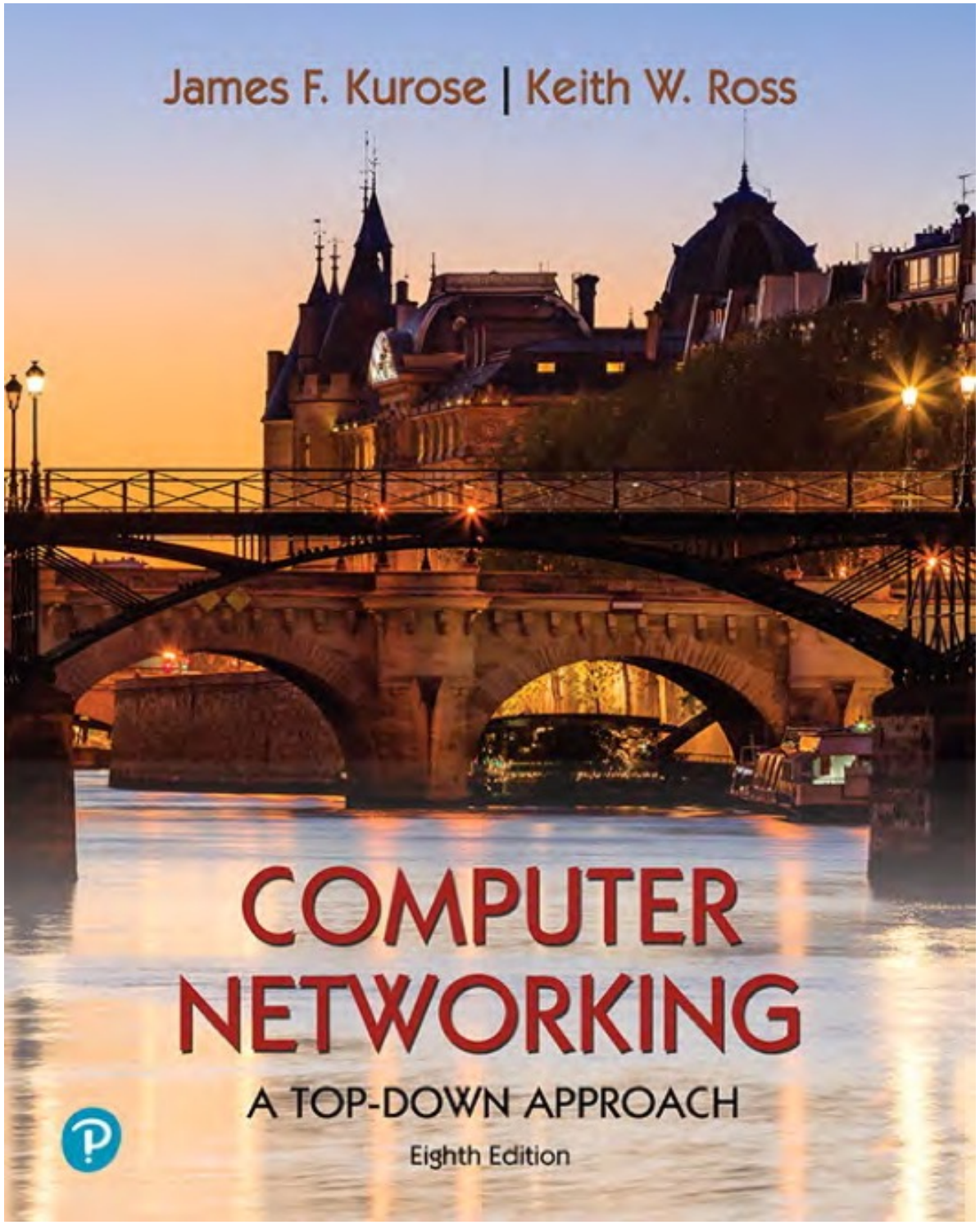
CS118:
Computer Network Fundamentals

Lecture-1: introduction

CS118: explains (roughly) how the Internet works

- Internet: a huge, complex network of networks
- Divide-and-conquer
  - Figure out how many major parts,
  - Learn one piece at a time
- Your job:
  - Read textbook, think, collect a list of questions
    - review every lecture slide deck after each class
  - Ask questions in class/office hours/via Piazza
  - Practice what you learn through homework and projects



Brief Contents

Chapter 1	Computer Networks and the Internet	1
Chapter 2	Application Layer	81
Chapter 3	Transport Layer	181
Chapter 4	The Network Layer: Data Plane	303
Chapter 5	The Network Layer: Control Plane	377
Chapter 6	The Link Layer and LANs	449
Chapter 7	Wireless and Mobile Networks	531
Chapter 8	Security in Computer Networks	607
	References	691
	Index	731

Course assignment and due schedule

Midterm	In-class, Wednesday Feb 5 (**Location TBD**)	
Final	3:00PM-6:00PM Saturday March 21 (**Location TBD**)	
Homework	**Release:** on Thursday of week 1, 3, 5, 7; **Due:** 11:59pm Tuesday of week 3, 5, 7, 9.	
Project 0	**Release**: Monday Jan 6, 2024 (Week 1) **Due:** 11:59pm Wednesday, Jan 15, 2024 (Week 2)	1.5 weeks
Project 1	**Release**: Thursday Jan 16, 2024 (Week 2) **Due:** 11:59pm Sunday, Feb 16, 2024 (Week 6) **Grading:** auto-grading script (sample tests will be provided to let everyone test their code before submission)	4.5 weeks
Project 2	**Release**: Monday, Feb 17, 2024 (Week 7) **Due:** 11:59pm Sunday, March 16, 2024 (Week 10) **Grading:** auto-grading script (sample tests will be provided to let everyone test their code before submission)	4 weeks

FOR ALL OTHER COURSE INFO, PLEASE SEE HTTPS://BRUINLEARN.UCLA.EDU/

Course workload and grading

- Bi-weekly homework assignments
- 3 programming projects,
  0. UDP socket (individual)
  1. Reliable data delivery (2-3 people team)
  2. Secured reliable data delivery (2-3 people team)
- Midterm and final exams (cheat sheets allowed, 2 pages double sided)
- **Strict Grading Policy**
  - Homework: do it yourself; no credit for late submission
  - Project: 20% credit reduction per late day
  - *No make-up exam*

Homework	20%
Programming Projects	25% (5/ 10/ 10)
Midterm	25%
Final exam	30%

2% extra credits based on piazza

1% extra credits course evaluation and TA/LA feedbacks

Class Policy

The following actions are **strictly prohibited**

- Posting/sharing/selling class material, with or without answers, to anyone outside this class, during or after this quarter.
- Use of old homework/midterm/finals in doing homework or exams
- Use ChatGPT in doing assignments
- Making your project code publicly available either during or *after* this quarter
  - *you must use private repository* on GitHub or GitLab

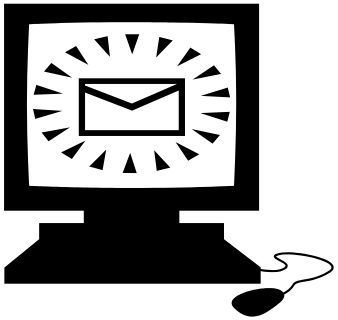
Hints for Getting Good Grade

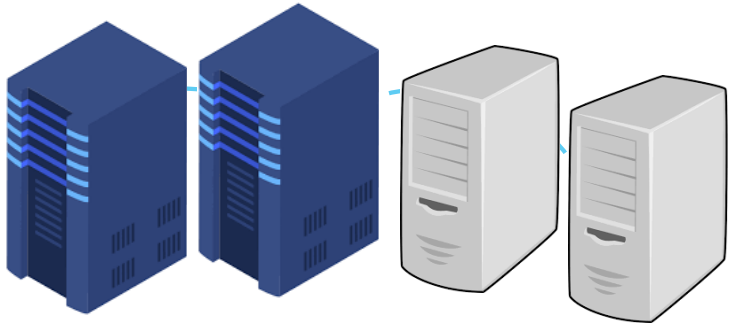
- Review previous lecture slides
- Read textbook before coming to each lecture
- *Ask questions*
- Get your work done early
  - Lecture slides uploaded to BruinLearn *one day before the lecture*
  - Get HWs and projects done ***before*** the deadline

Let’s get started

Today we cover the basic concepts in
Chapter 1 of the textbook

What is a Computer Network

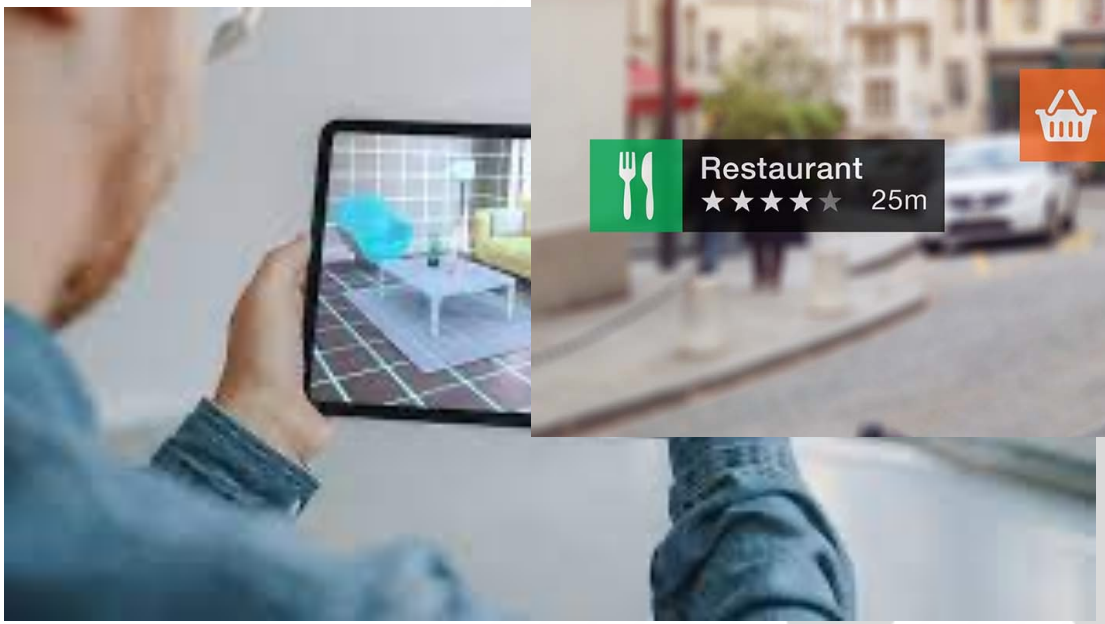


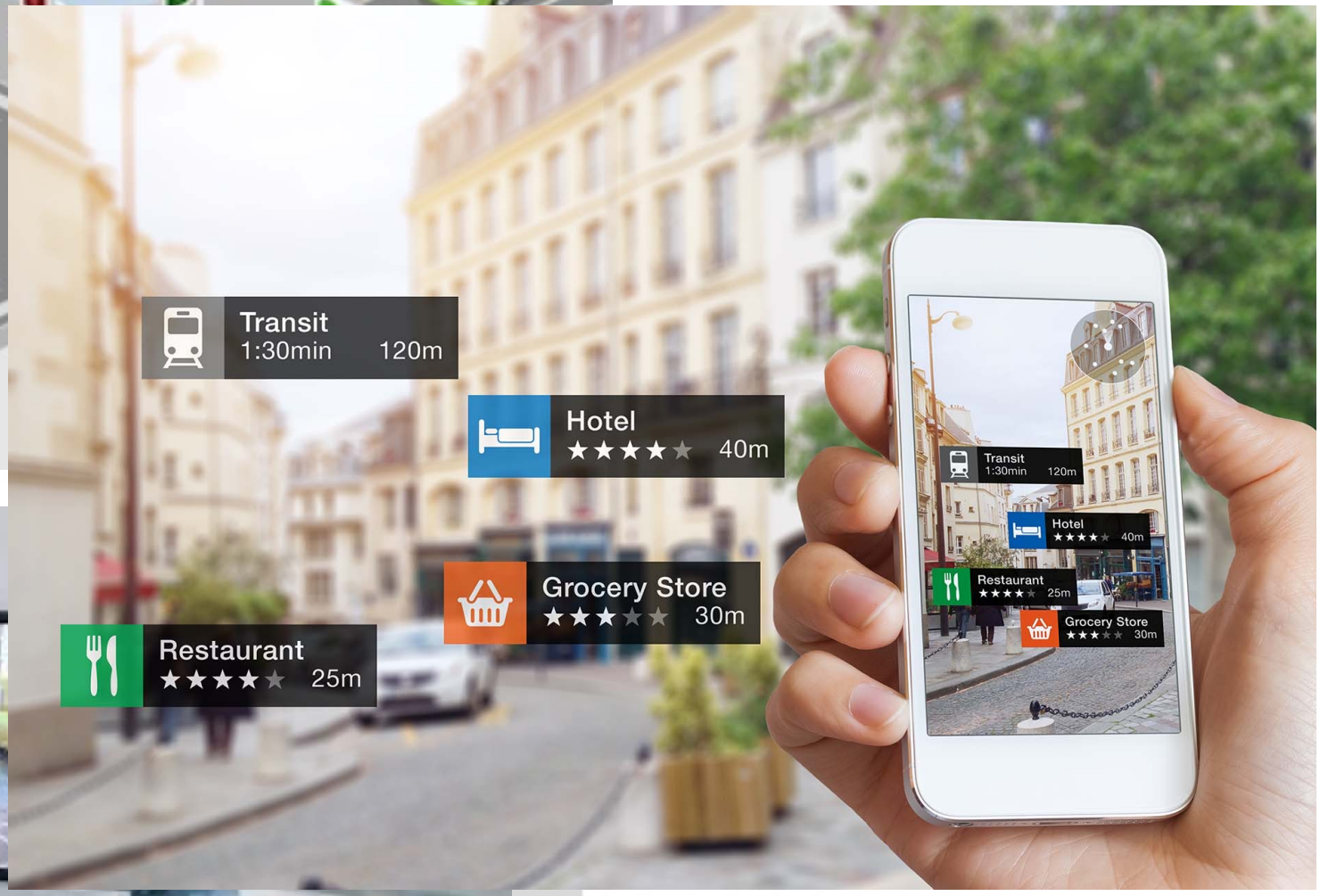


Internet

What is a Computer Network

Coming to you ....

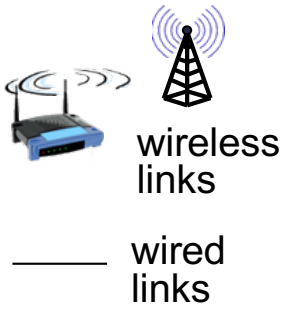




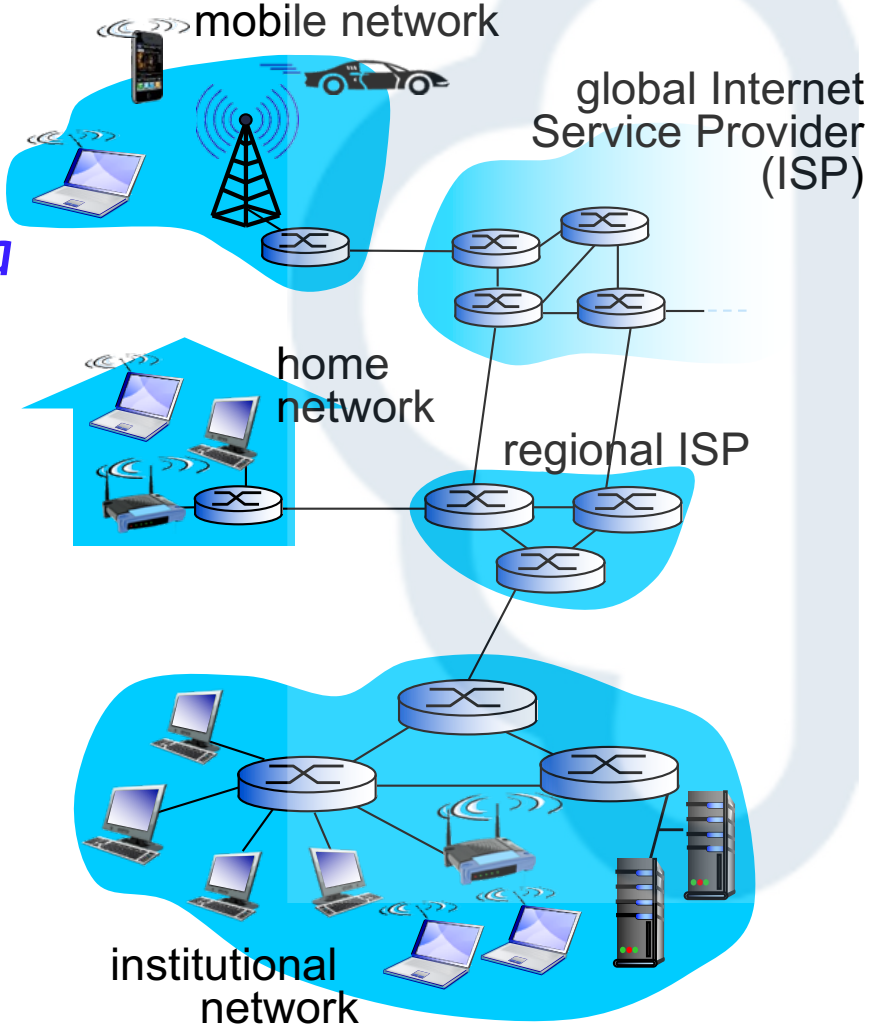
Terminology

- billions of connected computing devices:
  - *hosts* = *end systems*
  - running *network apps*
  - Apps send/receive *data packets*

- *Routers* = *packet switches* inside network



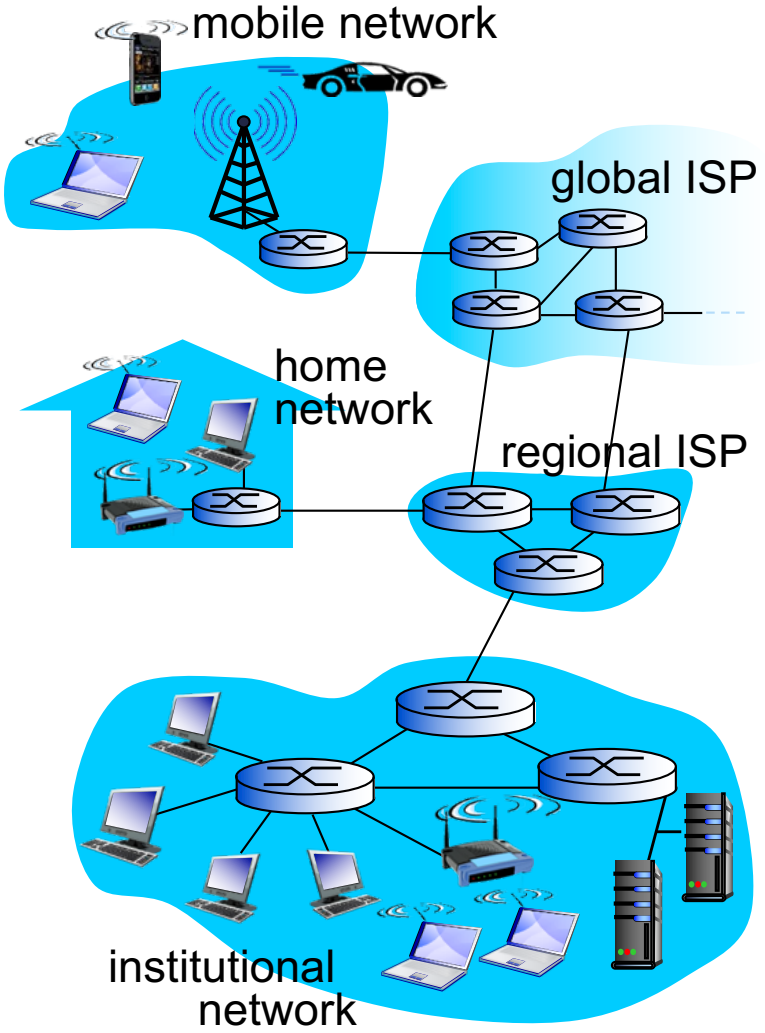
- *communication links*
  - fiber, copper, radio, satellite
  - transmission rate = *bandwidth (BW)*



Recent years witnessed rapid growth of giant cloud service providers

"Nuts and Bolts"

- *Internet:* "network of networks"
  - Interconnected ISPs, enterprise networks, now also cloud service providers
- *Protocols:* define how to send, receive packets
  - e.g., HTTP, TCP, IP, 802.11
- *Internet protocol standards*
  - RFCs: "Request for Comments"
    - https://www.rfc-editor.org/rfc-index.html
    - Developed by Internet Engineering Task Force (IETF)
  - IEEE Standards
  - W3C (World Wide Web Consortium), and others



What is a protocol?

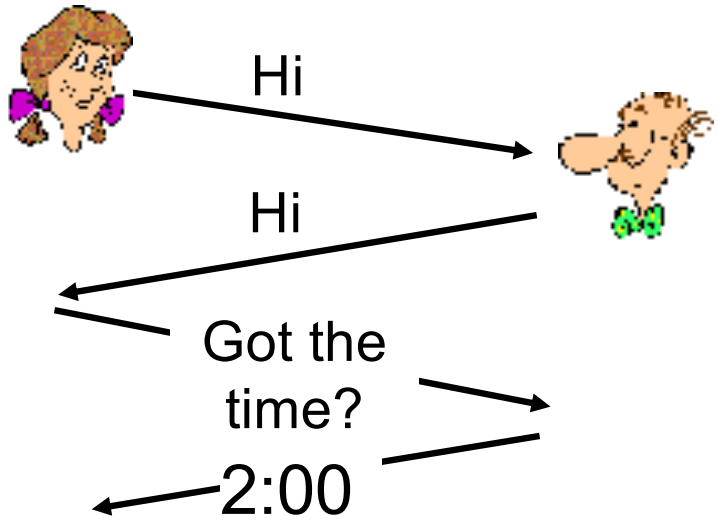
Traffic light protocol

- Green: go
- Red: stop
- Yellow: slow down - stop

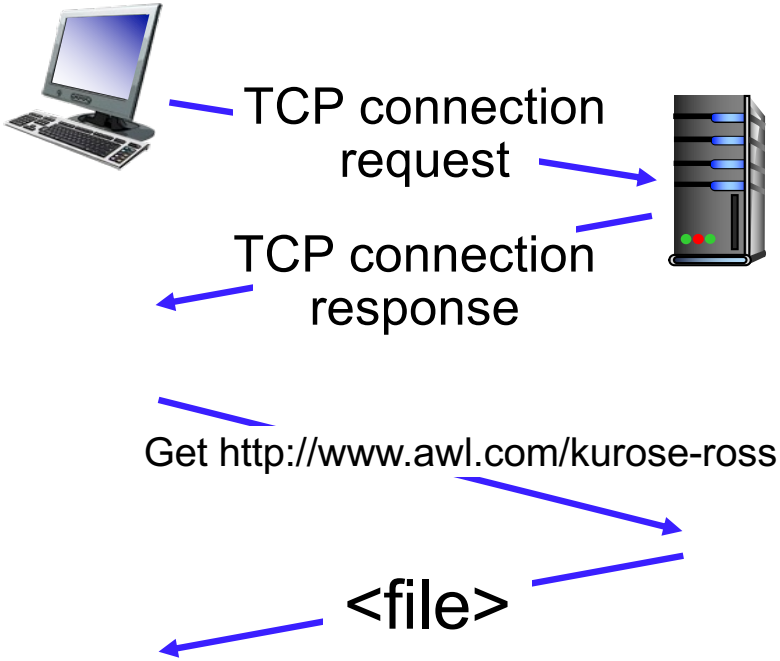
… specific messages sent

… specific actions taken when the messages received

human protocols:

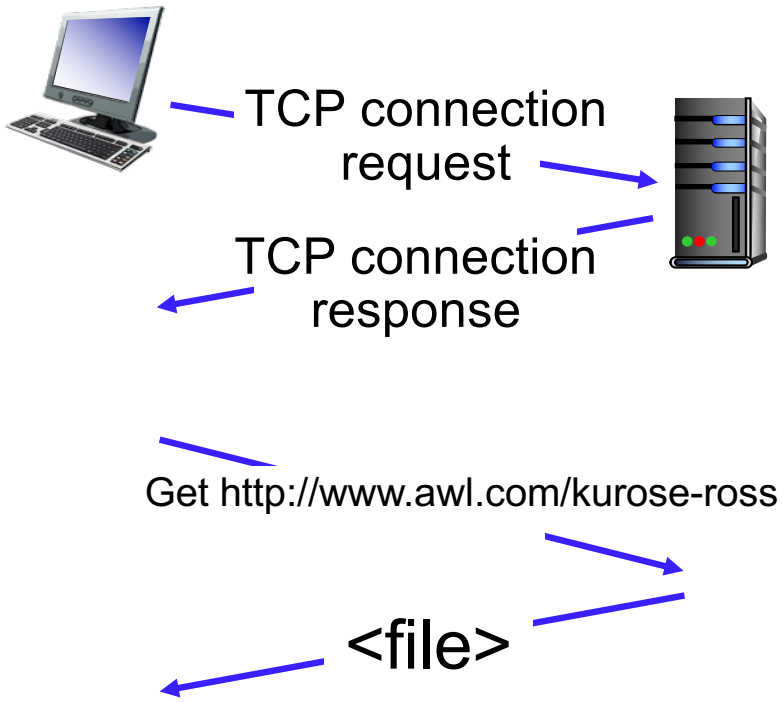


computer protocols:



Internet protocols

computer protocols:



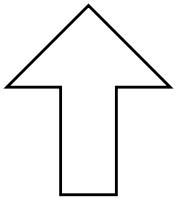
- Communication between machines rather than humans
- all communication activity governed by protocols

protocols define *format*, *order* of *packets sent and received* among network entities, and *actions taken* on packet transmission, receipt

Delivering data over the global Internet is a complicated process, involving many many steps

How to get the work done: divide and conquer

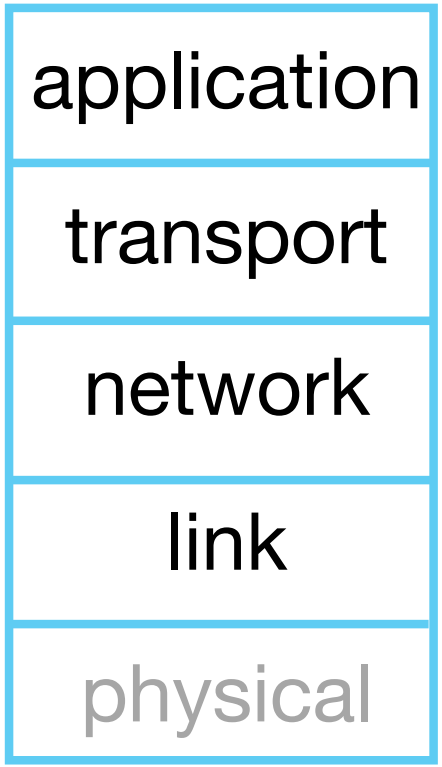
Group functions to a few modules



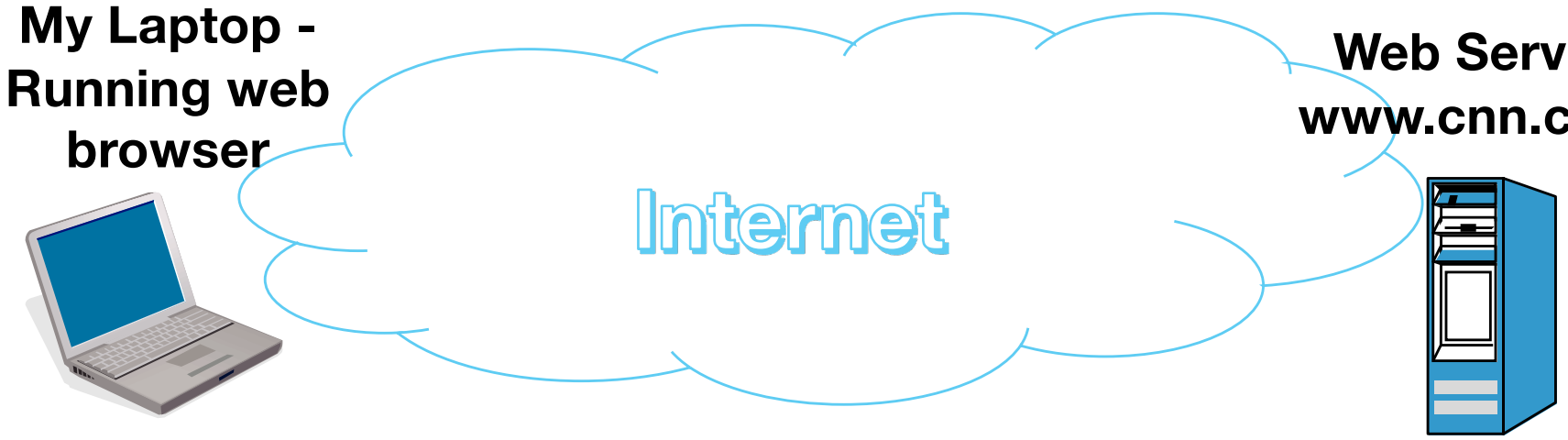
How many?

Internet protocol stack

- Application layer protocols
  - Support data exchange between application processes
  - Example: SMTP, HTTP, DNS
    (Simple Mail Transfer Protocol)
- Transport layer protocols
  - handling delivery reliability, multiplex within a host
  - Example: TCP, UDP
- Network layer protocols
  - forward packets from source to destination
  - Example: IP
- Link layer protocols
  - transfer data between directly connected network elements
  - Example: Ethernet protocol, WiFi
- Physical layer: bits “on the wire”



Application View



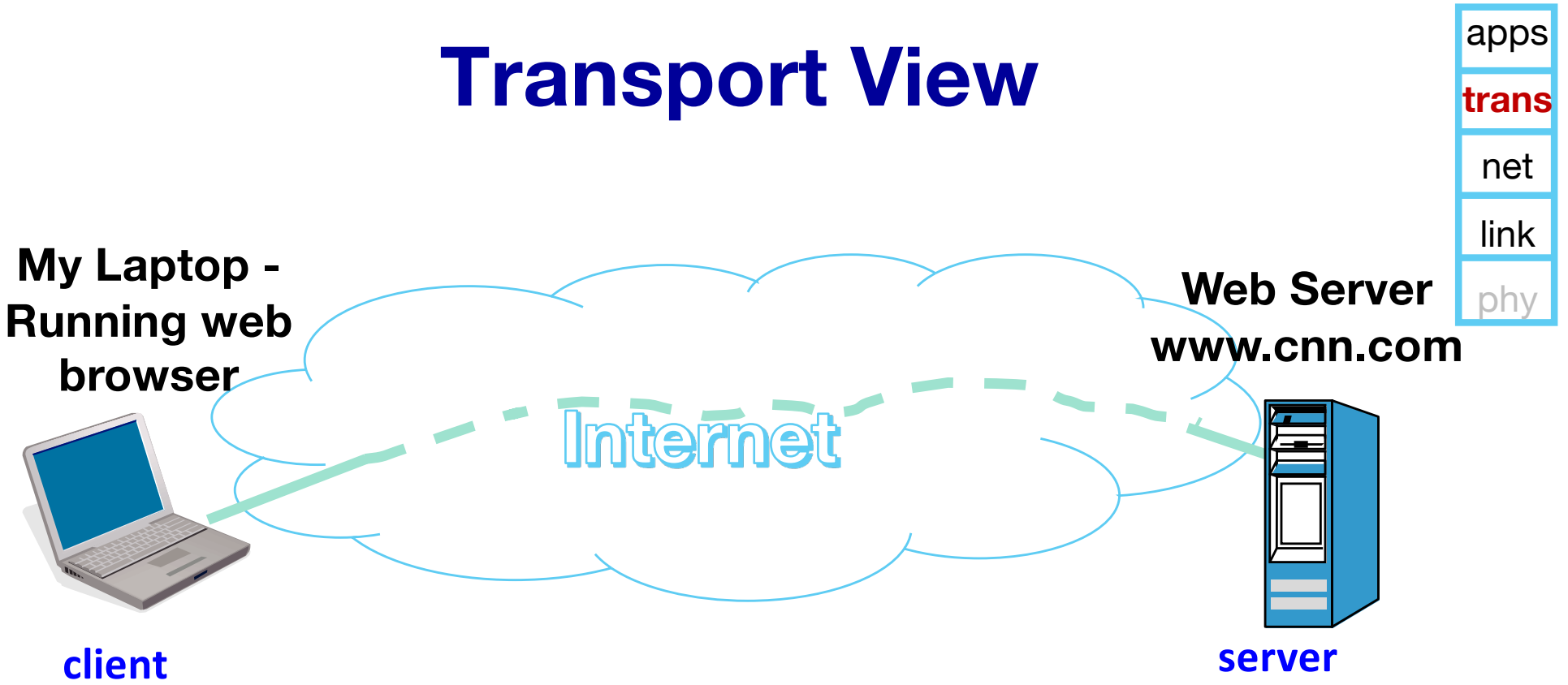
These are *application programs*

They talk to each other using *application protocols* (web protocol: HTTP)

Application protocols

- Assume network can send data to any hosts on the Internet
- Don't know/*care* how data is sent, and assume all data delivered reliably
- Runs on top of a transport protocol

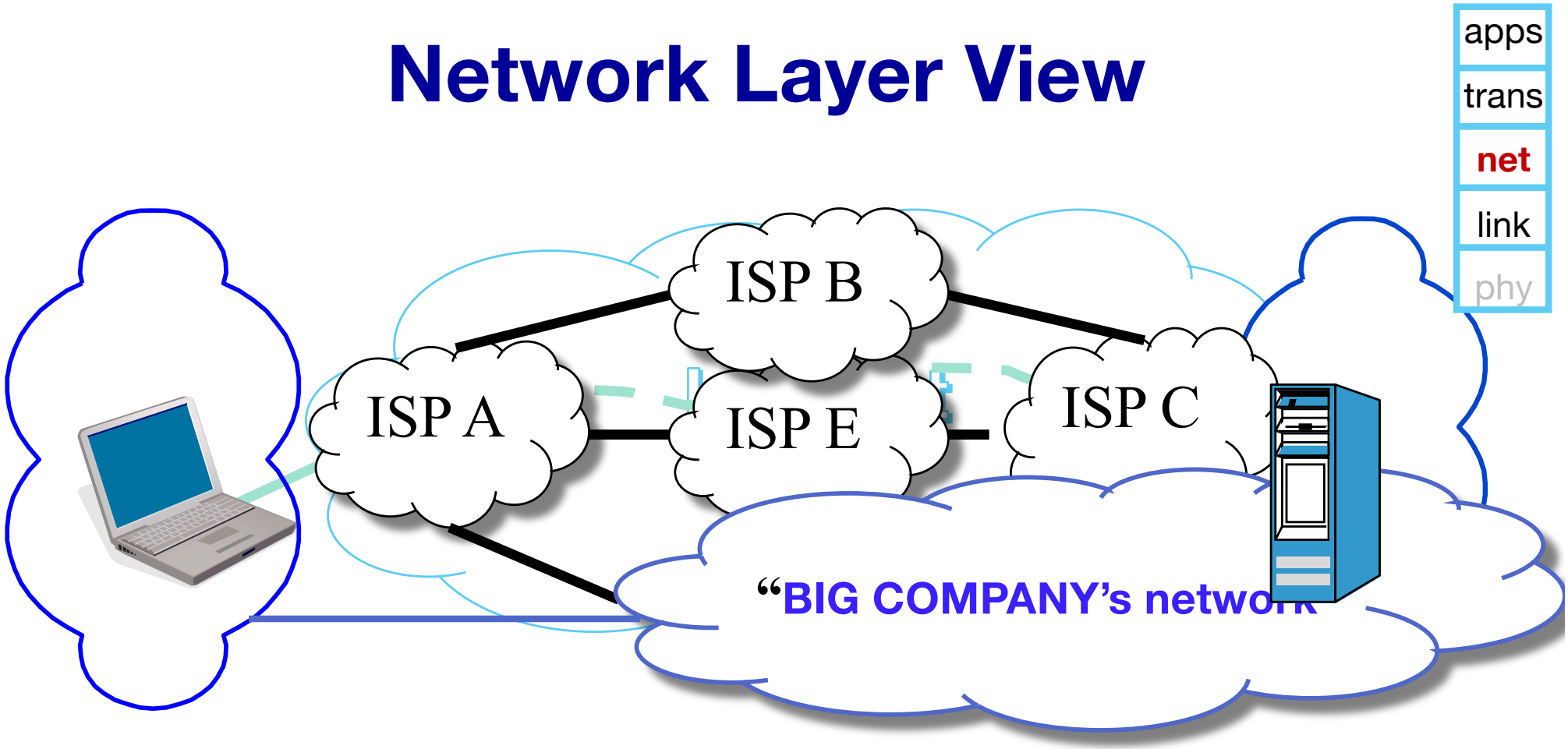
Transport View



- **A transport protocol**'s job: **delivering data** between the two communicating ends
  - *Don't know or care about which paths data may traverse through the network*
- Multiple transport protocols exist, each offers somewhat different functions (e.g. reliability, congestion control)

Actually, transport protocols don't do delivery → network protocol's job

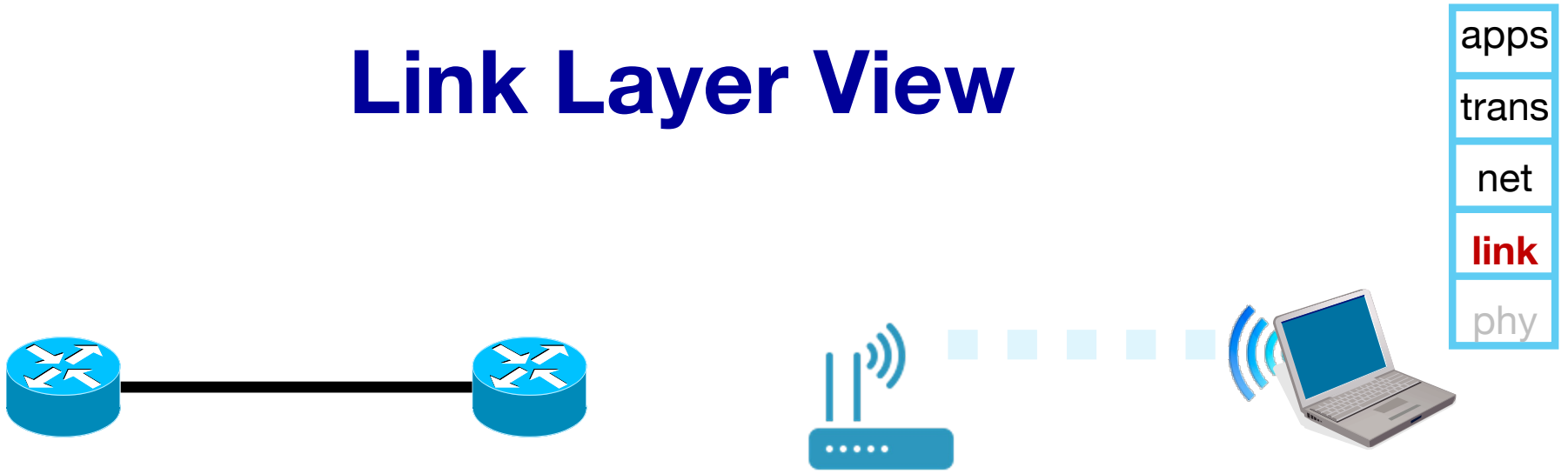
Network Layer View



- **network protocol**'s job: forward packets from source to destination host
- A really hard problem: the Internet is large, run by many different parties
  - connection from laptop to CNN.com:

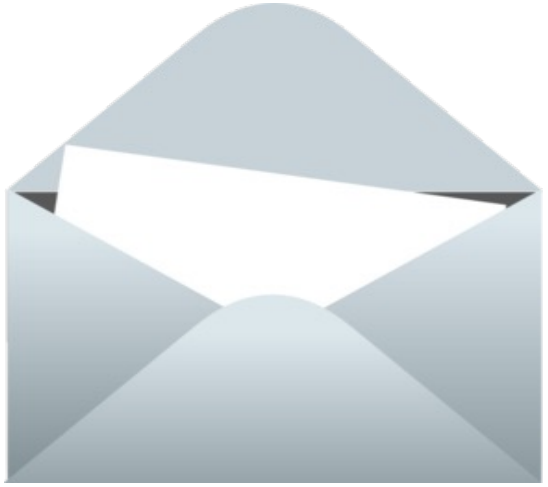
    WiFi → campus backbone → local ISP → other ISP → CNN website

Link Layer View



- **Link layer**'s job: Get a packet transmitted across some communication medium to **next hop**
- Different medium → different link layer protocol

What protocol “layer” really means



application

transport

network

link

physical

Link layer protocol

Network protocol

Transport protocol

Application protocol

Application data

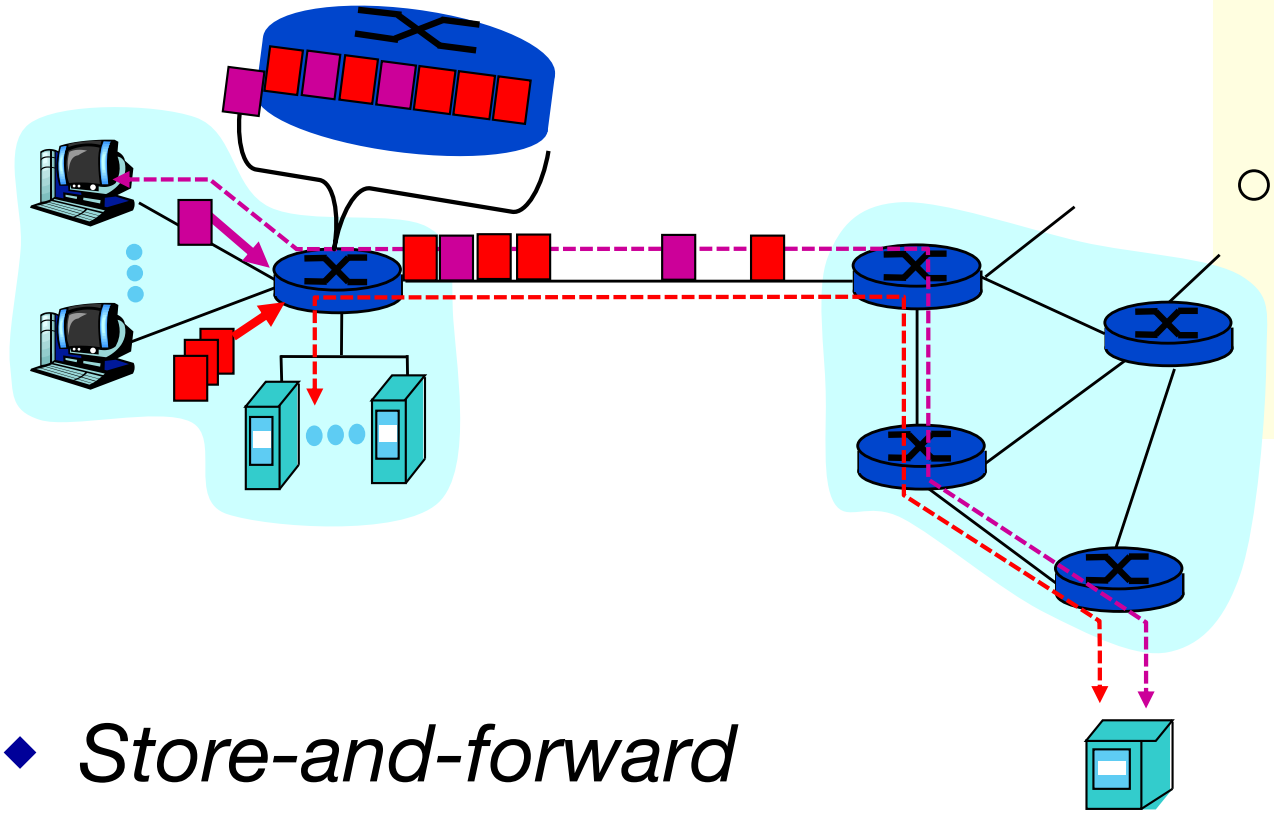
(Tentative) **Schedule of the Quarter**

Week:	1	2	3	4	5
Mon	**1/6** Course intro BW& delay	**1/13** HTTP	**1/20** Martin Luther King Jr. Day	**1/27** Transport protocols	**2/3** Congestion Control
Wed	**1/8** Socket programming, Web & HTTP	**1/15** DNS	**1/22** DNS	**1/29** TCP	**2/5** **Midterm**

	6	7	8	9	10	
Mon	**2/10** Security 101	**2/17** Presidents' Day	**2/24** Routing algorithms & protocols	**3/3** Routing in the Internet	**3/10** Hubs and switches	**3/21: Final Exam**
Wed	**2/12** Internet Protocol (IP)	**2/19** Addressing, NAT, IPv6	**2/26** Routing algorithms & protocols	**3/5** Link layer (Ethernet)	**3/12** Course review	

- The big yellow numbers indicate the chapter numbers in the textbook.

Packet Switching: *Statistical Multiplexing*



- Each node sends packets as soon as link available
- Receiver gets a full packet first, then forwards it towards the destination

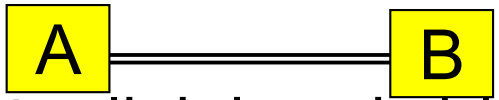
- *Store-and-forward*
- Packet switch can temporarily buffer up packets
  - Introduce *delay*
  - Packets get *dropped* when the queue is full

Network Performance

- 3 basic measurements
  - Throughput (bits/sec, Kbps=1000 bits/sec, Mbps)
  - Loss rate (% of packets lost)
  - Delay (sec, msec)

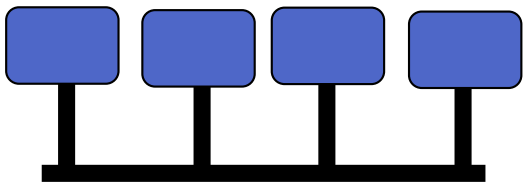
Throughput

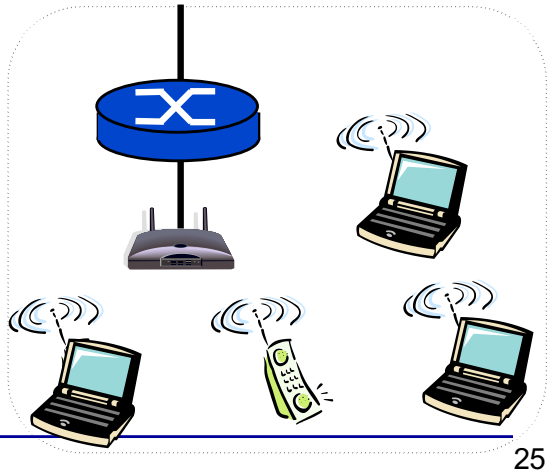
- over a single link: point-to-point



  - Pumping data into the pipe: throughput = link bandwidth
- Multi-access:

  a lot more difficult to measure, Why?

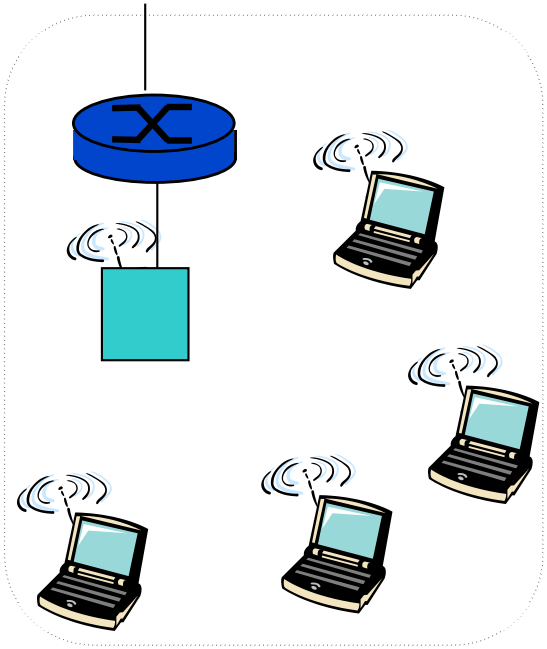




Packet Losses

- Wired links
  - Loss due to transmission errors
  - Loss due to congestion
- Wireless links
  - Limited transmission rate
  - Higher (than wire) bit error rate
  - Host mobility: high variance in the number of hosts sharing the same wireless channel

Do users know there are packet losses?
Do users' performance get affected by losses?



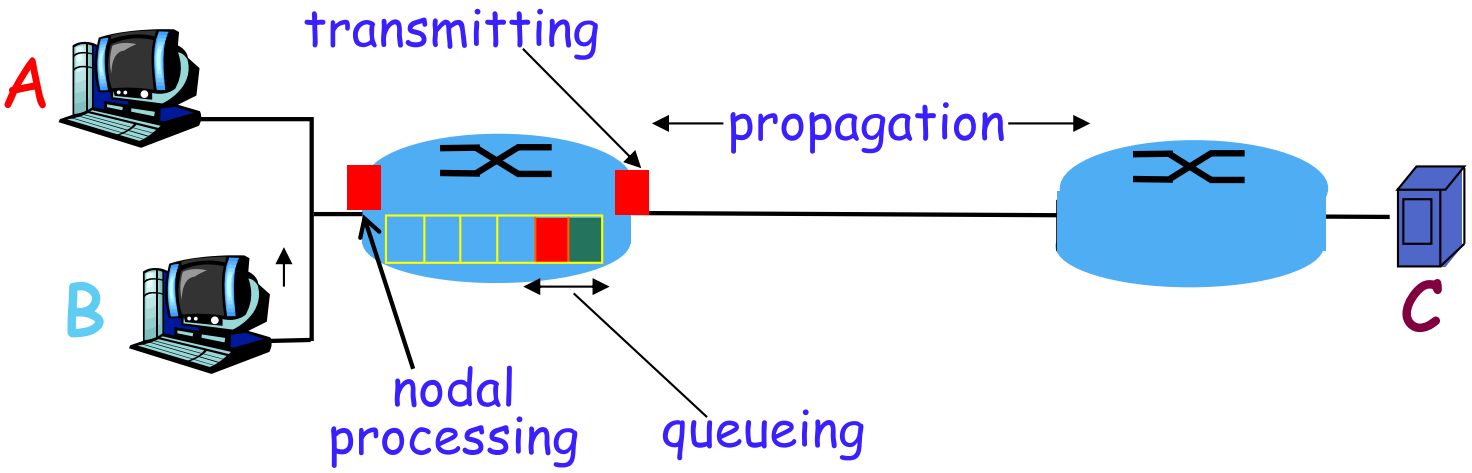
Delay in packet-switched networks

4 sources of delay at each hop

- node processing:
  - check bit errors
  - determine output link
- Queuing = #packets in queue X transmission time of each packet

Transmission = Length / rate
R = link bandwidth (bps)
L = packet length (bits)

Propagation = distance/sec
d = length of physical link
s = propagation speed in medium (~2x10^8 m/sec)

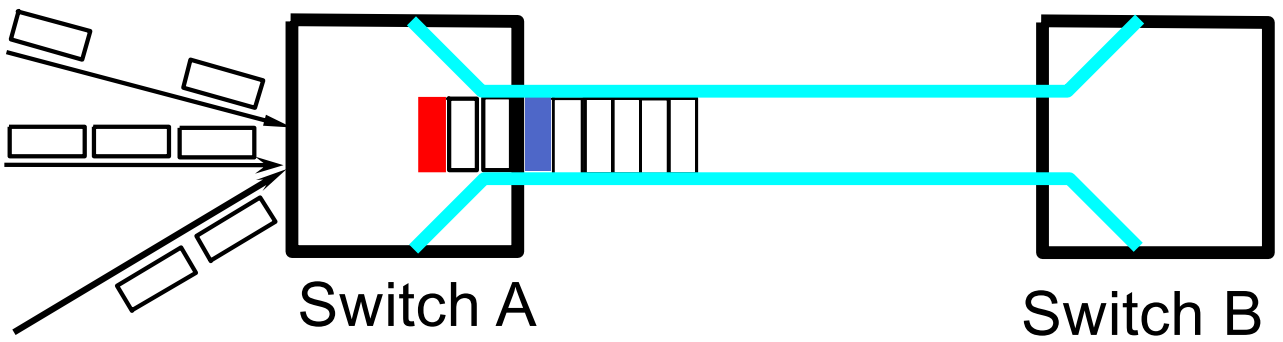


Example: calculating one hop delay

total delay (A—▮→B) = ?

- Queuing delay = ?
- transmission delay = ?
- Propagation delay = ?

link length = 100 km
Bandwidth= 1 Mbps
packet size= 1000 bits
(all pkts equal length)



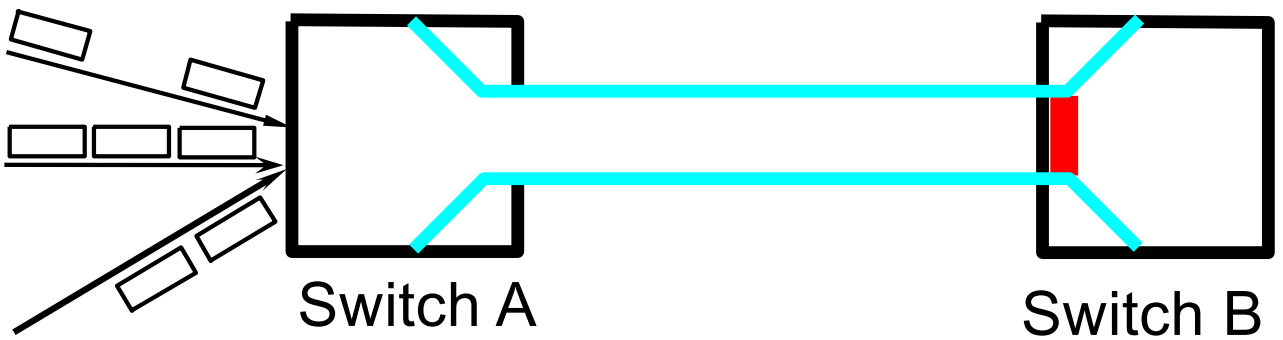
(2.0x10^8 meters/sec in a fiber)

Example: calculating one hop delay

total delay (A—▮→B) = $1ms \times 2 + 1ms + 0.5ms = 3.5ms$

- Queuing delay = **Waiting time for 2 pkts**
- transmission delay = $\frac{1000bits}{1000000bits/\sec}$ = **1 msec**
- Propagation delay = $\frac{100{,}000m}{2 \times 10^8 m/\sec}$ =**0.5 msec**

link length = 100 km
Bandwidth= 1 Mbps
packet size= 1000 bits
(all pkts equal length)

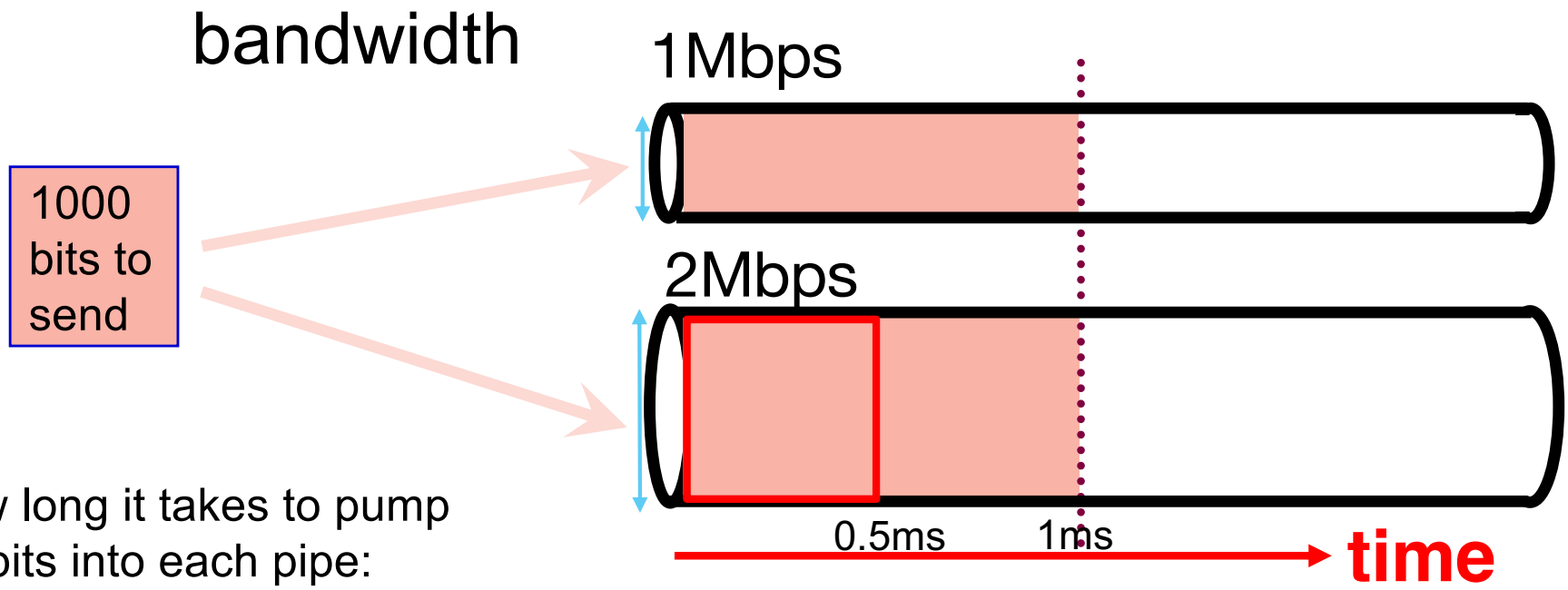


(2.0x10^8 meters/sec in a fiber)

Transmission vs. propagation delay

Transmission delay: L / R
R = link bandwidth (bit-per-second, bps)
L = packet length (bits)

Propagation: d / s
d = length of a physical link
s = signal's propagation speed in the medium (~2×10^8 meter/sec)



How long it takes to pump 1K bits into each pipe:

(1) $\frac{10^3 bits}{10^6\ bps}$ =1msec

(2) $\frac{10^3 bits}{2 \times 10^6\ bps}$ =0.5msec

What we covered today

- Internet: made of a huge number of hosts, routers, wired and wireless links
- Hosts: run application protocols to exchange data packets with each other
- Routers: run bunch of protocols to move all packets towards their destinations
- Why protocols are layered
- How to calculate packet delays as they move across a packet-switched network

Acknowledgment

- Slides adapted from S24 CS118 instructed by Prof. Lixia Zhang