# CS 352
# Internet Technology

Lecture 1: Introduction

http://www.cs.rutgers.edu/~sn624/352-S22

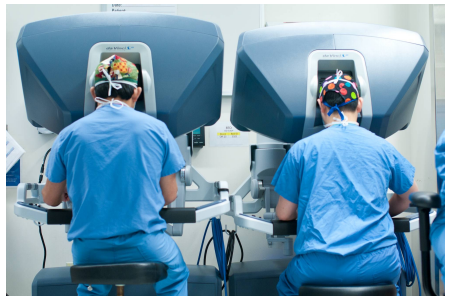
Srinivas Narayana

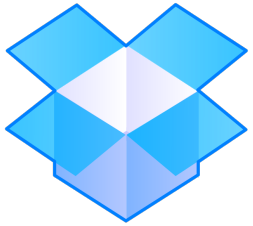
RUTGERS
UNIVERSITY | NEW BRUNSWICK

# The Internet is an exciting place







# The Internet has transformed everything

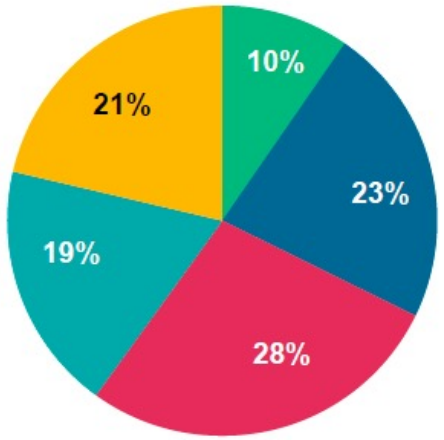
- How we communicate with other humans
- How we learn what's going on in the world
- How we learn and acquire knowledge
- How we transact and do business
- How we entertain ourselves
- How espionage and war is conducted

- In short how we live, especially through a pandemic.

# Internet growth

**1995**
**35MM+ Internet Users**
*0.6% Population Penetration*


**2014**
**2.8B Internet Users**
*39% Population Penetration*




USA China Asia (ex. China) Europe Rest of World

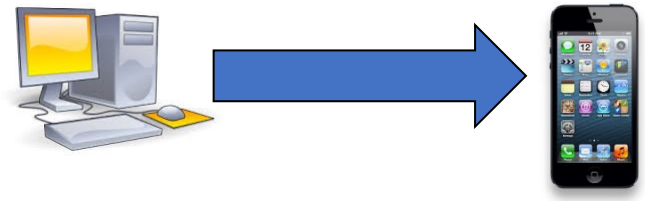
2020

4.8B users

(61% of the world's population)

https://www.broadbandsearch.net/blog/internet-statistics

# Evolution of Internet applications



1992
ftp
Web
email

1996
chat
Games
IM
Yahoo!

2000
news
Blog
Search
amazon

2004
Music
itunes
Games
search
Google

2008
Wikipedia
Craiglist
Youtube

2010-2020

2020--

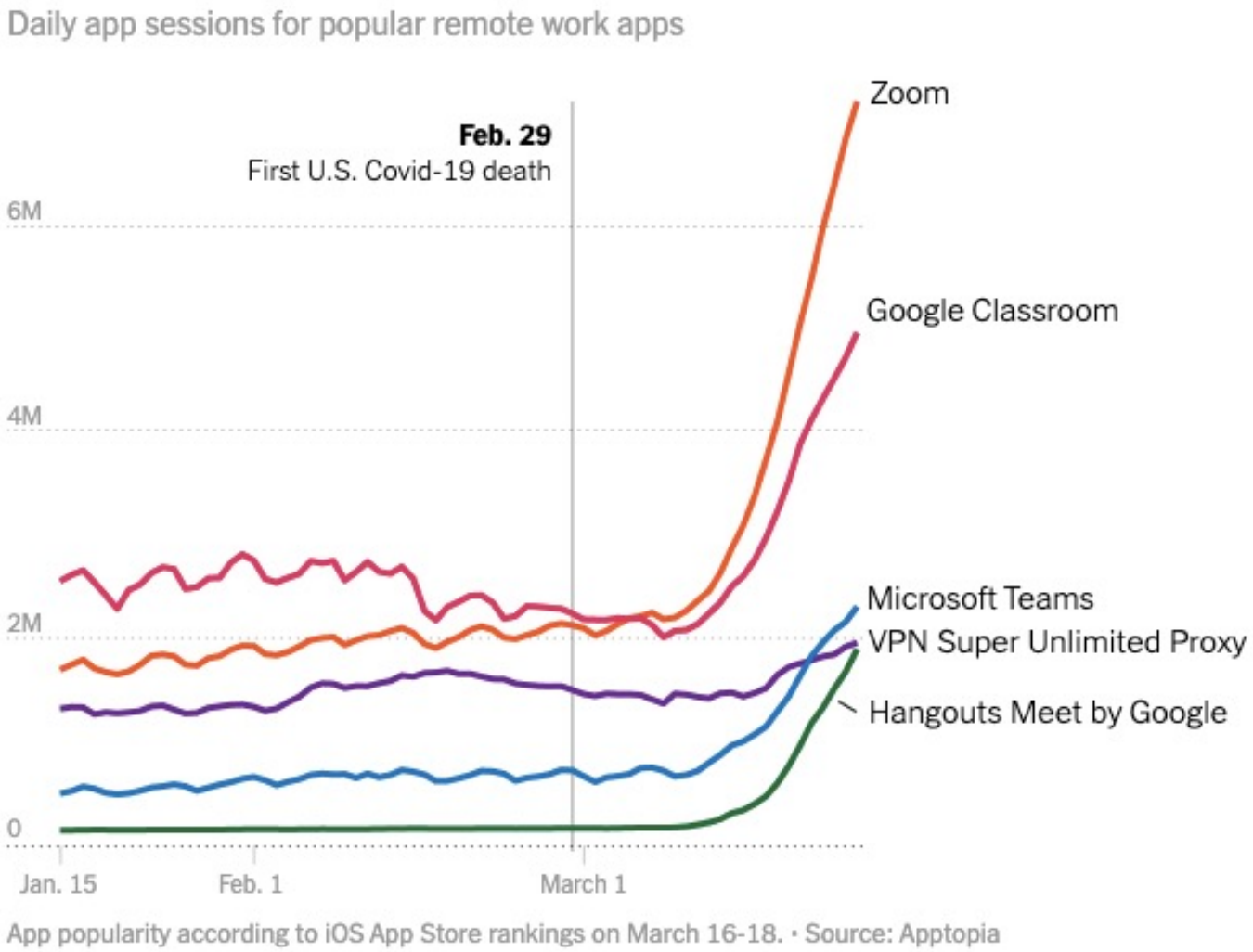
Text-heavy

Multimodal media

Augment physical world

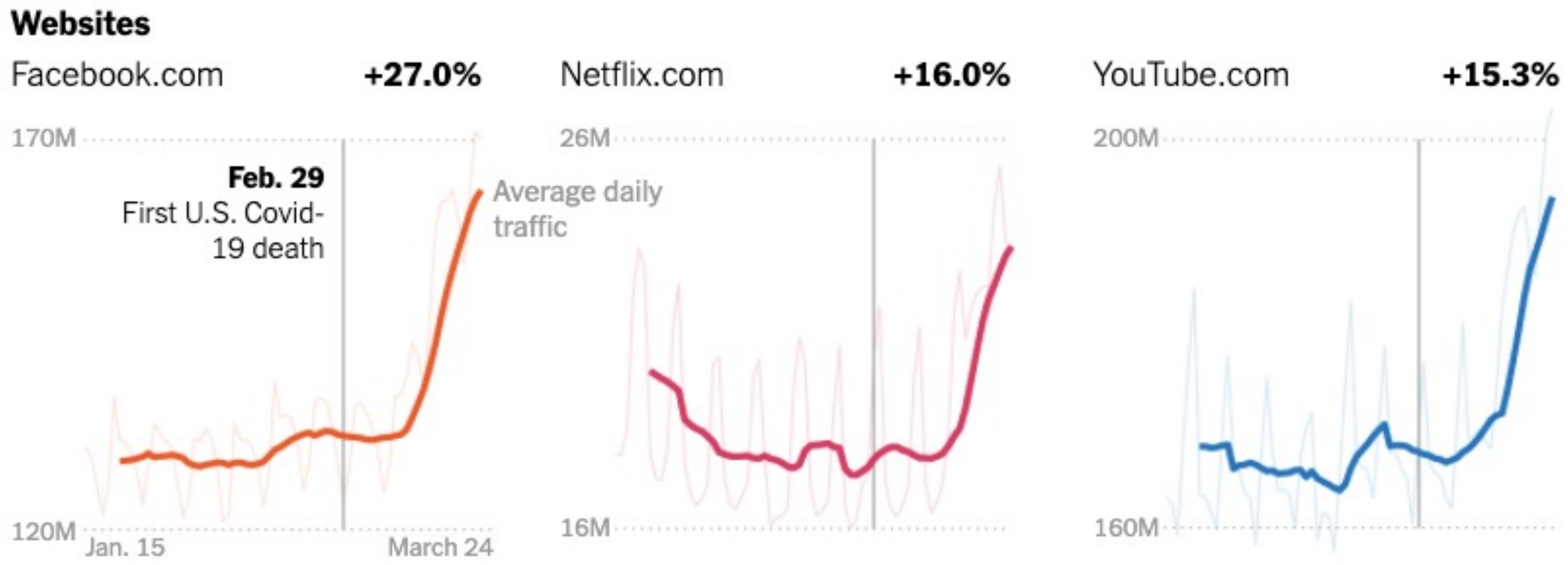
Replace phy world

# We relied on the Internet to work



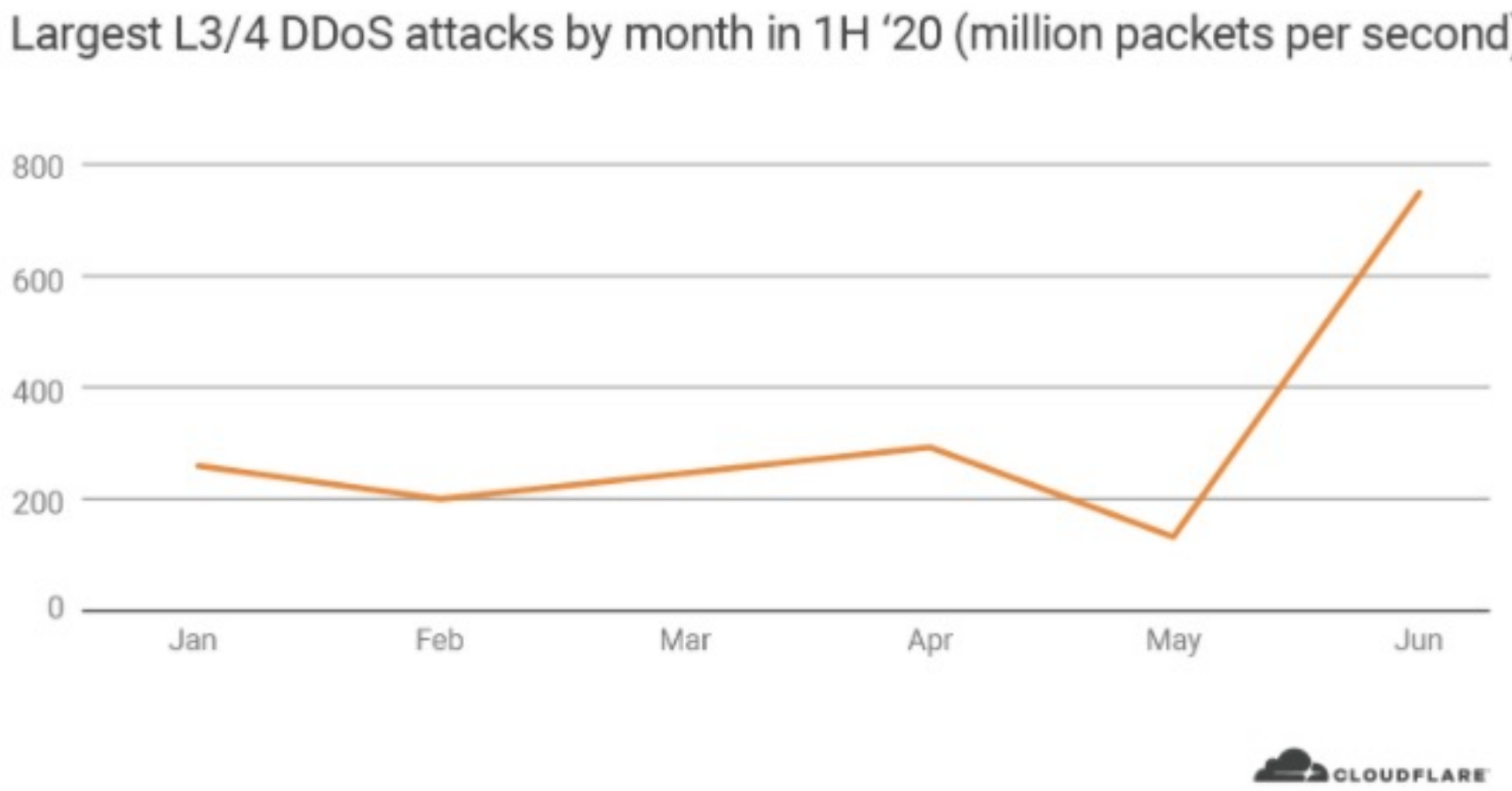

Data shows number of daily sessions in the US over a period in 2020. Source: nytimes

# We relied on the Internet to “play”!




Data shows number of daily sessions in the US over a period in 2020. Source: nytimes

# Threats on the Internet are growing, too




Source: CloudFlare blog

# Internet Technology: This course

- The study of how the Internet is designed
- The Internet is an example of a computer network

Technology is cool.

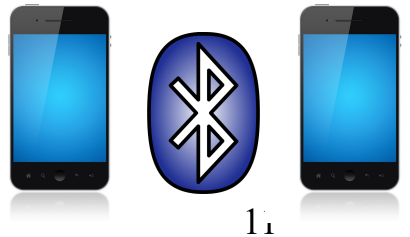
Learn fundamental principles that underlie Internet technology.

So that you can use and build technology for fun, altruism, and profit.

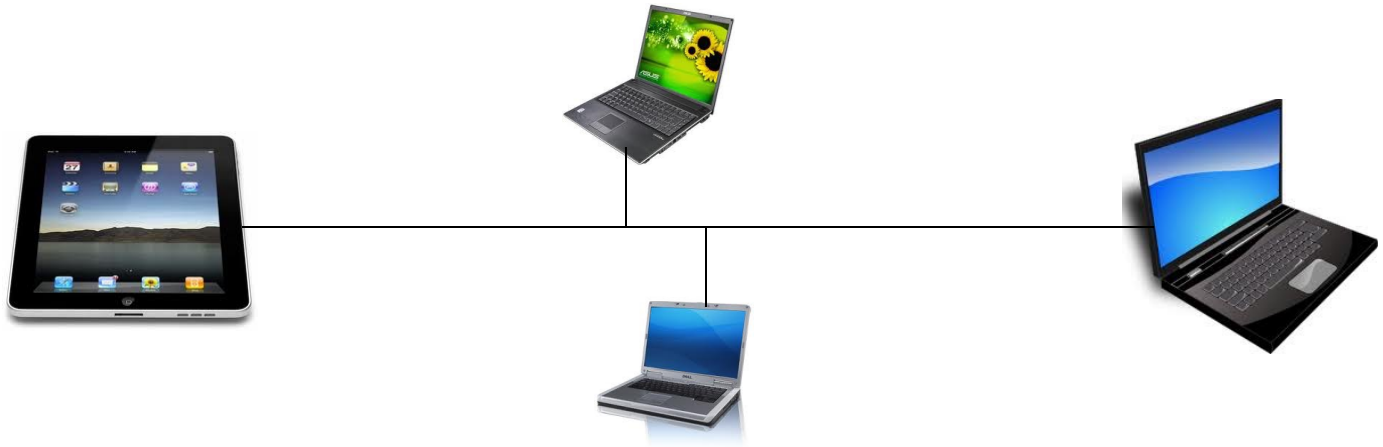
# What is a network?

- Carrier of information between two or more entities
- Entities may be hosts/endpoints (used interchangeably)
  - your laptop, cell phone, etc.
- Entities may also be devices in the middle of the network
  - For example, your WiFi router

- The interconnection between entities is any physical medium capable of carrying information: we call physical media links
  - Wireless links: cellular 4G/5G, wifi 802.11, bluetooth, satellite
  - Wired links: copper wire, lasers over optic fiber, coax cables

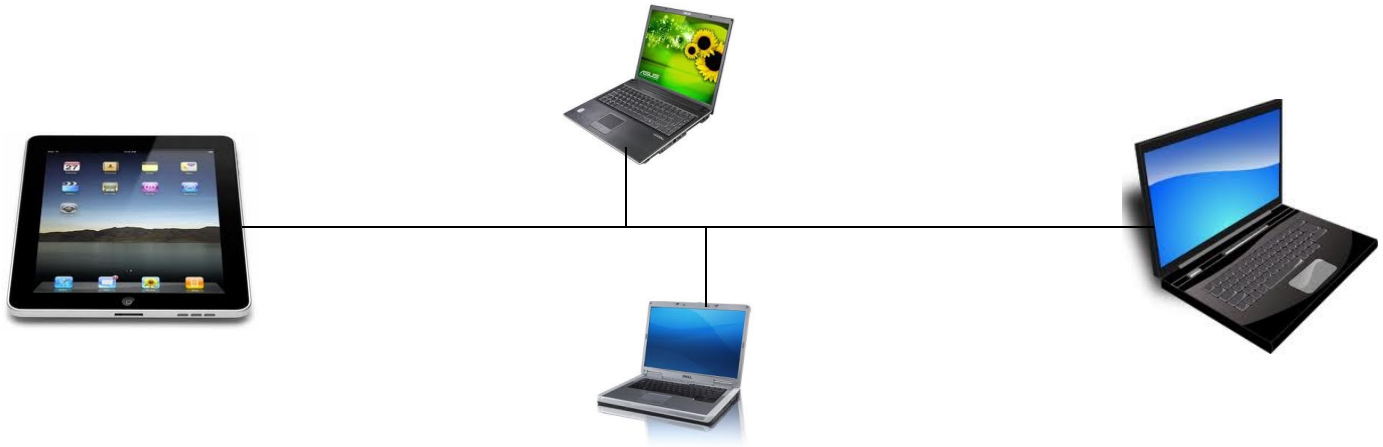


# A single link multiple access network



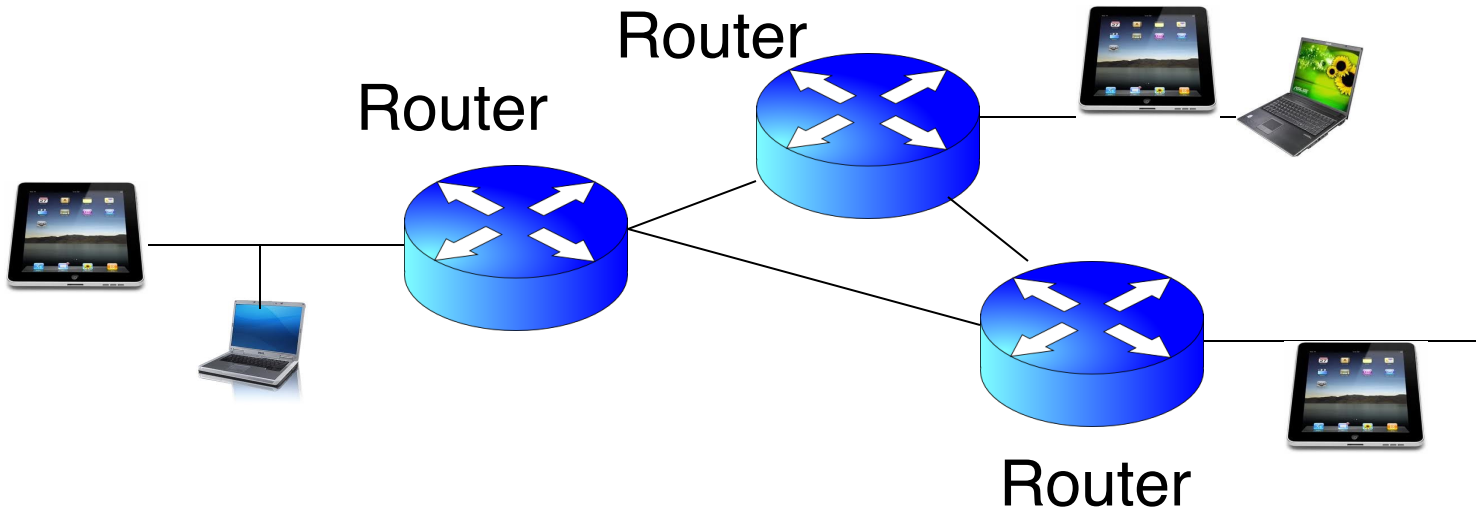
- Send bits of data in packets or frames
- How do we differentiate among many receivers?
- Every endpoint as a link level address: also called a *MAC* address
- Packets have a destination address on them
- However, can't have every computer in the world on the same link!
  - Physical limits on power / distance over which info travels over a single link

# A single link multiple access network



- Even on a single link, you need to worry about a few things:
- Converting digital data to physical signals over the medium (encode/decode)
- How do we decide who speaks? (medium access control problem)
- Detecting and correcting errors

# A multi-link network




- Connect multiple links via routers
- Need to figure out how to move packets from one host to another host, e.g., how to reach google.com from your laptop
- Known as the routing problem
- Key Q: How should packets be moved from A to reach B?

# In general, networks give no guarantees

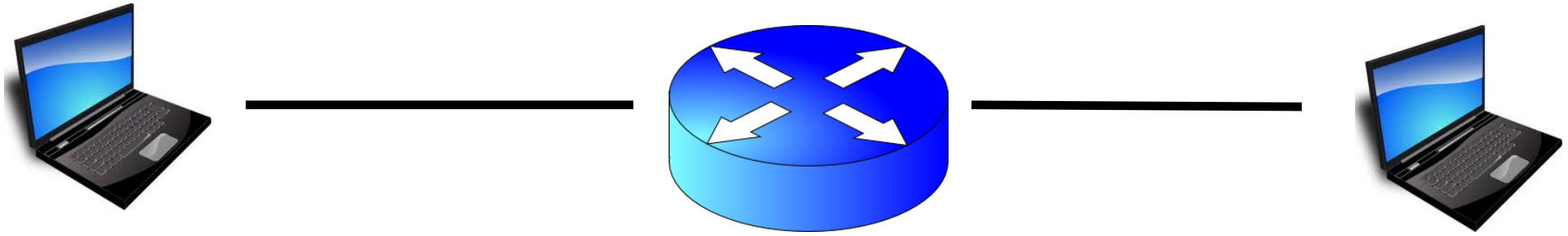
- Packets may be lost, corrupted, reordered, on the way to the destination
  - Best effort delivery

- Advantage: The network becomes very simple to build
  - Don’t have to make it reliable
  - Don’t need to implement any performance guarantees
  - Don’t need to maintain packet ordering
  - Almost any medium can deliver individual packets
    - Example: RFC 1149: “IP Datagrams over Avian Carriers”

- The early Internet thrived since (transient) disruptions are okay

# Guarantees for applications

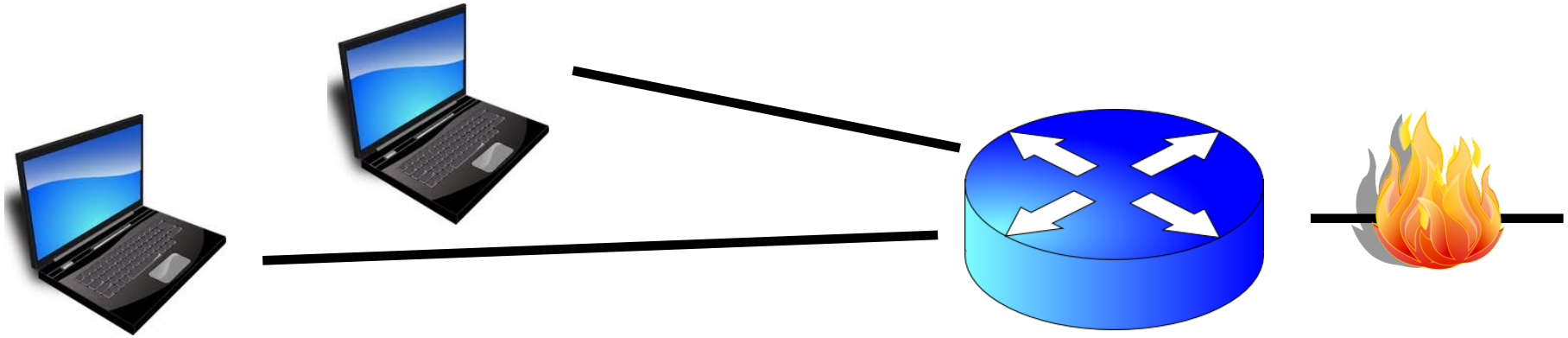
- How should endpoints provide guarantees to applications?



- Transport software on the endpoint oversees implementing guarantees on top of an unreliable network
- Need to solve the reliable data delivery problem
- For some applications, also need ordered delivery

# Sending data into a multi-link network

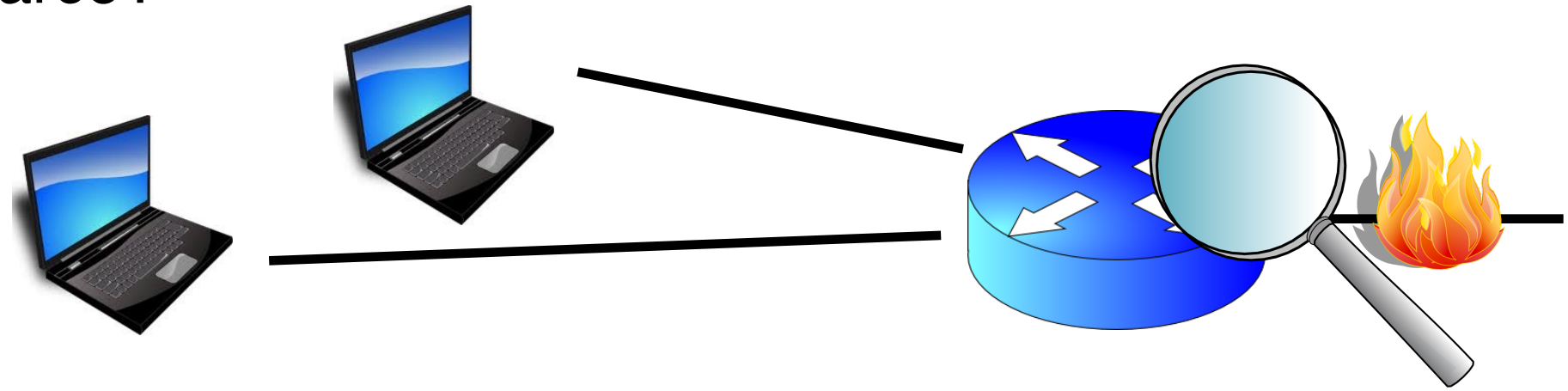
- How quickly should endpoints send data into a network?



- Known as the congestion control problem
- Congestion control algorithms at source endpoints react to remote network congestion. Part of the transport sw/hw stack.
- Key question: How to vary the sending rate based on network signals?

# Sending data into a multi-link network

- How should a router transmit packets when network resources are scarce?



- Known as the packet scheduling problem
- Key question: which packet to transmit over a constrained network link, and when?
  - Related: the buffer management problem

# Components of a network: Summary

- Link
  - Communication links for transmission
- Host/Endpoint
  - Computer running applications of end user
- Router
  - Computer for routing packets from input link to another output link
- Network
  - A group of hosts, links, routers capable of sending packets among its members

# Course Logistics

# About us

- Faculty Instructor: Srinivas Narayana
  - http://www.cs.rutgers.edu/~sn624
  - sn624@rutgers.edu
  - Office hours on Zoom (link on Canvas). Wed 9:30 – 10 am ET and Thu 2 - 3 pm ET
  - Lectures on Tue and Fri 8:30 – 9:50 am ET
- TAs and Recitations:
  - Section 5: Parvathi Mahesh pm850@scarletmail.rutgers.edu
  - Section 6: Chang Chen cc1547@cs.rutgers.edu
- Post questions on Piazza
- http://www.cs.rutgers.edu/~sn624/352-S22/

# Class philosophy

- We want you to learn and to be successful
- Ask questions on Piazza
- Attend office hours and recitations regularly to discuss material
- In summary, be proactive. Interact with us and with your fellow students and support each other
- Full video lectures from last offering (spr21) available

# Grading

- 32% programming projects
- 28% problem sets
- 18% mid-terms (2 * 9% each)
- 12% final exam
- 10% lecture questions

- Schedule of projects, problem sets, exams, etc. available at https://www.cs.rutgers.edu/~sn624/352-S22/syllabus.html

- This course uses absolute grading. There is no curve

# Programming projects (32%)

- Five programming projects
- Warmup/Socket programming intro (4 points)
- HTTP programming (7 points)
- Asynchronous sockets and load balancing (7 points)
- TCP analysis and configuration (7 points)
- IP network configuration (7 points)

- Tentative due dates 2/02, 2/16, 3/09, 3/30, and 4/20
  - Submit by 8 pm Eastern Time

# Programming projects (32%)

- Work in the same group of two students throughout semester
  - Only change groups under extenuating circumstances, at the discretion of the instructor
- Program and short write-up with responses required
- Background needed to get started
  - Python (211/214 level)
    - Get comfortable using data structures (tuples, arrays, dictionaries)
  - Unix (login, navigating folders, permissions, etc.)
- Use ilab machines or VMs (links provided) to run and test
- Hand projects in on Canvas

# Programming projects (32%)

- Please follow all project instructions carefully and exactly

- You will lose significant points if
  - We are unable to run your code
  - Your information (e.g., team member names and netids) is incorrect or incomplete
  - We do not receive your submission in a timely fashion

# Problem sets (28%)

- 6 problem sets
- Work individually
- Hand in a PDF file with solutions on Canvas
- Due dates: 1/26, 2/09, 3/02, 3/23, 4/13, and 4/27
- 8 pm Eastern Time

# Collaboration and Integrity policies

- Intellectual collaboration is welcome and encouraged
- Do
  - Ask questions on Piazza
  - Discuss projects and problem sets with each other
  - Read references (textbooks, Internet tutorials) widely
  - Acknowledge each other and all the references in problem sets & project reports
- Each problem set & project has a prompt on collaboration
  - Include who you talked to, references (including on the web) you consulted
  - Be as accurate and complete as possible

# Collaboration and Integrity policies

- All your written (coded) work must be your (team's) own
  - Understand the problem deeply and produce your own solutions
- Do not
  - blindly lift or incorporate other solutions
  - look at other people's code or solutions
  - copy code from the web (e.g., other people's GitHub projects)
  - post problem sets or projects (questions or solutions) on GitHub, Chegg, CourseHero, etc.
- Ask us for permission if you are ever in doubt

# Written exams (30%)

- Two mid-terms (9 + 9 = 18%) and a final exam (12%)
- Cheat sheet (1 page letter paper, both sides) allowed
  - It must be handwritten by you
- Calculators are allowed
- (Stating the obvious) you cannot collaborate or google solutions during exams

- Mid-terms tentatively scheduled on 2/22 and 4/08 during class
- Notify me as early as possible if you must miss scheduled mid-terms or final

# Writing answers

- In your answers to exams, problem sets, and project reports:
- Be as clear and concise as possible
- Vague and rambling answers will get zero credit
  - We must be able to understand your answer quickly
- 25% credit for questions if you leave the answer blank or clearly write “I don’t know”

# Lecture questions (10%)

- Each day of lecture, hand in responses on Canvas
  - Including today
- You can consult the lecture (and your notes)
- Responses due by 8 pm Eastern Time
- We will consider your 20 highest scores

# Late policy

- Don’t be late
- If you must be late, inform us in advance
- If you cannot inform us in advance (e.g., medical), provide a doctor’s note or other allowable documentation
- Unexcused late submissions will lose a significant fraction of points

# 24/7 Grading Policy

- You may not dispute a grade or request a regrade before 24 hours or after 7 days of receiving it

- Please contact us if you have a legitimate regrading request:

  - After 24 hours of receiving the grade: Please take the time to review your case before contacting the instructors

  - Before 7 days have elapsed: we don’t want to forget what the test/project was all about!

# Help, Accommodations, etc.

- We'll make every effort to accommodate reasonable requests that support your learning better

- sn624@cs.rutgers.edu

- The course staff is here for you

# Next steps

- Finish today's lecture questions
- Look out for problem set 1 and project 1 released by Wed
- Enroll on Piazza

- Contact me if interested: independent study & research opps

- See you at Friday's lecture