Lecture 1: Course Introduction

CSE 123: Computer Networks
Chris Kanich

Project 1 released tomorrow!

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: Chris Kanich
  - Office hours: Monday & Wednesday 2pm-3pm or by appt.
  - EBU3B 2106 (if not there, try 3140)

- TA: Neha Chachra
  - Discussion: Monday 2:25pm in Pepper Canyon 122
  - Office hours: 1pm-2pm Tuesday & Thursday
  - Responsible for project grading
Prerequisites

- 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 TA 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://www.piazza.com)
  - The place to ask questions about lecture, projects, etc.
  - Private or overly specific questions can be asked via email to ckanich@cs.ucsd.edu
Textbook


- This really actually is a **better book** than the 4th edition.
  - Reading mappings for the 4th edition available upon request.

Quizzes

- Think of these as mini-midterms to check your understanding and prepare for the midterm & final
- Format
  - Two quizzes throughout the session
  - One problem
  - First 20 minutes of class
  - Week 2 & Week 4
  - Specific dates will be announced 2 days in advance

Projects

- There will be two programming projects
  - You will have ~2 weeks to complete each of them
  - The first will be assigned TOMORROW

- 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 TA 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
  - Thursday July 14th
  - Covers first half of class
- Final
  - Friday, July 29th 8 AM – 11 AM (Sorry!)
  - Covers second half of class + selected material from first part
    - I will be explicit about the material covered
  - No makeup exams, early exams
  - If you have to ask, assume the answer will be no.
- 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
  - Exams will require you to apply material, not regurgitate it

Grading

- Quizzes: 10%
  - Think of these as mini-midterms to check your understanding and prepare for the midterm & final
    - In class, first 30 minutes, weeks 2&4
- Midterm: 20%
  - In class, Thursday July 14th
- Final: 30%
  - Material from second half of class will be twice as common as first half
- 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 prepare for the quizzes
  - It's only 10% of the grade
  - Excellent practice for the exams, and some quiz problems are exercises for helping with the project
  - 10% is actually a significant fraction of your grade (difference between an A and a B)

How Not To Pass (2)

- Do not ask questions
  - Chris 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
  - If your question is reasonable, we will work with you until you understand the answer.
- 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

How Not To Pass (3)

- Do not use piazza
  - All questions answered ASAP (ask people from Spring 127)
  - Helping other students understand the material is encouraged and helps incredibly with solidifying one's own understanding
  - If used, Piazza will be a great study guide by end of session
Class Web Page

http://cseweb.ucsd.edu/classes/su11/cse123-a/
http://www.piazza.com – search for UCSD, then 123

- Serves many roles…
  - Course syllabus and schedule (updated as quarter progresses)
  - Lecture slides
  - Announcements
  - Project information
  - Student-instructor and student-student discussion

Questions

- Before we start the material, any questions about the class structure, contents, 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.

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 (\( M=1KB \))
    - \( \frac{M}{R}=1\text{ms}, \ D=5\text{us} \)
  - 155 Mbps cross country ATM link (\( M=1KB \))
    - \( \frac{M}{R}=50\text{us}, \ D \approx 40-100\text{ms} \)
- 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.
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
- 62.5/125um 850nm MMF
- 1000BaseSX Ethernet (1000Mbps)
- To campus backbone

Link Layer (e.g. Ethernet)

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

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 \( \rightarrow \) \text{www.google.com} \( \rightarrow \) 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 \( (132.239.9.64) \) \( \rightarrow \) \text{Local DNS server (132.239.51.18)} \( \rightarrow \) What's the address for www.google.com \( \rightarrow \) 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

Global Network Addressing

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

Resource Allocation: Queues

- 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://cseweb.ucsd.edu/classes/su11/cse123-a/
  - http://www.piazza.com

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

- Drop now or plan to stick it out!
  - Last day to drop w/o a W is July 8th aka next Friday
  - Come see me if you are not yet officially enrolled

Prepare for an AWESOME FUN MONTH OF AWESOME