Today

• What is (this course on) the Internet about?

[quick break]

• Class logistics
• Internet
• Protocols
• Architecture
Two Meanings of “Internet”

- The infrastructure that ties together computing devices
  - TCP, IP, BGP, DNS, OSPF, ...

- The ecosystem of applications built on top of the above infrastructure
  - facebook, google, twitter, ....

- In this class, we use the first definition!
end-host

switch
The Internet transfers data between end hosts
• Internet

• Protocols

• Architecture
Facebook server

World of Warcraft server

Instant messaging

World of Warcraft client

Safari accessing Facebook
while (...) {
    message = ...
    send (message, ...);
}

while (...) {
    message = receive(...);
}
Alice

Bob

hello

hello

give me http://cs.berkeley.edu

here: ...
Alice

Bob

hello

give me http://...
give me http://...
give me http://...
give me http://...
give me http://...
give me http://...
give me http://...
Protocol

• A specification of the messages that communicating entities exchange
  - their syntax and semantics

• Very much like conversational conventions ... determining who should talk next and how they should respond

• Designing a good protocol is harder than it first seems!
• Internet

• Protocols

• Architecture
Why study the Internet?
The Internet has and is transforming everything

• The way we do business ...
  • retail, advertising, cloud computing

• The way we have relationships
  • Facebook, twitter,

• The way we learn
  • Wikipedia, search engines, MooCs

• The way we govern
  • E-voting, censorship, cyber-warfare

• The way we cure disease
  • digital health, remote surgery
What’s your formal model for the Internet? -- theorists

Aren’t you just writing software for networks? – OS community

You don’t have performance benchmarks?? – hardware folks

But why is the Internet interesting?

What’s with all these TLA protocols? – everyone

But the Internet seems to be working now ... – my parents
A few defining characteristics of the Internet...
Network versus “The Internet”

- There are many kinds of network technologies (switches and links)
  - Ethernet, optical, wifi access points, DSL modems, Infiniband switches, ...

- The Internet is not a new/particular kind of network technology

- Instead, the Internet ties different networks together
  - The Internet
A federated system

Interoperability is the Internet’s most important goal!

The Internet interconnects over 40,000 independently operated networks
A federated system

• Fundamental challenge: how do you interconnect competing entities?
  • Competing network providers must cooperate to serve their customers!

• Leads to a constant tussle between business and technical factors
  • Real-world incentives determine topology, path selection, diagnostics, and more

• And complicates innovation
  • How do you differentiate when interoperability relies on supporting a common protocol?
  • Upgrading “the Internet” is not an option
Tremendous scale

• >5 Billion users (51% of world population)

• 1.24 Trillion unique URLs (web pages)

• Every second, we generate >6000 tweets, >60,000 Google queries, >3M emails

The phrase “Internet scale” is now used refer to such systems
Enormous diversity and dynamic range

- **Technology**: optical, wireless, satellite, copper,…
- **Communication latency**: microseconds to seconds \((10^6)\)
- **Bandwidth**: 1Kbits/second to 1 Terabit/second \((10^8)\)
- **Packet loss**: 0 – 90%
- **Endpoint devices**: sensors, cell phones, datacenters,…
- **Applications**: skype, live video, gaming, remote medicine,…
- **Users**: the governing, governed, operators, selfish, malicious, naïve, savvy,…
Asynchronous Operation

• Fundamental constraint: *speed of light*

• Consider: how many cycles does your 3GHz CPU in Berkeley execute before it can possibly get a response for a message it sends to a server in NY?
  • Berkeley to New York: 4,125 km
  • Traveling to NY and back at 300,000 km/s: 27.5 milliseconds
  • 3,000,000,000 cycles/sec * 0.0275 = 84,000,000 cycles!

• Thus, communication feedback is always *dated*
Prone to Failure

• Many components along a path
  • software, switches, links, network interface cards, wireless access points, modem,…

• Consider: 50 components, that work correctly 99% of time → 39.5% chance communication fail
  • Plus asynchrony → takes a long time to hear (bad) news

Handling failure at scale was dealt with for the first time in the context of the Internet!
Constant evolution

1970s:
• 56,000 bits/second links
• < 100 computers in the US
• Copying files is the “killer” app

Today
• $10^{12}$ bits/second links
• 8B+ devices, all over the globe
• 2.45B people use facebook

Cannot design for a fixed target!
Recap: The Internet is ...

- A federated system ...
- of enormous scale ...
- with tremendous dynamic range and diversity ...
- that is asynchronous in operation ...
- failure prone ...
- and constantly evolving
Recap: The Internet is ...

- Too complex for theoretical models
- “Working code” needn’t mean much
- Performance benchmarks are too narrow

The creation of the Internet required a new design paradigm
(One that changed computer science!)
The Internet design paradigm

- Decentralized control
- A best-effort service model
- “Route around trouble”
- Dumb infrastructure (w/ smart endhosts)
- The end-to-end design principle
- Layering
- Federation via a “narrow waist” interface

A radical departure from systems at the time
Example: a best-effort service model

• Fundamental question: what’s the right service model that a network should support?
  • “contract” between network and its users/end-hosts

• Some possibilities:
  • “guarantee that data will be delivered”
  • “guarantee that data will be delivered within X time”
  • “return a confirmation of successful delivery or an error”

• Instead, what the Internet supports: “best effort” delivery of data
  • No guarantee on whether or when data will be delivered
  • No notification of outcome!
The Internet design paradigm

- Decentralized control
- A best-effort service model
- “Route around trouble”
- Dumb infrastructure (w/ smart endpoints)
- The end-to-end design principle
- Layering
- Federation via a “narrow waist” interface

A radical departure from systems at the time

Now routinely adopted in modern systems (e.g., cloud services)
The Internet design paradigm

• Decentralized control → SDN: centralize? → dSDN: (re)decentralize?
• A best-effort service model
• “Route around trouble”
• Dumb infrastructure (w/ smart endpoints) → NFV: richer in-network services?
• The end-to-end design principle → Edge computing?
• Layering → cross-layer coding
• Federation via a “narrow waist” interface

But it is just one design ...

... and we’re still debating the big questions
Backing up a level

• The Internet poses a design challenge like no other
• From its creation emerged a new design paradigm
• That shaped how we reason about the design of complex systems
  • What’s the right prioritization of goals?
  • What are fundamental constraints?
  • How do we decompose a problem?
  • What abstractions do we need?
  • What are the tradeoffs?
• In short, a lesson in how to architect a (networked) system
• Internet

• Protocols

• Architecture
Network architecture*

- More about thinking rigorously than doing rigorous math
- More about understanding tradeoffs than running benchmarks
- More about practicality than optimality

Done right, can be a powerful thing
What (I hope) CS 168 will teach you

• How the Internet works

• Why it works the way it does

• How to reason through a complex (networking) design problem
Let’s take a quick break
Today

• What is (this course on) the Internet about?

[quick break]

• Class logistics
Teaching Staff (see course website for office hours and sections)
Instructor: Sylvia Ratnasamy

• Background
  • PhD from UC Berkeley
  • Worked in industry ~10 years
  • Returned to UCB to join the faculty in 2011
  • Co-founded a startup in 2016; spent 2021-22 at Google
  • Networking has been my focus throughout

• My teaching style
  • I’m a much better teacher when you engage with my questions!!
  • I talk too fast -- the more bored you look, the faster I talk!
Head TAs (see cs168.io for office hours and sections)

- Sean Kim
- Alex Krentsel
Project TAs (see cs168.io for office hours and sections)

• Silvery Fu
• Tenzin Ukyab
• Kenneth Lien
• Zhihong Luo
Section TAs (see cs168.io for office hours and sections)

• Sarah McClure
• Mark Theis
• Narek Galstyan
Enrollment and wait list

• Class size will not increase

• Wait-listed students will be admitted as and when registered students drop the class
  • Course staff do not process the waitlist!
  • If you’re planning to drop, please do so soon!

• Concurrent enrollment students will be admitted after the wait list is processed
Recordings

• Lectures will be recorded and posted online
  • We will make every effort to release videos on the same day as the lecture
  • See Sean’s post on Ed

• One section will be recorded

• Best-effort attempt to livestream the lecture on zoom, for the first few weeks
  • Will assess as we go - please do not rely on this
Sections

• All sections on Monday
  • Will cover material from the previous week’s lectures

• Go to whichever one you want, but please register your choice online
Class workload

1. Two projects (see website for deadlines)
   - One on routing, one on transport protocol design
   - Goal is to learn networking, not programming
   - No partners
2. Self-tests after class
3. One homework based on a research paper we’ll read
4. Exams: midterm and final
**Grading**

- Course grades curved according to recent guidelines
  - But I reserve the right to grade towards the lower end of the spectrum

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project 1</td>
<td>20%</td>
</tr>
<tr>
<td>Project 2</td>
<td>20%</td>
</tr>
<tr>
<td>Self-Tests</td>
<td>5%</td>
</tr>
<tr>
<td>Homework</td>
<td>5%</td>
</tr>
<tr>
<td>Midterm exam</td>
<td>25%</td>
</tr>
<tr>
<td>Final exam</td>
<td>25%</td>
</tr>
</tbody>
</table>
Extensions / Late Policy

- If you are a DSP student or have extenuating circumstances, fill out the extension form

- No extensions for self-quizzes

- For projects and homework, late assignments are penalized as →

- You may only submit once if you are more than 3 days late

- Projects turned in after Dec 9, 11:59pm receive no credit

- If you receive <50% on any project, you may redo it ONCE for up to 50% of the maximum score

<table>
<thead>
<tr>
<th>Lateness</th>
<th>Penalty</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 24 hrs</td>
<td>-10%</td>
</tr>
<tr>
<td>&lt; 48 hrs</td>
<td>-20%</td>
</tr>
<tr>
<td>&lt; 72 hrs</td>
<td>-40%</td>
</tr>
<tr>
<td>&gt;= 72 hrs</td>
<td>-50%</td>
</tr>
</tbody>
</table>
Self-tests

• Self-test quiz posted after each lecture
  • Available by 5pm the day of the lecture and will remain open for a week

• Scores are not important, but you must try!
  • This will help us, and you, identify what topics need explanation

• Participation counts for 5% of your grade
  • Participation → fill out the form in some sensible manner
  • Can skip up to 3 self-tests without penalty
Exams

• All exams are closed book, open crib sheet

• Exam dates and time can be found on the schedule at http://cs168.io/

• Alternate exams will be offered in time slots directly following the regular exam
  • You must let us know and receive approval from us to take the alternate exam
  • See Sean’s post on Ed for requesting an alternate exam
  • DSP students will be accommodated as needed
  • There will be no additional alternates
Lectures and participation

• Class will be recorded but attendance is **highly** recommended
  • My slides will be available on the class website a few minutes before lecture

• Ask and answer questions!
  • It helps you understand
  • It helps others understand
  • It helps you stay awake
  • It helps me stay awake
  • It’s just more fun for all of us ...

• Do sit towards the front and limit electronic access and **BE QUIET!!**
Questions answered in real-time!

• One or more TAs will be on Ed during lecture

• If you have a quick question and don’t want to ask me, then ask on the real-time thread

• Don’t use this for deep conceptual questions:
  • Ask those of me because you can’t be the only one who is confused

• But if you missed something in passing, ask online in real-time!
Class communications

• Website: cs168.io
  • Assignments, lecture slides, announcements

• Use Ed for intra-class communication as much as possible

• Email cs168@berkeley.edu with any questions
  • Reaches me, Alex, and Sean
Course Material

• Disclaimer: we’re still figuring out how to teach system architecture

• Focus on fundamental questions and tradeoffs
  • The broader design space, rather than the details of the solutions implemented today
  • Ideally, we do this together as a joint design exercise

• You will also have to learn the current design
  • But with a good understanding of where and why it falls short

• You will end up with a mix of the “big picture” and “details”
Fundamental questions

• How do you architect the Internet?
• How do you find a path from source to destination? (routing)
• How do you build reliable communication on top of an unreliable network? (transport)
• How do you share network resources across users? (congestion control)
• How do you federate a set of competing network providers?
• ....
First half of course: basics

• General overview
• Architectural principles
• Routing
• Reliable data transfer
• Naming and Addressing
• Etc.
Second half of course: advanced topics

• Congestion control
• Inter-domain issues

Newer topics:
  • SDN and network management (guest lecture by Scott Shenker)
  • Datacenter networks
  • Cellular networks
  • Read a research paper!
  • 2 guest lectures from the lead architects of Google’s global network
What you will not learn...

• How to setup or operate real networks

• Tiny details of current network protocols or the Linux networking stack

• Instead, you will learn about the fundamental challenges in designing the Internet
  • And quite a bit about how the Internet currently addresses these

• Make sure this is what you’re looking for!
Textbook

  • 5th and 6th editions ok, but translate the reading assignments

• You will not be tested on material we didn’t cover in lecture or section
  • Use as a reference and a source of examples
For next time...

• If you plan to drop, please do so ASAP

• See Sean’s post on Ed regarding alternate exams

• Discussion sections will start on August 29

• Self quizzes start on August 30