

# Chapter 5

## Link Layer

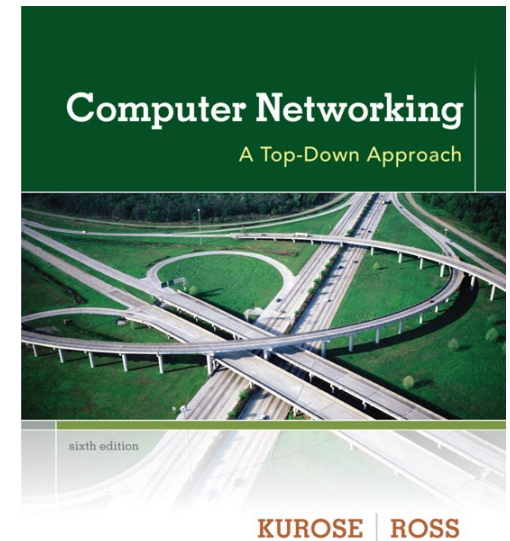
### A note on the use of these ppt slides:

We're making these slides freely available to all (faculty, students, readers). They're in PowerPoint form so you see the animations; and can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a lot of work on our part. In return for use, we only ask the following:

- ❖ If you use these slides (e.g., in a class) that you mention their source (after all, we'd like people to use our book!)
- ❖ If you post any slides on a www site, that you note that they are adapted from (or perhaps identical to) our slides, and note our copyright of this material.

Thanks and enjoy! JFK/KWR

© All material copyright 1996-2012  
J.F Kurose and K.W. Ross, All Rights Reserved



*Computer  
Networking: A Top  
Down Approach  
6<sup>th</sup> edition  
Jim Kurose, Keith Ross  
Addison-Wesley  
March 2012*

# Link layer, LANs: outline

5.1 introduction, services

5.2 error detection,  
correction

5.3 multiple access  
protocols

5.4 LANs

- addressing, ARP
- Ethernet
- switches
- VLANs

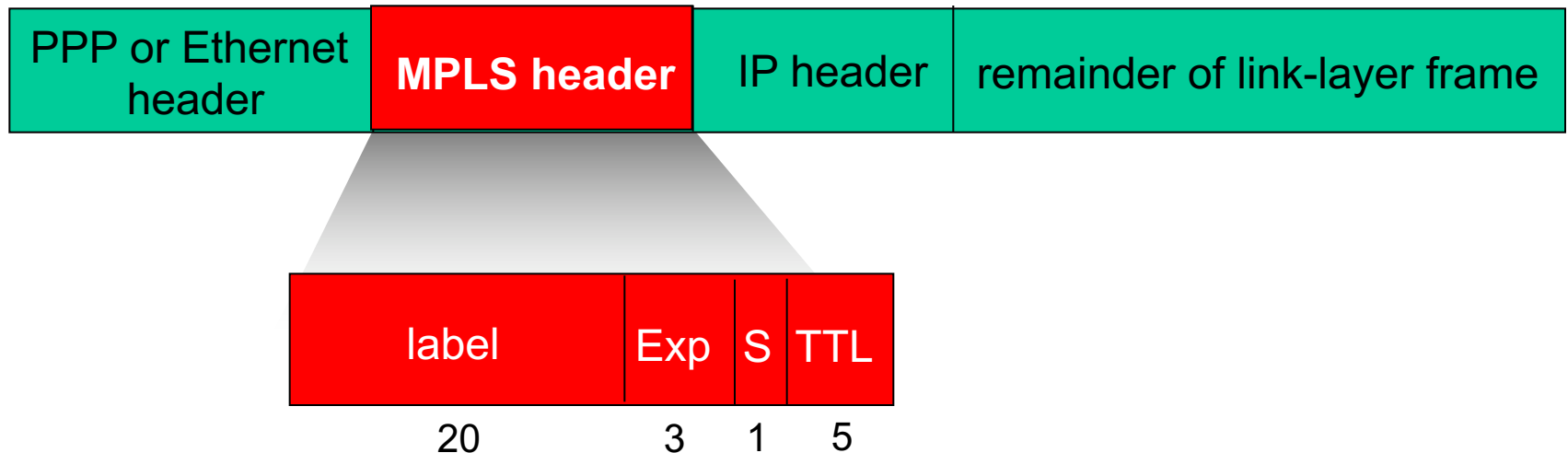
5.5 link virtualization:  
MPLS

5.6 data center  
networking

5.7 a day in the life of a  
web request

# Multiprotocol label switching (MPLS)

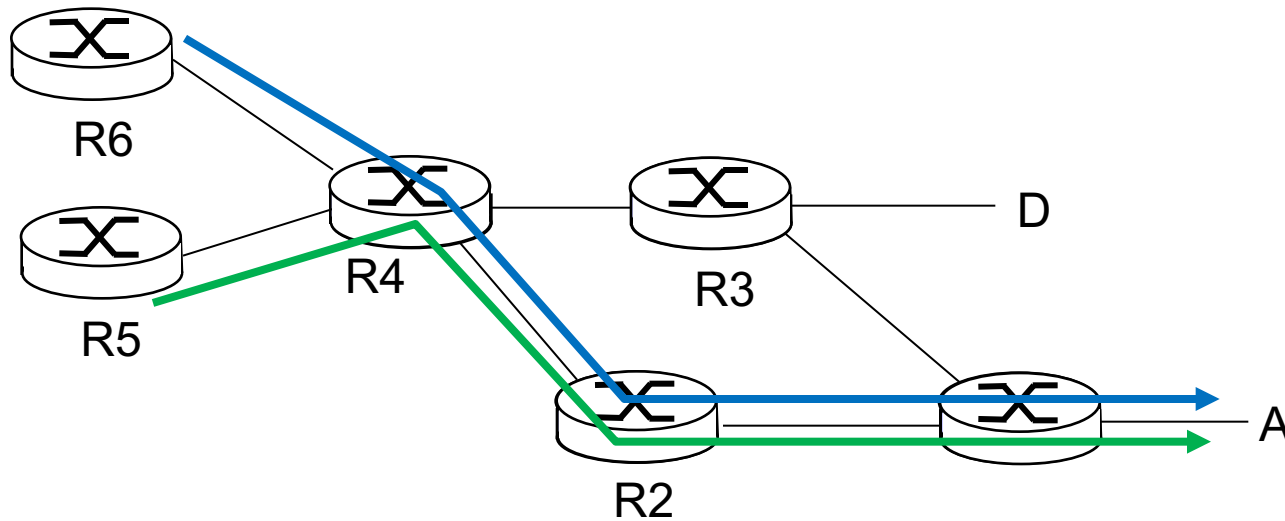
- ❖ initial goal: high-speed IP forwarding using fixed length label (instead of IP address)
  - fast lookup using fixed length identifier (rather than shortest prefix matching)
  - borrowing ideas from Virtual Circuit (VC) approach
  - but IP datagram still keeps IP address!



# MPLS capable routers

- ❖ a.k.a. label-switched router
- ❖ forward packets to outgoing interface based only on label value (*don't inspect IP address*)
  - MPLS forwarding table distinct from IP forwarding tables
- ❖ *flexibility*: MPLS forwarding decisions can *differ* from those of IP
  - use destination *and* source addresses to route flows to same destination differently (traffic engineering)
  - re-route flows quickly if link fails: pre-computed backup paths (useful for VoIP)

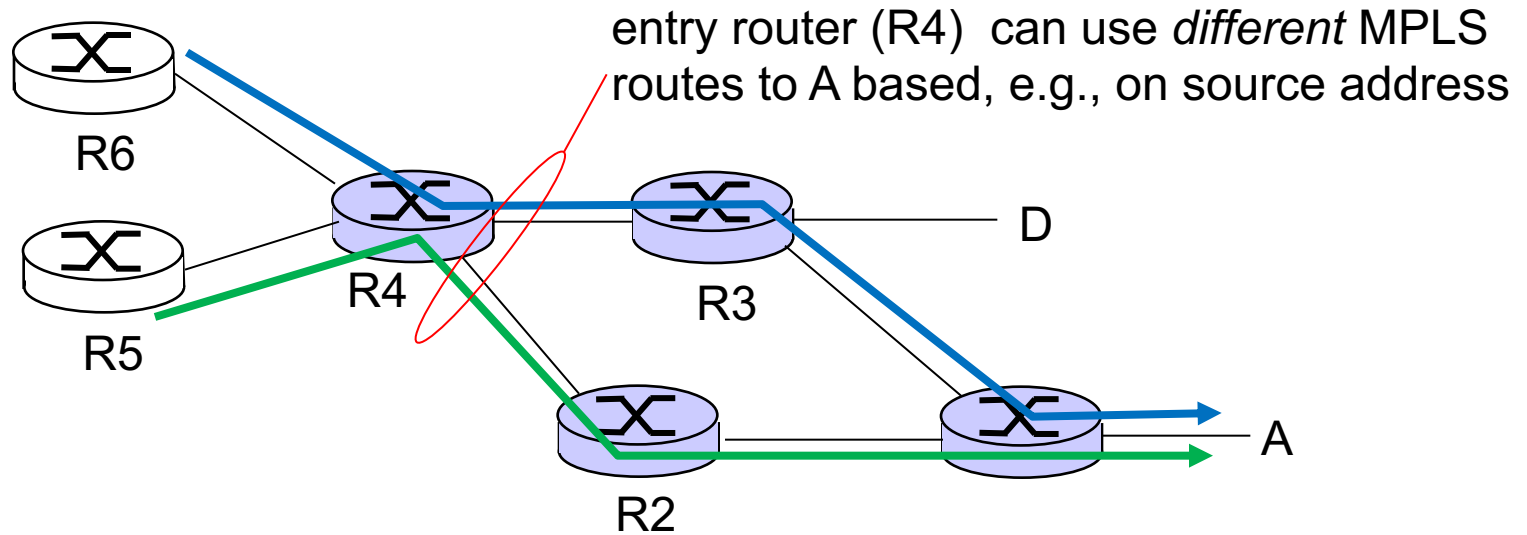
# MPLS versus IP paths



❖ *IP routing: path to destination determined by destination address alone*



# MPLS versus IP paths



❖ **IP routing:** path to destination determined by destination address alone



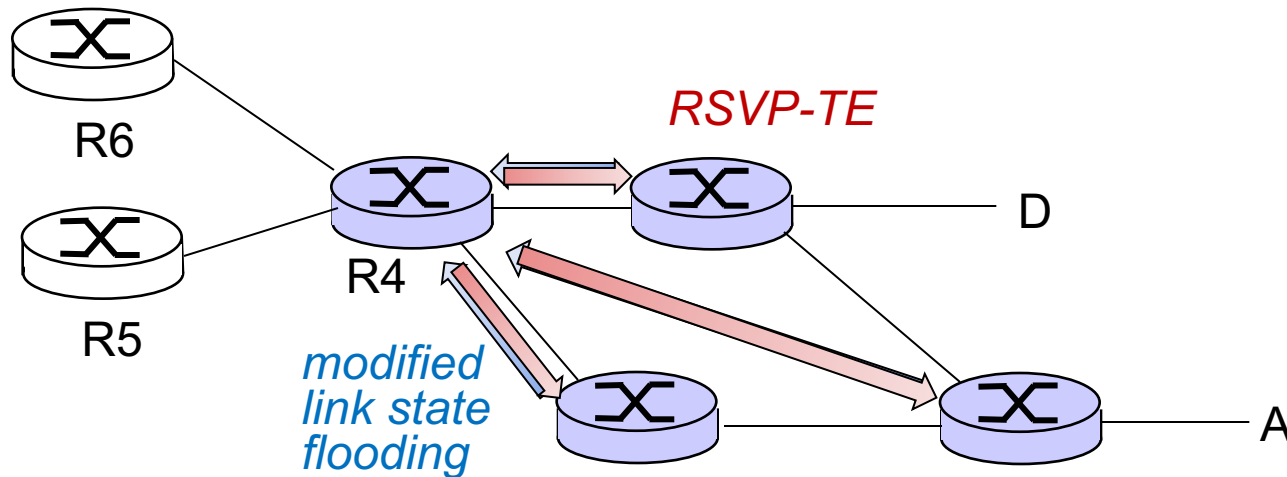
❖ **MPLS routing:** path to destination can be based on source *and* dest. address



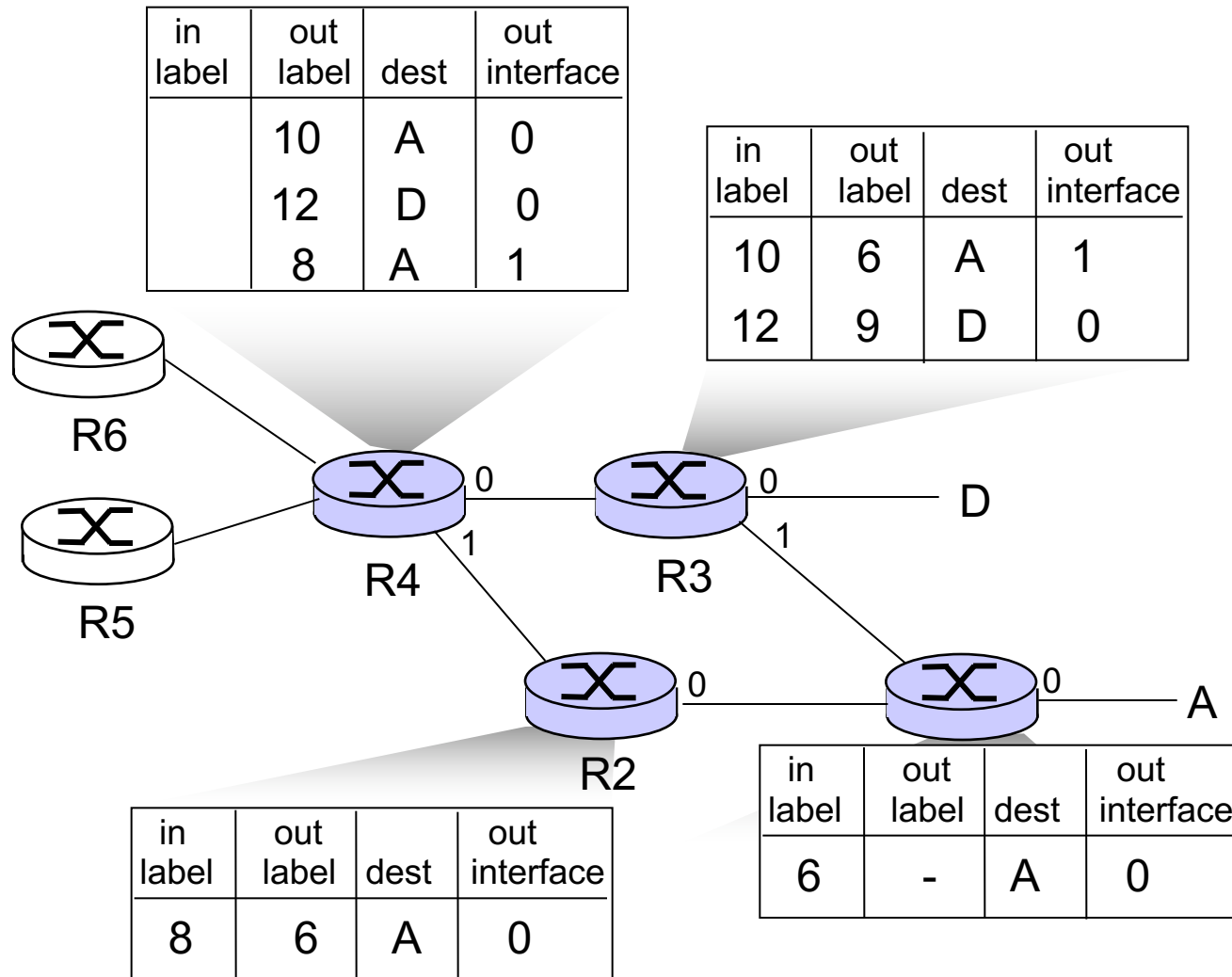
- **fast reroute:** precompute backup routes in case of link failure

# MPLS signaling

- ❖ modify OSPF, IS-IS link-state flooding protocols to carry info used by MPLS routing,
  - e.g., link bandwidth, amount of “reserved” link bandwidth
- ❖ entry MPLS router uses *RSVP-TE signaling protocol* to set up MPLS forwarding at downstream routers



# MPLS forwarding tables





# Link layer, LANs: outline

5.1 introduction, services

5.2 error detection,  
correction

5.3 multiple access  
protocols

5.4 LANs

- addressing, ARP
- Ethernet
- switches
- VLANs

5.5 link virtualization:  
MPLS

5.6 data center  
networking

5.7 a day in the life of a  
web request

# Data center networks

- ❖ 10's to 100's of thousands of hosts, often closely coupled, in close proximity:
  - e-business (e.g. Amazon)
  - content-servers (e.g., YouTube, Akamai, Apple, Microsoft)
  - search engines, data mining (e.g., Google)
- ❖ challenges:
  - multiple applications, each serving massive numbers of clients
  - managing/balancing load, avoiding processing, networking, data bottlenecks

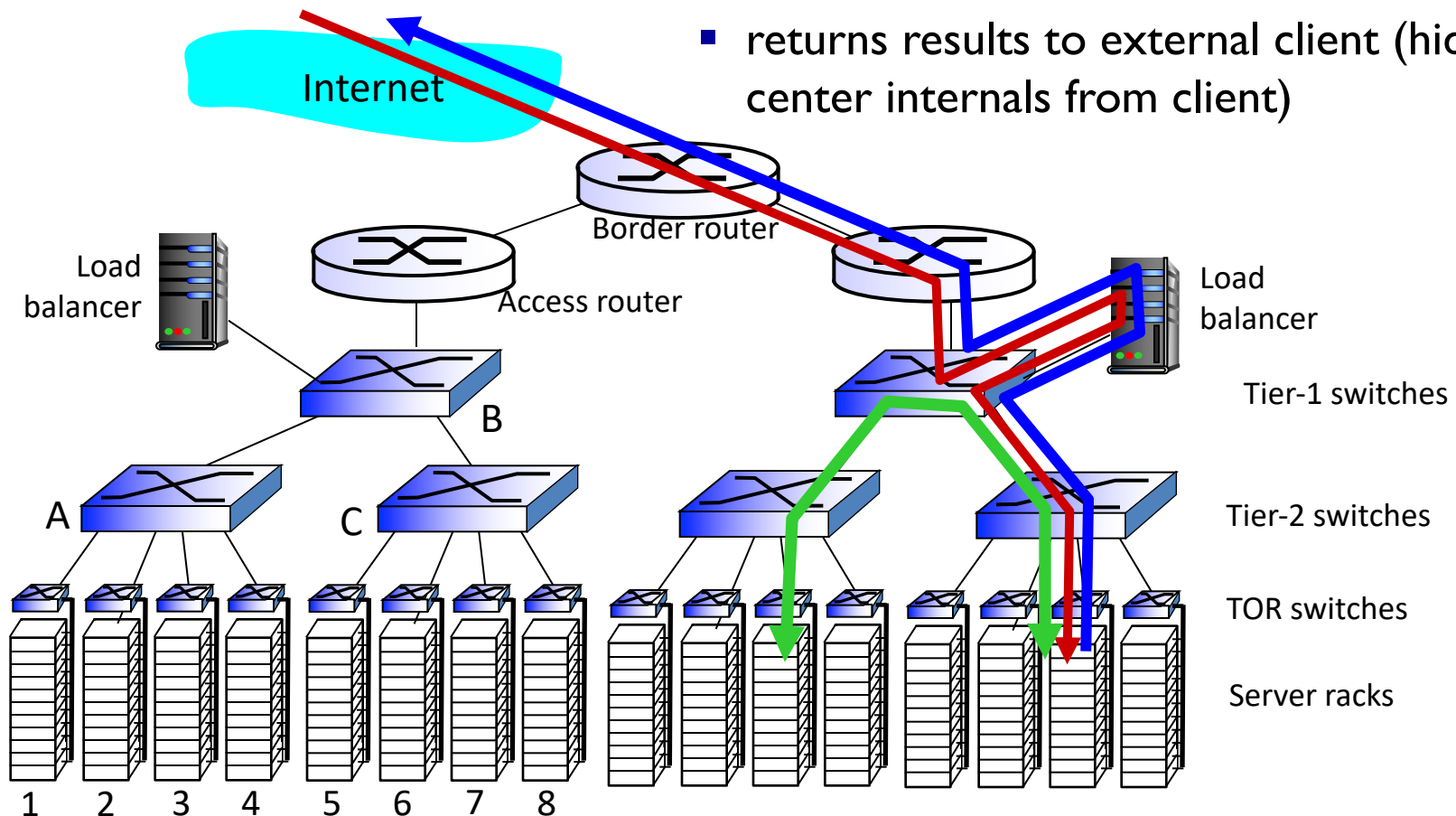


Inside a 40-ft Microsoft container,  
Chicago data center

# Data center networks

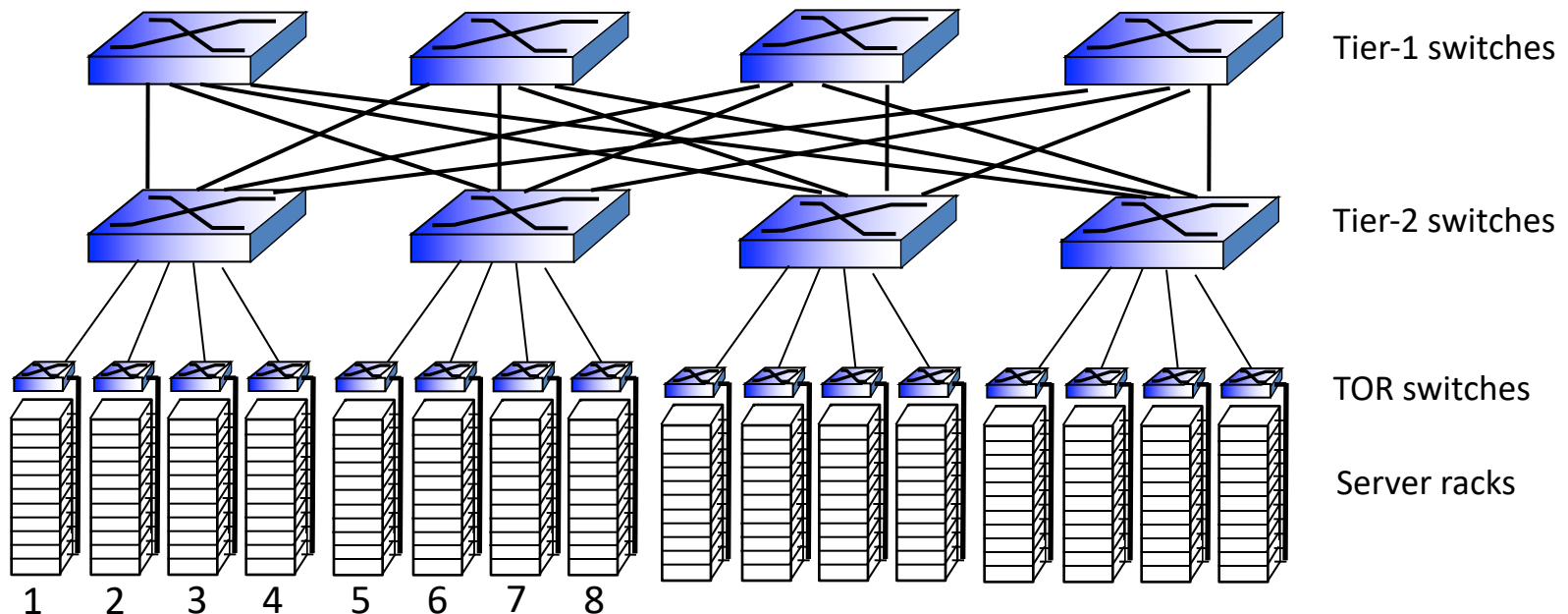
*load balancer: application-layer routing*

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



# Data center networks

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



# Link layer, LANs: outline

5.1 introduction, services

5.2 error detection,  
correction

5.3 multiple access  
protocols

5.4 LANs

- addressing, ARP
- Ethernet
- switches
- VLANs

5.5 link virtualization:  
MPLS

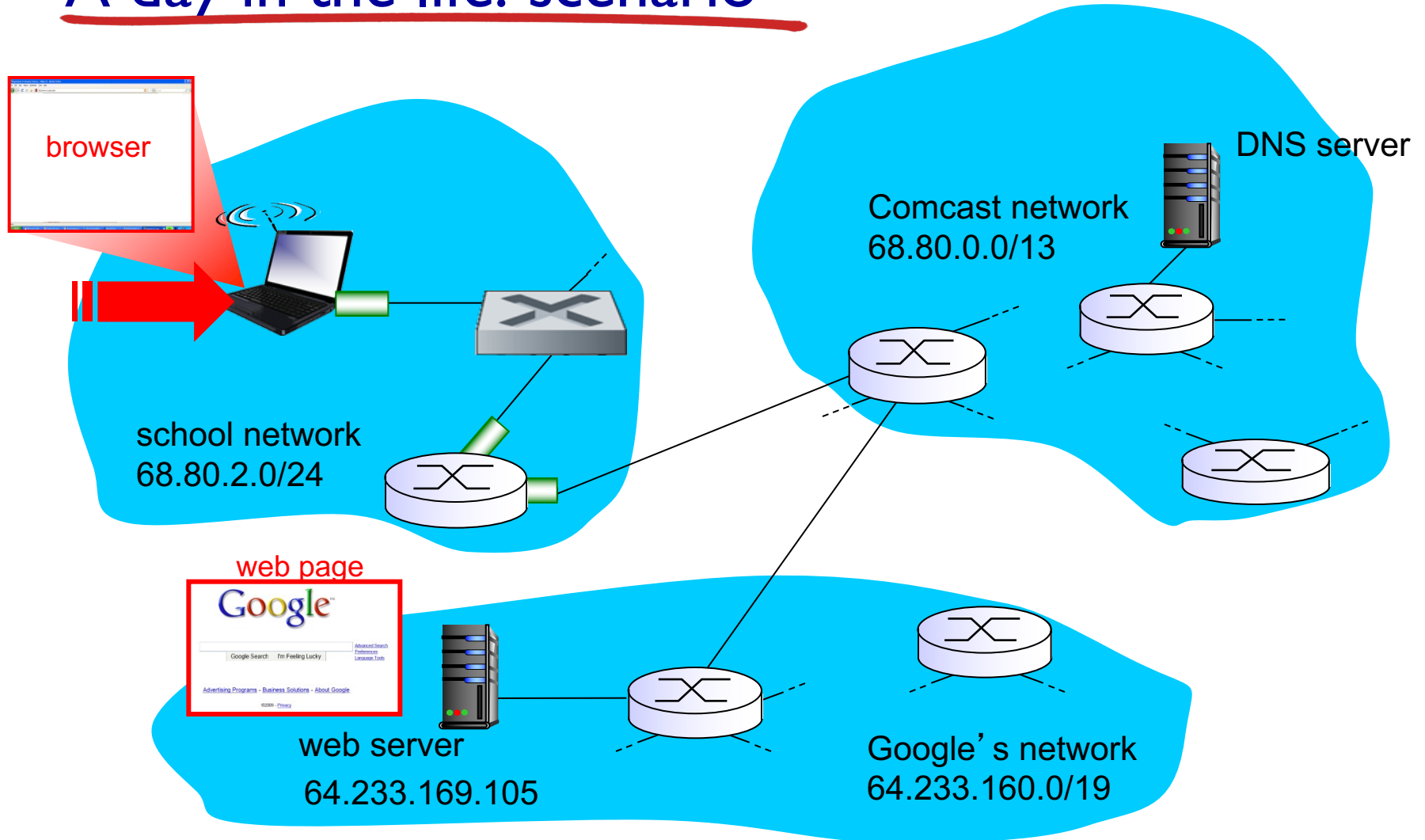
5.6 data center  
networking

5.7 a day in the life of a  
web request

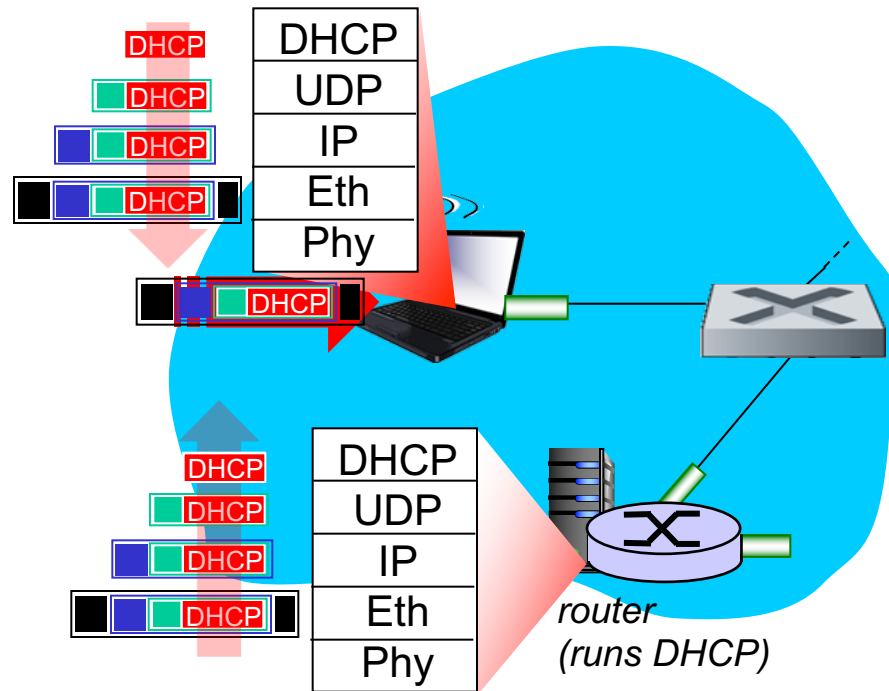
## Synthesis: a day in the life of a web request

- ❖ 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](http://www.google.com)

# A day in the life: scenario



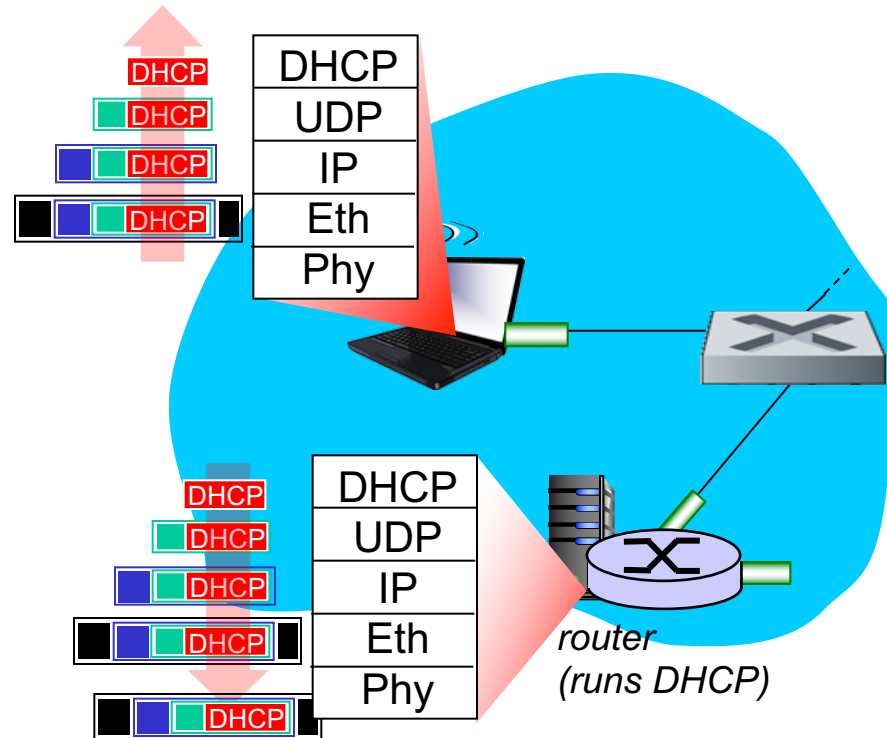
# A day in the life... connecting to the Internet



- ❖ 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: FFFFFFFFFFFFFFFF) on LAN, received at router running *DHCP* server
- ❖ Ethernet *demuxed* to IP demuxed, UDP demuxed to DHCP



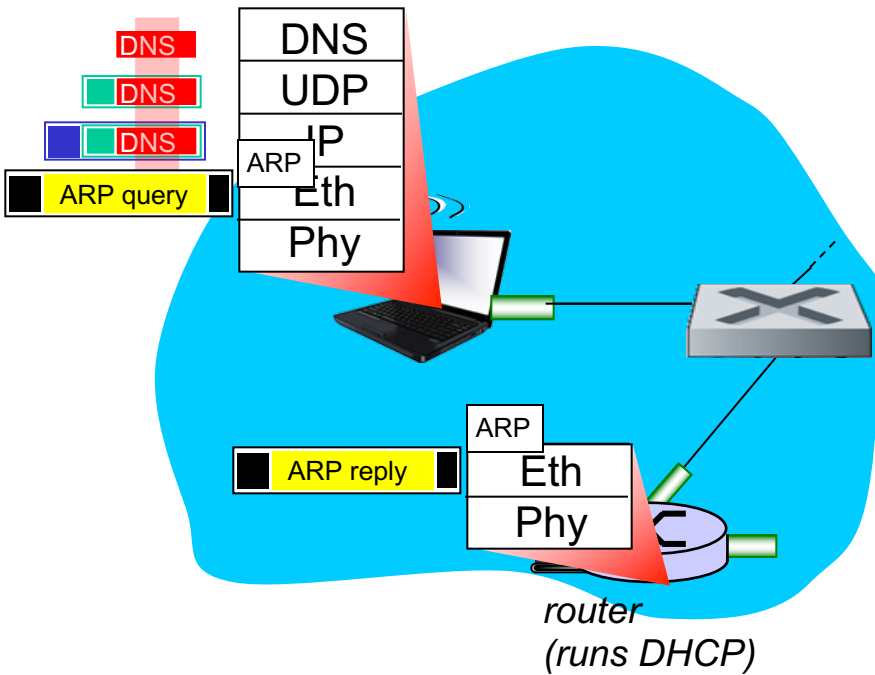
# A day in the life... connecting to the Internet



- ❖ 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

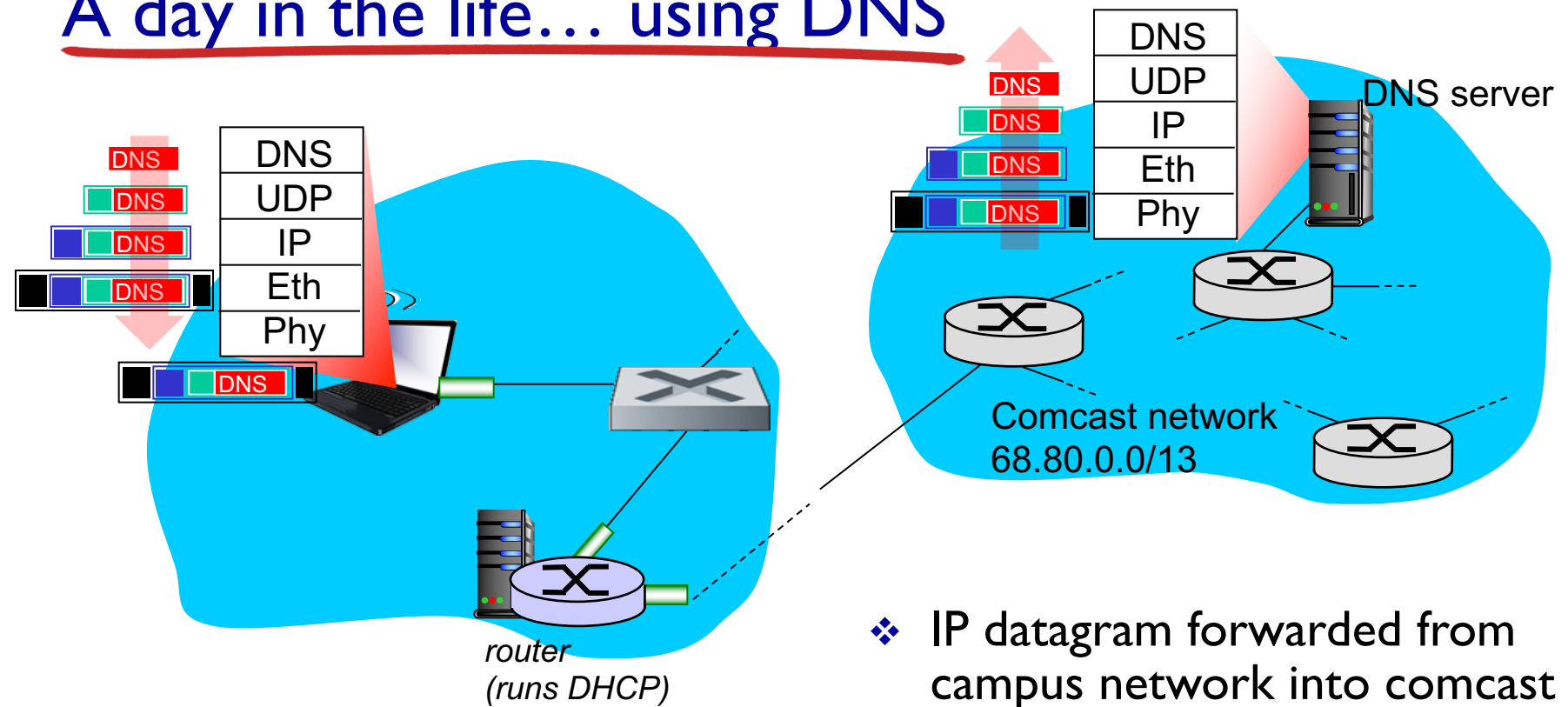
*Client now has IP address, knows name & addr of DNS server, IP address of its first-hop router*

# 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

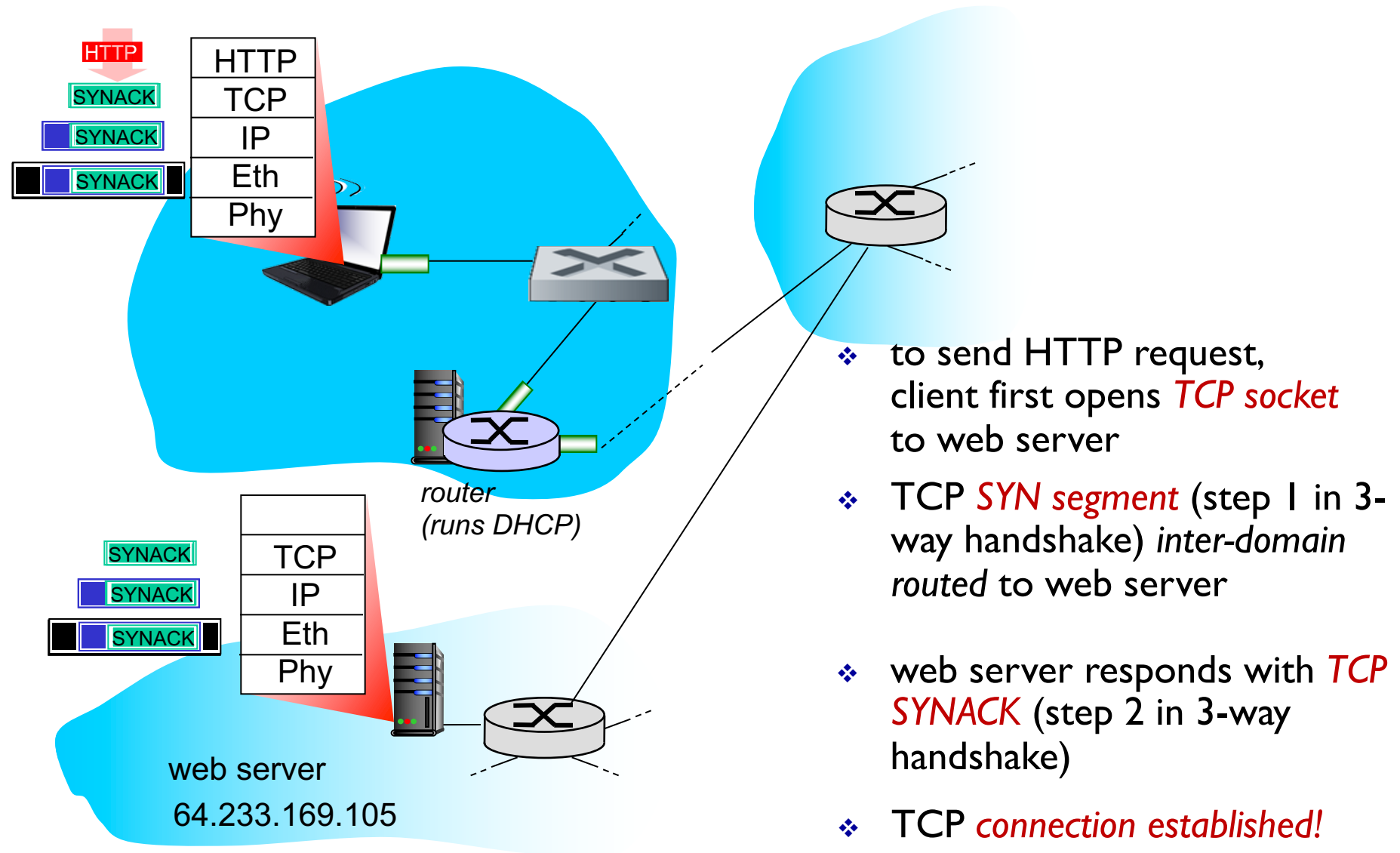
# A day in the life... using DNS



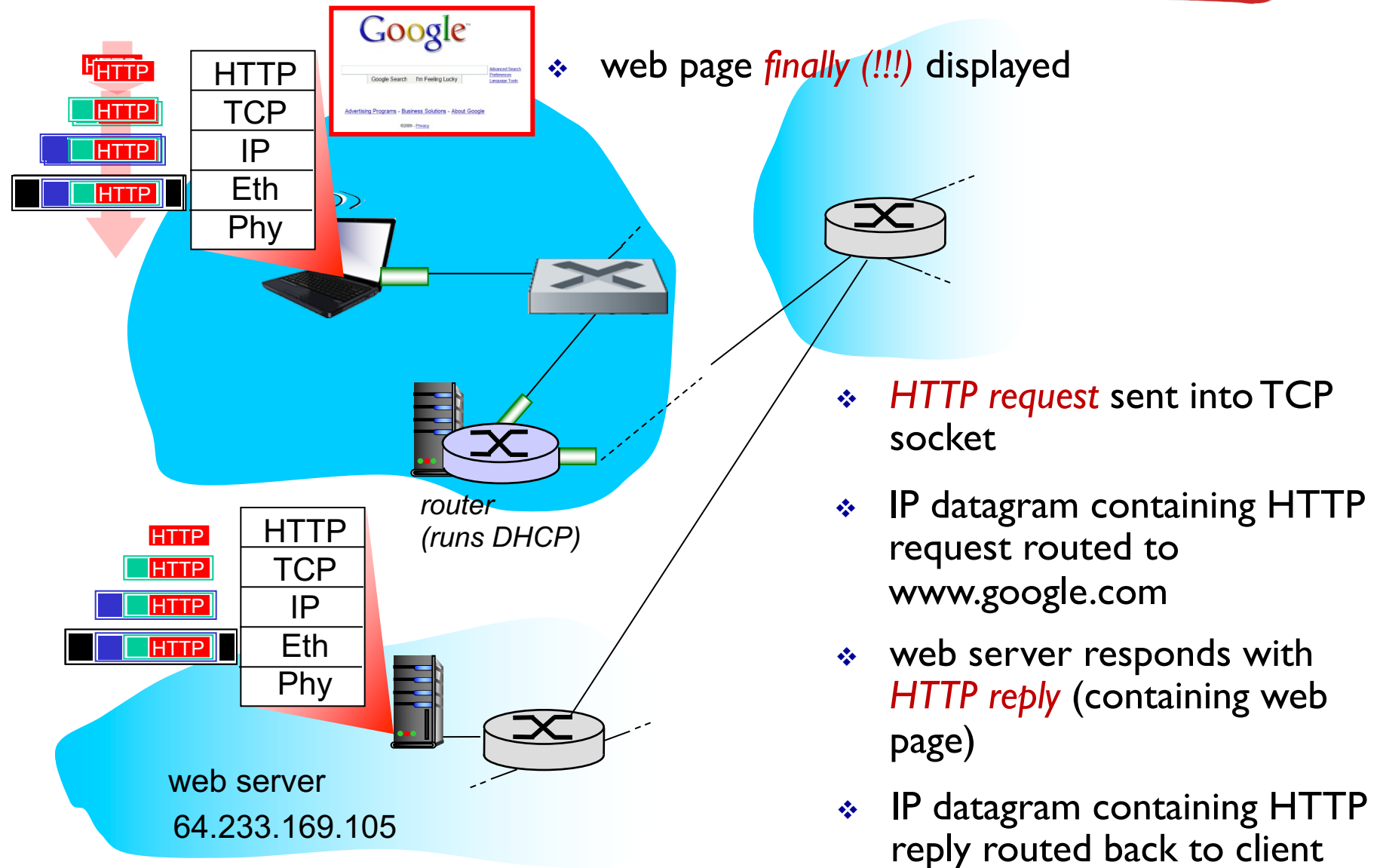
- ❖ IP datagram containing DNS query forwarded via LAN switch from client to 1<sup>st</sup> hop router

- ❖ IP datagram forwarded from 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](http://www.google.com)

# A day in the life...TCP connection carrying HTTP



# A day in the life... HTTP request/reply



# Chapter 5: Summary

- ❖ principles behind data link layer services:
  - error detection, correction
  - sharing a broadcast channel: multiple access
  - link layer addressing
- ❖ instantiation and implementation of various link layer technologies
  - Ethernet
  - switched LANS, VLANs
  - virtualized networks as a link layer: MPLS
- ❖ synthesis: a day in the life of a web request

# Chapter 5: let's take a breath

- ❖ journey down protocol stack *complete* (except PHY)
- ❖ solid understanding of networking principles, practice
- ❖ ..... could stop here .... but *lots* of interesting topics!
  - wireless
  - multimedia
  - security
  - network management