CS 1652: Data Communication and Computer Networks

Syllabus

Course Website

http://www.cs.pitt.edu/~jacklange/teaching/cs1652-f14

Instructor

Jack Lange, Assistant Professor
Office: Sennott Square, Office #5407
Phone: 412-648-0168
Email: jacklange@cs.pitt.edu
Office Hours: Weds. 2-4PM

Teaching Assistant

TBA
Office: TBA
Email: TBA
Office Hours: TBA

Location and Time

Tues./Thurs., 4:00 - 5:15PM
Sennott Square, Room 6110

Prerequisites

Required
CS1550
Knowledge of C and C++
Unix development experience (gcc, gdb, make, etc)
Unix systems programming experience

Textbooks

James Kurose and Keith Ross
Computer Networking: A Top-Down Approach
This will be the primary textbook for the course. Required readings and homework assignments will come from this book.

Richard Stevens
TCP/IP Illustrated, Volume I: The Protocols
Addison Wesley, 1994
This book will be used as a secondary reference source. You will find it extremely useful during the projects.
Recommended books

Larry L. Peterson and Bruce S. Davie  
**Computer Networks - A Systems Approach**  
Morgan Kaufmann, 2003  
Good supplementary material to the required textbooks

Richard Stevens  
Prentice Hall  
Describes the details of socket programming and IPC on Unix.

Richard Stevens  
**Advanced Programming in the Unix Environment**  
Addison-Wesley, 1992  
A basic book for anyone writing programs that run under Unix

Bjarne Strousup  
**The C++ Programming Language, Special Edition**  
Addison-Wesley, 2000

Overview

This course introduces the underlying concepts behind networking using the Internet and its protocols as examples. There are three goals: (1) to give you an understanding of how networks, especially the Internet, work, (2) to give you experience with large scale systems, and (3) to teach you network programming.

We will cover the first five chapters of Kurose in detail, working our way down the network stack from the application layer to the data-link layer. Concurrent with the lectures, you (in groups of two) will be building a functional TCP/IP stack and a small web server that will run on it. What you build will be ‘real’ your code will interoperate with other TCP/IP stacks and you’ll be able to talk to your web server using any browser on any TCP/IP stack.

This class places an equal emphasis on practical experience as well as theoretical foundations. You will interact directly with parts of our Internet infrastructure and implement several core components. It will be a lot of work, but it will also be a lot of fun, provided you enjoy this sort of thing. We will assume that you do and that you will make a good faith effort. After finishing the course, you will be able to do the following.

- Understand the Internet protocols
- Build implementations of the Internet protocols
- Generalize this knowledge to other networking protocols.
- Be a competent network and systems programmer.
- Think like a networking practitioner
- Read and judge articles on networking in trade magazines
- Begin to read and judge research and technical articles on networking
- Create simplicity and reliability out of complexity and unreliability
- Structure and design software systems to achieve that simplicity and reliability
Communication

- **Website** - Announcements will be made both in class and via the course website. Please check it regularly for clarifications and corrections. Homework and project materials will also be published there.

- **Discussion Group** - A google group has been created for class discussions: [pitt-cs1652-f14@googlegroups.com](mailto:pitt-cs1652-f14@googlegroups.com). Students will be granted membership via their email address. Web address: [http://groups.google.com/group/pitt-cs1652-f14](http://groups.google.com/group/pitt-cs1652-f14)

Projects

Over the course of the quarter, you will implement a user-level TCP/IP stack and a small web server that runs on top of it. Your code will not implement the full functionality of HTTP or TCP/IP, but it will implement enough of it to be able to interoperate with other, complete implementations. In keeping with the top-down approach of Kurose and Ross, you will build this from the web server down instead of from the network card up. I will initially provide you with the whole stack (as object code) and you will implement the web server. Next, we will move lower down the stack, leaving you to implement your own version of TCP. Each layer will have well-defined interfaces that you will fill out.

- 20% Web server (Sockets)
- 50% TCP
- 30% Routing

The implementation language will be C++ and the platform will be Linux. The projects are designed to use a GCC 4.x.x compiler and, make as the build tool. Because you will be working in groups you are strongly encouraged to use CVS for a version control system. You may also find that the C++ standard template library will make your life easier. We have configured a set of machines on a private network in the networking lab for use. You are also welcome to use other machines, but we must be able to compile and run your code on our machines. Note that the Ethernet layer of the code requires that your kernel supports the Berkeley packet filter interface and that you can run your Ethernet card in promiscuous mode to extract and inject raw packets. (i.e. you will need either sudo access or the root password)

To evaluate your project, we will spot-check your source code, compile it, and run randomized testcases on it. When appropriate, we will supply you with examples of such testcases. We will also interview both members of each group.

Homework

Each lecture covers material from the textbook which you will be responsible for reading before you come to class.

There will also be four homework sets that will be periodically assigned to help you improve your understanding of the material.

Late Policy

Unless otherwise indicated, homework and projects are due by midnight on their due date. If you submit an assignment late, a 10% penalty will be accrued for each day it is late.

Exams

There will be a midterm exam and a final exam. The final exam will not be cumulative.
Grading

- 50% Project
- 20% Midterm
- 20% Final
- 10% Homework

Cheating

Since cheaters are mostly hurting themselves, we do not have the time or energy to hunt them down. We much prefer that you act collegially and help each other to learn the material and to solve development problems than to have you live in fear of our wrath and not talk to each other. Nonetheless, if we detect blatant cheating, we will deal with the cheaters as per university guidelines.

Schedule (Tentative)

<table>
<thead>
<tr>
<th>Date</th>
<th>Lecture</th>
<th>Reading</th>
<th>HW/Proj</th>
</tr>
</thead>
<tbody>
<tr>
<td>8/25</td>
<td>Class overview and introduction</td>
<td>1.1 - 1.3</td>
<td></td>
</tr>
<tr>
<td>8/27</td>
<td>No Lecture</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9/2</td>
<td>Introduction cont’d</td>
<td>1.4 - 1.7</td>
<td>Proj 1/HW 1 out</td>
</tr>
<tr>
<td>9/4</td>
<td>Web, HTTP and FTP</td>
<td>2.1 - 2.3</td>
<td></td>
</tr>
<tr>
<td>9/9</td>
<td>Email, DNS, and P2P</td>
<td>2.4 - 2.6</td>
<td></td>
</tr>
<tr>
<td>9/11</td>
<td>Sockets</td>
<td>2.7 - 2.8</td>
<td></td>
</tr>
<tr>
<td>9/16</td>
<td>Lab session</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9/18</td>
<td>Transport intro, multi/demultiplexing, UDP</td>
<td>3.1 - 3.3</td>
<td>Proj 1 in, Proj 2 out</td>
</tr>
<tr>
<td>9/23</td>
<td>Reliable transport principles</td>
<td>3.4</td>
<td></td>
</tr>
<tr>
<td>9/25</td>
<td>Connection oriented transport: TCP</td>
<td>3.5</td>
<td></td>
</tr>
<tr>
<td>9/30</td>
<td>Lab session</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10/2</td>
<td>Congestion Control</td>
<td>3.6 - 3.7</td>
<td>HW 1 in, HW 2 out</td>
</tr>
<tr>
<td>10/7</td>
<td>Midterm review</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10/9</td>
<td>Midterm</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10/14</td>
<td>Lab Session</td>
<td></td>
<td>Proj 2a due</td>
</tr>
<tr>
<td>10/16</td>
<td>Midterm grading review</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10/21</td>
<td>Network Layer introduction</td>
<td>4.1 - 4.3</td>
<td>HW 2 in, HW 3 out</td>
</tr>
<tr>
<td>10/23</td>
<td>Network Layer: IP and ICMP</td>
<td>4.4</td>
<td></td>
</tr>
<tr>
<td>10/28</td>
<td>Lab session</td>
<td></td>
<td>Proj 2b due</td>
</tr>
<tr>
<td>10/30</td>
<td>Routing algorithms</td>
<td>4.5</td>
<td></td>
</tr>
<tr>
<td>11/4</td>
<td>Hierarchical routing, RIP, OSPF, BGP</td>
<td>4.6 - 4.7</td>
<td></td>
</tr>
<tr>
<td>11/6</td>
<td>Data link layer, error detection and correction multiple access protocols</td>
<td>5.1 - 5.3</td>
<td></td>
</tr>
<tr>
<td>11/11</td>
<td>Link layer addressing, Ethernet, hubs and switches</td>
<td>5.4 - 5.6</td>
<td>Proj 2 in, Proj 3 out</td>
</tr>
<tr>
<td>11/13</td>
<td>Lab session</td>
<td></td>
<td>HW 3 in, HW 4 out</td>
</tr>
<tr>
<td>11/18</td>
<td>Wireless and mobile networks (wireless links, 802.11)</td>
<td>6.1 - 6.3</td>
<td></td>
</tr>
<tr>
<td>11/20</td>
<td>IPv6 and related protocols</td>
<td>7.1 - 7.8</td>
<td></td>
</tr>
<tr>
<td>11/25</td>
<td>Thanksgiving Break (No Classes)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/27</td>
<td>Thanksgiving Break (No Classes)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12/2</td>
<td>Advanced Topics</td>
<td>8</td>
<td>HW 4 in, Proj 3 in</td>
</tr>
<tr>
<td>12/4</td>
<td>Final Review</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12/13</td>
<td>Final Exam, 10:00 AM - 12:00 PM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>