Course on Computer Communication and Networks

Lecture 16
Synthesis, Summary/flashback and Projection
(related topics – continuation of study)

EDA344/DIT 423, CTH/GU

Based on the book Computer Networking: A Top Down Approach, Jim Kurose, Keith Ross, Addison-Wesley.

Important for the exam

When/where: wednesday March 14, 14.00-18.00, SB-building

You may have with you:

- English-X dictionary
- no calculators, PDAs, etc (if/where numbers matter, do rounding)

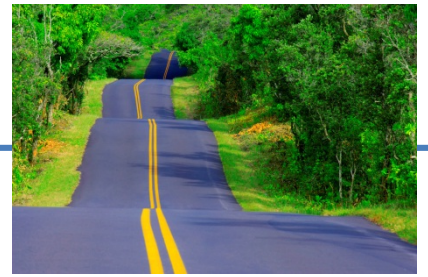
Grading

- 30-40, 41-50, 51-60 (out of 60)= 3, 4, 5 (CTH)
- 30-44, 45-60 (out of 60) = G, VG (GU)

To think during summary-study

Have overview, critical eye; explain; ask yourselves: why is this so? / how does it work (or not work)?

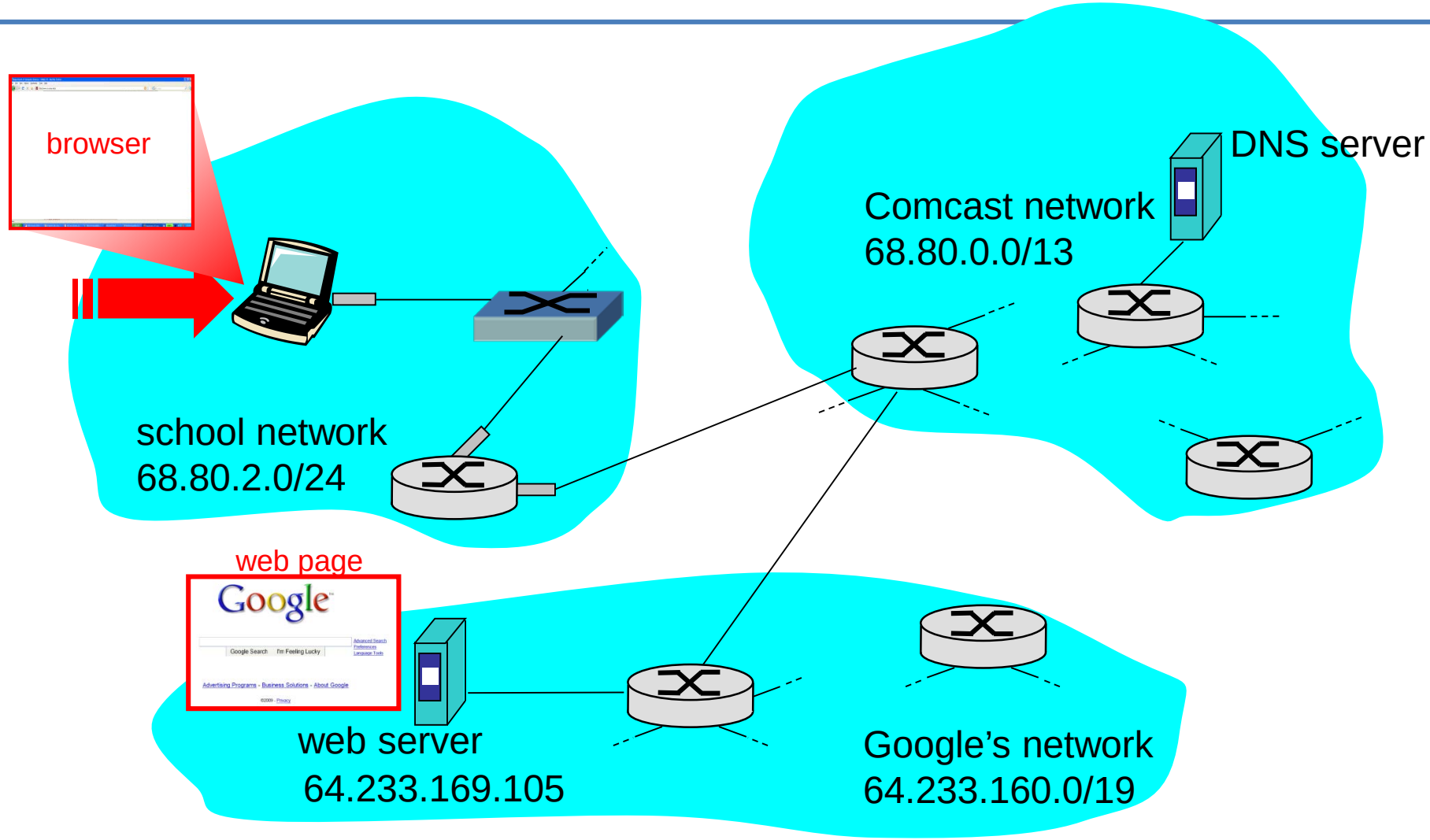
Roadmap



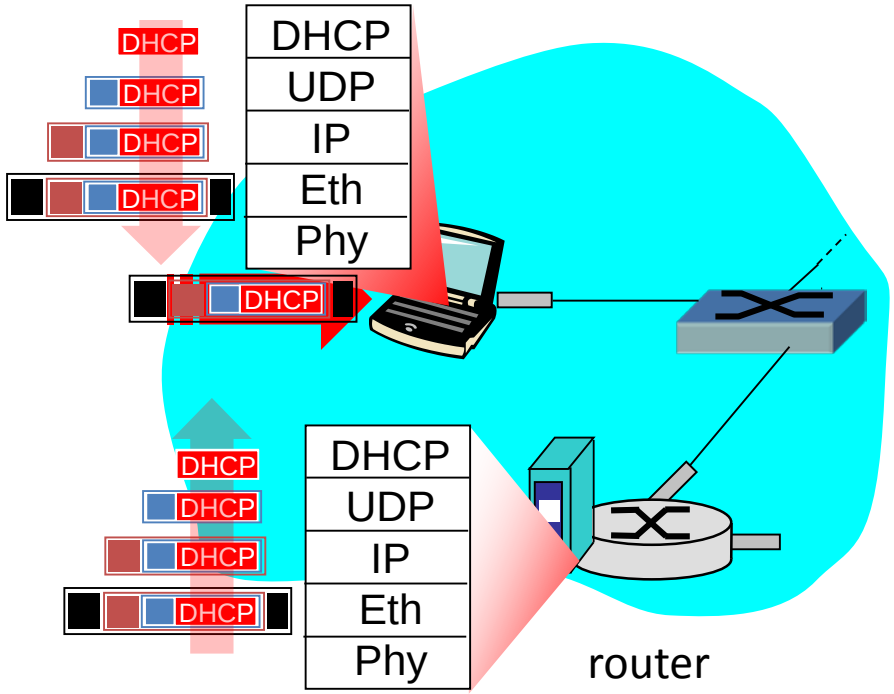
Synthesis:

- Putting lots-of-what-we-learned together: **a day in the life of a web request**
  - *goal:* identify, review protocols (at all layers) involved in seemingly simple scenario:
  - *Scenario* requesting www page*:* student attaches laptop to campus network, requests/receives www.google.com
- **…….**

A day in the life …. : scenario



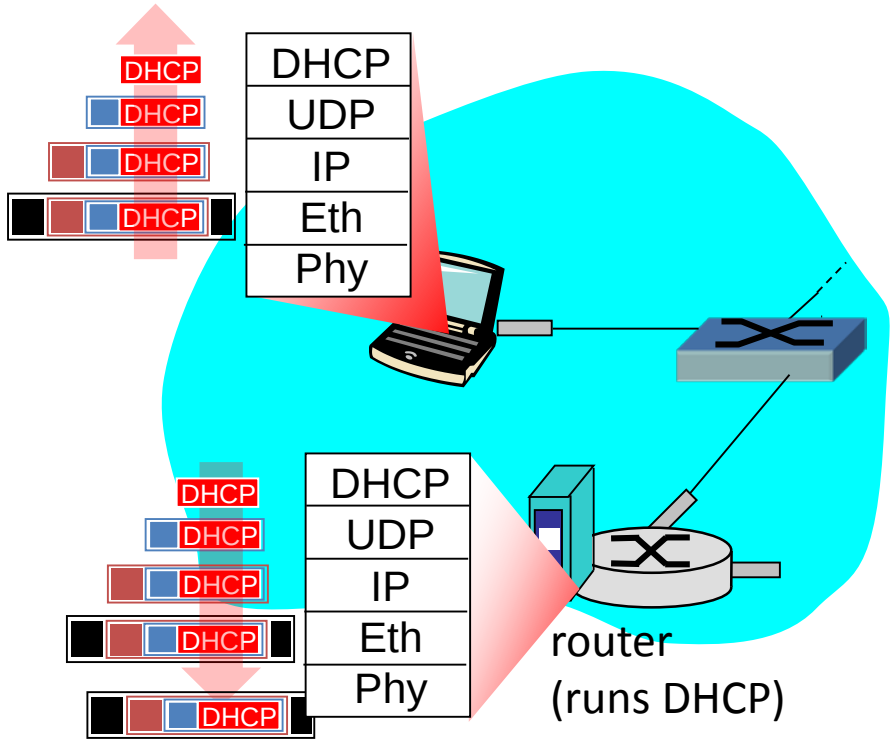
A day in the life… connecting to the Internet



connecting laptop needs to get its own IP address: use ***DHCP***

- DHCP request encapsulated in UDP, encapsulated in IP, encapsulated in Ethernet
- Ethernet frame broadcast (dest: FFFFFFFFFFFF) on LAN, received at router running DHCP server
- Ethernet demux'ed to IP demux'ed to UDP demux'ed to DHCP

A day in the life… connecting to the Internet

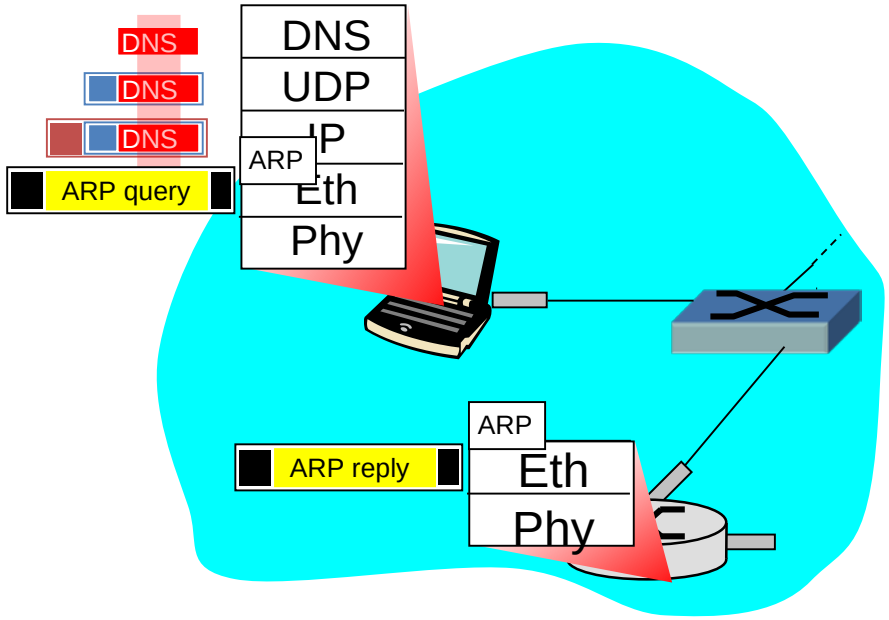


DHCP server formulates ***DHCP ACK*** containing client's IP address (**and also IP address of first-hop router for client, name & IP address of DNS server**)

- 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

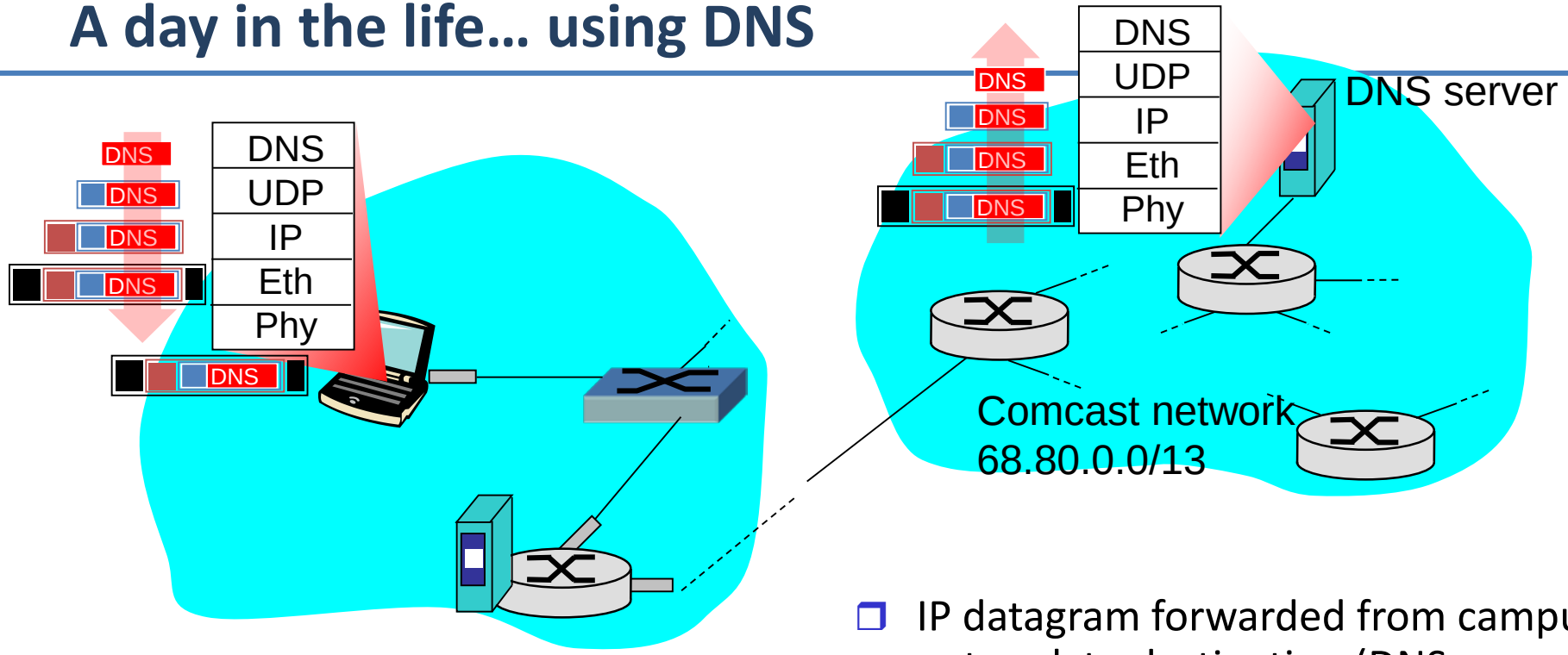
A day in the life… ARP (before DNS, before HTTP)



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

- DNS query created, encapsulated in UDP, encapsulated in IP, encasulated in Eth. In order 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

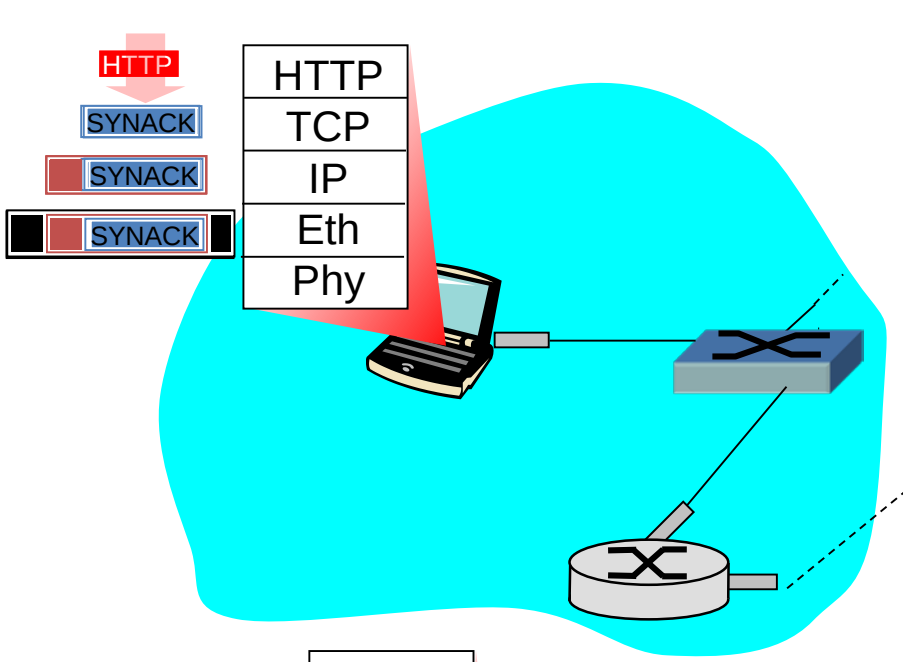
A day in the life… using DNS

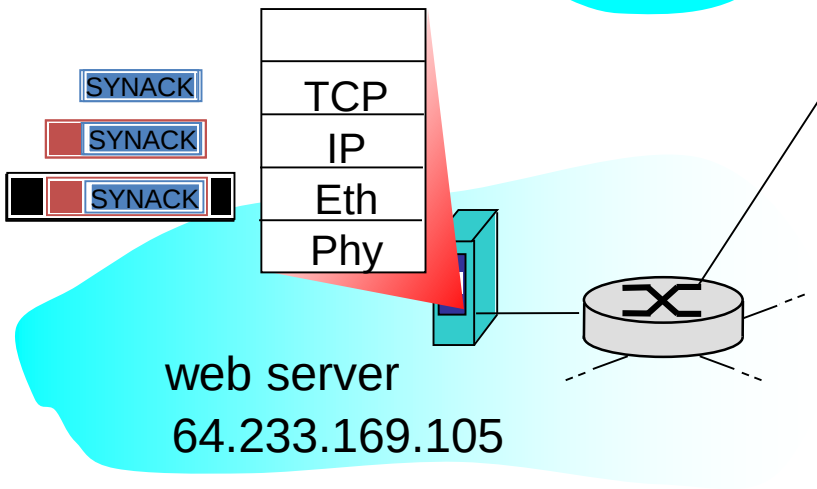


- IP datagram containing DNS query forwarded via LAN switch from client to 1st hop router

- IP datagram forwarded from campus network to destination (DNS-server) network, routed (tables created by OSPF and BGP routing protocols) to DNS server
- demux'ed to DNS server
- DNS server replies to client with IP address of www.google.com

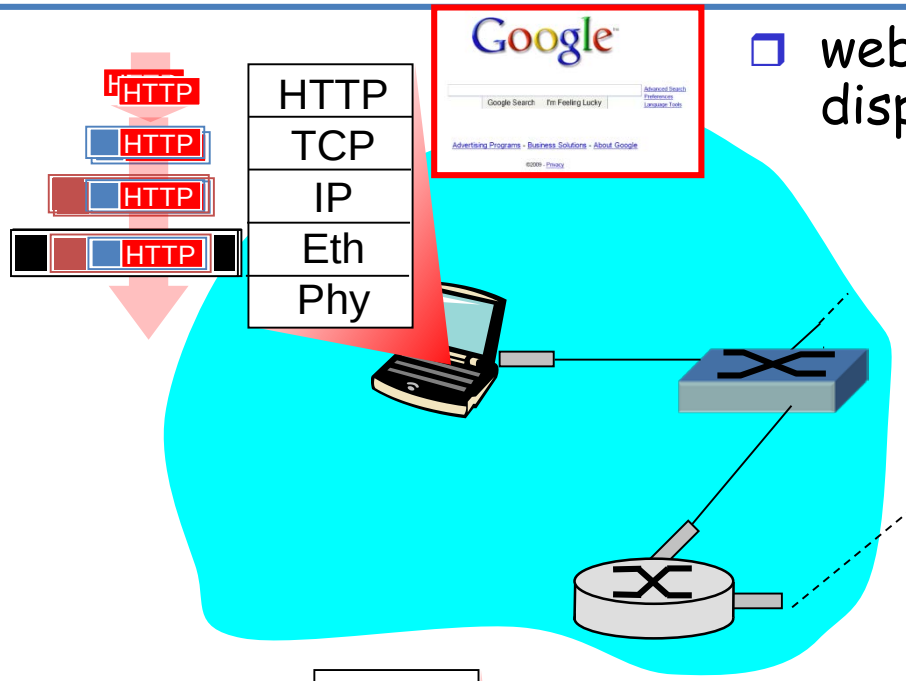
A day in the life… TCP connection carrying HTTP

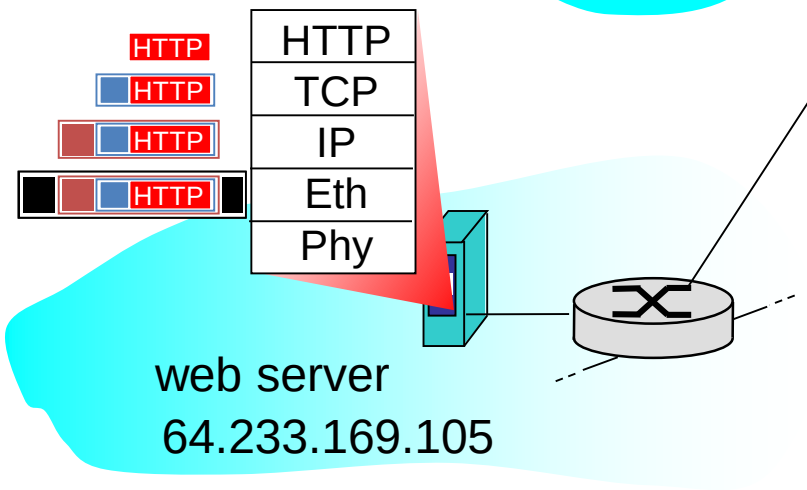




- to send HTTP request, client first opens TCP socket to web server
- TCP SYN segment (step 1 in 3-way handshake) inter-domain routed to web server
- web server responds with TCP SYNACK
- TCP connection established!

A day in the life… HTTP request/reply

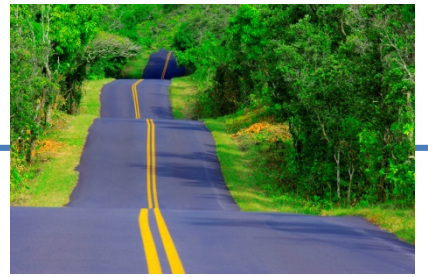




- web page finally (!!!) displayed

- 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 datgram containing HTTP reply routed back to client

Roadmap



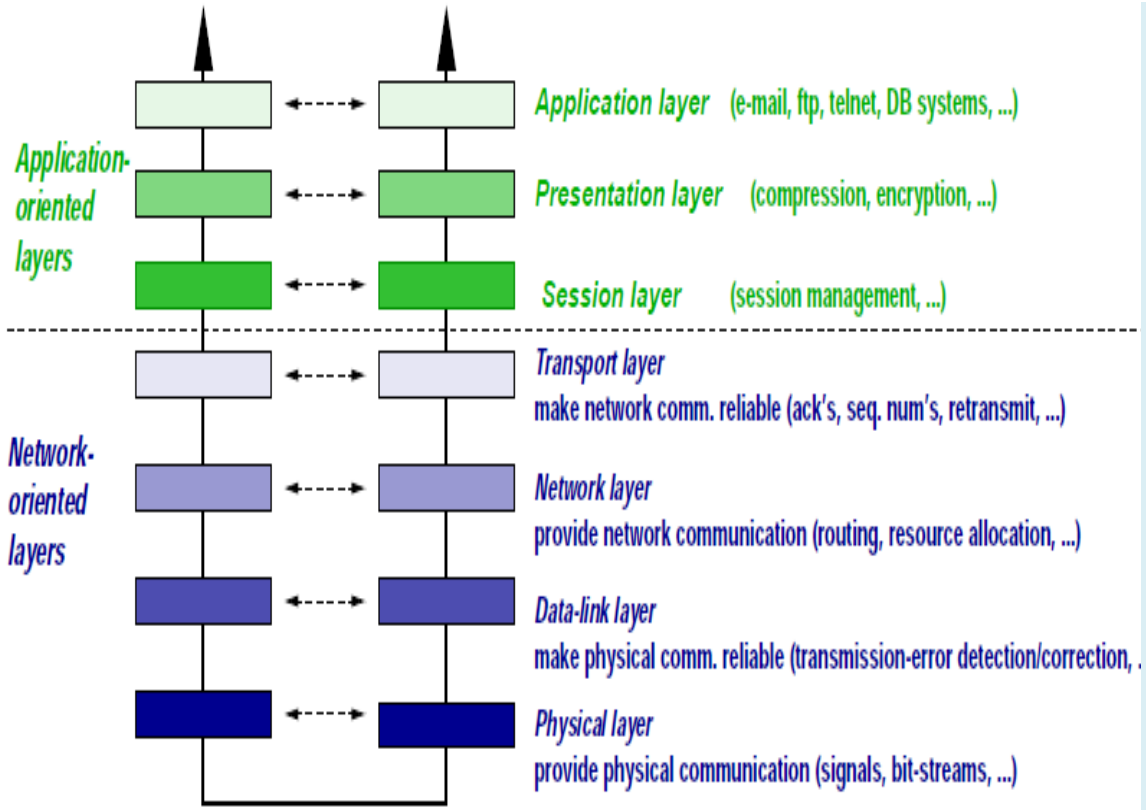
Synthesis: a day in the life of a web request

- Putting lots-of-what-we-learned together: synthesis!
  - *goal:* identify, review protocols (at all layers) involved in seemingly simple scenario: requesting www page
  - *scenario:* student attaches laptop to campus network, requests/receives www.google.com

Highlights

…..

Principles, Organisation

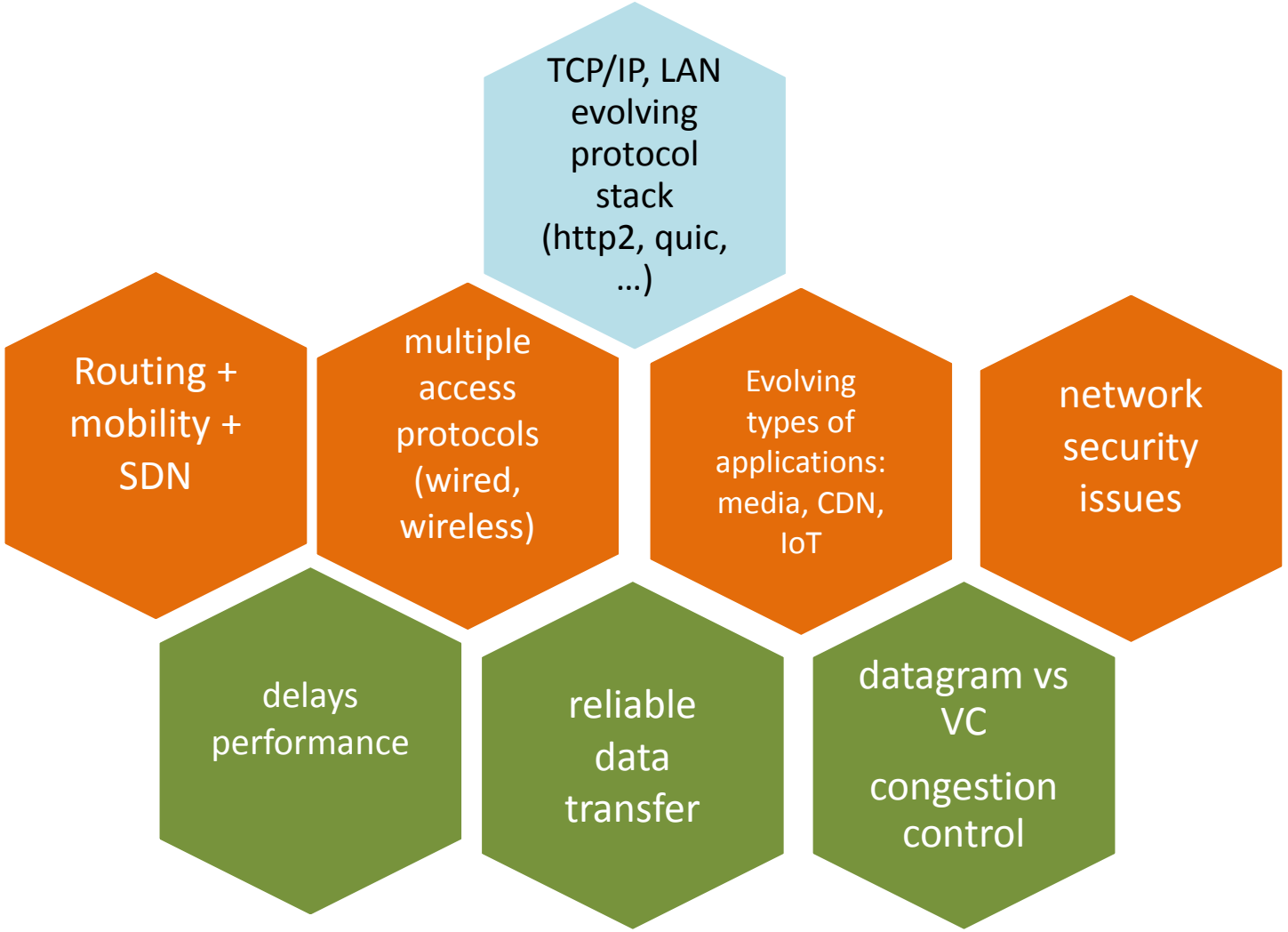


Network Problems

- Continuously evolving applications, mobility, performance, security, …, …
- serving different types of traffic,
- connecting transparently different networks,
- routing, congestion control,
- access to shared (broadcast) transmission medium
- producer-consumer problems, flow and error control

Layering : principle, why

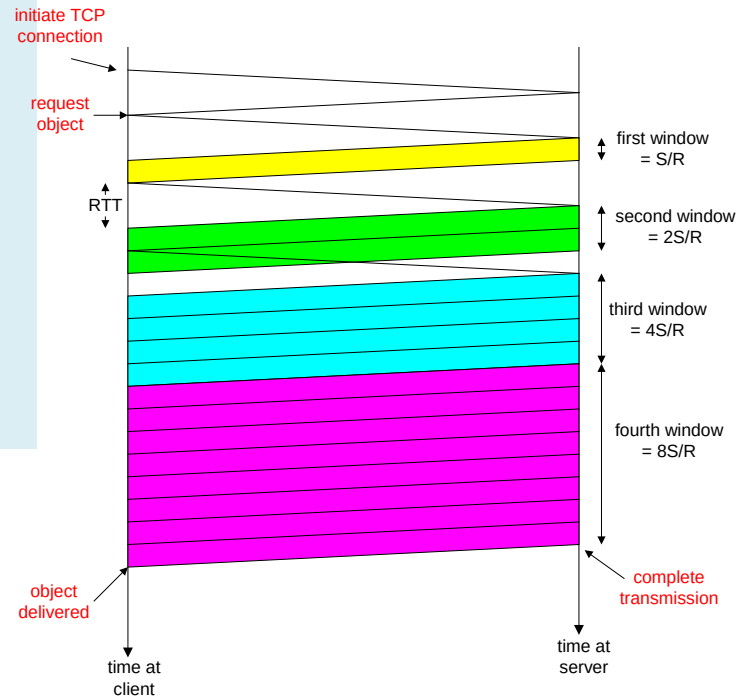
Highlights

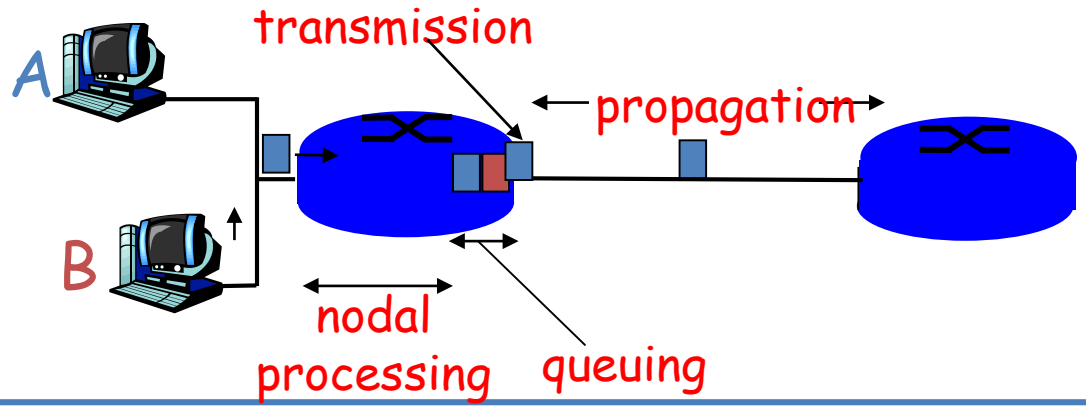

TCP/IP, LAN evolving protocol stack (http2, quic, …)
Routing + mobility + SDN
multiple access protocols (wired, wireless)
Evolving types of applications: media, CDN, IoT
network security issues
delays performance
reliable data transfer
datagram vs VC
congestion control

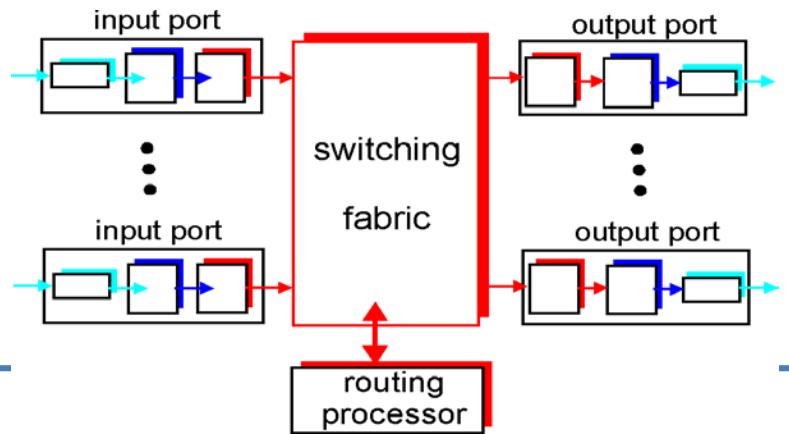
Types of delay; performance

delays performance

- Propagation, transmission, queueing, processing
- Throughput (effective bandwidth), utilization -- efficiency
- Packet-switching: impact of store&forward, pipelines, space-time diagrams
- Sliding windows performance
- Relation between delays-losses







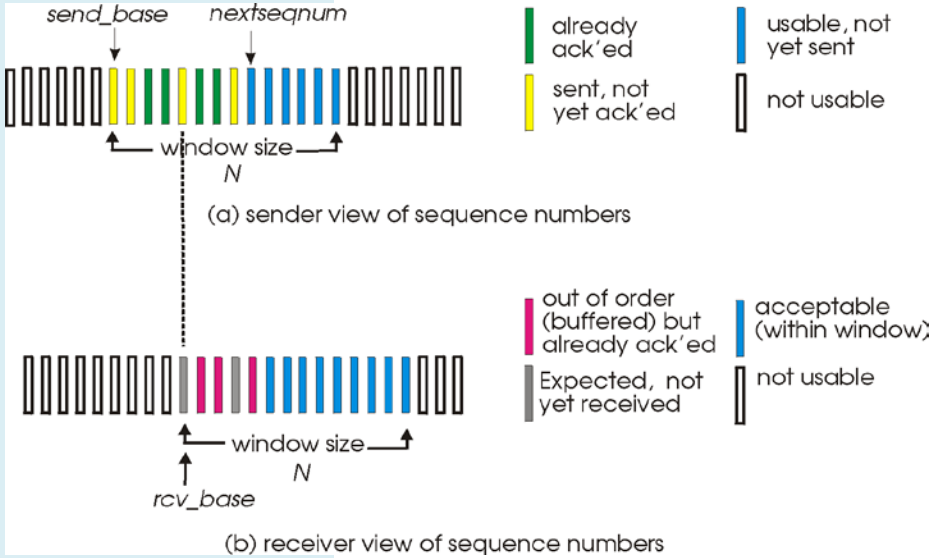
Reliable data transfer

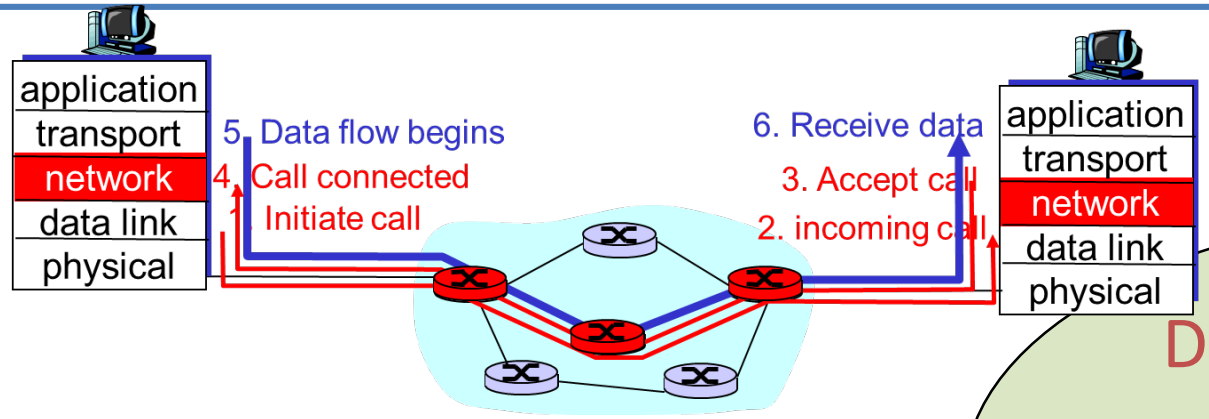
reliable data transfer

Guaranteed, in-order, correct delivery:

- stop&wait
- sliding windows
- sequence numbers
- window sizes
- dynamic windows (TCP)
- flow control

- **Error detection**: checksums
- **Error control**: go-back-n, selective repeat, FEC methods





Datagram vs VC end-to-end comm.

Conceptual differences

Congestion Control

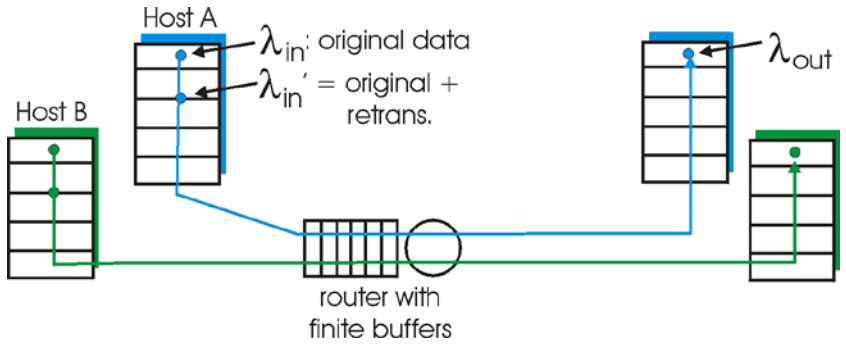
RT traffic/streaming

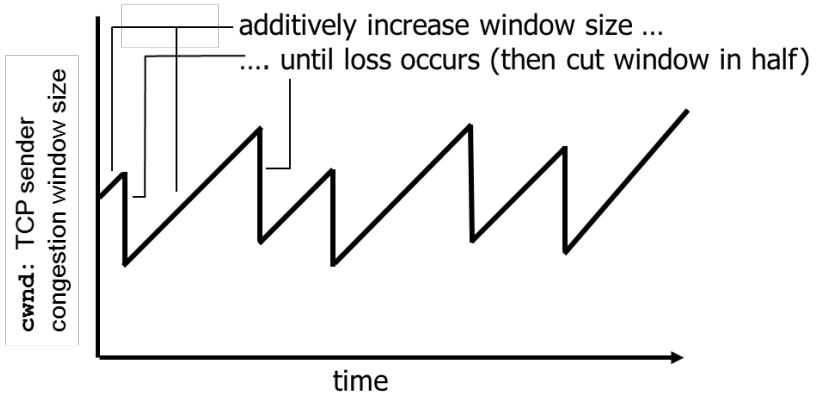
Congestion control (CC)

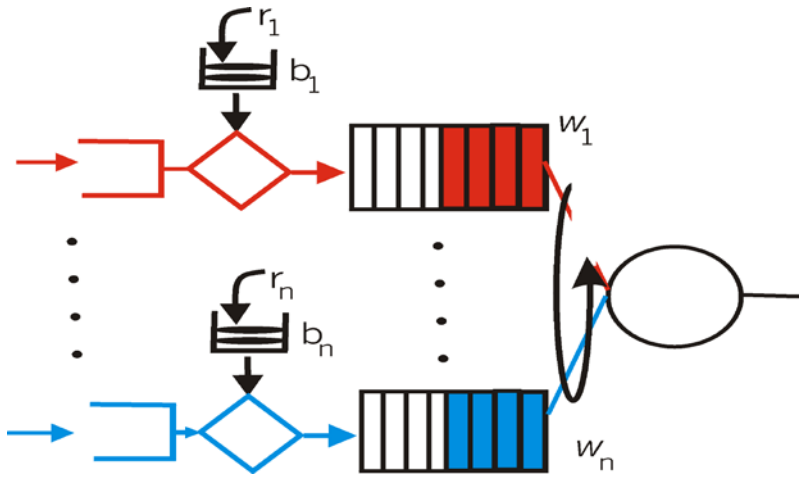
datagram vs VC

congestion control

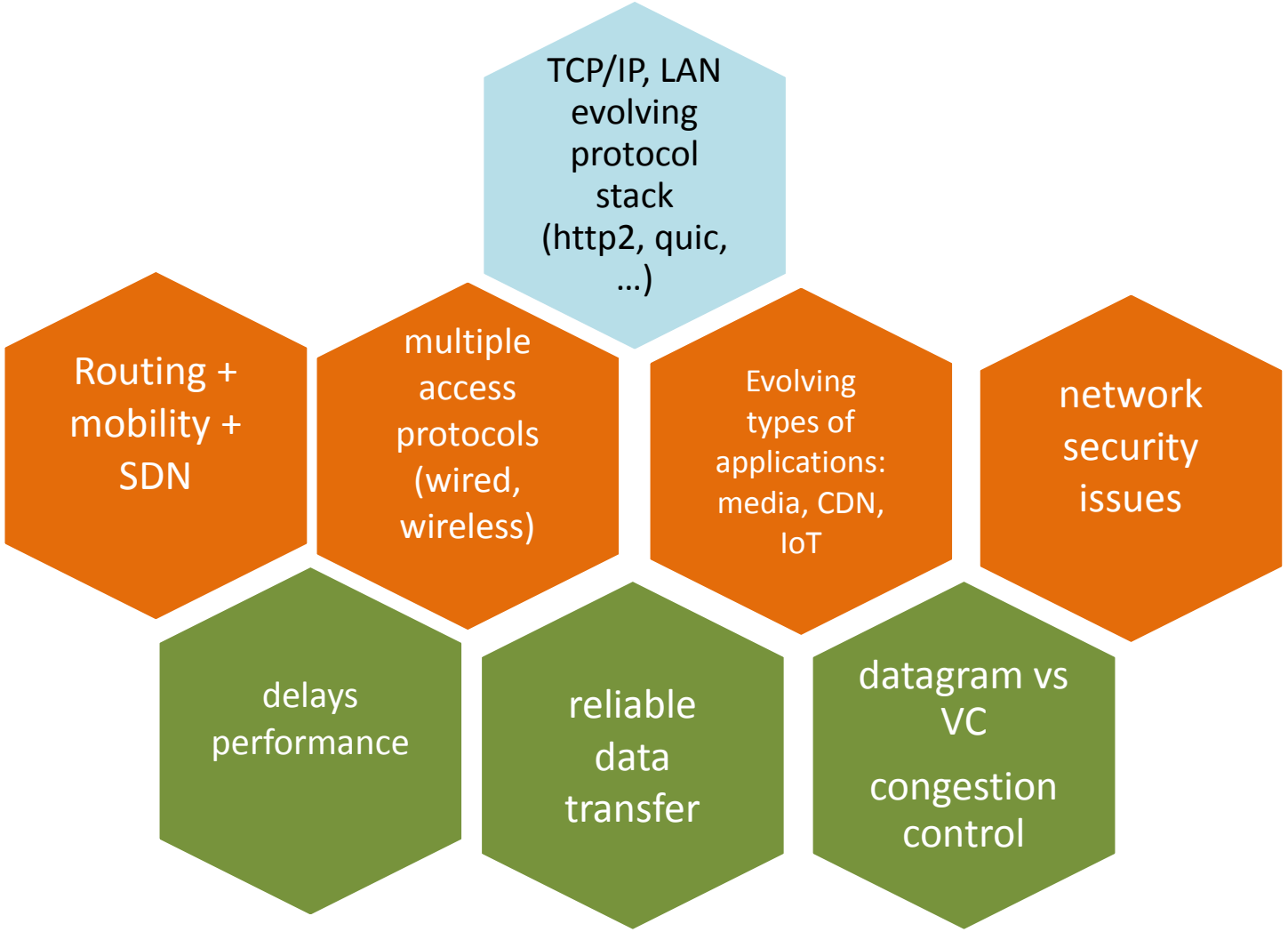
- why, how congestion occurs
- CC in TCP and performance; implied weaknesses
- CC in other ways, e.g. VC-based networks
  - Real-time (RT)-traffic resource reservation:
    traffic shaping and policing
  - rate-based







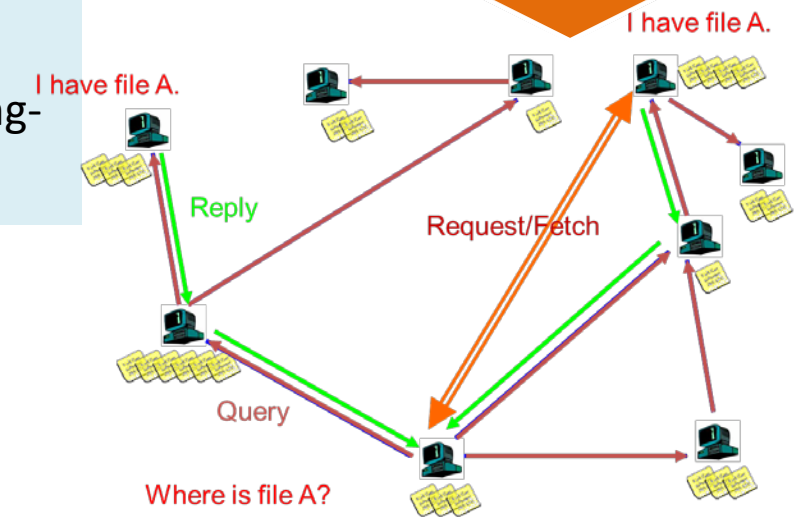
Highlights

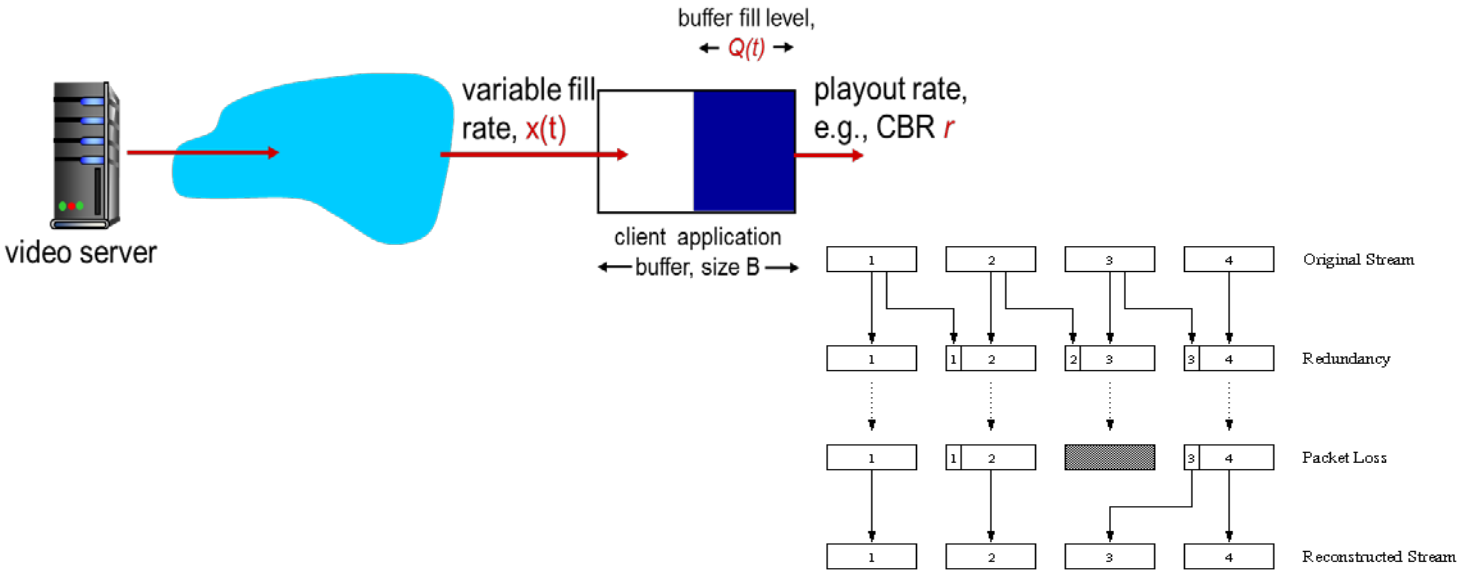

TCP/IP, LAN evolving protocol stack (http2, quic, …)
Routing + mobility + SDN
multiple access protocols (wired, wireless)
Evolving types of applications: media, CDN, IoT
network security issues
delays performance
reliable data transfer
datagram vs VC
congestion control

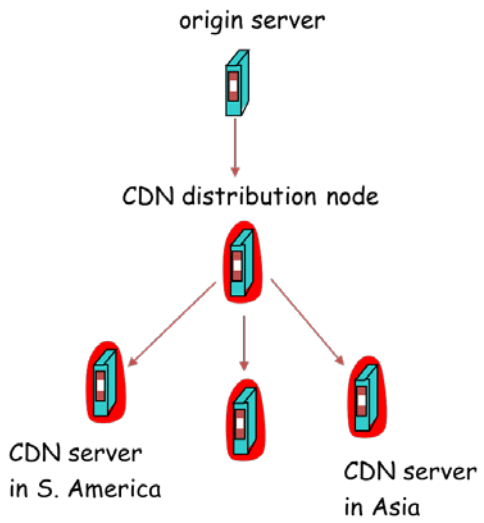
Evolving types of applications ++

Evolving types of applications: media, CDN, IoT

- P2P/streaming applications-infrastructure (application-layer networking)
- Varying approaches for MM apps, besides conceptual, VC-related
  - Application-level solutions (playout delay, forward-error-control, caching-CDN, FEC)



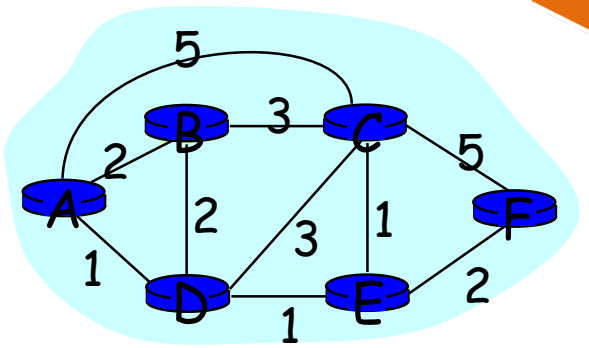


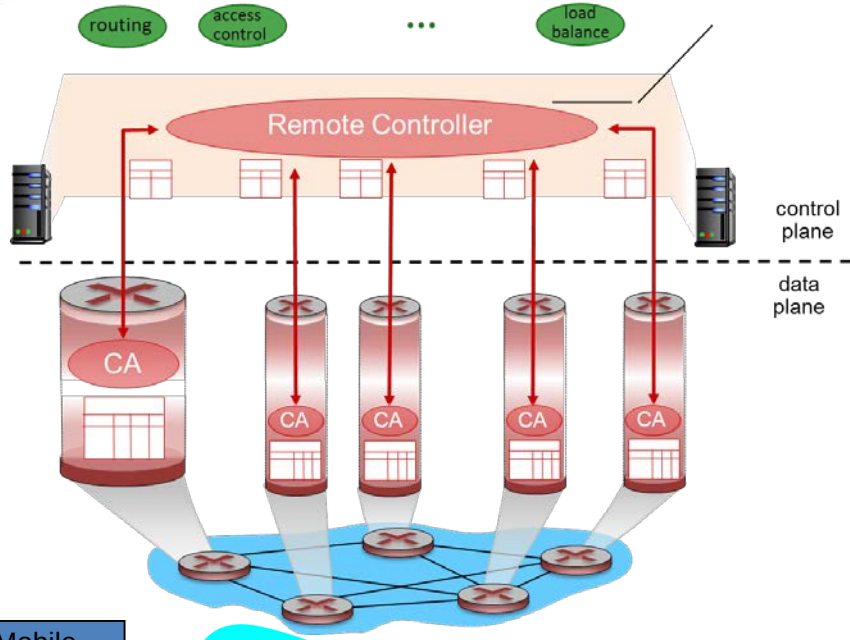


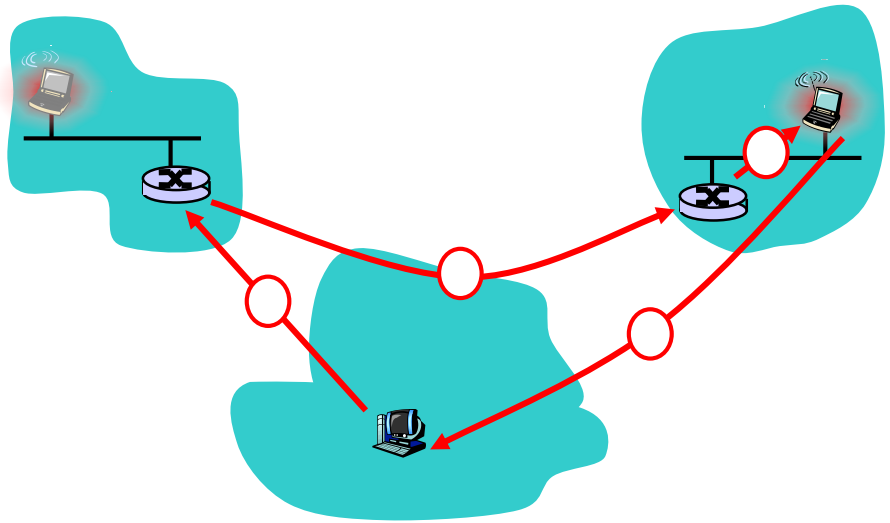
Routing + mobility + SDN

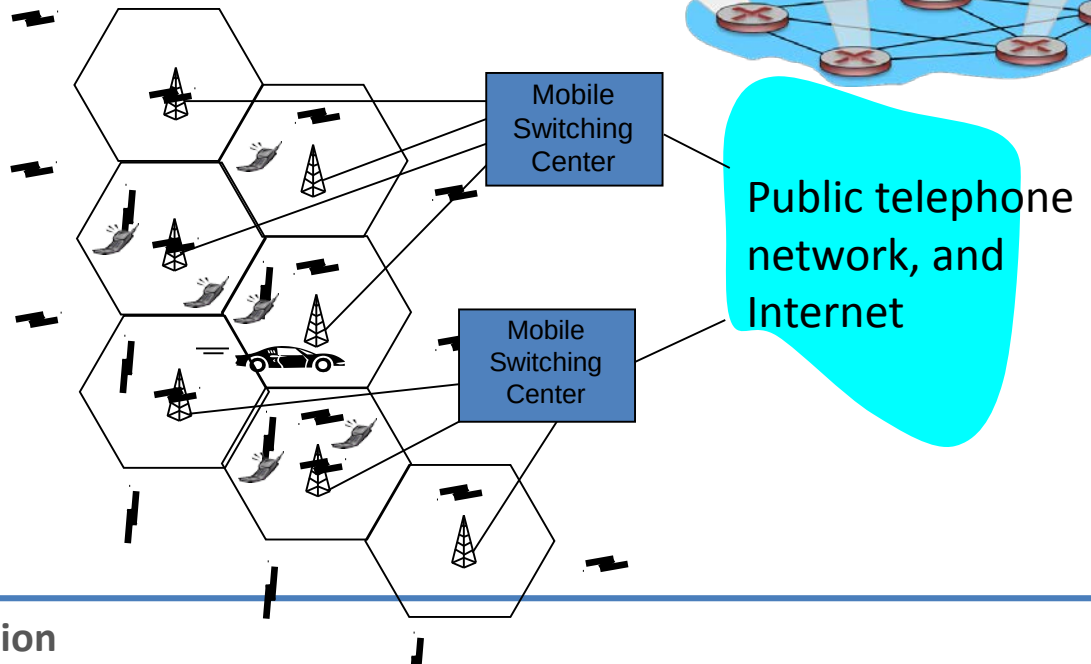
Routing + mobility + SDN

- Routing algorithms, protocols
- Forwarding in routers
- Resource, policy issues
- Data plane and control plane in SDN
- Addressing mobility, tunneling







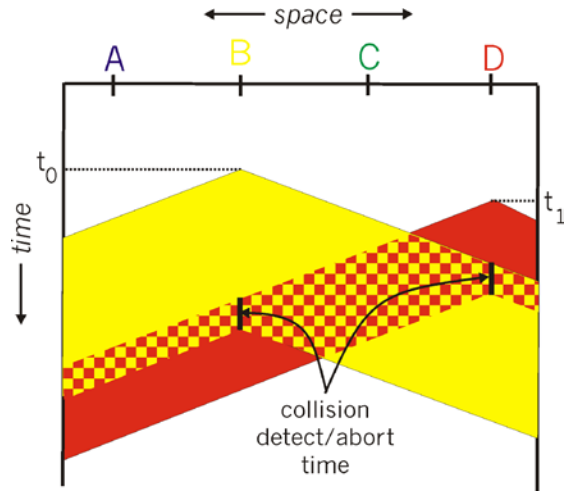


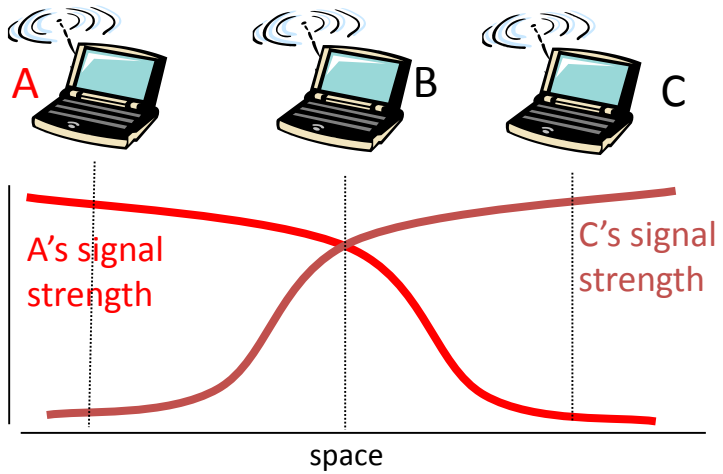
Medium access: multiple access methods

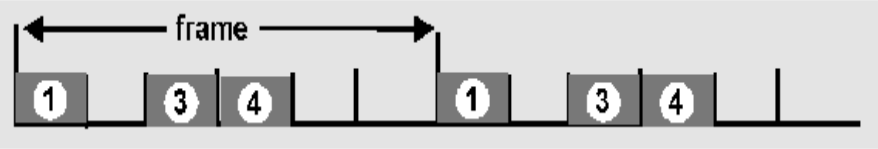
multiple access protocols (wired, wireless)

Strategies: (functionality, appropriateness)

- **Contention-based (random access), wired/wireless:**
  - Aloha, CSMA(CD/CA)
- **Collision-free:**
  - **Channel partitioning:** TDMA, FDMA, CDMA
  - **Taking turns:** e.g. tokens, reservation-based



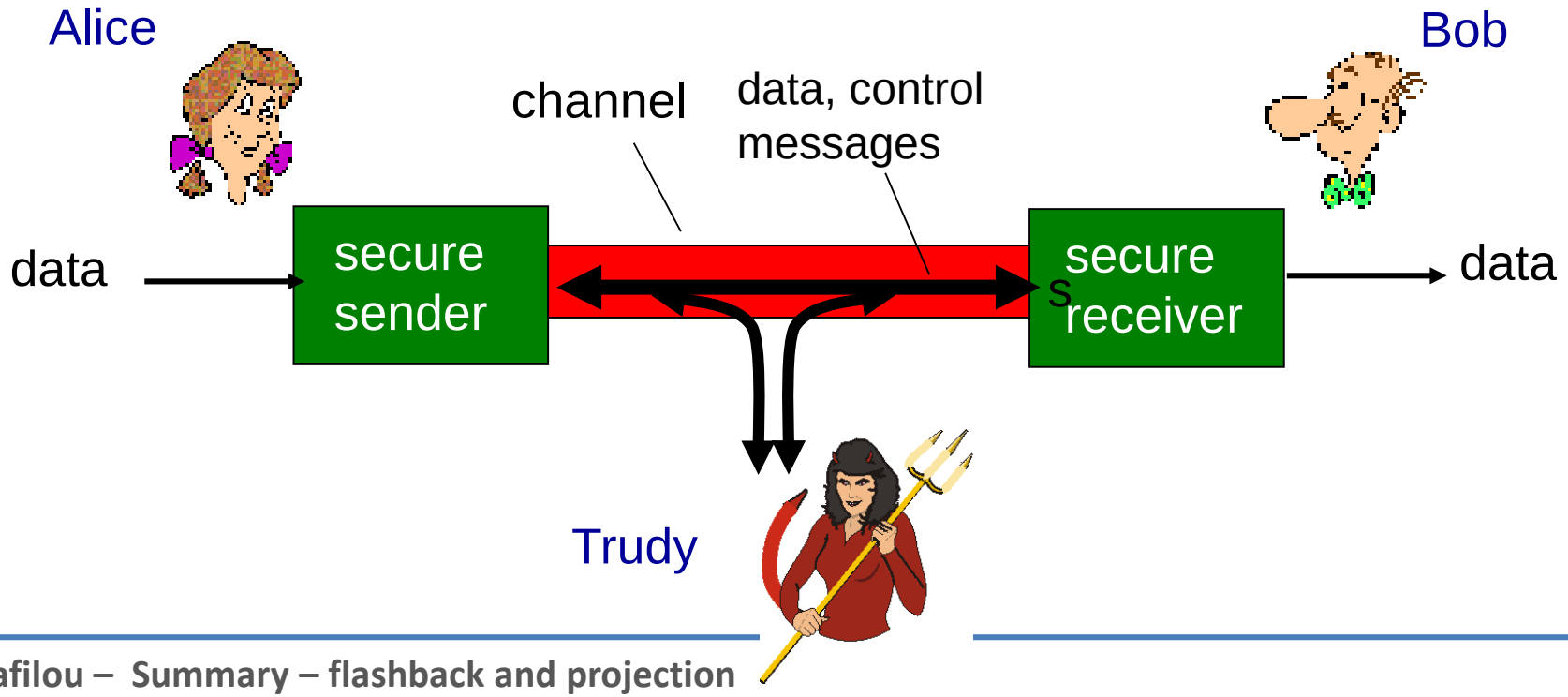




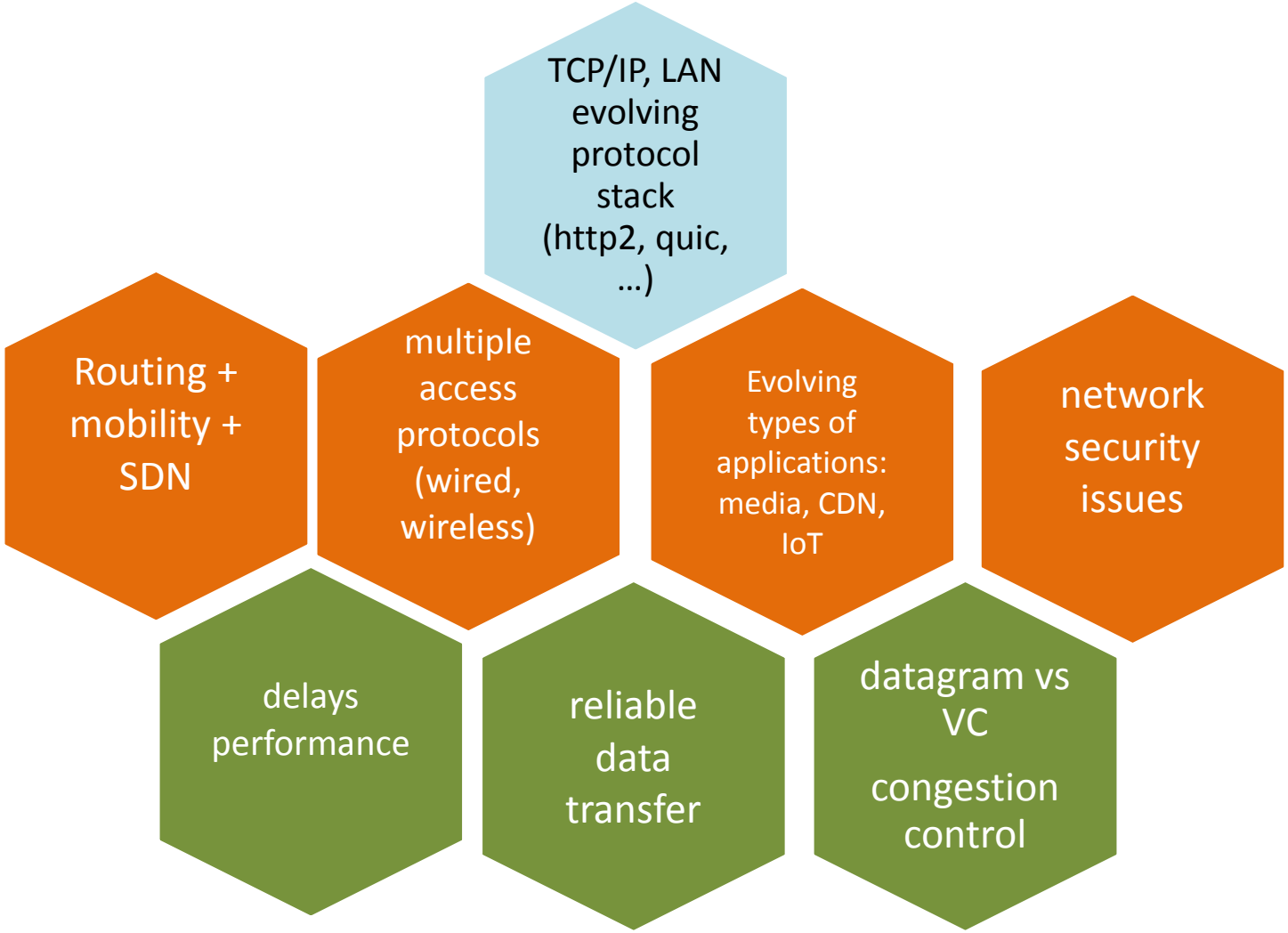
Security issues

network security issues

- **C, I, A** and methods to achieve them
  - The language of cryptography
  - Message integrity, signatures
- Instantiation in Internet: SSL, IPsec
- Firewalls



Highlights

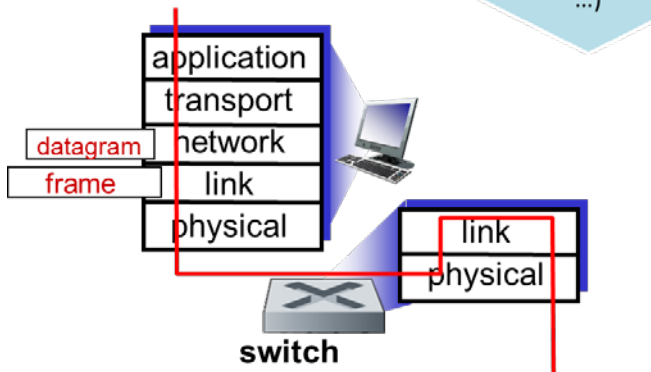

TCP/IP, LAN evolving protocol stack (http2, quic, …)
Routing + mobility + SDN
multiple access protocols (wired, wireless)
Evolving types of applications: media, CDN, IoT
network security issues
delays performance
reliable data transfer
datagram vs VC
congestion control

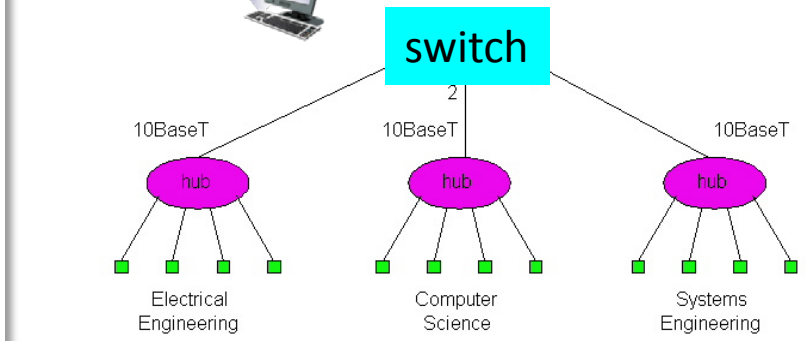
TCP/IP protocol stack, applications, evolution

- Instantiation of network solutions, advantages, limitations, updates
  - http, DNS, SMTP,
  - TCP. UDP, IP, routing protocols
  - LAN Protocol Examples: wired, wireless: Ethernet, 802.xy, GSM:
  - Connecting devices: functionalities and differences (Hubs, switches)
  - Algorithms for switch-"routing": learning& forwarding of packets
  - ARP

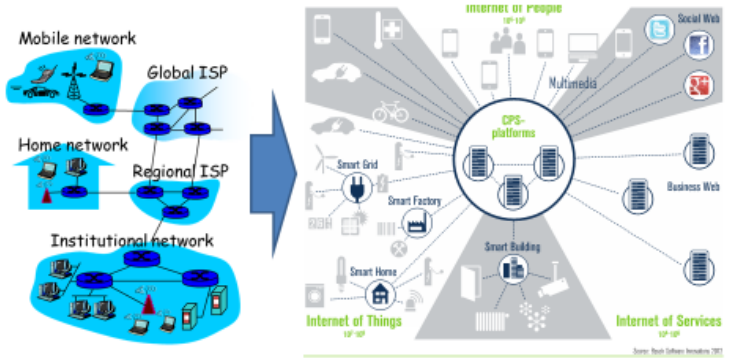
- New types of applications and how they function in the Internet
  - Changing NW-edge: Intserv/diffserv, http2, quic, IoT-related protocols…

TCP/IP, LAN evolving protocol stack (http2, quic, …)





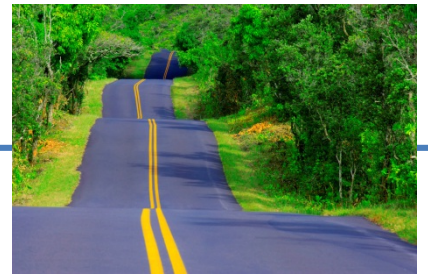
Recall: Internet & its context….



approx 10 yrs ago

continuous evolution ….

Roadmap



Synthesis: a day in the life of a web request

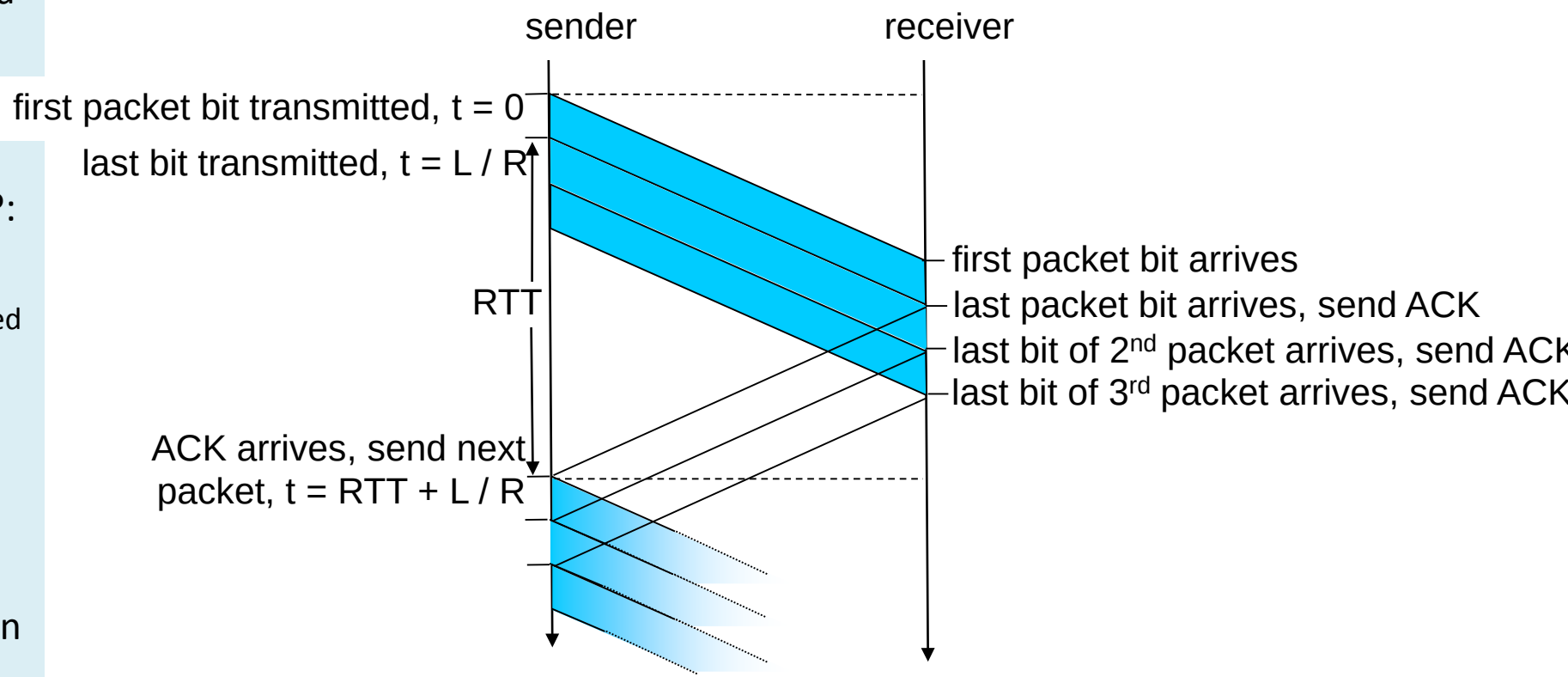
- Putting lots-of-what-we-learned together: synthesis!
  - *goal:* identify, review protocols (at all layers) involved in seemingly simple scenario: requesting www page
  - *scenario:* student attaches laptop to campus network, requests/receives www.google.com

Highlights

Some questions asked by you

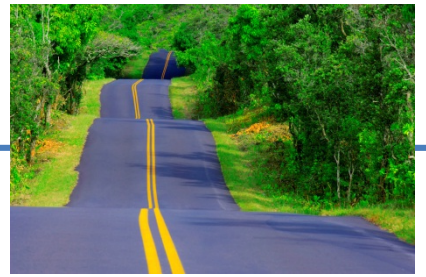
some questions asked by you:

1. How can we have reliable data transfer on top of UDP
   - Implementation at app-layer
2. RDT with only NACK possible?:
   - no-lossy channel
     - Timeout => ok, msg received
   - lossy channel:
     - Cannot tell difference between late or lost….
3. Working with time-space diagrams
   - Eg. what is a full utilization window?



E.g. for 100% utilization, calculate how many packets can fill in RTT + L / R, ie (RTT + L / R)/ (L / R)

Roadmap



Synthesis: a day in the life of a web request

- Putting lots-of-what-we-learned together: synthesis!
  - *goal:* identify, review protocols (at all layers) involved in seemingly simple scenario: requesting www page
  - *scenario:* student attaches laptop to campus network, requests/receives www.google.com

Highlights

Some questions asked by you

Reflections, prespectives

…….

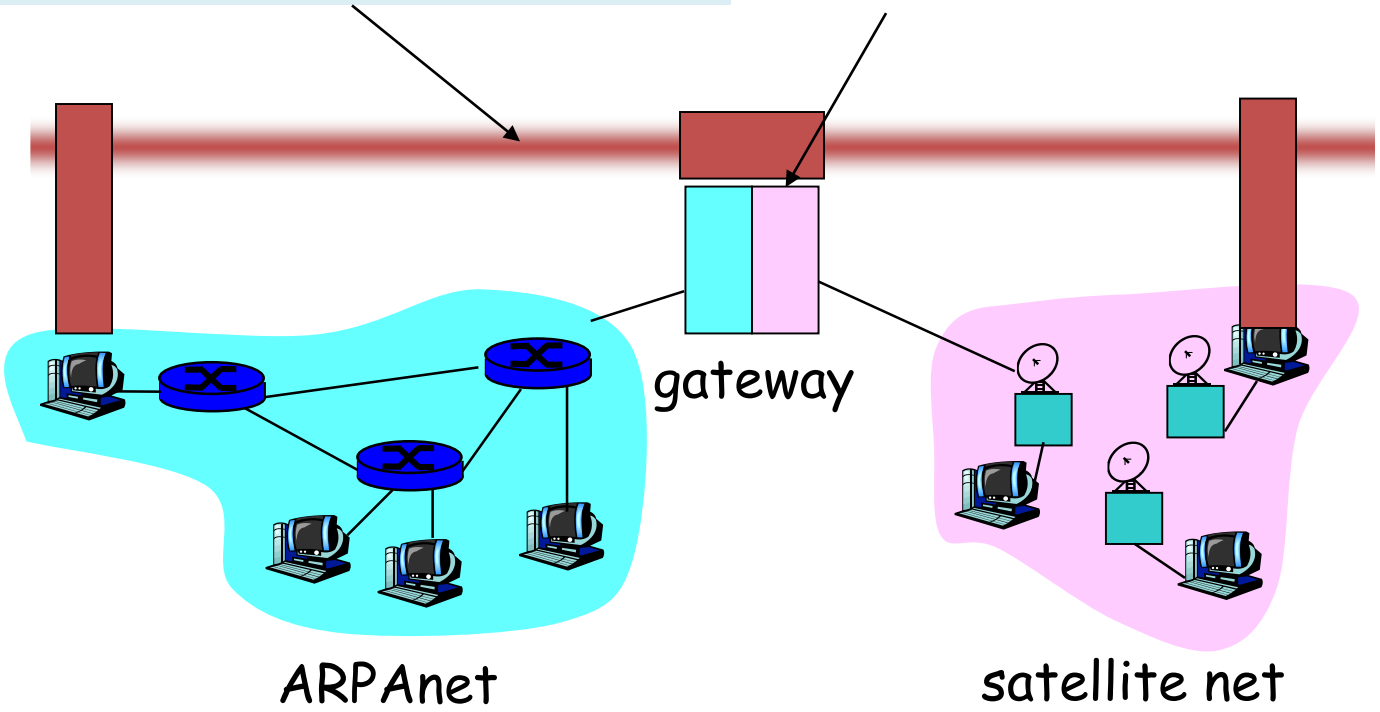
The Internet: virtualizing networks

Internetwork layer (IP):

- addressing: internetwork appears as single, uniform entity, despite underlying local network heterogeneity
- network of networks

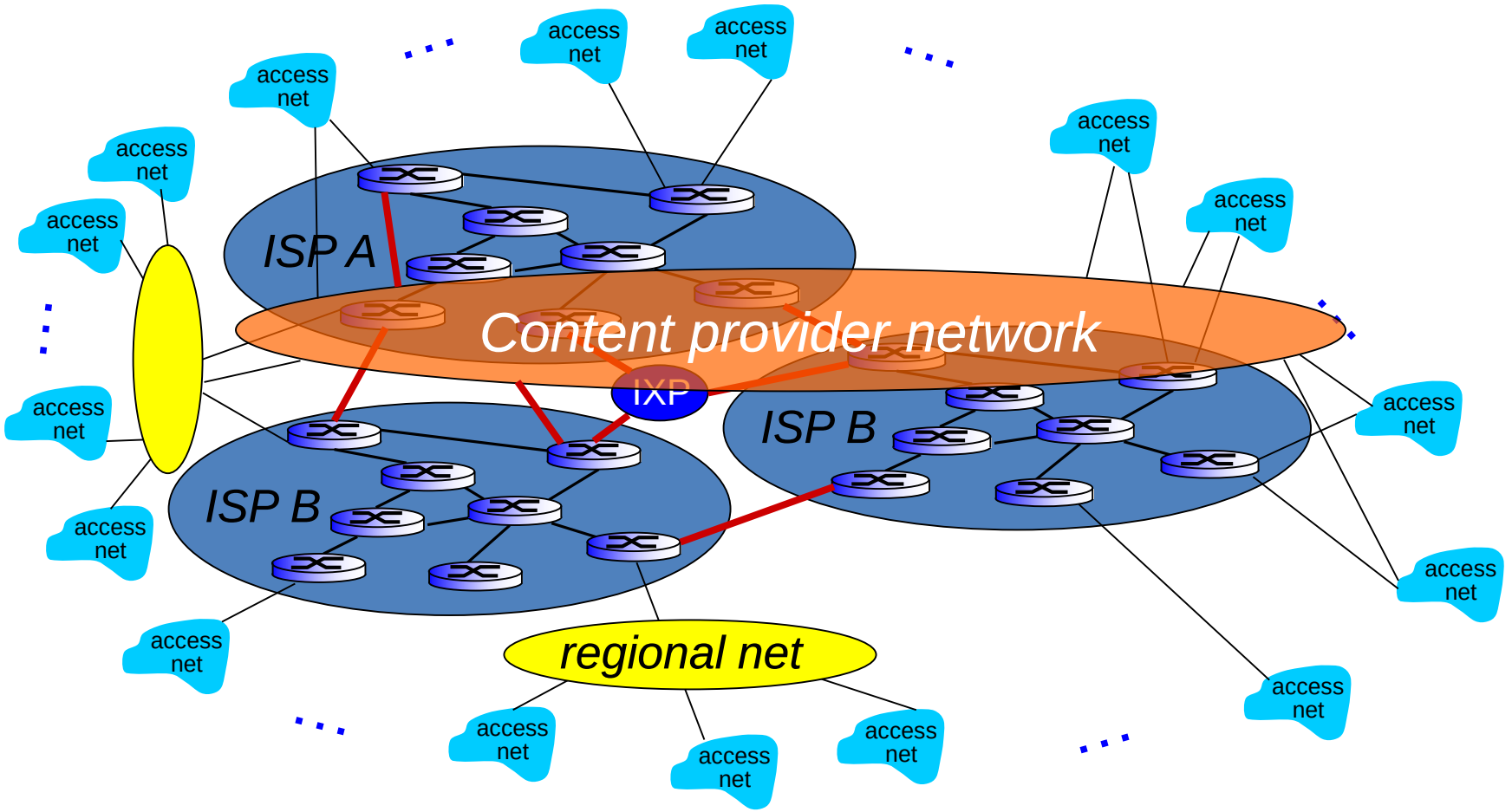
Gateway:

- "embed internetwork packets in local packet format"
- route (at internetwork level) to next gateway

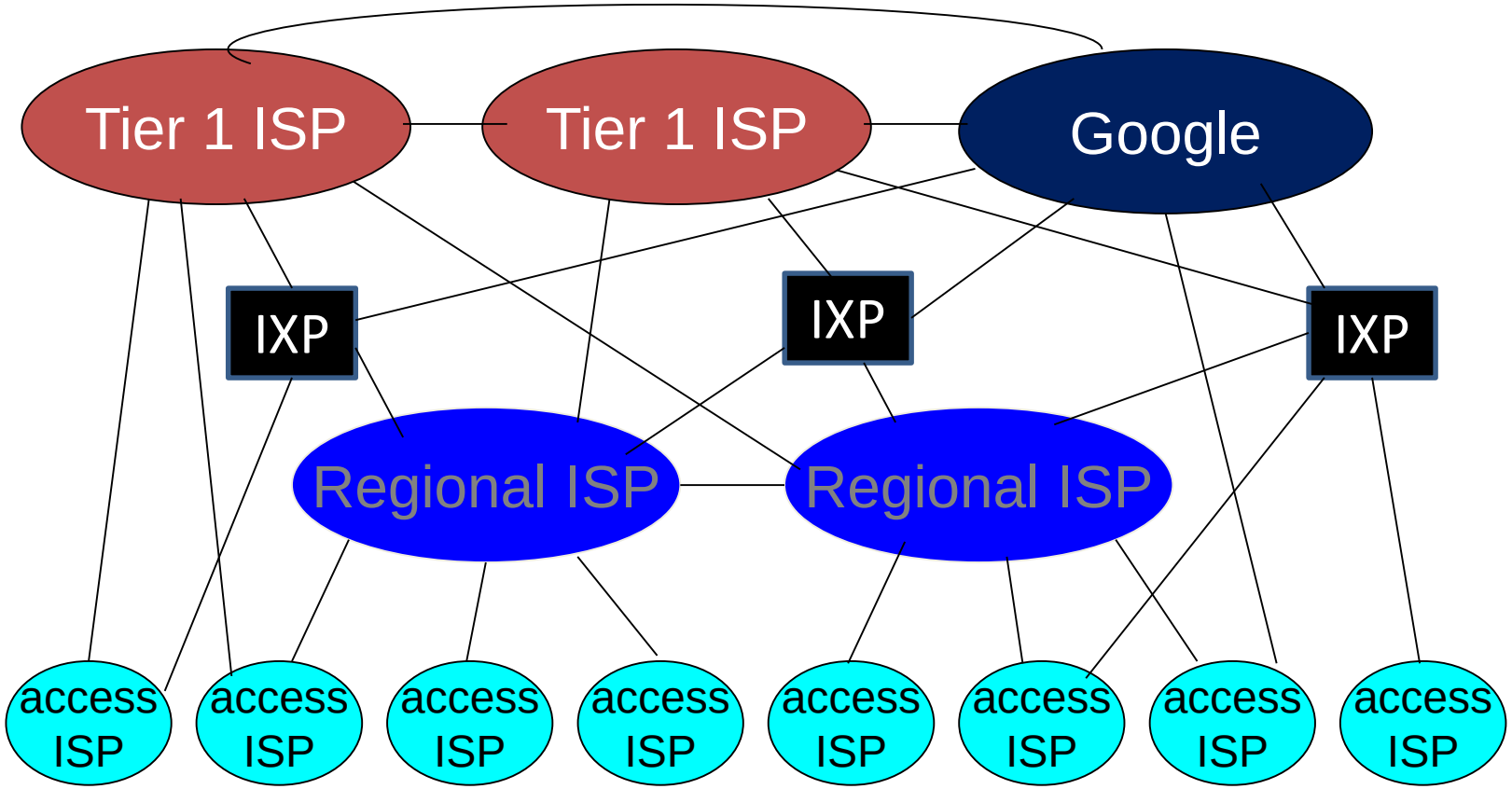


Internet structure: network of networks

… and content provider networks (e.g., Google, Microsoft, Akamai) may run their own network, to bring services, content close to end users



Internet structure: network of networks



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, networking, data bottlenecks

Inside a 40-ft Microsoft container, Chicago data center

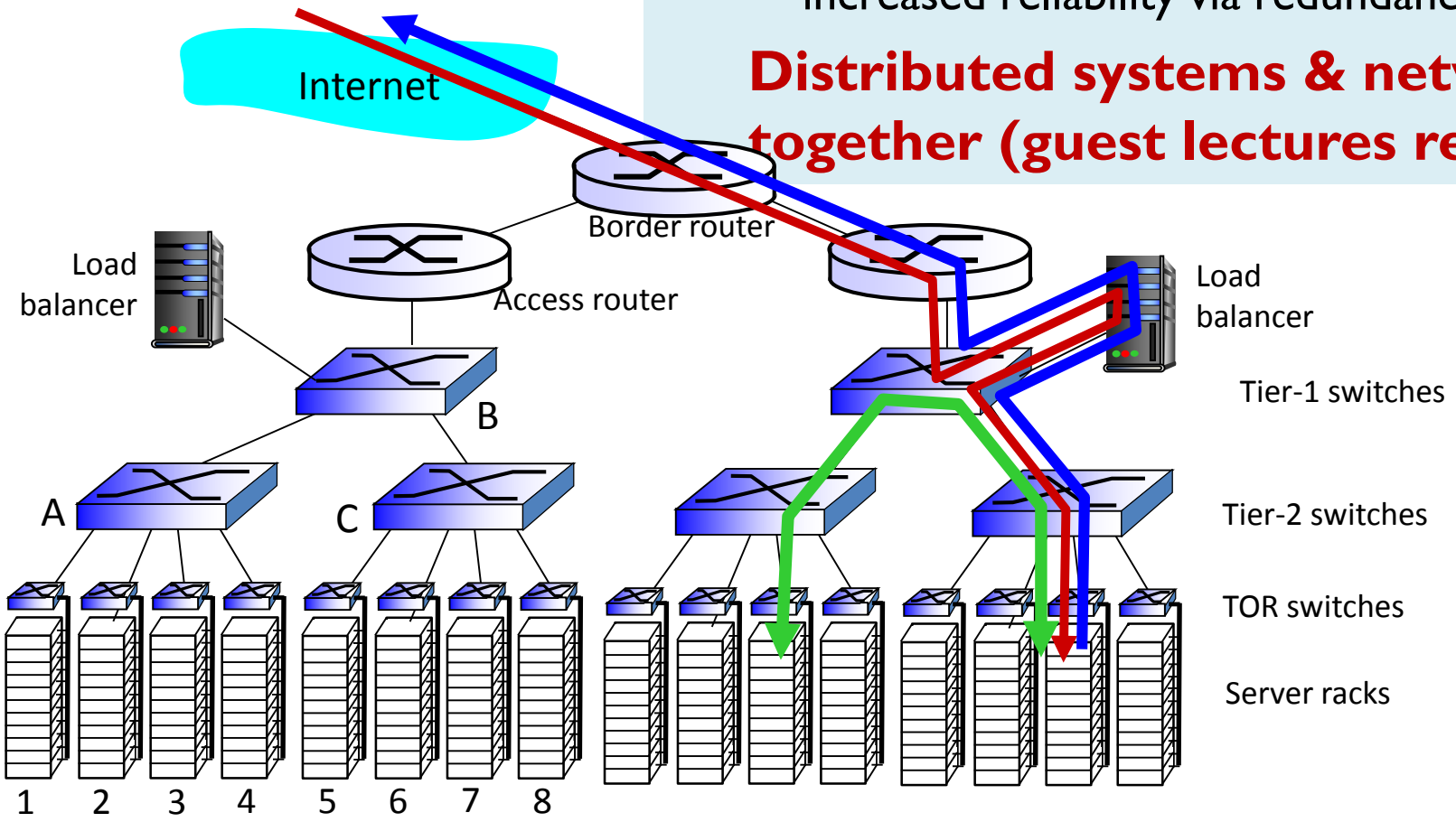
Data center networks

load balancer: application-layer routing

- directs client requests (workload) within data center

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

Distributed systems & networks working together (guest lectures relevance)



Thank you!

Synthesis: a day in the life of a web request

- Putting lots-of-what-we-learned together: synthesis!
  - *goal:* identify, review protocols (at all layers) involved in seemingly simple scenario: requesting www page
  - *scenario:* student attaches laptop to campus network, requests/receives www.google.com

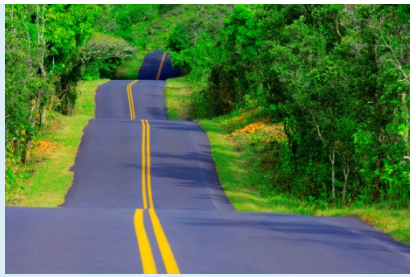
Highlights

Some questions asked by you

Reflections, prespectives

Recall, important for the exam:

When/where: wednesday March 14, 14.00-18.00, SB



You may have with you:

- English-X dictionary
- no calculators, PDAs, etc (if/where numbers matter, do rounding)

To think during last, summary-study

Overview; critical eye; explain; ask yourselves: why is this so? / How does it work?

Good luck with all your efforts!!!

"If you hear a voice within you say 'you cannot paint,' then by all means paint, and that voice will be silenced." – Vincent Van Gogh