CS118:
Computer Network Fundamentals

Lecture-1: introduction

CS118: explains how the Internet works

- Internet: a huge, complex system
- Divide-and-conquer
  - Figure out how many major parts, then learn one part at a time
- Your job:
  - Read textbook, think, collect a list of questions
  - Ask questions in class/office hours/via Canvas
  - Practice what you learn through homework and projects

Course workload and grading

- Bi-weekly homework assignments
- 2 programming projects,
  plus a warmup exercise
  0. Install VM, learn GIT (individual)
  1. Simple web server (individual)
  2. IP router (team of 2-3 people)
- In-class quizzes (3 total, online)
- Midterm and final exams (cheat sheets allowed)
- **Strict Grading Policy**
  - Homework: do it your self; no credit for late submission
  - Project: 20% credit reduction per late day
  - *No make-up exam*

Homework	20%
Programming Projects	25% (0/ 10/ 15)
In-class quiz	15%
Midterm	15%
Final exam	25%

We strongly encourage class participation
- Join class discussions on Piazza, ask question and help address issues raised by others
- Actively participate in recitation sessions
- Other means to improve the learning experience for all

Course assignment and due schedule

Midterm	In-class, Thursday May 4 (**Location TBD**)
Final	11:30AM-2:30PM Saturday June 10 (**Location TBD**)
Homework	**Release:** on Thursday of Week 1, 3, 5, 7; **Due:** 11:59pm Monday of week 3, 5, 7, 9.
Quizzes	In-class, at end of Thursday lecture of week 2, 4, 7.
Project 0	**Release**: Monday Apr 3, 2023 (Week 1) **Due:** 11:59pm Monday, Apr 10, 2023 (Week 2)
Project 1	**Release**: Monday Apr 10, 2023 (Week 2) **Due:** 11:59pm Friday, Apr 28, 2023 (Week 4) **Grading:** auto-grading script (sample tests will be provided to let everyone test their code before submission)
Project 2	**Release**: Monday, May 8, 2023 (Week 5) **Due:** 11:59pm Monday, Jun 5, 2023 (Week 10) **Grading:** auto-grading script (sample tests will be provided to let everyone test their code before submission)

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

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, except those provided by instructor/TAs
- Making your project code publicly available during or after this quarter
  - you must use private repository on either GitHub or GitLab

Hints for Getting Good Grade

- Read textbook before coming to each lecture
- *Ask questions*
  - One may earn extra credit for in-class participation
    - come to me after class, so that I record your name
- Get your work done early
  - Lecture slides uploaded to Canvas *by Sunday* each week
  - Get HWs and projects done ***before*** the deadline

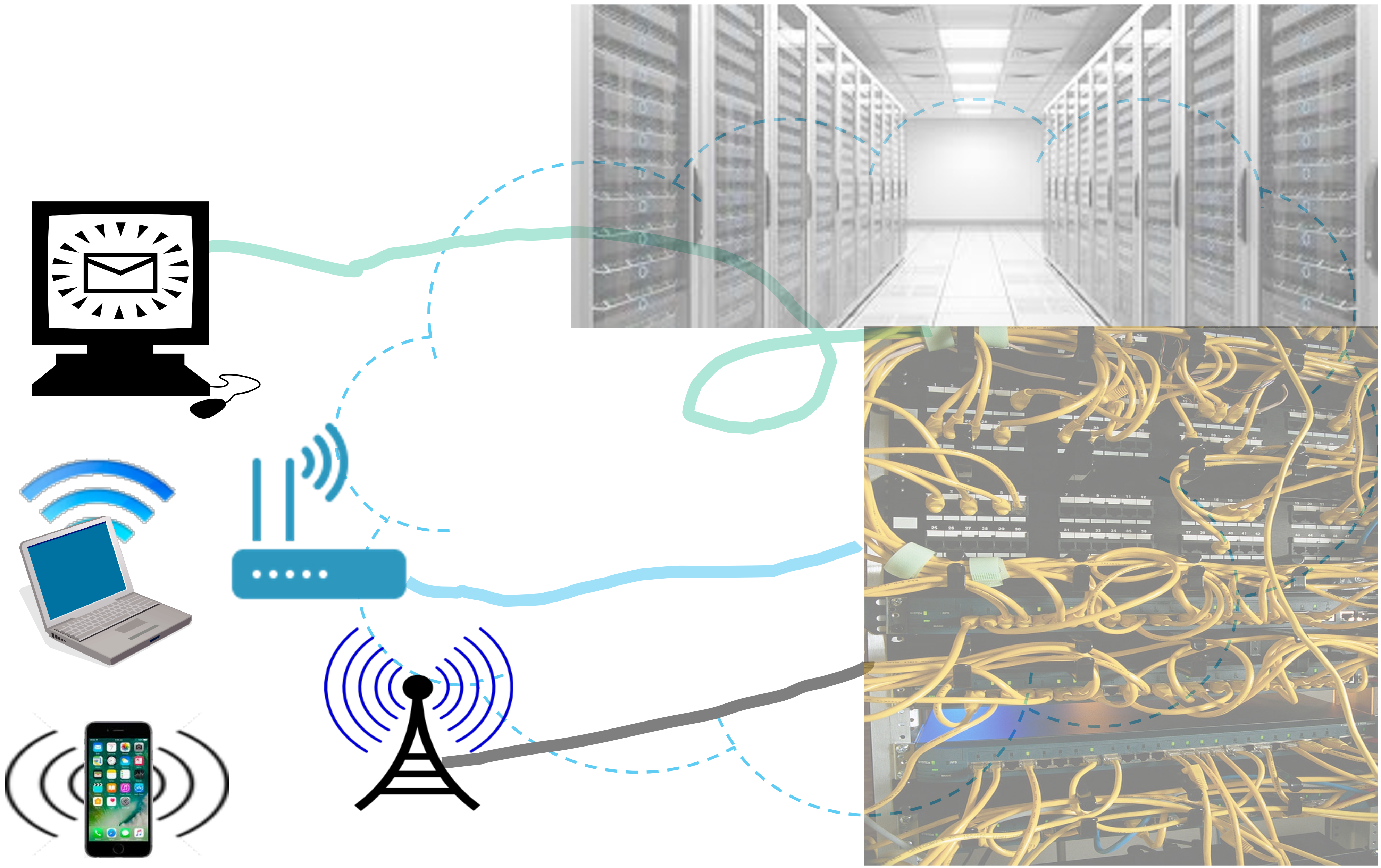
In addition, if anyone needs a recommendation letter later: make sure that I get some chance to know you

Let's get started

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

What is a Computer Network

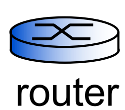
What is a Computer Network



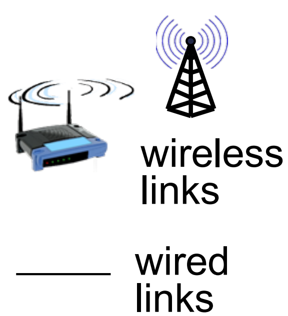

Transit
1:30min 120m
Hotel
40m
Grocery Store
30m
Restaurant
25m

Terminology

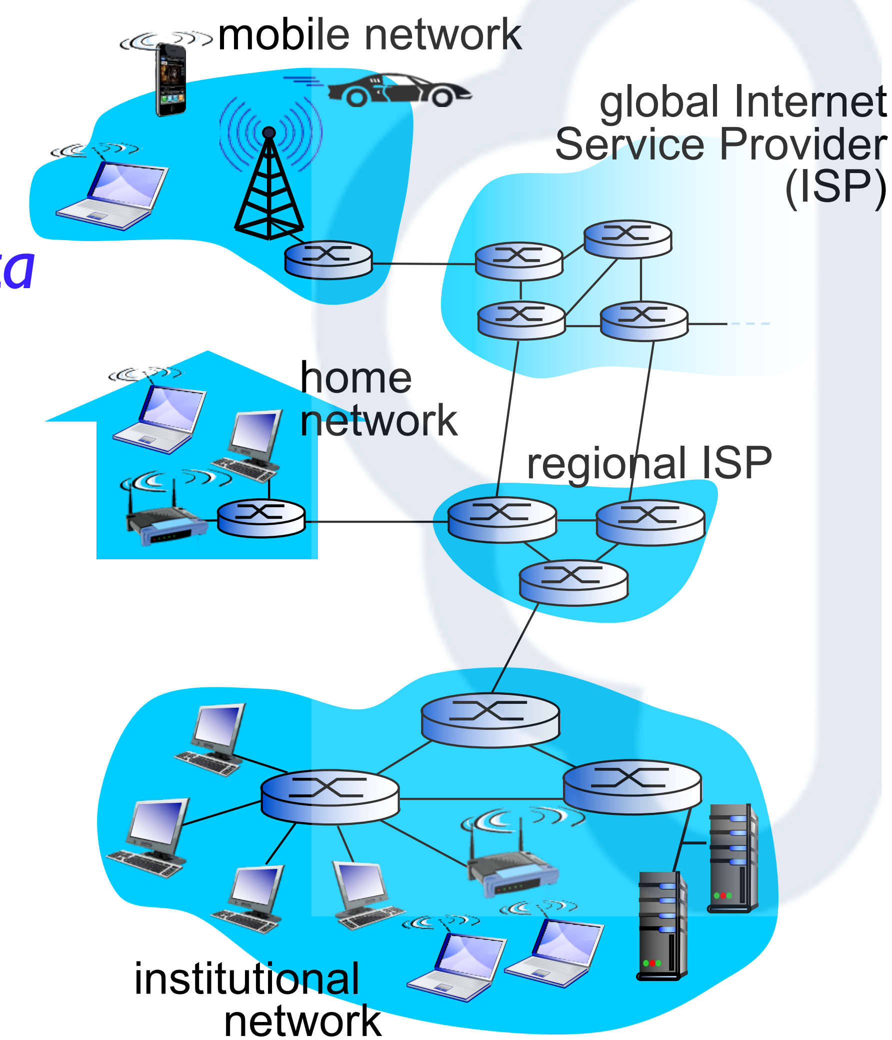
- millions of connected computing devices:
  - *hosts* = *end systems*
  - running *network apps*
  - Apps send/receive *data packets*



- *Routers* = *packet switches* inside network



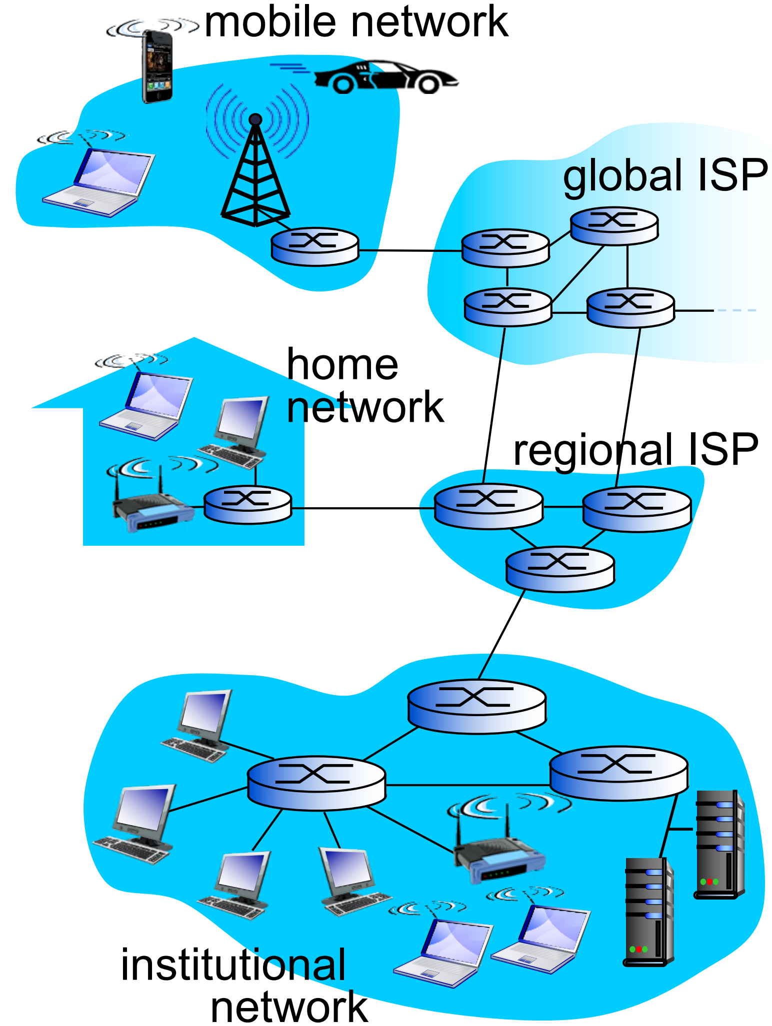
- *communication links*
  - fiber, copper, radio, satellite
  - transmission rate = *bandwidth (BW)*



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?

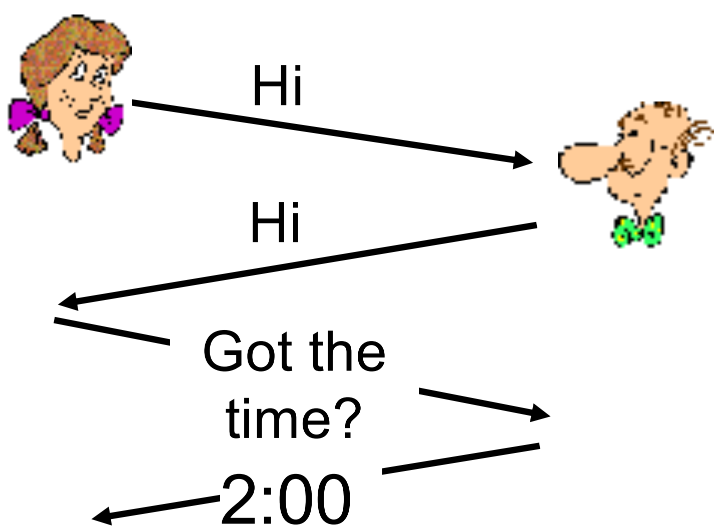
Traffic light protocol

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

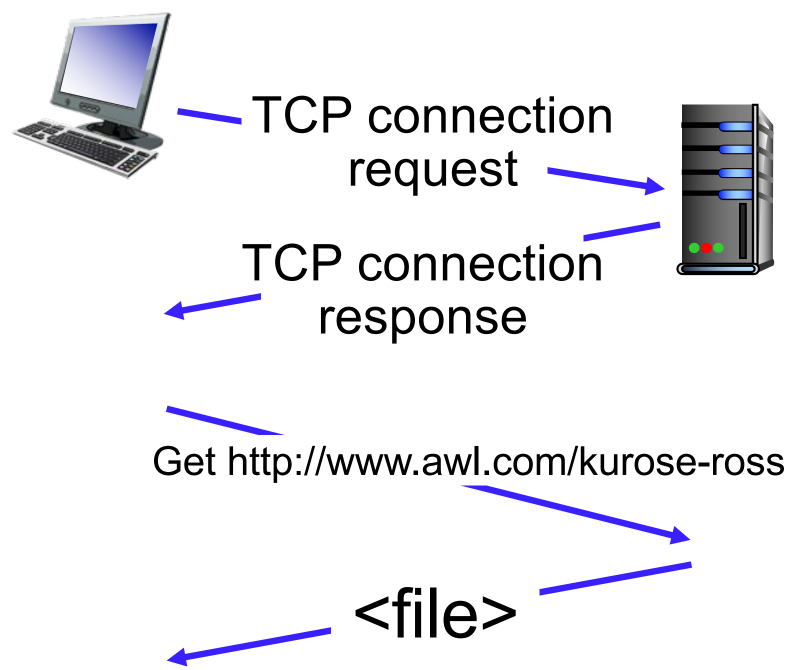
… specific messages sent

… specific actions taken when the messages received

human protocols:

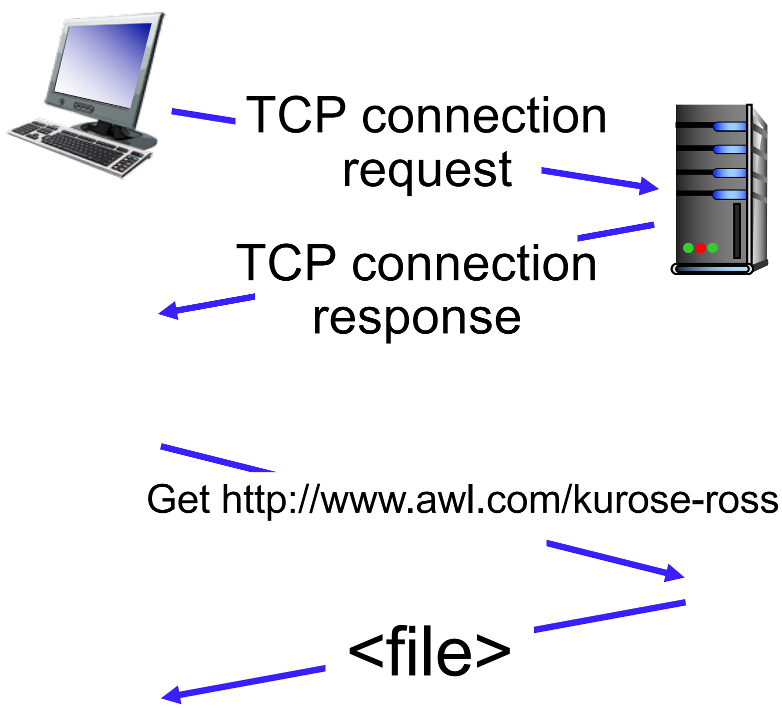


computer protocols:



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

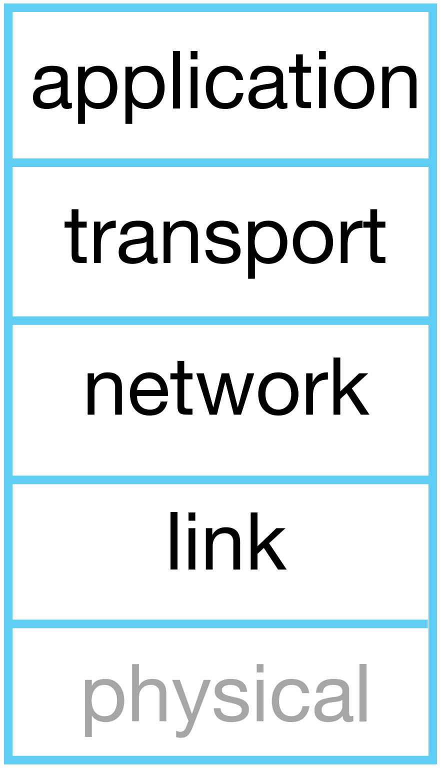
How to get the work done: divide and conquer

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"

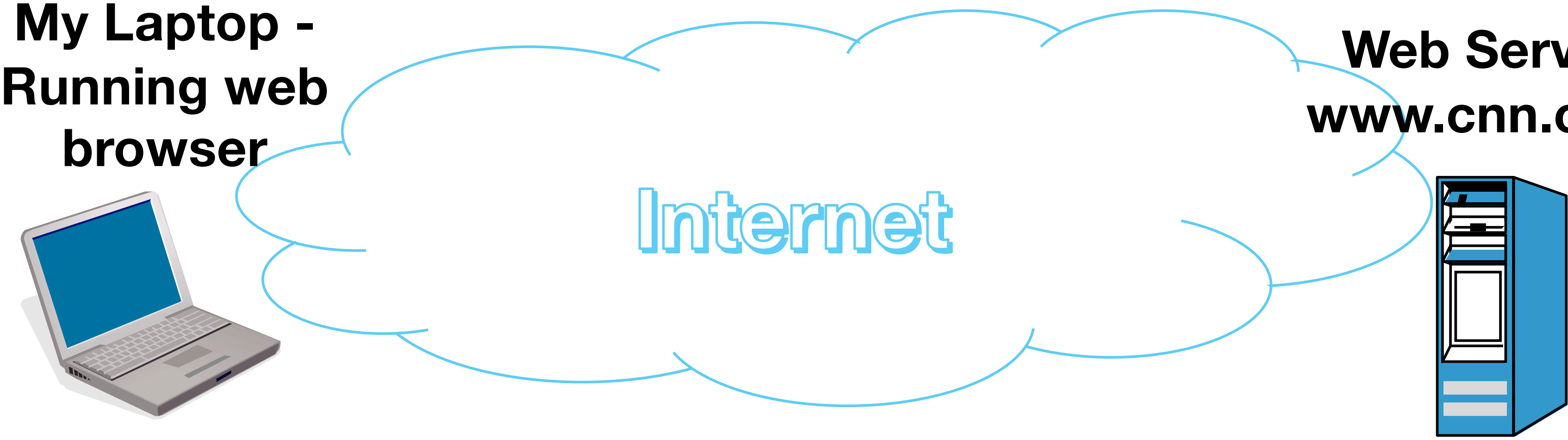


Application View

apps
trans
net
link
phy

My Laptop - Running web browser

Web Server www.cnn.com



CLIENT
(Chrome, Safari, Firefox, ...)

SERVER
(Apache, GWS, ...)

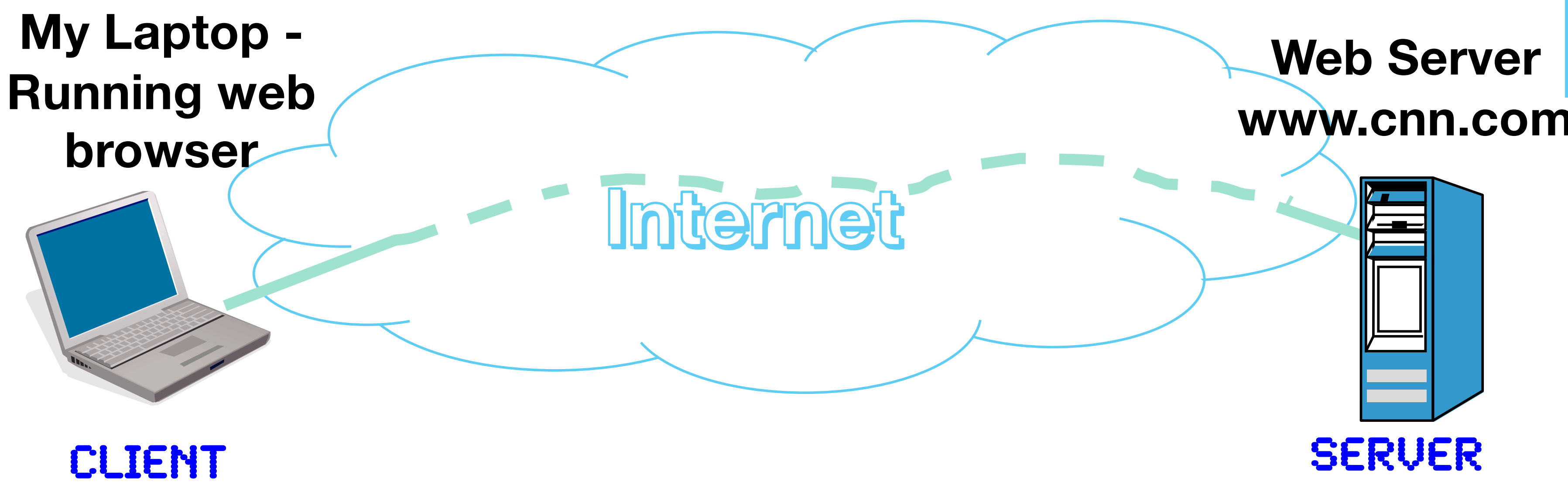
These are *application programs*

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

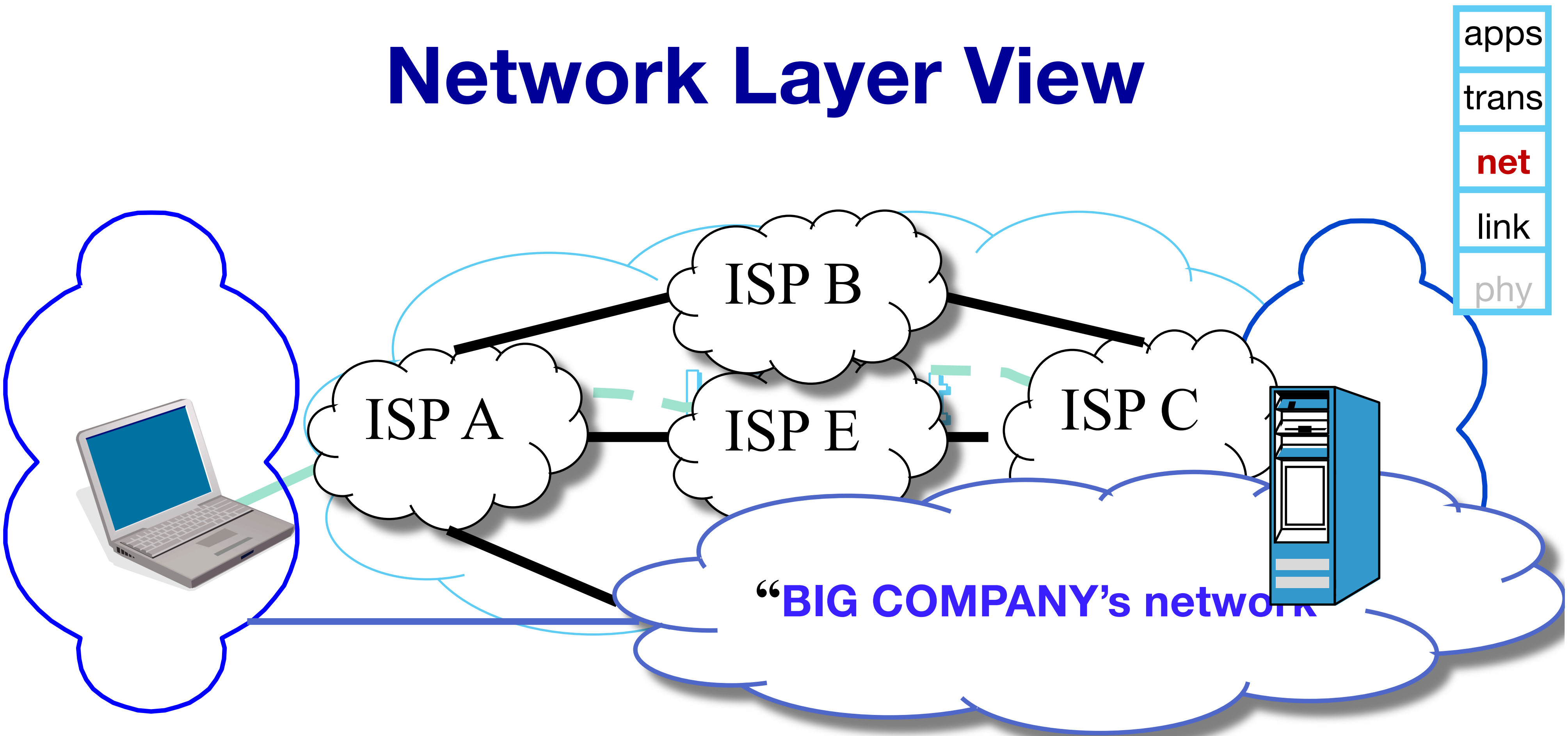


- **A transport protocol**'s job: **delivering data** between the two communicating ends
  - *Don’t know or care about which paths data may traverse through the network*

- 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:

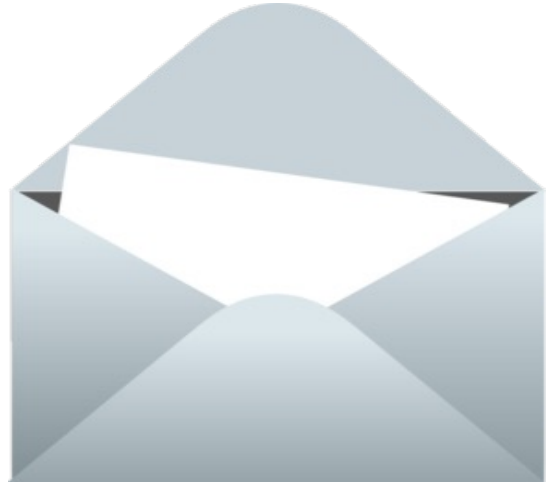
    WiFi → campus backbone → local ISP → other ISP → CNN website

Link Layer View

apps
trans
net
link
phy

- **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

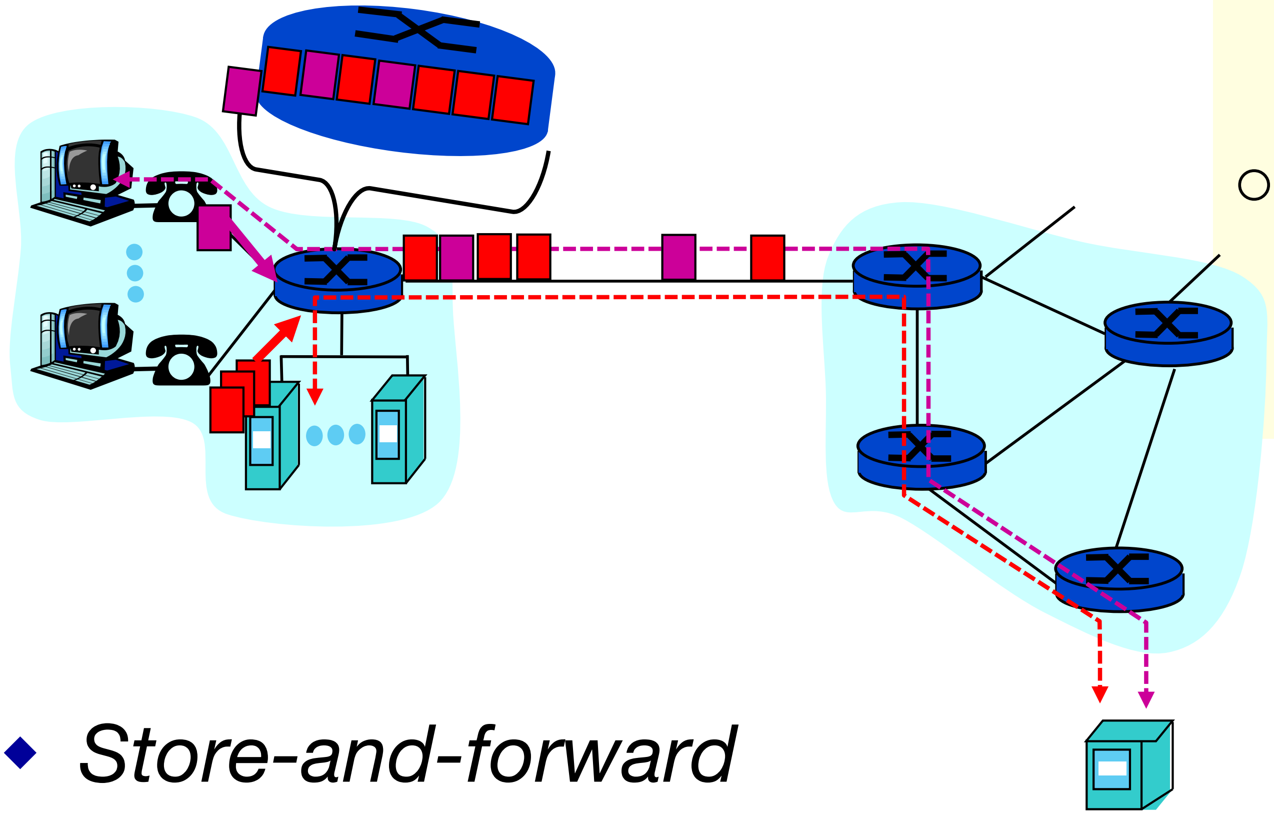
(Tentative) **Schedule of the Quarter**

Week:	1	2	3	4	5
Tue	**4/4** Course intro BW& delay	**4/11** HTTP	**4/18** DNS	**4/25** TCP	**5/2** QUIC
Thu	**4/6** Socket programming, Web & HTTP	**4/13** DNS	**4/20** Transport protocols	**4/27** Congestion Control	**5/4** **Midterm**

	6	7	8	9	10	
Tue	**5/9** Internet Protocol (IP)	**5/16** Routing algorithms & protocols	**5/23** Routing in the Internet	**5/30** Wireless and mobility	**6/6** Security 101	**6/10: Final Exam**
Thu	**5/11** Addressing, NAT, IPv6	**2/18** Routing algorithms & protocols	**5/25** Link layer (Ethernet)	**6/1** Hubs and switches	**6/8** 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

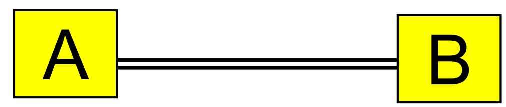
Network Performance

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

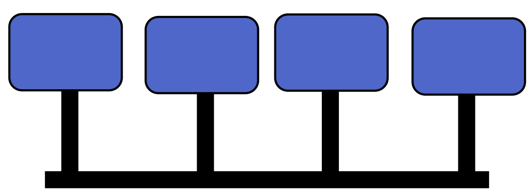
This is how network people measure performance.
How do you (user) judge?

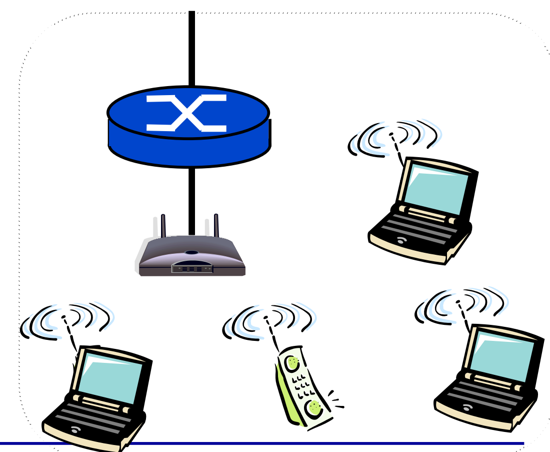
Throughput

- over a single link: point-to-point



  - Pumping data into the pipe: throughput = link bandwidth
- Multi-access:
  a lot more difficult to measure, Why?



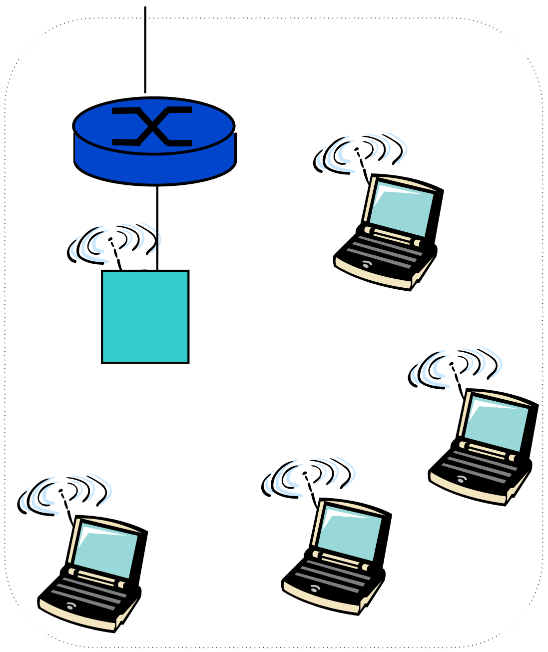


Packet Losses

- 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

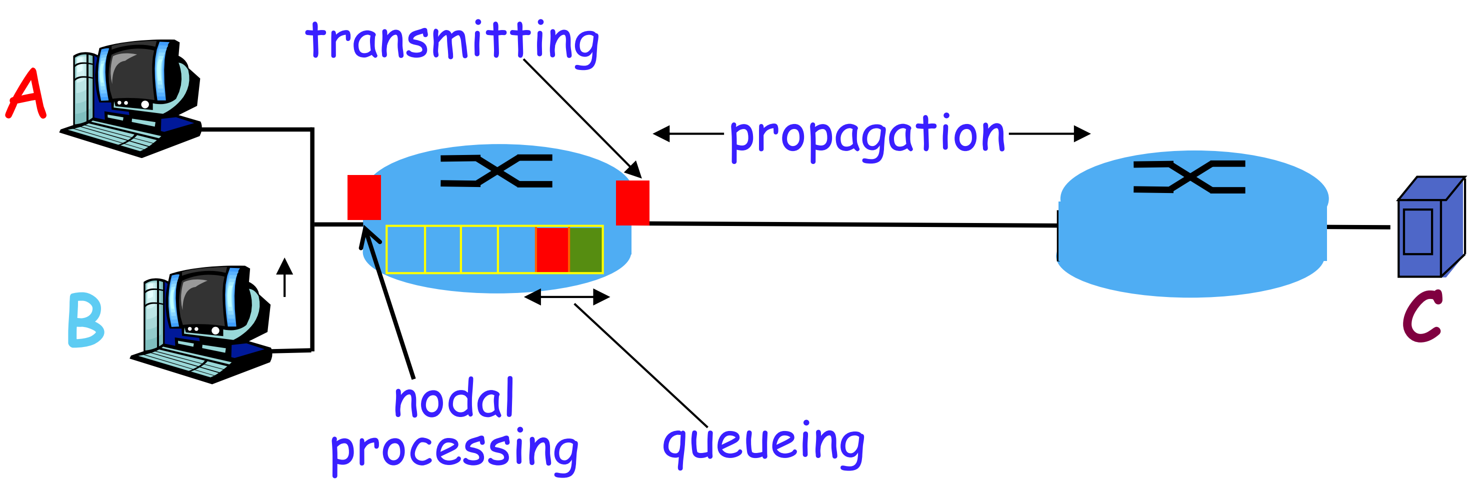
R = link bandwidth (bps)

L = packet length (bits)

Propagation = distance/sec

d = length of physical link

s = propagation speed in medium (~2×10^8 m/sec)

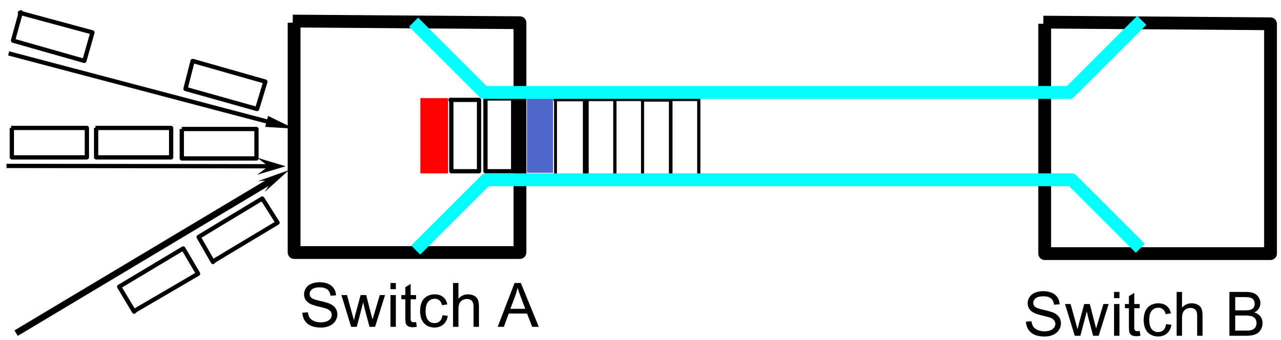


Example: calculating one hop delay

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)



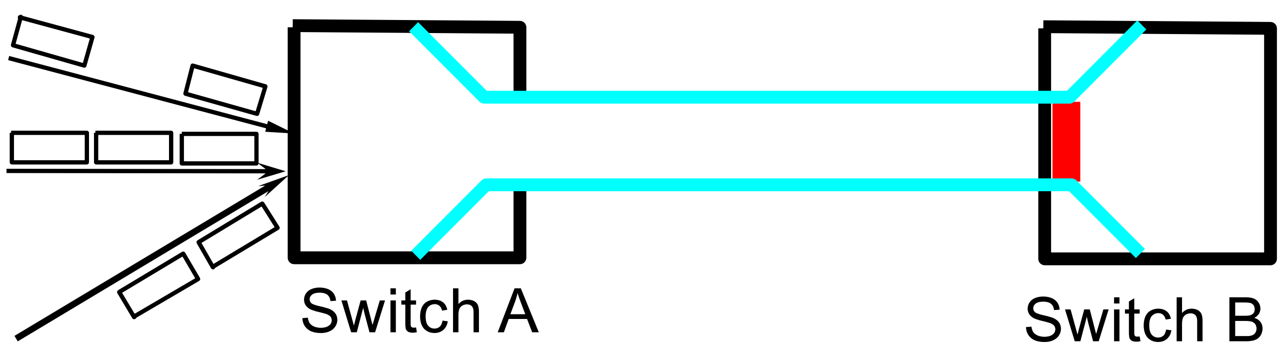
(2.0x10^8 meters/sec in a fiber)

Example: calculating one hop delay

total delay (A→B) = $\boxed{1ms \times 2 + 1ms + 0.5ms = 3.5ms}$

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

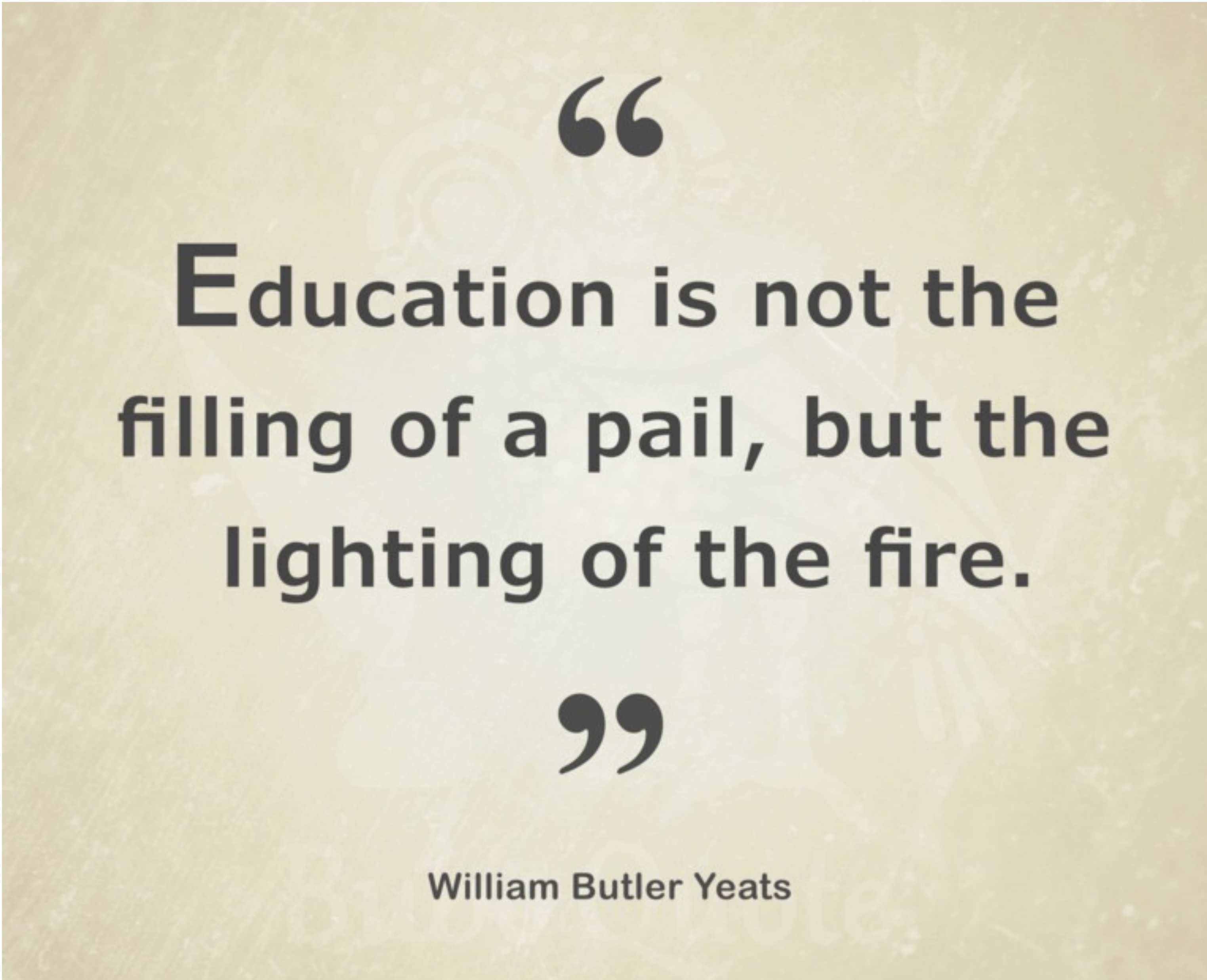
link length = 100 km
Bandwidth= 1 Mbps
packet size= 1000 bits
(all pkts equal length)



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

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


"
Education is not the filling of a pail, but the lighting of the fire.
"
William Butler Yeats