Lecture 1:
Course Introduction

CSE 123: Computer Networks
Alex C. Snoeren
Lecture 1 Overview

- Class overview
  - Expected outcomes
  - Structure of the course
  - Policies and procedures

- A brief overview of Computer Networking
  - High-level concepts
  - An end-to-end example
Personnel

- Instructor: Alex C. Snoeren
  - Office hours Thursdays 11am-12 noon or by appointment
  - EBU3B 3114

- Project TAs: Shreeja Kumar and Victor Zhu
  - Office hours Th 2-4pm in EBU3b B240a
  - Office hours Tu 9-11am in EBU3b B260a

- Homework TA: Brajesh Kushwaha
  - Office hours Fri 2-4pm in EBU3B B240a
Prereqs

- CSE120
  - I’ve been instructed to enforce prerequisites
  - If there is room after the add deadline (i.e., no one is on the waitlist with prerequisites filled) then I will consider exceptions
    » Sorry, but I can’t predict what enrollment is going to look like

- Programming experience
  - We will be assigning programming projects in C/C++
  - This course will not teach you C. The TAs will help, but you need to learn it on your own if you don’t already know it.
Expected Outcomes

- This course *will* teach you the *fundamentals* of computer networks:
  - Layering, signaling, framing, MAC, switching, routing, naming, Internetworking, congestion control, router design, etc.
  - At the end of this course you should completely understand what’s actually happening when you view a Web page

- This course *will not* teach you signals and coding
  - Take an EE course to learn about modulation, encoding, etc. on different hardware technologies

- Similarly, we will not cover Internet apps/services
  - CSE124 covers application layer protocols, Web, etc.
  - You *will be able to* pick this up on your own with Google
CSE 123 Class Overview

- Course material taught through class lectures, textbook readings, and discussion sections

- Course assignments are
  - Homework questions (based on lecture)
  - Two substantial programming projects

- Discussion sections are a forum for asking questions
  - Help you get started on the projects
  - Lecture material and homework
  - Additional networking topics

- Discussion board (Piazza.com)
  - The place to ask questions about lecture, hw, projects, etc.
Textbook

Homeworks

- There will be 4 homeworks throughout the quarter
  - Reinforce lecture material…no better practice

- Collaboration vs. cheating
  - You *should* discuss homework problems with others
    » You can learn a lot from each other
  - But there is a distinction between collaboration and cheating
  - Rule of thumb: Discuss together in library, walk home, and write up answers independently
  - Cheating is copying from other student’s homeworks or solution sets, searching for answers on the Web, etc.
  - Suspicious homeworks will be flagged for review
Projects

- There will be at least two programming projects
  - You will have four weeks to complete each of them
  - The first will be assigned NEXT FRIDAY

- The projects must be completed in C/C++
  - We will provide skeleton code for you to use
  - Your job is to fill in the interesting/hard parts
  - The TAs will be available to help with coding

- The projects are INDIVIDUAL assignments
  - All code must be your own
  - OK to discuss design ideas, NOT OK to share/look at code
Espresso Prize
Computer Labs

- You are welcome to use any Linux machine in the labs in the basement of the CSE/EBU3B building
  - Linux running on Intel machines

- You can also use your home machine
  - The project source will work on Windows/OS X (with caveats)
  - Graders will test on ieng6 machines
  - Be sure to test your projects there as well
Exams

- Midterm
  - Friday, October 30th
  - Covers first half of class

- Final
  - Friday, Dec 11th (8am-11am)
  - Covers second half of class + selected material from first part
    » I will be explicit about the material covered

- No makeup exams
  - Unless dire circumstances

- Closed book with crib sheet
  - You can bring one double-sided 8.5x11” page of notes to each exam to assist you in answering the questions
  - Not a substitute for thinking
Grading

- **Homeworks**: 20%
  - Think of these collectively as a take-home midterm

- **Midterm**: 15%

- **Final**: 25%

- **Projects**: 40%
  - Divided evenly among the projects
How *Not* To Pass CSE 123

- **Do not come to lecture / discussion**
  - It’s nice out, class is early, the slides are online, and the material is in the book anyway
  - Lecture material is the basis for exams and directly relates to the projects
  - Besides, the professor thinks he’s funny

- **Do not do the homework**
  - It’s only 20% of the grade
  - Excellent practice for the exams, and some homework problems are exercises for helping with the project
  - 20% is actually a significant fraction of your grade (easily the difference between an A and a C)
How Not To Pass (2)

- Do not ask questions in lecture, office hours, or email
  - Professor is scary, I don’t want to embarrass myself
  - Asking questions is the best way to clarify lecture material at the time it is being presented
  - Office hours and email will help with homeworks, projects

- Wait until the last couple of days to start a project
  - We’ll have to do the crunch anyways, why do it early?
  - The projects cannot be done in the last couple of days
  - Repeat: The projects cannot be done in the last couple of days
Class Web Page

http://www.cs.ucsd.edu/classes/fa15/cse123-a/

- Serves many roles...
  - Course syllabus and schedule (updated as quarter progresses)
    - Lecture slides
  - Announcements
  - Homework handouts
  - Project information
Questions

- Before we start the material, any questions about the class structure, contents, etc.?
This Class in One Slide

- **Protocols & Layering**
  - Manage complexity by decomposing the tasks
  - Standardizing syntax and semantics to support interoperability

- **Naming**
  - Agreeing on how to describe a host, application, network, etc.

- **Switching & Routing**
  - Deciding how to get from here to there
  - Forwarding messages across multiple physical components

- **Resource Allocation**
  - Figuring out how to share finite bandwidth, memory, etc.
A “Simple” Task

- Send information from one computer to another

- Endpoints are called **hosts**
  - Could be computer, iPod, cell phone, etc.

- The plumbing is called a **link**
  - We don’t care what the physical technology is: Ethernet, wireless, cellular, etc.
Actually Quite Complicated

- ROUGHLY, what happens when I click on a Web page from UCSD?

My computer → ? → Internet → www.google.com
Web request (HTTP)

- Turn click into HTTP request

GET http://www.google.com/ HTTP/1.1
Host: www.google.com
Connection: keep-alive
...
Name resolution (DNS)

- Where is www.google.com?

My computer (132.239.9.64) → Local DNS server (132.239.51.18)

What’s the address for www.google.com

Oh, you can find it at 66.102.7.104
Data transport (TCP)

- Break message into packets (TCP segments)
- Should be delivered reliably & in-order

```
GET http://www.google.com HTTP/1.1
Host: www.google.com
Connection: keep-alive
...
```

“and let me know when they got there”
Global Network Addressing

- Address each packet so it can traverse network and arrive at host

My computer (132.239.9.64)

www.google.com (66.102.7.104)
Network Routing

- Each router forwards packet towards destination
Link management (Ethernet)

- Break message into frames
- Media Access Control (MAC)
  - Can I send now? Can I send now?
- Send frame
Physical layer

2.4Ghz Radio
DS/FH Radio
(1-11Mbps)

802.11b Wireless
Access Point

Cat5 Cable (4 wires)
100Base TX Ethernet
100Mbps

Ethernet switch/router

To campus backbone

62.5/125um 850nm MMF
1000BaseSX Ethernet
1000Mbps
For Next Class…

● Browse the course web
  ♦ http://www.cs.ucsd.edu/classes/fa15/cse123-a/

● Read Chapter 1 and start Chapter 2 (up to 2.2)

● Next class: links and signaling

● Drop now or plan to stick it out!