Internet Technology

RUTGERS
UNIVERSITY | NEW BRUNSWICK

The Internet

The Internet has transformed how we live

- Communicating with other human beings
- Processing vast quantities of data
- Learning what’s going on in the world
- Education
- Transacting and doing business
- Entertaining ourselves
- Living sanely in an isolated world
- Moving and operating in the physical world

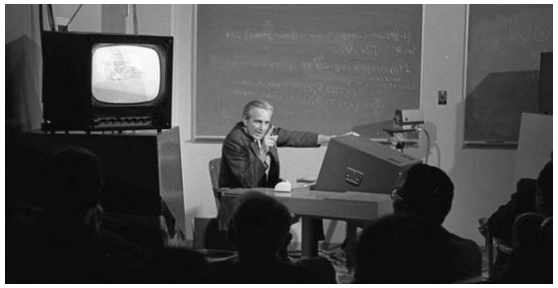
Internet growth

1968

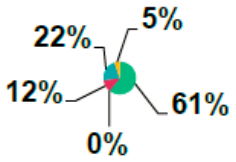
DARPA research experiment

San Francisco to Menlo Park video call

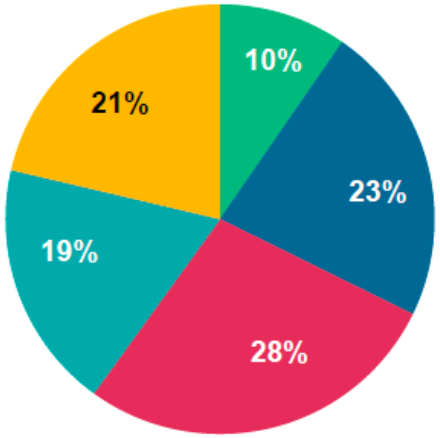
“Mother of all demos”



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

2023

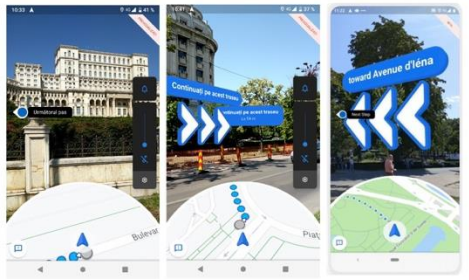
5.2B users

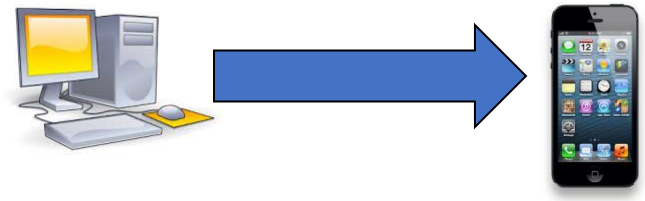
(65% of the world’s population)

Also 5.2 billion smartphone users worldwide

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

Evolution of Internet applications





2010-2020

2020--

1992	1996	2000	2004	2008
ftp Web email	chat Games IM Yahoo!	news Blog Search amazon	Music itunes Games search Google	Wikipedia Craiglist Youtube

UBER

airbnb

NETFLIX

Spotify

NVIDIA

OMNIVERSE

FORTNITE

OpenAI

MINECRAFT

Gather

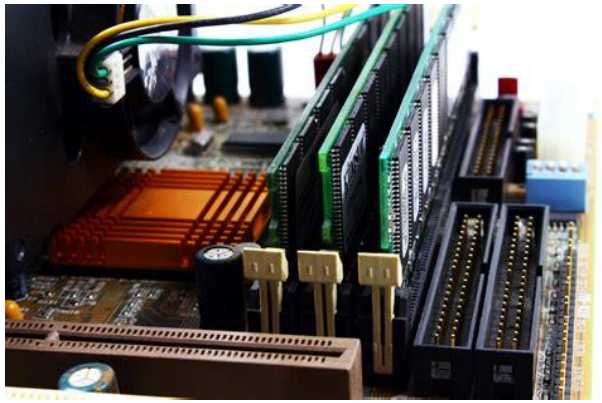
Text-heavy

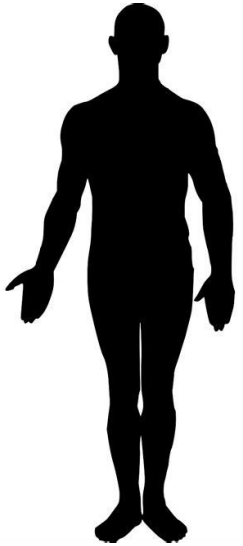
Multimodal media

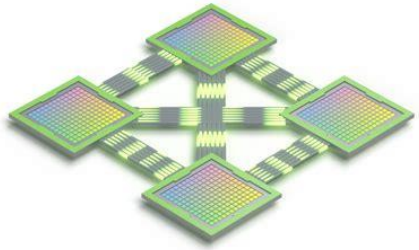
Augment physical world

Replace phy world

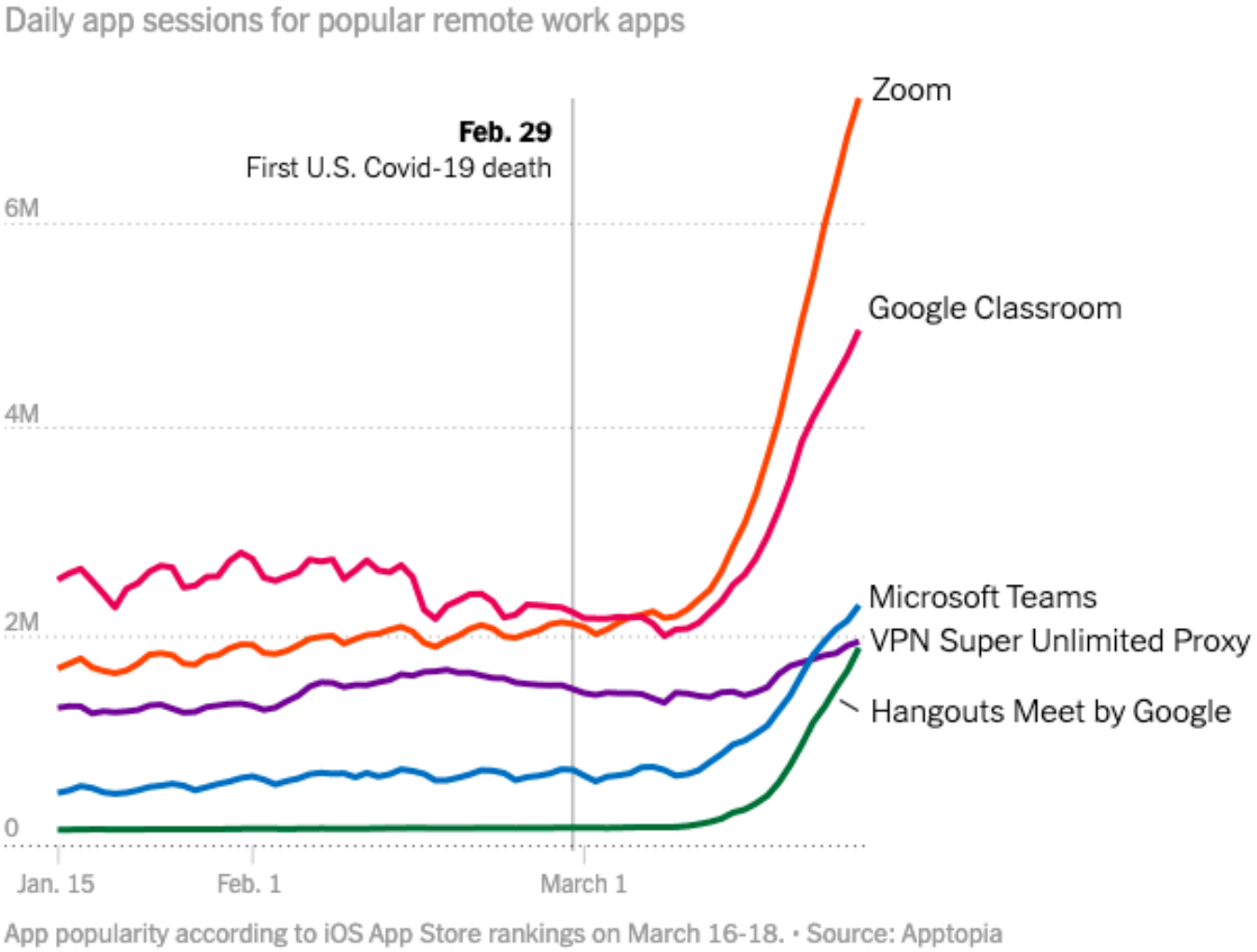
Emerging avenues







We rely on the Internet to work



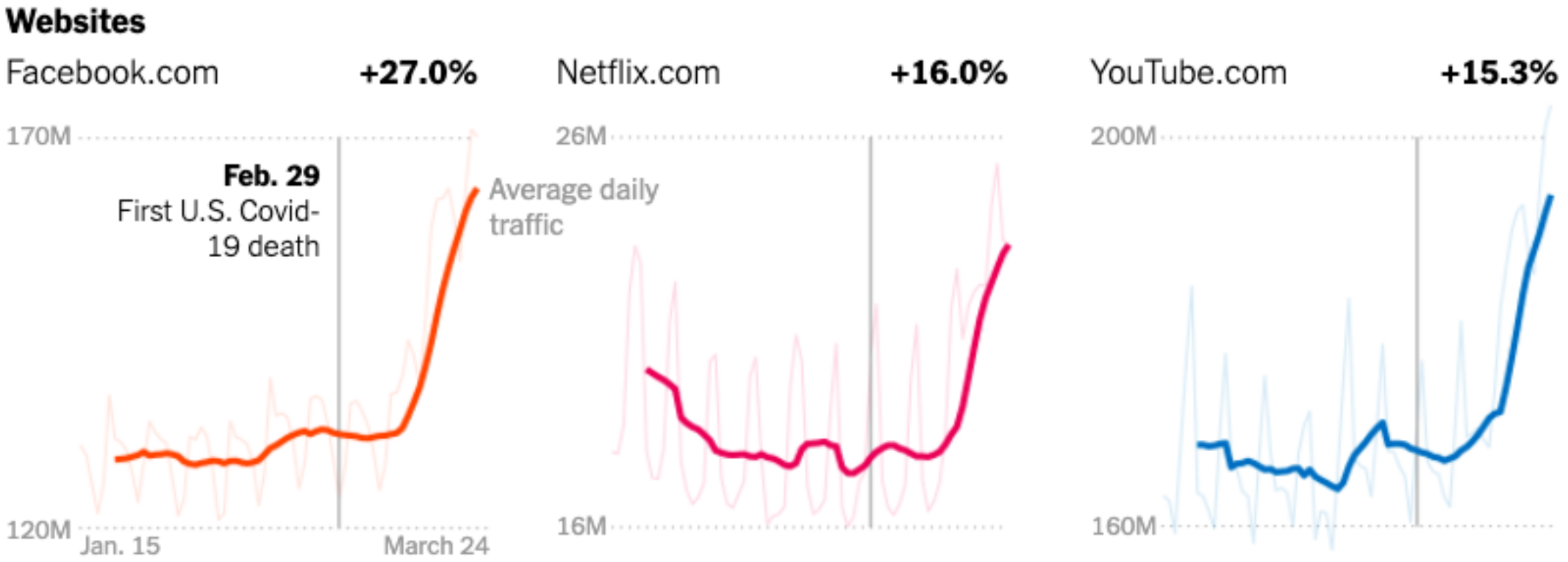
Data shows number of daily sessions in the US over a period in 2020. Source: nytimes

Remote work, learning, meetings are now routine

1 in 5 workers working remotely, 98% of workers want to work remote at least some of the time

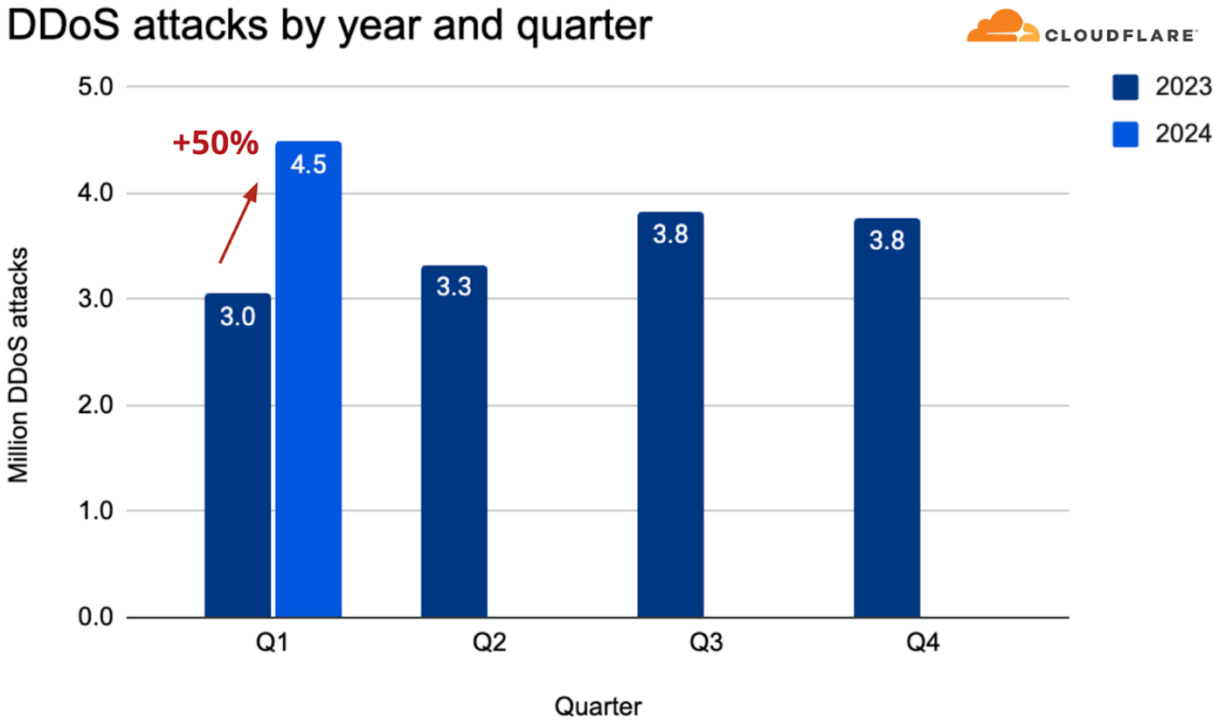
https://www.forbes.com/advisor/business/remote-work-statistics/

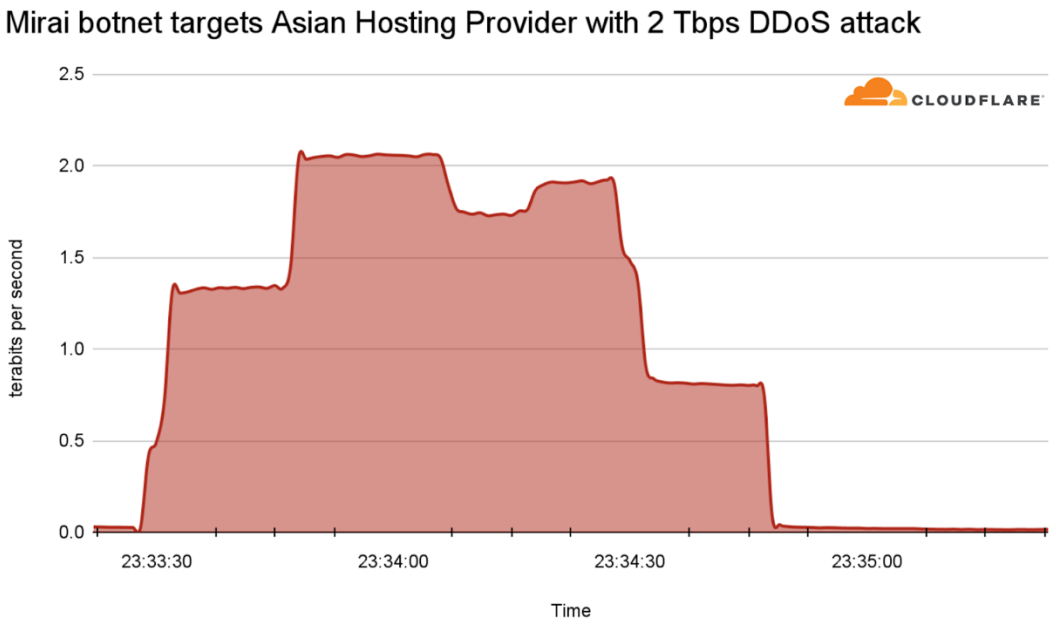
We rely on the Internet to play & socialize



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

Threats on the Internet are growing, too





https://blog.cloudflare.com/ddos-threat-report-for-2024-q1/

Threats on the Internet are growing, too

- Phishing
- Data breaches
- Identity theft

Internet Technology: This course

- The Internet is an example of a computer network
- A computer network is a collection of computers exchanging information

Learn fundamental principles and artifacts that underlie the Internet

… so that you can use and build on this technology for fun, profit, or societal benefit.

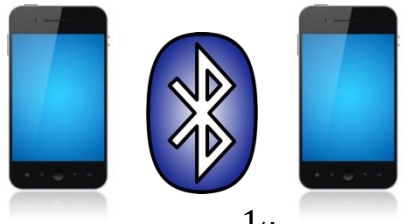
Some questions we will study

- How do you as an app developer use networking capabilities?
- How are messages delivered reliably?
- How is the Internet designed to share limited resources across growing and variable user demands on the fly?
- How do messages find their way to the destination (despite frequent equipment failures)?
- (if time permits) How is communication secured on the Internet?

What is a network?

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

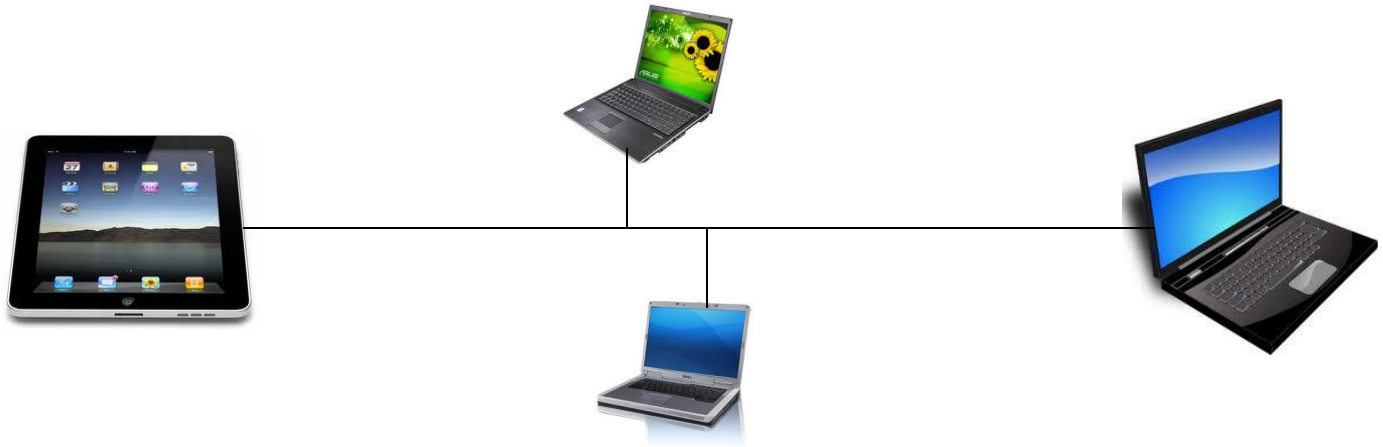
- The interconnection between entities is any physical medium capable of carrying information: we call physical media links
  - Wireless links: cellular 4G/5G, wifi 802.11, bluetooth, satellite
  - Wired links: copper wire, optic fiber



A single link multiple access network

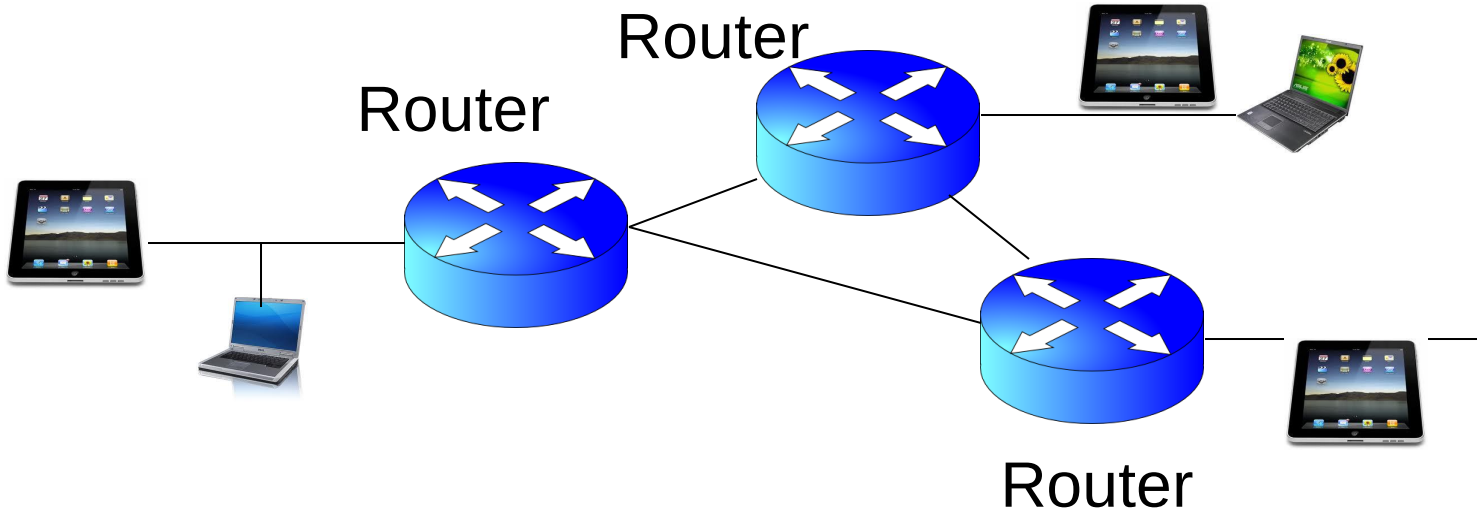
- Send bits of data in packets or frames
- How do we differentiate among many receivers?
- Every endpoint has a link level address: also called a *MAC* address
- Packets have a destination address on them
- However, can't have every computer in the world on the same link!
  - Physical limits on power & distance over which physical information travels
- There are many kinds of addresses. We will see why we need as many later

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)
- Detecting and correcting errors

A multi-link network



- Connect multiple links via routers
- Need to figure out how to move packets from one endpoint to another endpoint, e.g., reaching google.com from your laptop
- Known as the routing problem
- Key Q: At every router, how should packets be moved towards the destination (on the packet)?

In general, networks give no guarantees

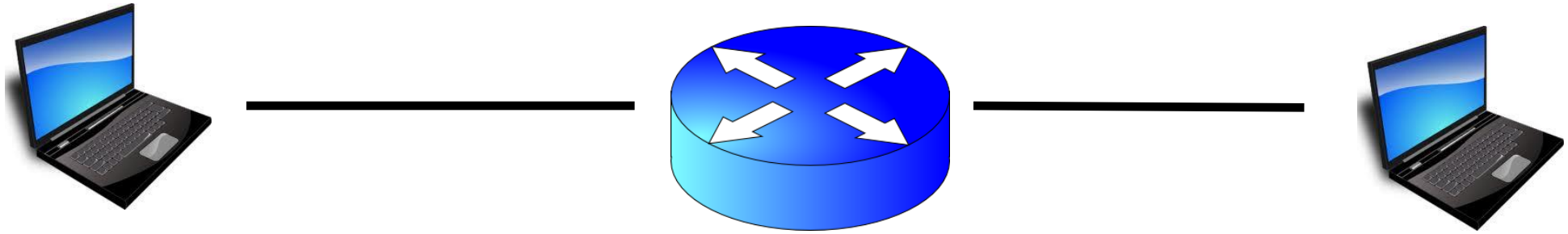
- Packets may be lost, corrupted, reordered, on the way to the destination
  - Best effort delivery

- 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”

- Early Internet thrived: easy to engineer, no guarantees to worry about

Guarantees for applications

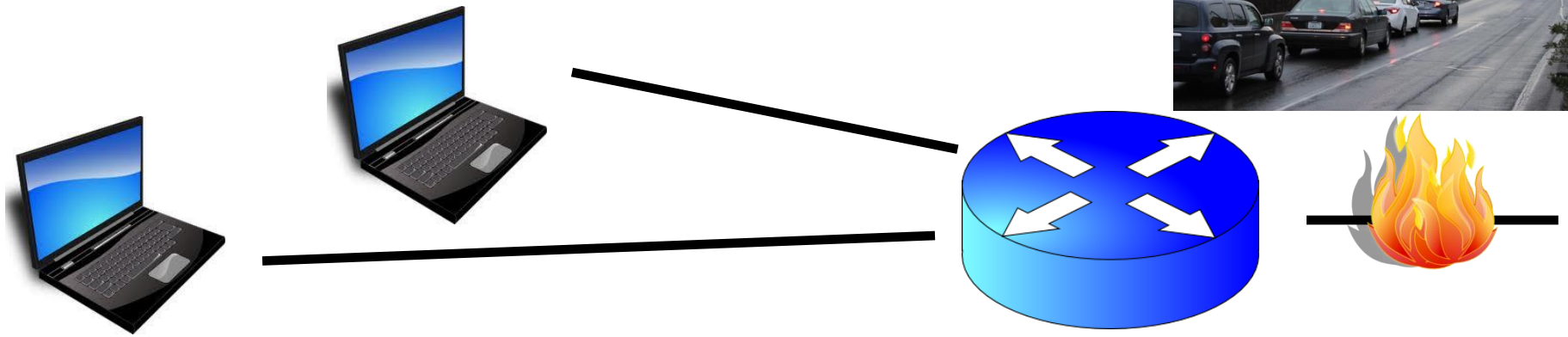
- How should endpoints provide guarantees to applications?



- Transport software on the endpoint oversees implementing guarantees on top of an unreliable network
- Reliable data delivery: applications should know that the data got through to the other side, or get notified of failure
- For some applications, also need ordered data delivery

Managing resources in a shared network

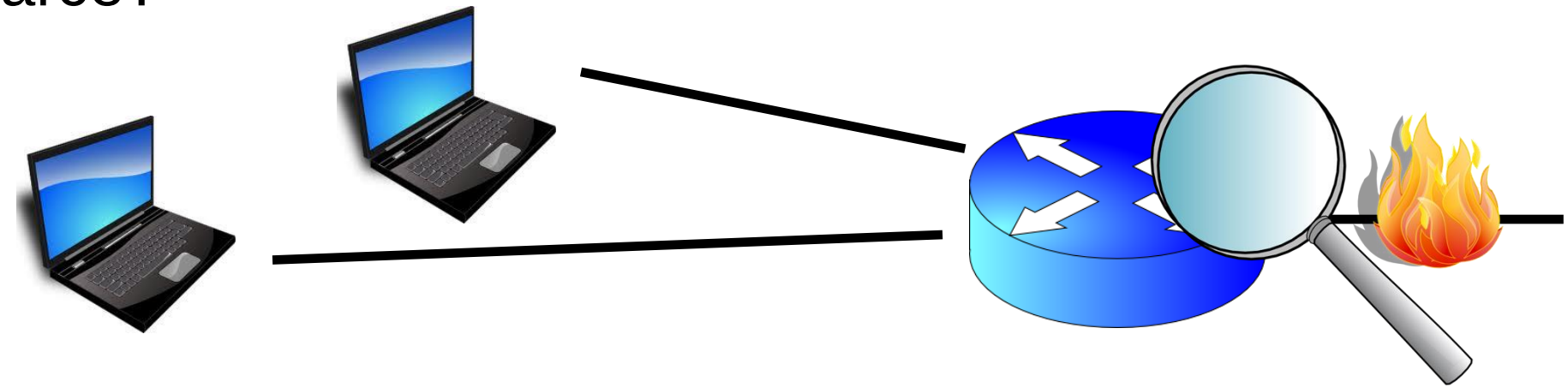
- How quickly should endpoints send data?



- 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?

Managing resources in a shared 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?

Components of a network: Summary

- Link
  - Communication links for transmission
- Endpoint (or) Host
  - Computer running applications of end user
- Router
  - Computer for routing packets from an input link to an 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 Tuesdays from 4--5 pm (Subject to change)
  - Lectures on Tue and Fri 12:10 – 1:30 pm ET
- TAs and Recitations: Four sections
  - Chang, Negin, Ajay, Yen-Lin
- Post q's to Piazza (see Canvas announcement to sign up)
- Class info: http://www.cs.rutgers.edu/~sn624/352-F24

Class philosophy

- We want you to learn and to be successful
- Attend recitations and office hours regularly to discuss
- Be proactive: interact, ask, support.
  - Use Piazza
  - Happy to help supplement your learning with more resources
- Full recorded video lectures will be made available
  - NOT A SUBSTITUTE for attending lecture

Grading

- 40% programming projects
- 15% problem sets
- 18% mid-terms (2 * 9% each)
- 12% final exam
- 15% lecture questions

- Schedule of projects, problem sets, exams, etc. will be made available https://www.cs.rutgers.edu/~sn624/352-F24/syllabus.html

- This course uses absolute grading. There is no curve

Programming projects (40%)

- Five programming projects. Python3
- P1: Warmup/Socket programming intro
- P2: HTTP programming
- P3: Asynchronous sockets and load balancing
- P4: Reliable data delivery
- P5: IP network and routing configuration

- Due dates will be made available on the class web page

Programming projects (40%)

- Work in the same group of two students throughout semester
  - Only change groups or work solo under extenuating circumstances
  - Discretion of the instructor. Talk to us
  - Sign up team on Google forms (to be announced on Canvas)
- Program and short write-up with responses required
- Background needed to get started
  - Python3: Get comfortable using data structures (tuples, arrays, dictionaries)
  - Unix (login, navigating folders, permissions, etc.)
- We will provide instructions to use class VMs
- Hand projects in on Canvas

Programming projects (40%)

- Please follow all 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 (15%)

- 3 problem sets
- Work individually
- Hand in a PDF file with solutions on Canvas
- Typically, due one week prior to major written exams
- Exact due dates will be made available on the class web page

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 or typed up by you
- Calculators are allowed
- (Stating the obvious) you cannot collaborate or google solutions during exams. No mobile phone use permitted

- Mid-term dates will be posted on the class website shortly
- Please notify us right away if you must miss scheduled written exams

Lecture questions (15%)

- Day before each lecture, hand in responses on Canvas
  - Includes the upcoming Thursday
  - Last lecture question due date will be announced separately
- You can consult the lecture (and your notes)
- No collaboration or searching for answers on the Internet
- We will consider your 20 highest scores (out of 26)

Class integrity and Grading policies

Collaboration and Integrity policies

- Intellectual collaboration is welcome and encouraged
- All your written (coded) work must be your (team's) own
- Understand the problem deeply and produce your own solutions
- Ask us for permission if you are ever in doubt

Collaboration and Integrity policies

- Do
  - Ask questions on Piazza
  - Discuss projects and problem sets with us and 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

- You can talk to others, however all submitted work (code) must be your (your team's) own
- Do not blindly lift code from GitHub, genAI, stack overflow, …
- Do not look at other students' code or solutions
- Do not use the solutions from previous cohort(s)
- Do not post solutions on GitHub, Chegg, CourseHero, etc.
- Free to learn from any publicly available sources and produce your own solutions
- State collaboration & references in the attached report
- We will run plagiarism checks on your submissions

Rutgers takes academic dishonesty very seriously.

Consequences include suspension and expulsion.

We will run plagiarism detection tools on all submitted materials.

If you are ever in doubt, ask us first.

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”

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 official medical note of absence through the University
- Unexcused late submissions will result in losing 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 and Accommodations

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

- sn624@cs.rutgers.edu

- Course staff is committed to help you succeed

See you on Friday

Next steps

- Finish lecture questions by Thursday (soon up on Canvas)
- Look out for project 1 released later this week
  - Starting early significantly helps your project grade (40% of total)
- Look out for project team signup form
  - We will share VM access instructions once we know your teams
- Sign up for class Piazza (link TBA on canvas)
- Play around with Python3 (ilab available)