Lecture 1: Course Introduction

CSE 123: Computer Networks
Alex C. Snoeren

First Discussion Monday 9/27
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 Tuesdays 11-12pm or by appointment
  - EBU3B 3114

- Discussion TA: Chris Kanich
  - Monday 2-2:50pm in Pepper Canyon 122
  - Office hours Wed 11-12pm EBU3B B275
  - Homework grader

- Project TA: Marti Motoyama
  - Office hours Mon 3-4pm in EBU3B B275
Prereqs

- CSE120
  - I will approve enrollment for students who have not taken it,
  - But, several parts of the course will be especially challenging
    » You are responsible for doing the extra reading on your own

- 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.

- 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
  - Lecture material and homework
  - Additional networking topics
- Discussion board ([http://webboard.ucsd.edu](http://webboard.ucsd.edu))
  - 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 two programming projects
  - You will have four weeks to complete each of them
  - The first will be assigned NEXT TUESDAY

- The projects must be completed in C/C++
  - We will prove 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
Labs

- We will use the uAPE (B230) lab 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 uAPE machines
  - Be sure to test your projects there as well
Exams

- Midterm
  - Tuesday, November 2nd
  - Covers first half of class

- Final
  - Thursday, December 9th
  - Covers second half of class + selected material from first part
    » I will be explicit about the material covered

- No makeup exams
  - Unless dire circumstances (we all want to start vacation early)

- 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%
  - Each project is 20% of your final grade
How *Not* To Pass CSE 123

- Do not come to lecture
  - 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 (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/fa10/cse123/

- 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 Iphone, etc.

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

- How fast?
  - Bandwidth measured in bits per second
  - Often talk about KBps or Mbps – Bytes vs bits
- How long was the wait?
  - Delay (one-way or round trip) measured in seconds
- How efficiently?
  - Overhead measured in bits or seconds or cycles or…
- Any mistakes?
  - Error rate measured in terms of probability of flipped bit
How long to send a message?

- Transmit time $T = \frac{M}{R} + D$
  - 10 Mbps Ethernet LAN ($M=1\text{KB}$)
    » $\frac{M}{R}=1\text{ms}$, $D \sim 5\text{us}$
  - 155 Mbps cross country ATM link ($M=1\text{KB}$)
    » $\frac{M}{R} = 50\text{us}$, $D \sim 40-100\text{ms}$

- Where are the bits in the mean time?
  - In transit inside the network

- R*D is called the bandwidth delay product
  - How many bits can be “stored” be stored in transit
  - Colloquially, we say “fill the pipe”
Is Not Really So Simple
Layering: A Modular Approach

- Sub-divide the problem
  - Each layer relies on services from layer below
  - Each layer exports services to layer above

- Interface between layers defines interaction
  - Hides implementation details
  - Layers can change without disturbing other layers

- Interface among peers in a layer is a protocol
  - If peers speak same protocol, they can interoperate
Protocol Standardization

- Communicating hosts speaking the same protocol
  - Standardization to enable multiple implementations
  - Or, the same folks have to write all the software

- Internet Engineering Task Force
  - Based on working groups that focus on specific issues
  - Produces “Request For Comments” (RFCs)
    - Rough consensus and running code
    - After enough time passes, promoted to Internet Standards

- Other standards bodies exist
  - ISO, ITU, IEEE, etc.
TCP/IP Protocol Stack

- Application Layer
  - HTTP

- Transport Layer
  - TCP
  - IP

- Network Layer
  - Ethernet interface
  - SONET interface

- Link Layer

CSE 123 – Lecture 1: Course Introduction 24
Encapsulation

HTTP
TCP
IP
Ethernet interface

Payload
Headers

HTTP
TCP
IP
Ethernet interface
Internet Protocol Suite

The Hourglass Model

"Thin Waist"

Applications
Transport
Data Link
Physical

The Hourglass Model

FTP
HTTP
NV
TFTP
TCP
UDP
IP
NET_1
NET_2
...
NET_n
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

CSE 123 – Lecture 1: Course Introduction
Link Layer (e.g. Ethernet)

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

CSE 123 – Lecture 1: Course Introduction
Connecting links

- **Routers/Switches**: moves bits between links
  - *Circuit switching*: guaranteed channel for a session (Telephone system)
  - *Packet switching*: statistical multiplexing of independent pieces of data (Internet)
Putting this all together

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

My computer  www.google.com

Internet
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

http://www.google.com
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)

CSE 123 – Lecture 1: Course Introduction
Sharing access to limited resources
- E.g., a link with fixed service rate

Simplest case: first-in-first out queue
- Queue/serve packets in the order they arrive
- Drop packets when the queue is full

Anybody hear of “Network Neutrality”?
For Next Class...

- Browse the course web
  - http://www.cs.ucsd.edu/classes/sp10/cse123/

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

- Drop now or plan to stick it out!
  - Come see me if you are not yet officially enrolled

- Have a great weekend!