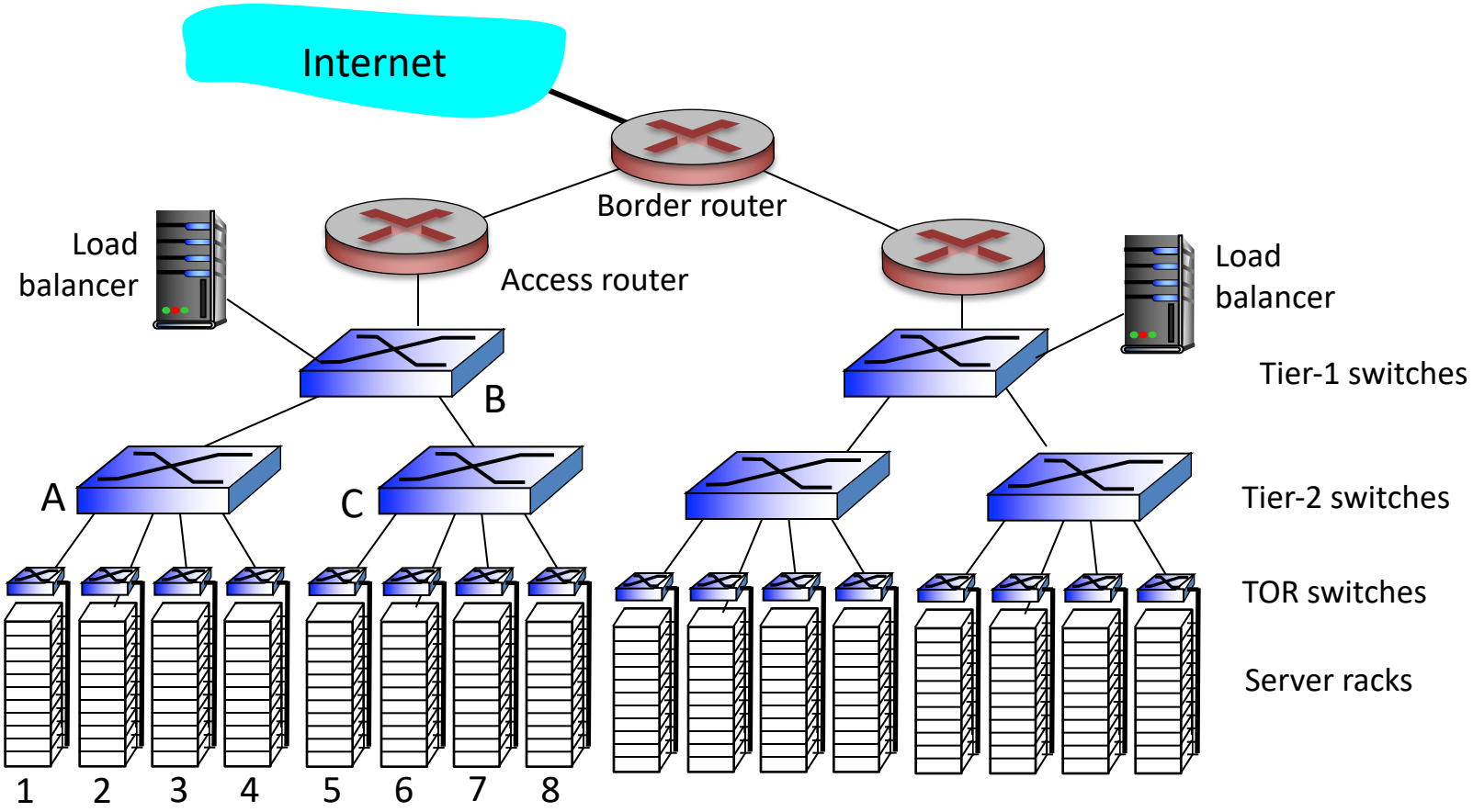
Assume all are 10GbE links, how much total traffic can be supported if N servers in rack 1 want to send to N other servers in the same rack?



Data center networks (early generation)

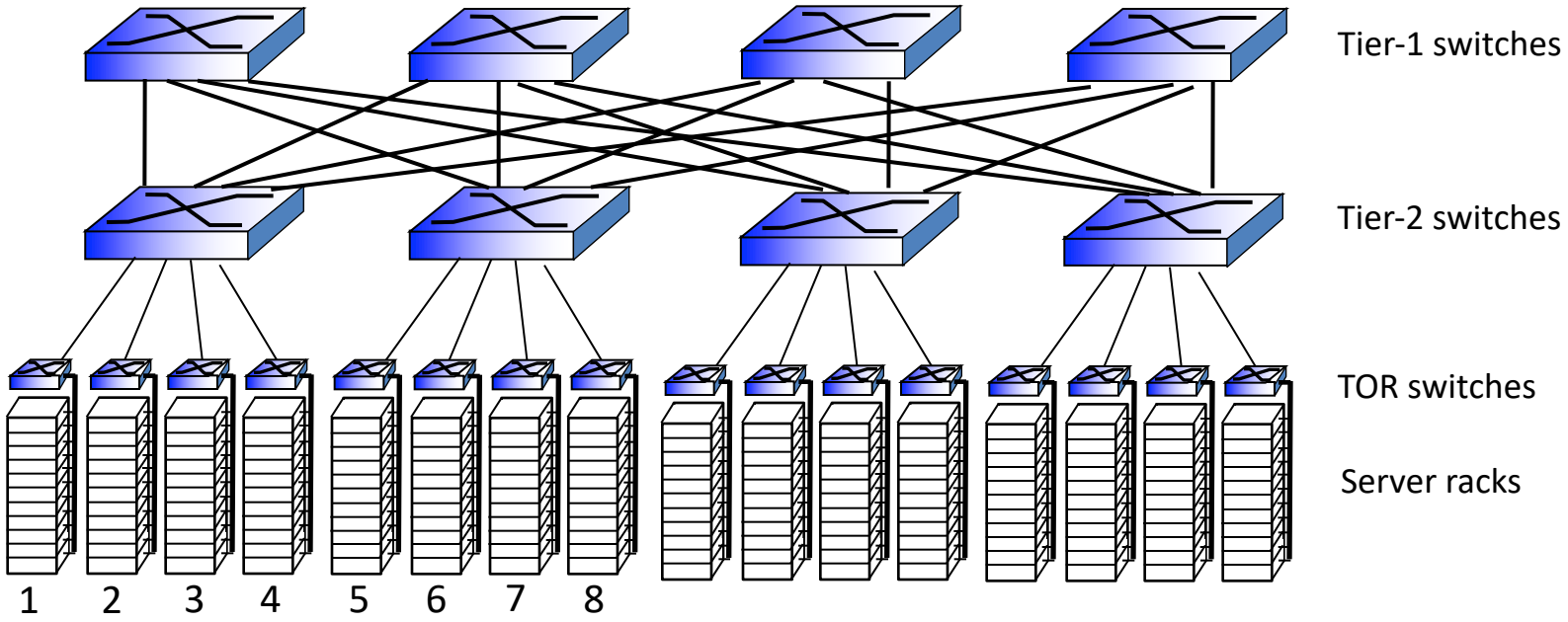
Performance questions:

Assume all are 10GbE links, how much total traffic can be supported if N servers in rack 1 want to send to N servers in another rack (e.g., rack 2, or rack 5)?



Data center networks - evolved

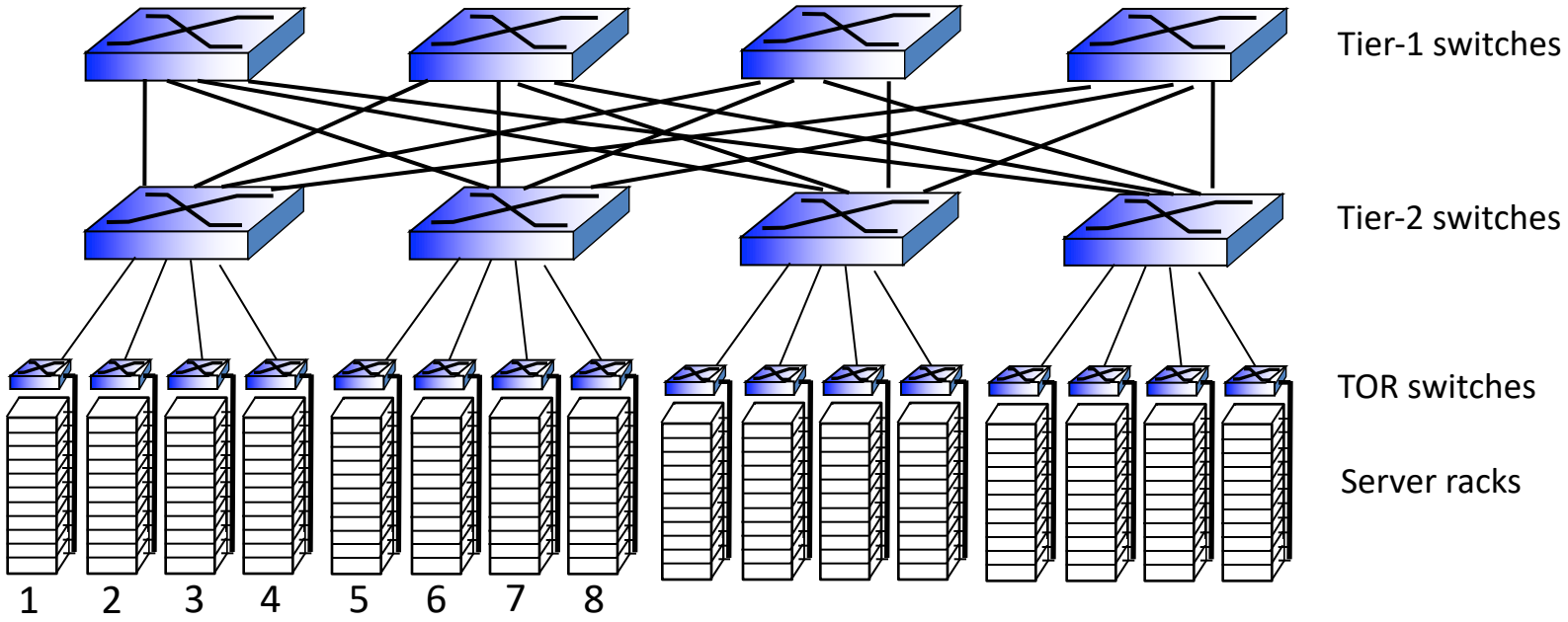
- rich interconnection among switches, racks:
  - increased throughput between racks (multiple routing paths possible)
  - increased reliability via redundancy



Data center networks - evolved

Performance questions:

Assume 10GbE links between ToR switch and servers, 160GbE from ToR switch to tier-2 switch, and 160 GbE from tier-2 to tier 1 switch, what maximum total throughput can be supported if 64 servers (16 each in rack 1, 2, 3, and 4) want to talk to 64 other servers (16 each in rack 5, 6, 7, and 8)?



Outline

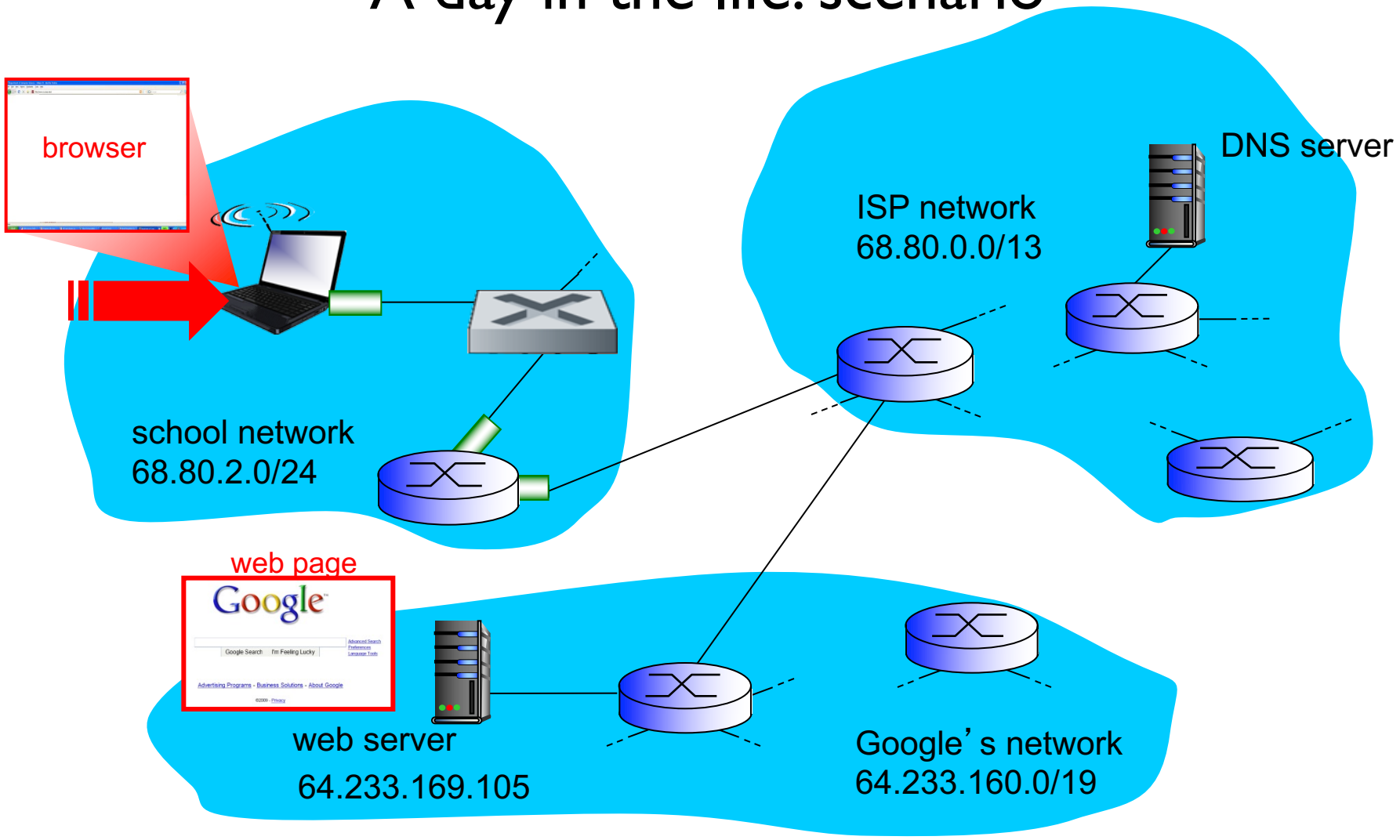
Data center networking

Synthesis: a day in the life of a web request

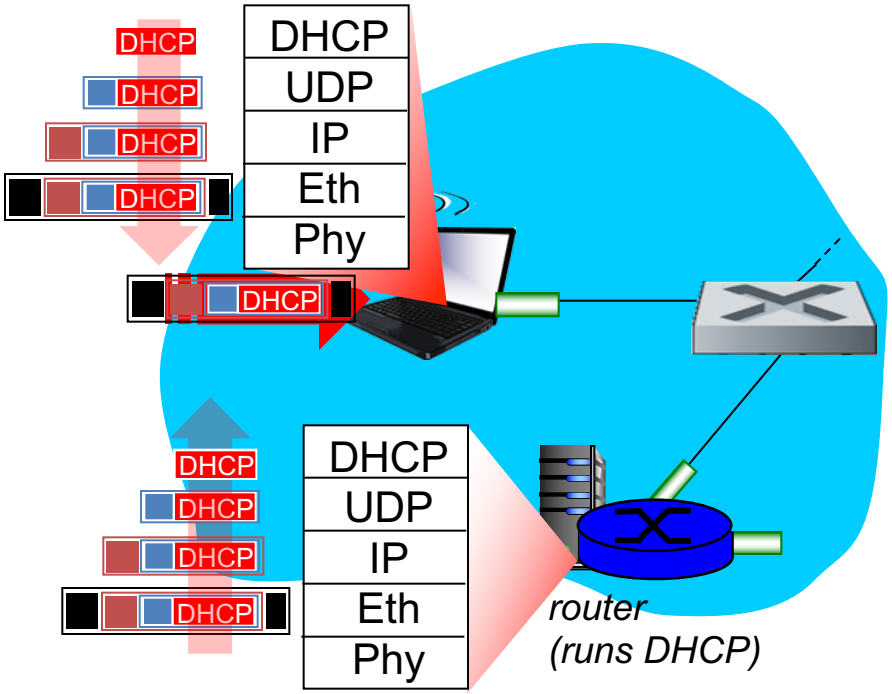
Synthesis: a day in the life of a web request

- 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



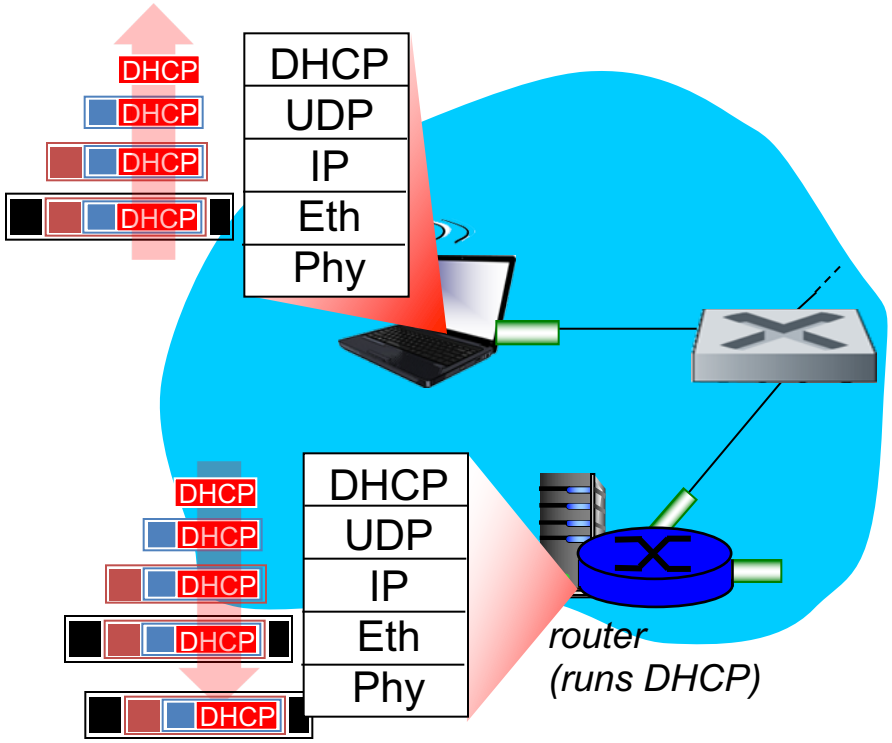
A day in the life… connecting to the Internet



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

- 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

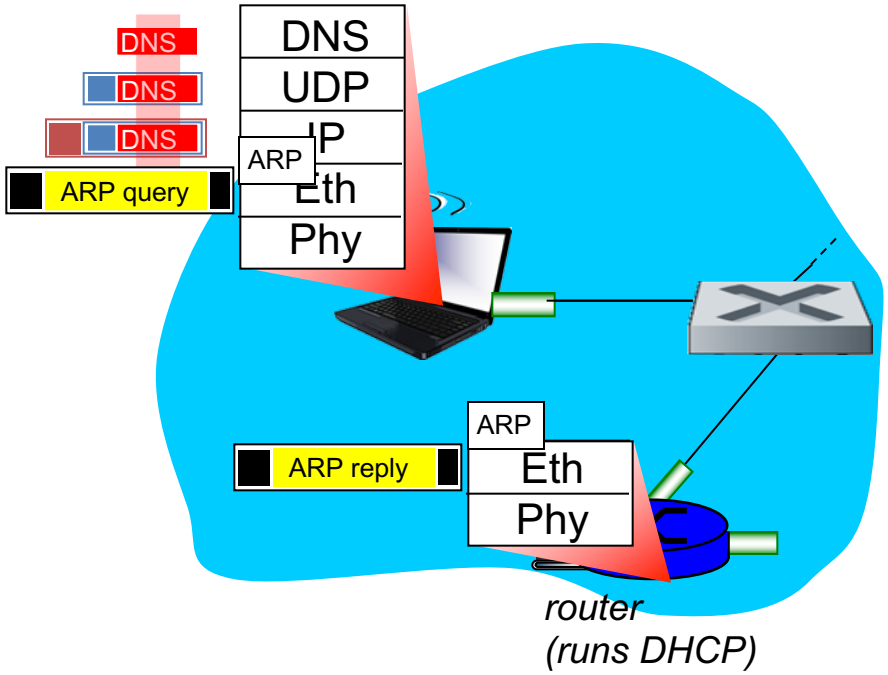
A day in the life… connecting to the Internet



- DHCP 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 forwarded (switch learning) through LAN, demultiplexing at client
- DHCP client receives DHCP ACK reply

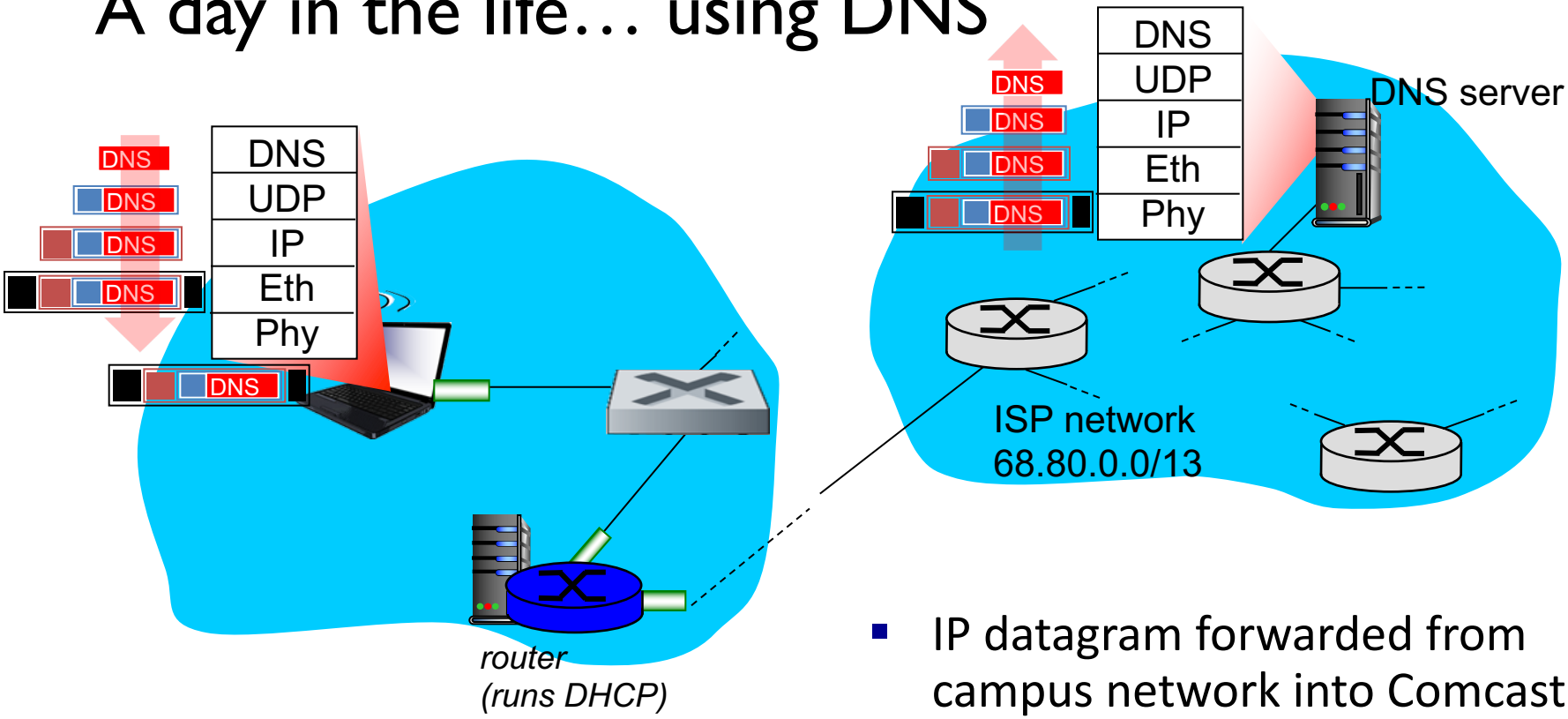
Client now has IP address, knows name & addr of DNS server, IP address of its first-hop router

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



- 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 router, need MAC address of router interface: 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

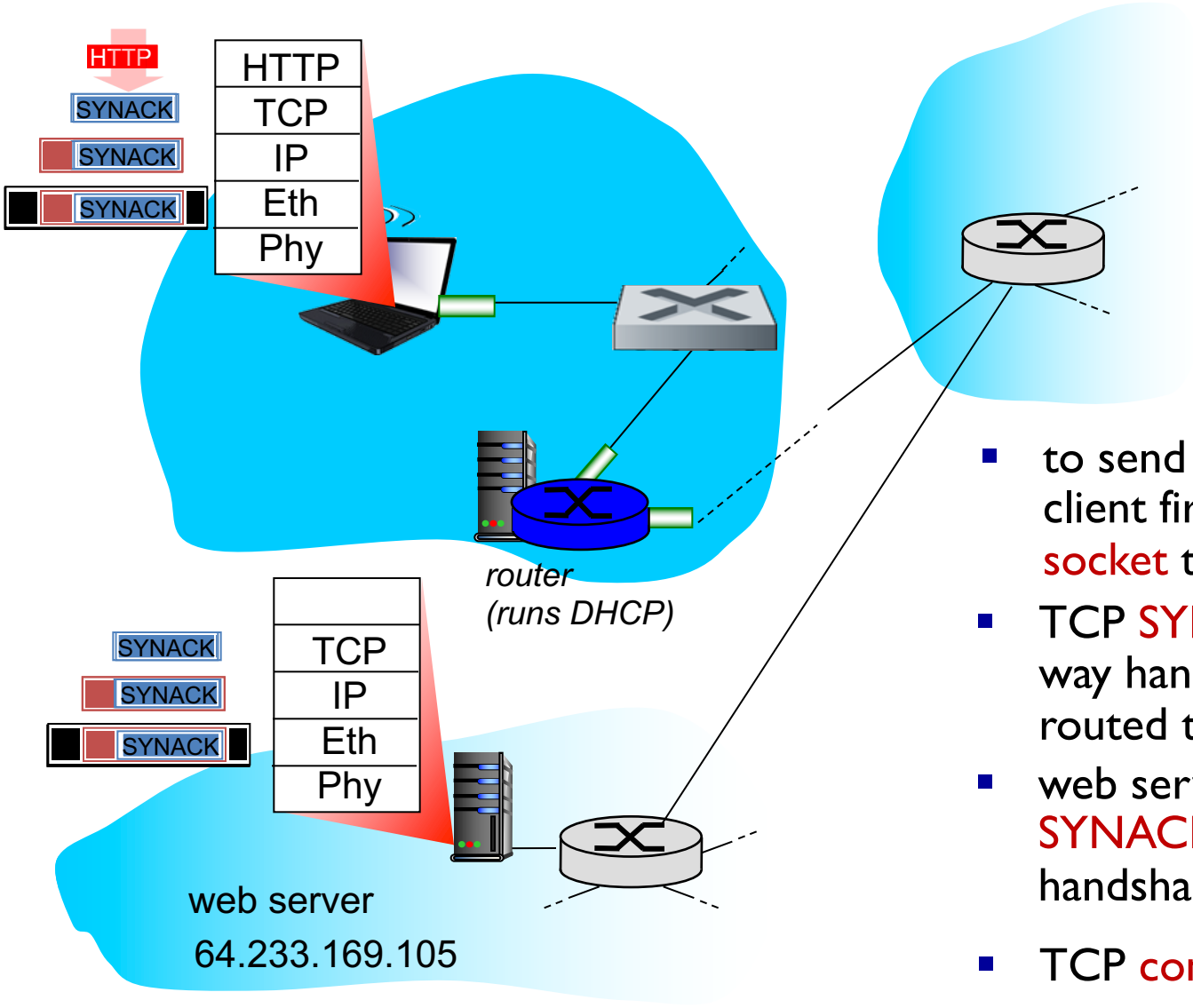
A day in the life… using DNS



- IP datagram containing DNS query forwarded via LAN switch from client to 1st hop router

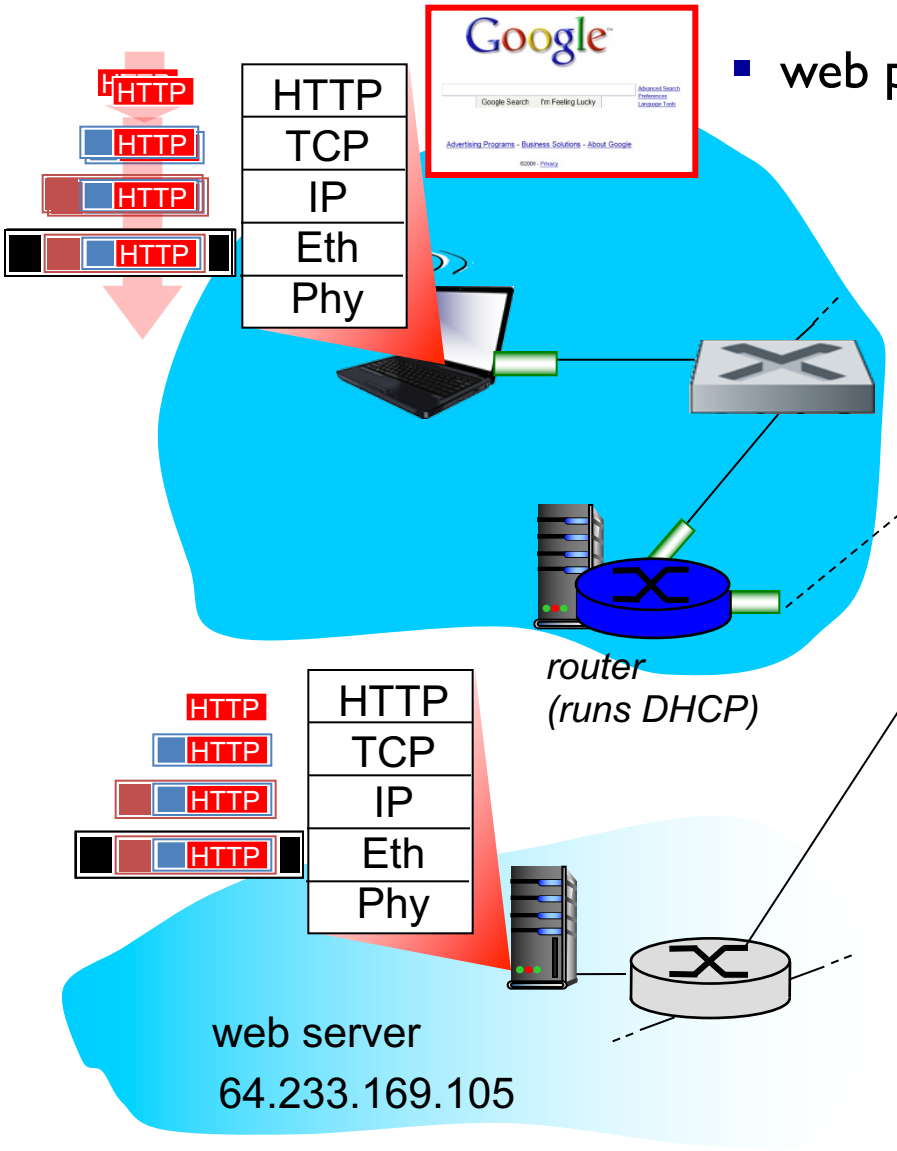
- IP datagram forwarded from campus network into Comcast network, routed (tables created by OSPF, IS-IS and/or BGP routing protocols) to DNS server
- demuxed to DNS server
- DNS server replies to client with IP address of www.google.com

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 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