Internet Technology

RUTGERS
UNIVERSITY | NEW BRUNSWICK

Introduction

Lecture 1

Srinivas Narayana

http://www.cs.rutgers.edu/~sn624/352-F22

RUTGERS
UNIVERSITY | NEW BRUNSWICK

The Internet is an exciting place

ROBLOX

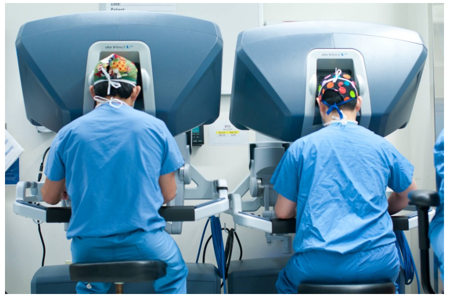
aws

Instagram

NETFLIX

WhatsApp

Spotify

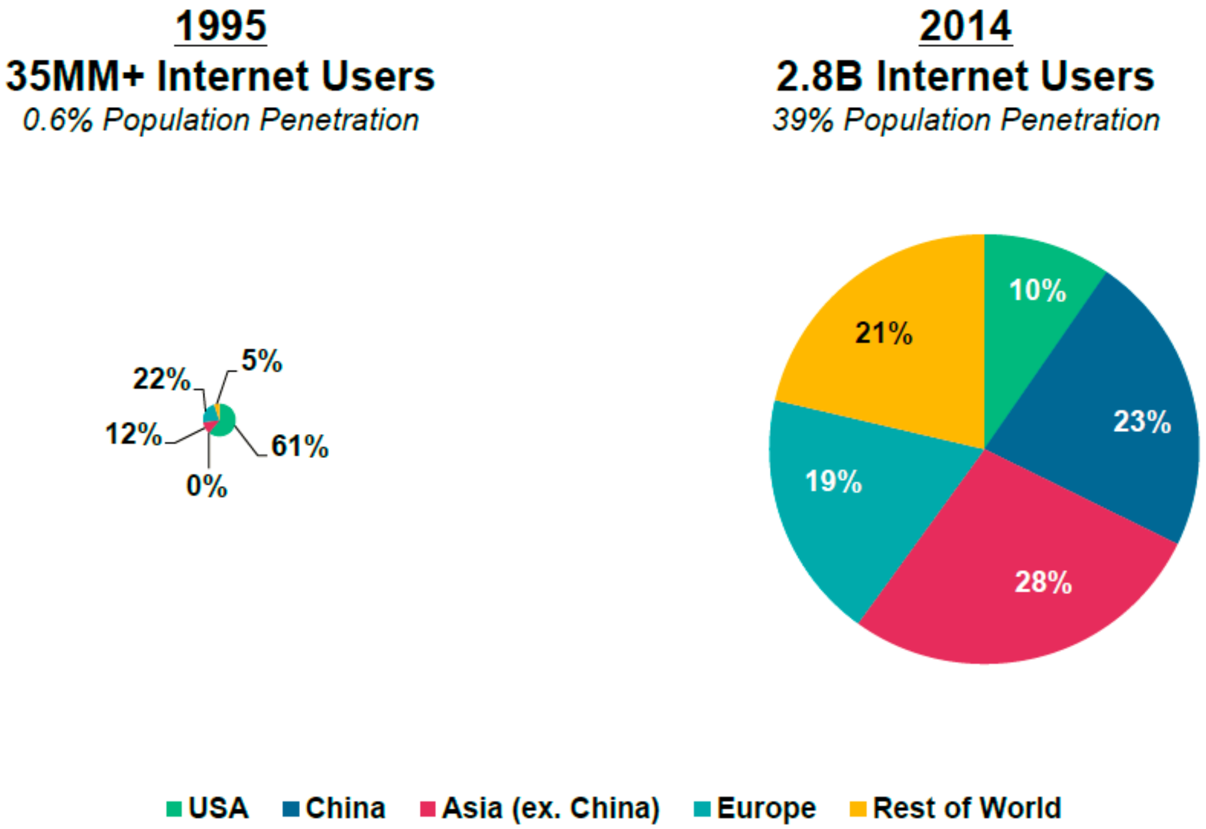


The Internet has transformed everything

- 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

Internet growth



1995
35MM+ Internet Users
0.6% Population Penetration

2014
2.8B Internet Users
39% Population Penetration

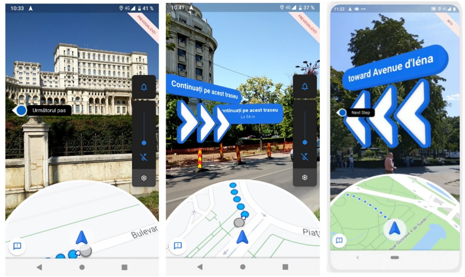
2020

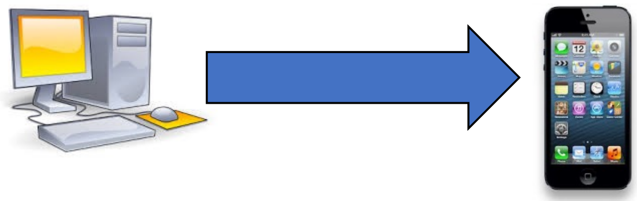
4.8B users

(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

UBER

airbnb

NETFLIX

Spotify

2020--

NVIDIA

OMNIVERSE

FORTNITE

MINECRAFT

Gather

Text-heavy

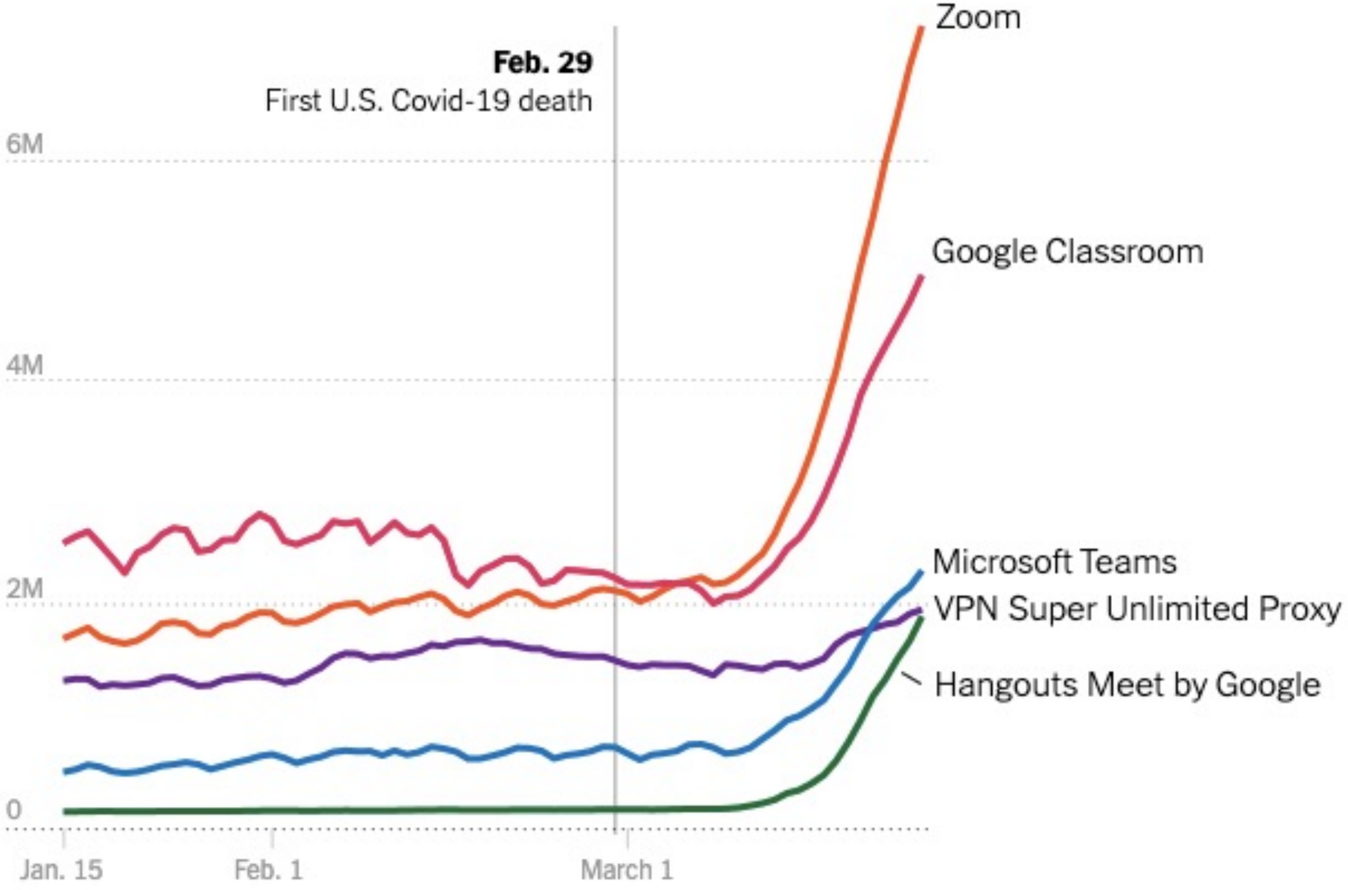
Multimodal media

Augment physical world

Replace phy world

We relied on the Internet to work

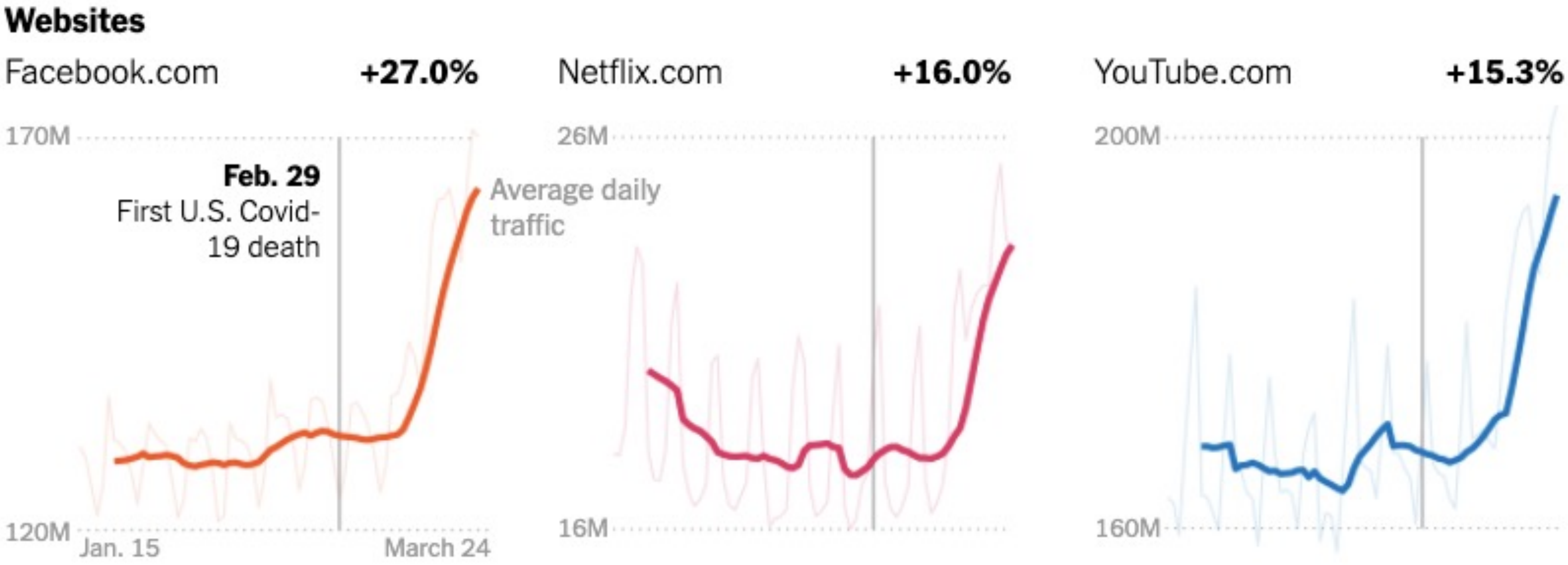
Daily app sessions for popular remote work apps



App popularity according to iOS App Store rankings on March 16-18. • Source: Apptopia

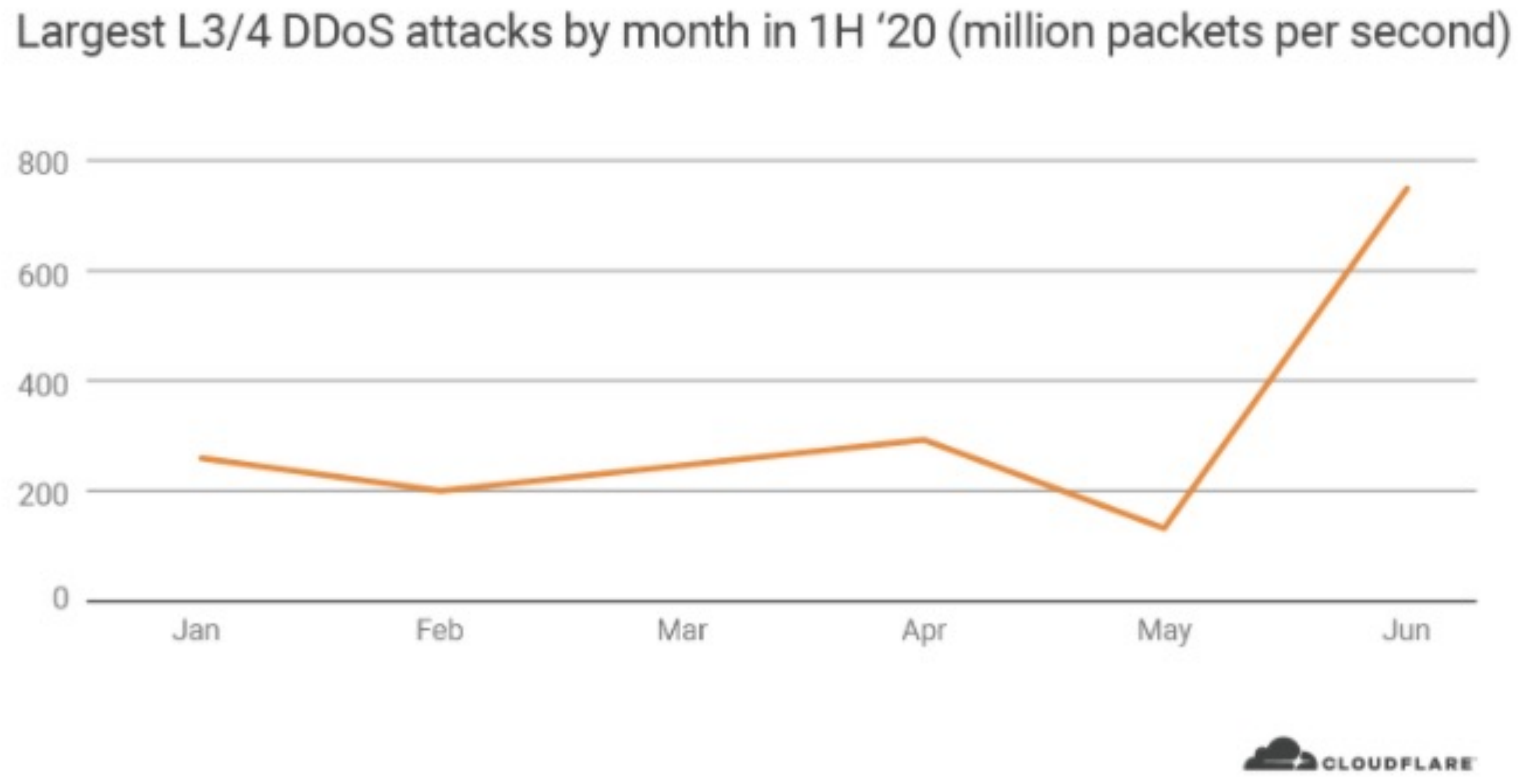
Data shows number of daily sessions in the US over a period in 2020. Source: nytimes

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



Largest L3/4 DDoS attacks by month in 1H '20 (million packets per second)

Source:
CloudFlare
blog

Internet Technology: This course

- The study of how the Internet is designed

- The Internet is an example of a computer network

Learn fundamental principles and artifacts that underlie the Internet.

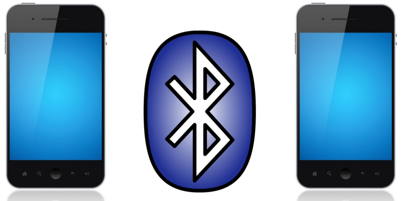
So that you can use and build technology for fun, profit, or altruism.

What is a network?

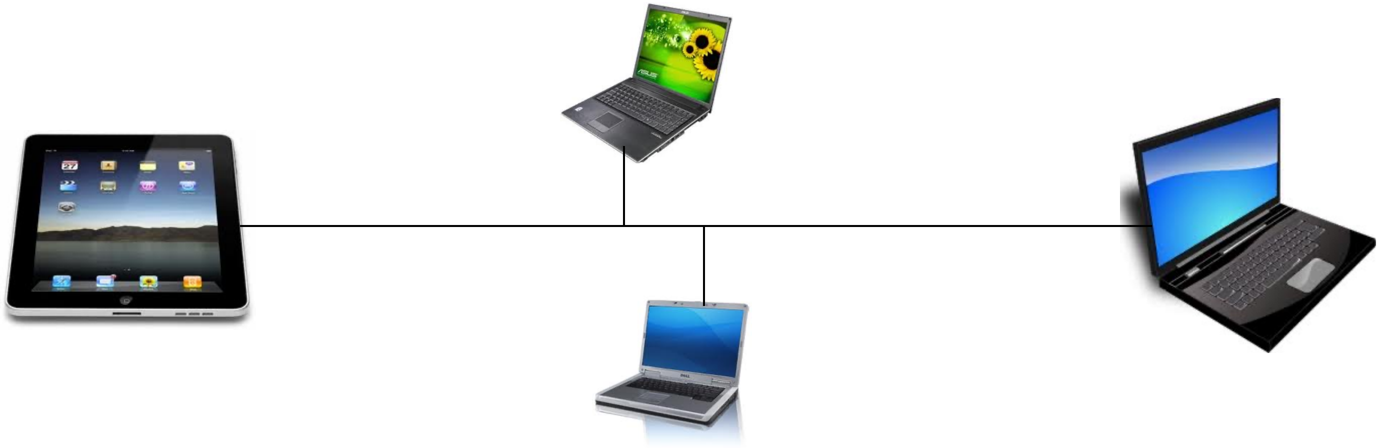
- Carrier of information between two or more entities
- Entities may be hosts/endpoints (used interchangeably)
  - your laptop, cell phone, etc.

- 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

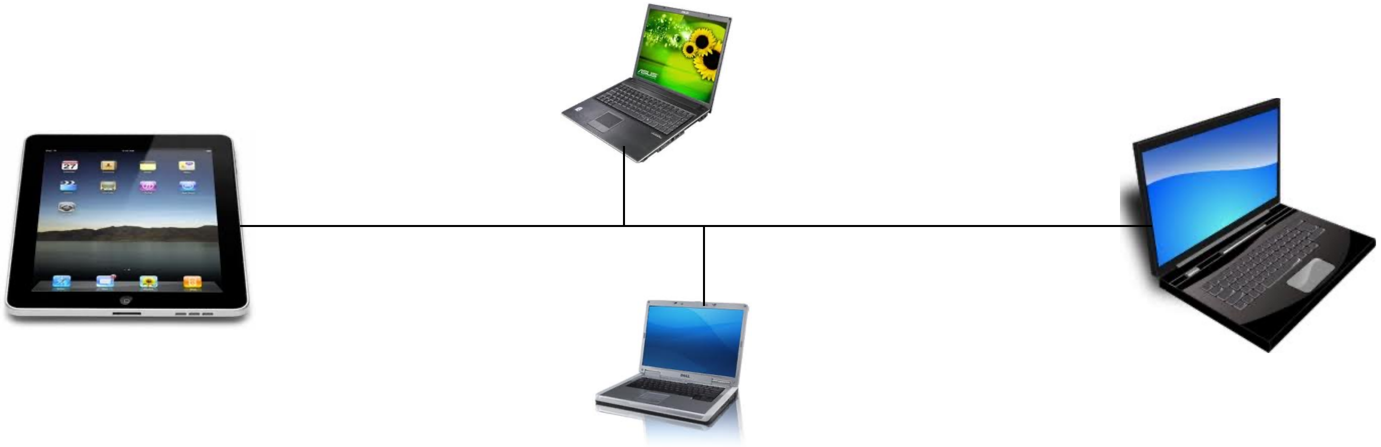


A single link multiple access network



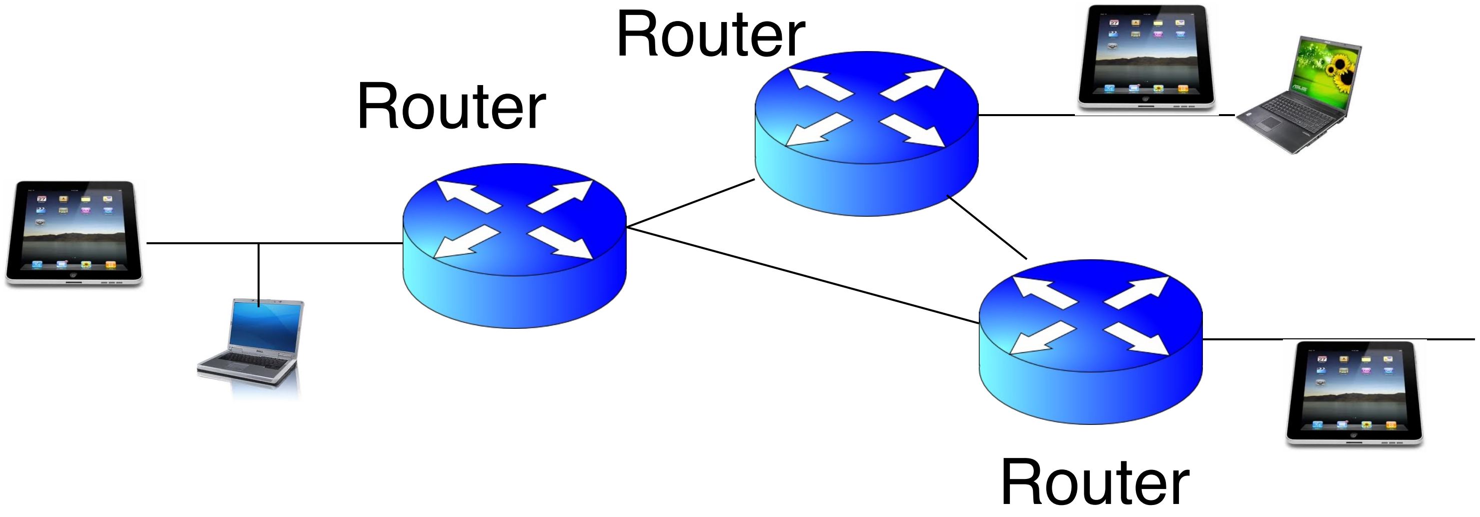
- Send bits of data in packets or frames
- How do we differentiate among many receivers?
- Every endpoint as a link level address: also called a *MAC* address
- Packets have a destination address on them
- However, can’t have every computer in the world on the same link!
  - Physical limits on power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

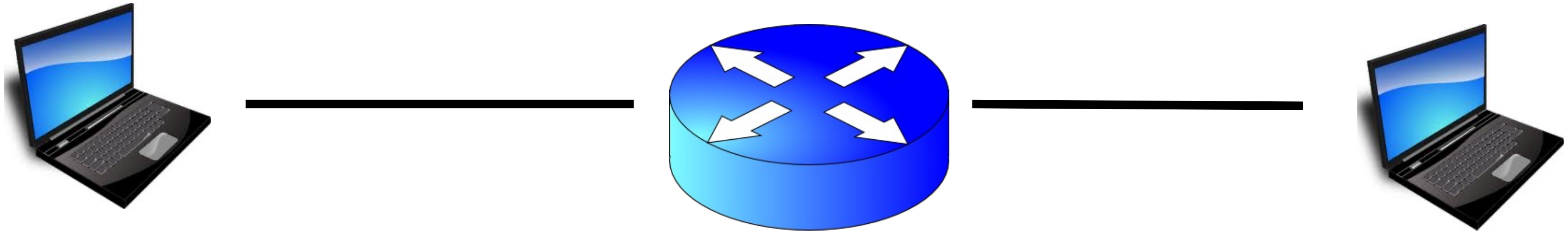
- Packets may be lost, corrupted, reordered, on the way to the destination
  - Best effort delivery

- 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”

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

Guarantees for applications

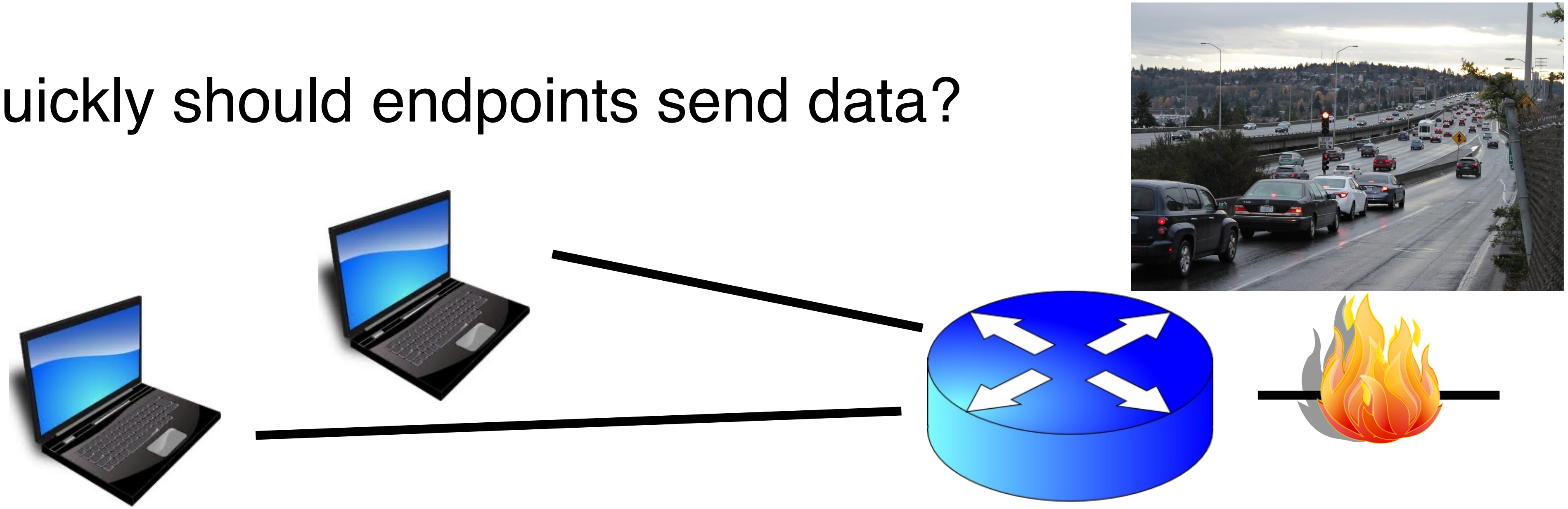
- How should endpoints provide guarantees to applications?



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

Sending data into a multi-link network

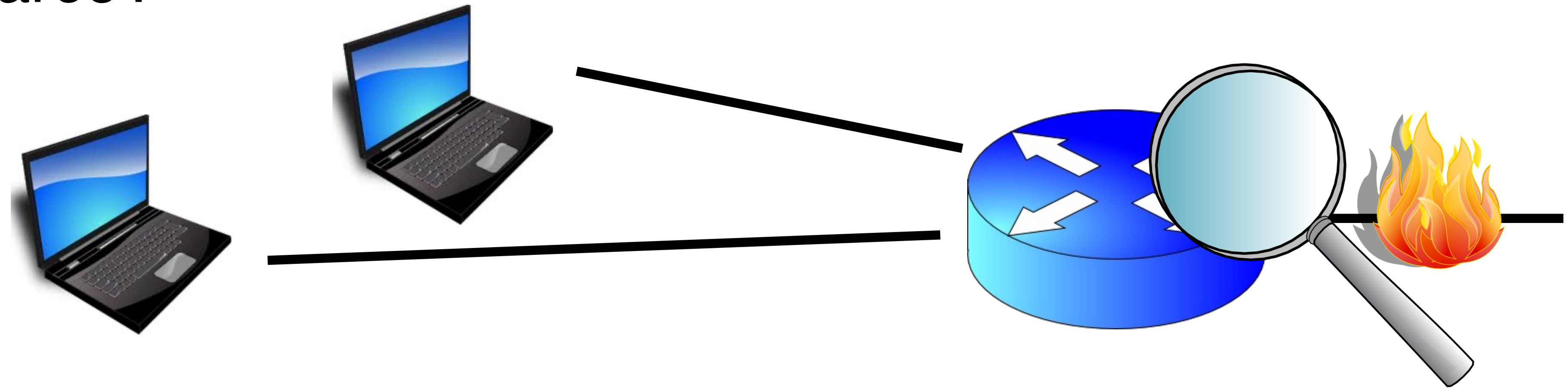
- How quickly should endpoints send data?



- 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). Tue 10 – 11 am ET and Wed 9 - 10 am ET
  - Lectures on Tue and Fri 8:30 – 9:50 am ET
- TAs and Recitations: Three sections
  - Chang, Parvathi, and Negin
- Post q's to Piazza (see Canvas announcement to sign up)
- Class info: http://www.cs.rutgers.edu/~sn624/352-F22/

Class philosophy

- We want you to learn and to be successful

- Attend recitations and office hours regularly to discuss material

- Be proactive: interact, ask, support.
  - Use Piazza

- Full video lectures from 3 offerings (spr21, 22, fall22) available

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. available at https://www.cs.rutgers.edu/~sn624/352-F22/syllabus.html

- This course uses absolute grading. There is no curve

Programming projects (40%)

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

- Tentative due dates 9/23, 10/14, 10/28, 11/18, and 12/02
  - Submit by 8 pm Eastern Time

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.
- 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 (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
- Due dates: 9/30, 11/04, and 12/09
  - Submit at 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 us and each other
  - Read references (textbooks, Internet tutorials) widely
  - Acknowledge each other and all the references in psets & 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 or typed up by you
- Calculators are allowed
- (Stating the obvious) you cannot collaborate or google solutions during exams

- Mid-terms tentatively scheduled on 10/07 and 11/11 in class
- Notify us ASAP if you must miss scheduled exams

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 (15%)

- Each day of lecture, hand in responses on Canvas
  - Includes today!
- You can consult the lecture (and your notes)
- No collaboration or searching for answers on the Internet
- Submit by 8 pm Eastern Time
- We will consider your 20 highest scores (out of 26)

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, Accommodations, etc.

- 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

Next steps

- Finish today's lecture questions (up on Canvas)

- Look out for project 1 released later this week
    - Starting early significantly helps your project grade (40% of total)

- Sign up for class Piazza (link on canvas announcement)

- Contact me if interested: independent study & research opp's

- See you at Friday's lecture