

CS144 Course Intro (Winter 2025)



Keith Winstein



Wireless Communications: 1964

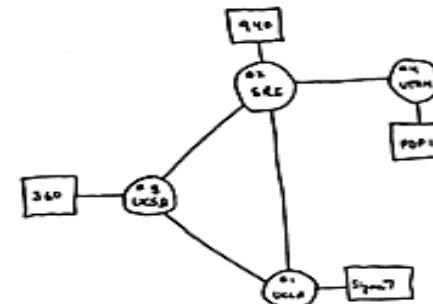
*“A network to
survive nuclear attack.”*



Paul Baran

1st network
connects two
computers

Four nodes connected
(UCLA, SRI, UCSB, Utah)



US Government
starts “ARPANET”
project

1964

1965

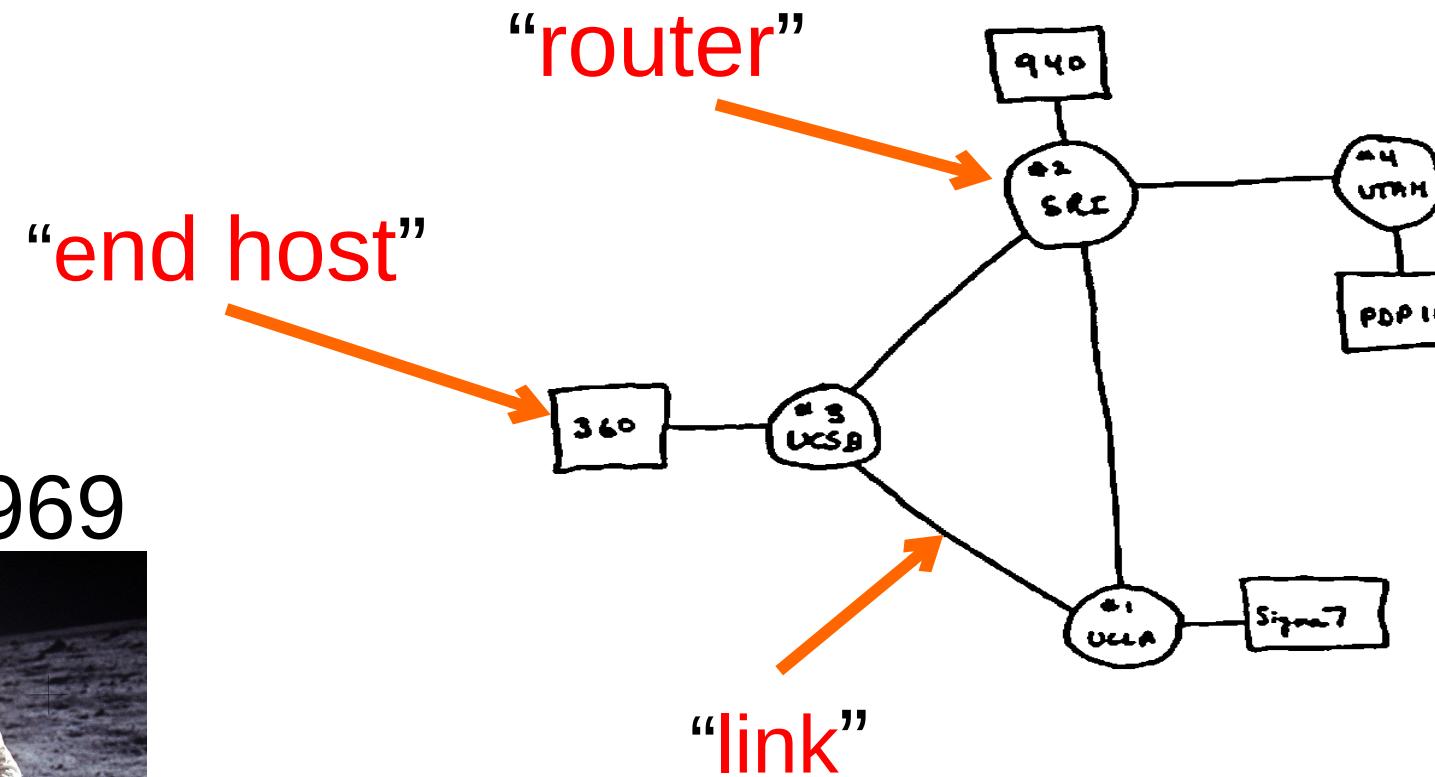
1966

1968

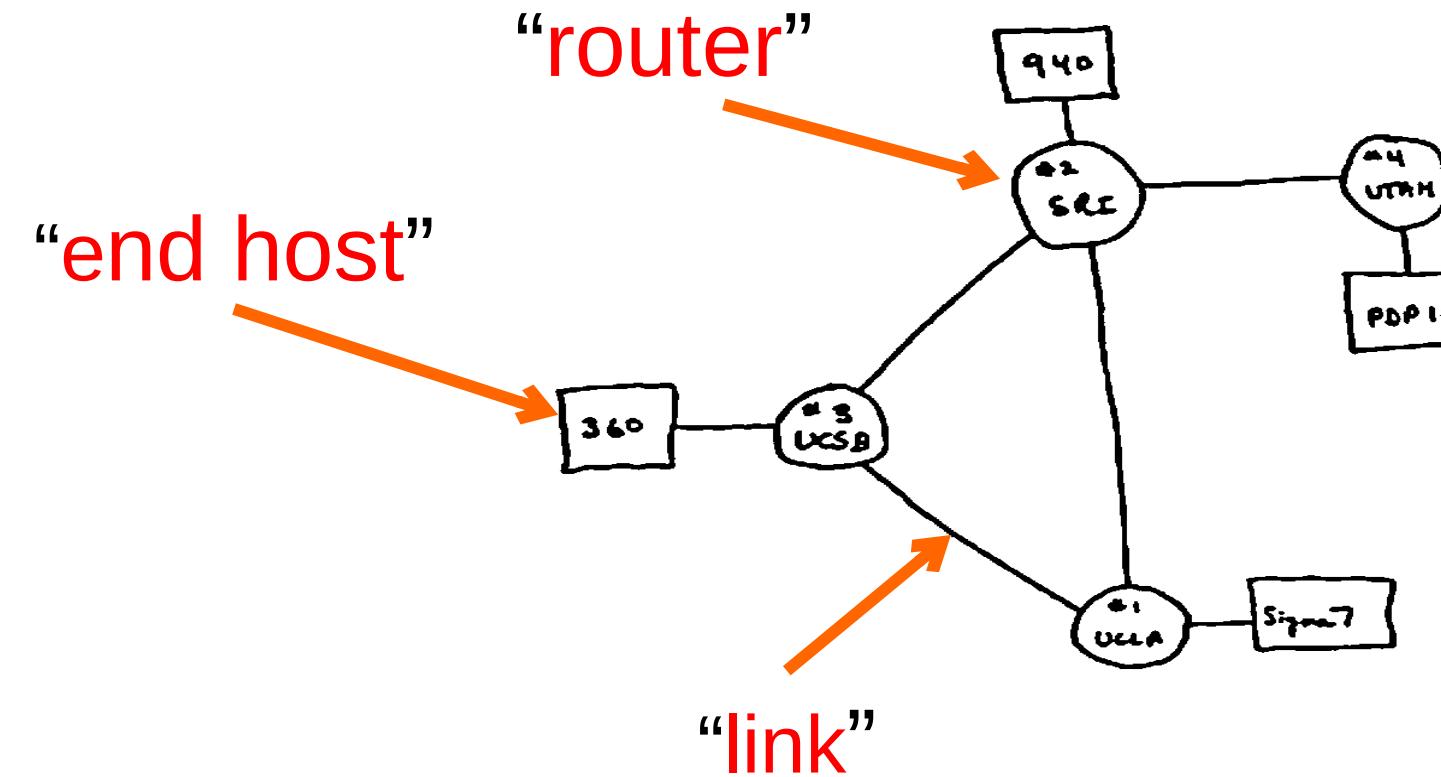
1969

The Internet in 1969

Also in 1969



What did they use it for?



1. Sending files between scientists: "*Here is a big file of astronomy data!*"
2. Email: "*Where shall we have lunch today?*"
3. Remote login to another computer.

1971

First email typed here

“QWERTYUIOP”

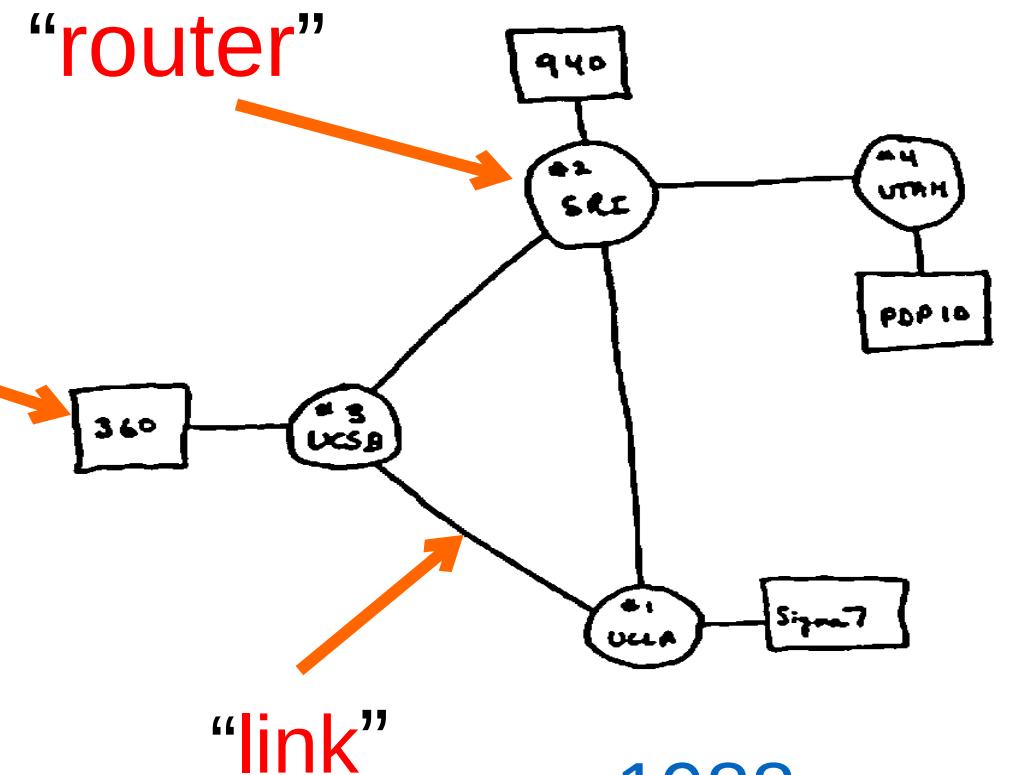
...and printed here





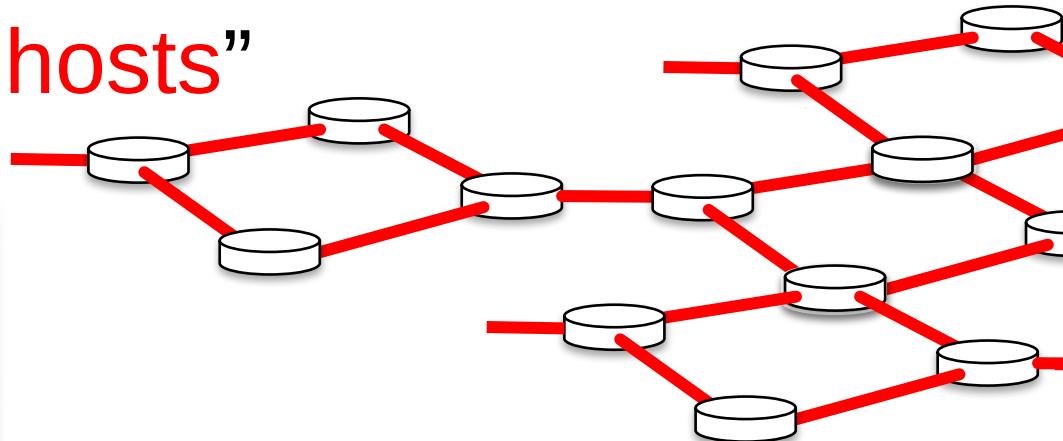
1969

4 “end hosts”



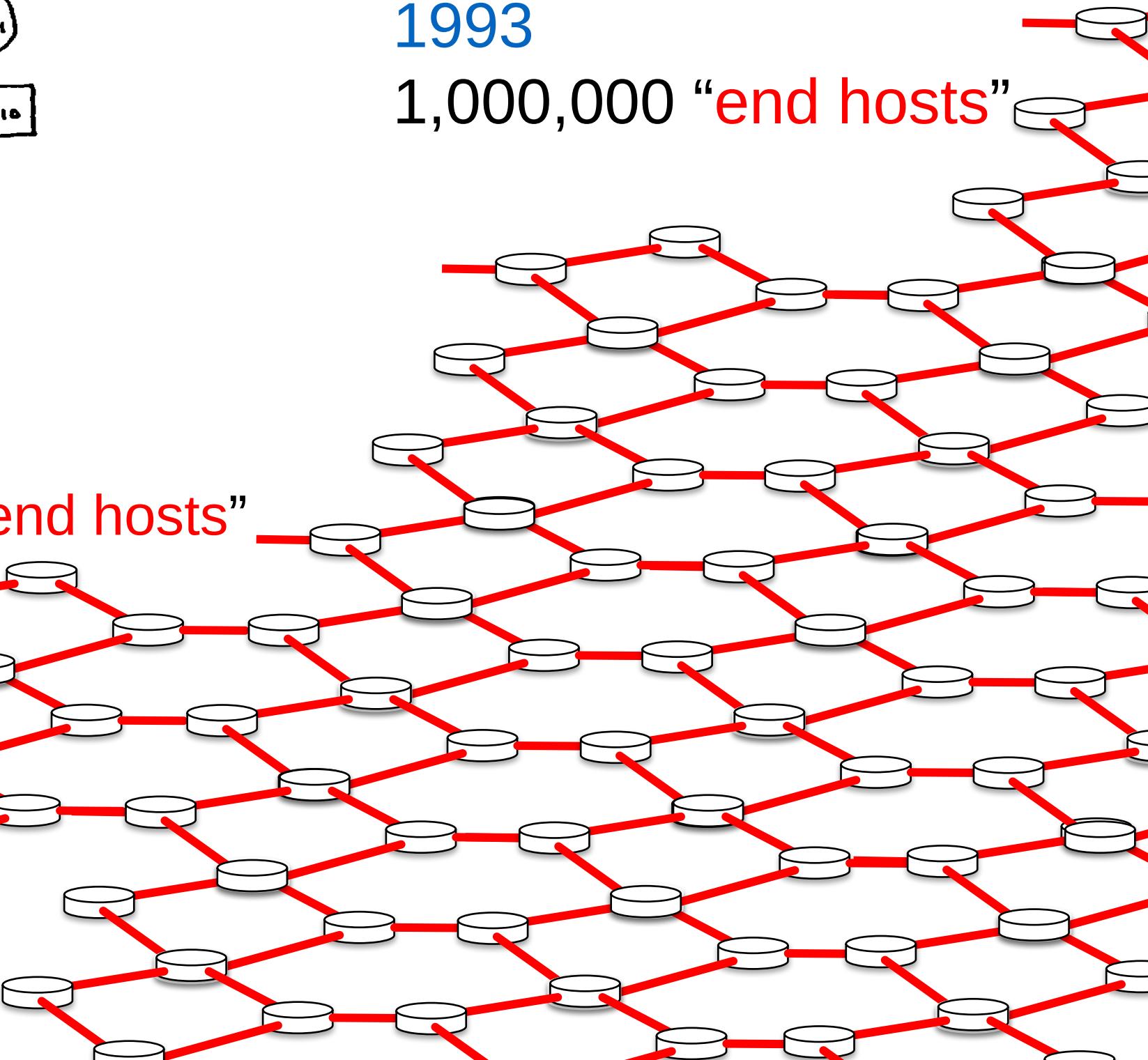
1988

10,000 “end hosts”



1993

1,000,000 “end hosts”



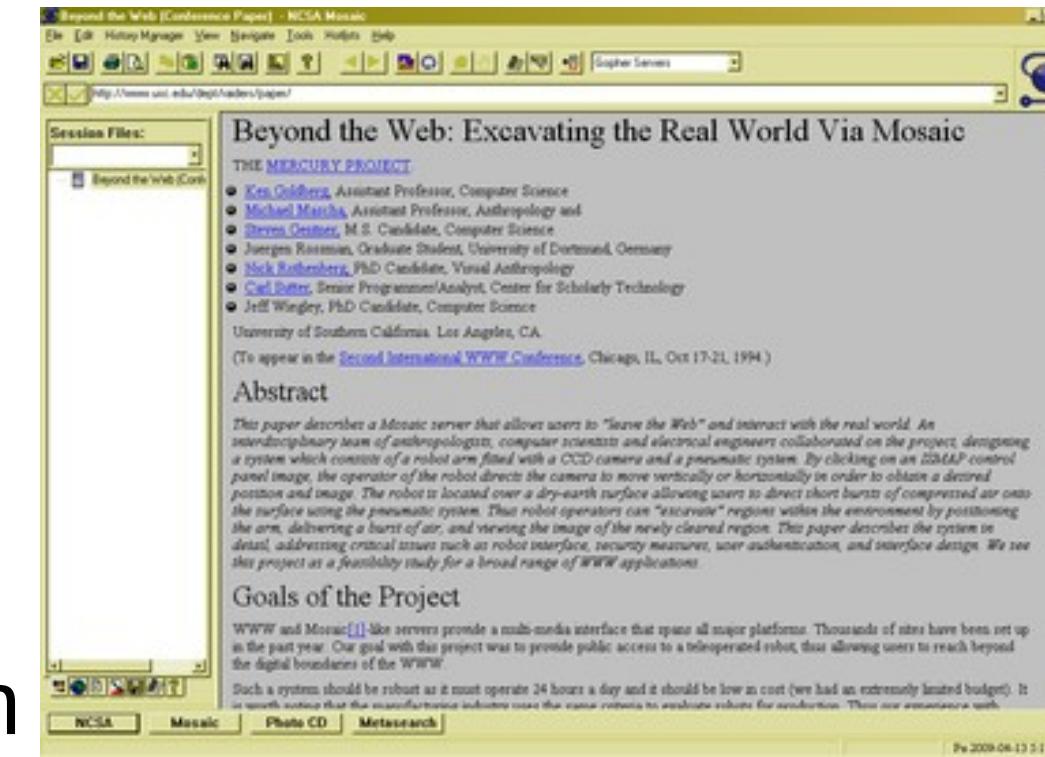
Network Systems: 1984

Then in 1993 something
even BIGGER happened!!!

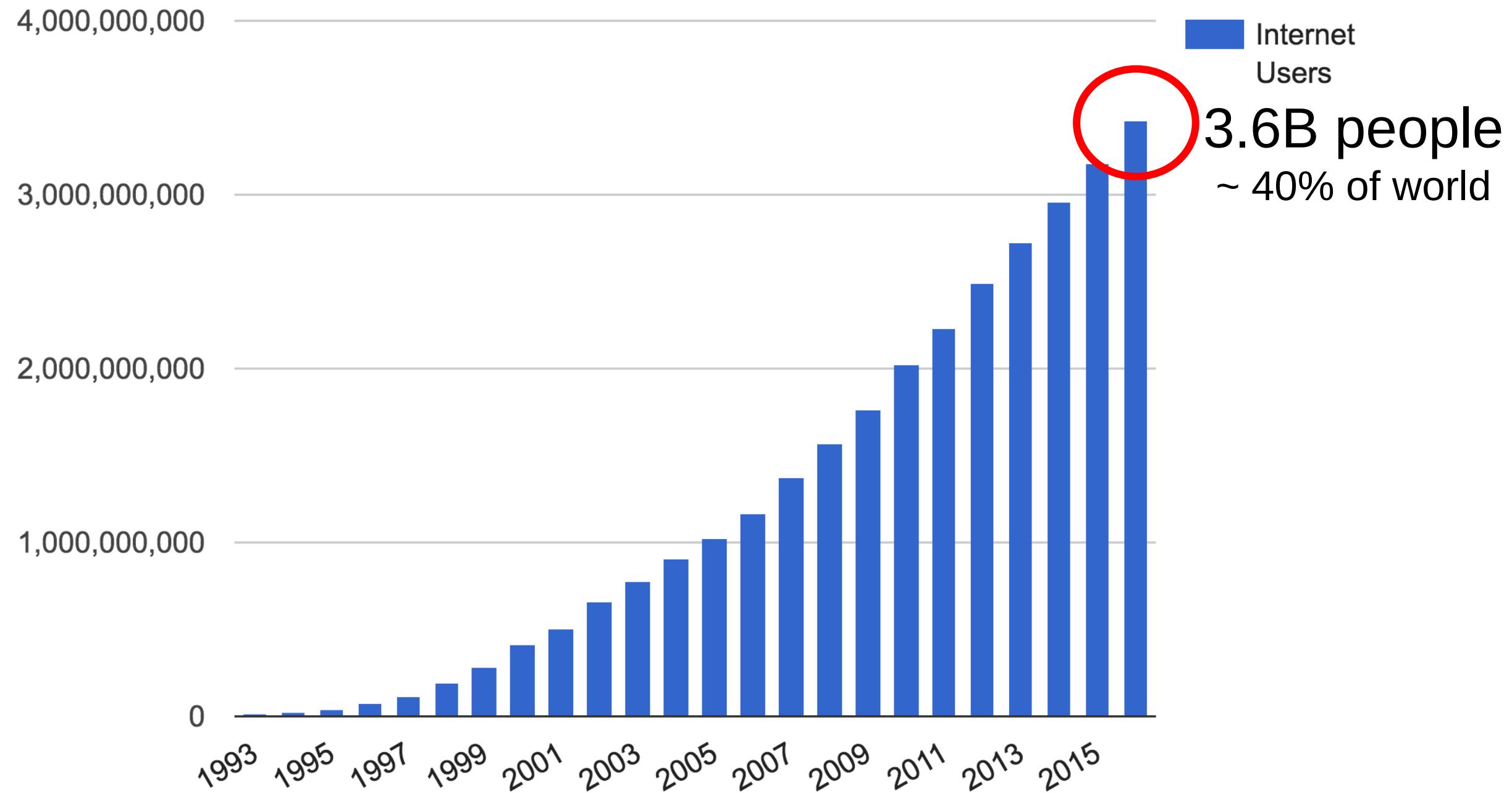
1993: The first web browser (Mosaic)



Marc Andreessen



The number of Internet users in the world





How does it all work?

Why was it designed this way?

CS144

Isn't it really difficult...?

Goals

1. To learn how the Internet works.
2. To learn why the Internet was designed this way.
3. To learn how to use the Internet.
4. To build some significant pieces of the Internet.
5. To learn the underlying principles and technologies of networking.

Class overview

CS144 is a fast-paced lab class.

Each week has THREE lectures and ONE lab session.

1:30-2:20 p.m. (Please no distracting laptop usage!)

The course is divided into week-long units, devoted to a particular topic.

e.g. the 1st week is about Basic Principles, then (2) Transport, (3) Packet Switching, (4) Congestion Control, (5) Routing, (6) Lower Layers, (7) Applications, ...

There is ONE BIG LAB ASSIGNMENT (“you build most of the Internet”) with 8 “checkpoints,” #0 through #7.

Laptops

- We discourage laptop use in class, especially use that will distract others behind you.
- Except for specific in-class exercises (we will ask you to bring your laptop).



How we calculate your grade

1. Participation 3%

- ❑ Helping others on Ed or in the lab sessions
- ❑ Credit in somebody else's lab writeup

2. Quizzes 12%

- ❑ Best 6 out of 8: 2% each

3. The lab 45%

- ❑ Correctness: 27%
- ❑ Checkpoints 0-7 subjective grades, best 6 out of 8: 18% (3% each)

4. Exams 40%

Midterm (Feb. 12): 20% (lab slot)

Final (March 19): 20% (duration TBA)

Q: “What does it take to earn an A?”

Exam Policy

Exams are closed-book, closed-note, closed-laptop etc.

But you may bring 1 double-sided piece of 8.5" x 11" paper of your own design to the Midterm, and 2 double-sided pieces of paper to the Final.

COVID

This quarter may ~~suck~~ present lousy circumstances. :-(

My job is to help you learn. I really want that to happen.

We're going to have to react to curveballs and we will do our best.

If you have to quarantine, we will be there for you (online) to help you avoid falling behind.

Lab

- Programming is in modern C++
- CS111 and C++ comfort are prerequisites.
- You want to go to the lab sessions + have started already
- Deadline policy:
 - Lab checkpoints are due Sundays, 11:59 p.m. (except week 6)
 - If you turn checkpoint in on time, we will get you subjective style/grade and feedback by Friday 11:59 p.m.
 - Late/extension deadline is Wednesday, 7 p.m. (just before lab session). We can't guarantee feedback timeline for these.
 - Functionality/correctness: no-penalty regrades until end of class.
 - *If more than a week behind, lab attendance is mandatory.*



Win Certificates!!!

1. Certificate for BEST checkpoint submission
2. Certificate for BEST help (credited in lab writeups)
3. Certificate for BEST help (on Ed)

Workload



- This is a 4-unit workload, which means a workload of about 12hrs/week
- Our estimate based on previous years
 1. Class time: 3hrs/week
 2. Labs: Avg 6hrs/week
 - *wide variance between checkpoints and between students*
 - *read the whole checkpoint BEFORE coding*
 3. Average overall ~12hrs/week

Contact

For anything non-private: **Ed**

If private: Private Ed posting, or
email cs144-staff@cs.stanford.edu

If it's personal (e.g. a medical emergency):
email Keith keithw@cs.stanford.edu

(If it's a mega-emergency, Keith's cell phone number is on his Web page.)

The Honor Code

- We take it seriously and we expect you to take it seriously too.
- Last year was a bad year with several CS144 students cheating. None of them exactly set out to cheat: They had other priorities. They had a sick parternet. They asked friends for “help” and got tempted by their screens. Or they copied an assignment off the web, then tried to modify it.
- We use tools to compare solutions against past ones.
- Please, let’s have a zero-violation year. The Honor Code is an amazing commitment by the students to govern and deter cheating themselves because you **don’t** want the faculty to police you. Please show that student self-government and mutual deterrence works so we can keep faculty out of it.

The Honor Code

Permitted Collaboration: The following items are encouraged and allowed at all times for all students in this class:

- Discussion of material covered during lecture, problem sessions, or in handouts
- Discussion of the requirements of an assignment
- Discussion of the use of tools or development environments
- Discussion of general approaches to solving problems
- Helping others at the lab session (**without** looking at code!)
- Discussion of general techniques of coding or debugging

The Honor Code

Unpermitted Collaboration: All submissions must represent **original, independent work**. Some examples of activities that do not represent original work include:

- Copying solutions from others or knowingly allowing others to copy your solution.
 - Use of solutions posted to websites.
 - Looking at another student's code (unless handwritten, not CS144-specific, **and** credited).
 - Placing your source code in a public repository where others can copy it.
 - Debugging code for someone else.
 - Collaborating on or discussing the online graded quizzes before you have completed them.
 - Using output from a code-synthesis tool (e.g. an LLM) trained on past students' solutions.
-
- *Honor code exercise: can you stump the teaching staff?*

What to do next

- Look around and get familiar with <https://cs144.stanford.edu>
- Start setting up your VM for Lab 0! It is due next Tuesday.
- See you Wednesday!

TCP/IP Header Formats in Lego

