Introduction to Networking

Syllabus

Web Page

http://www.cs.northwestern.edu/~pdinda/netclass-f00

Instructor

Peter A. Dinda
1890 Maple Avenue, Room 326
847-467-7859
pdinda@cs.northwestern.edu
Office hours: 2-4pm on Fridays or by appointment

Teaching assistant

Kevin Dill
1890 Maple Avenue, Room 378
847-467-1976
dill@ils.northwestern.edu
Office hours: 2-4pm on Tuesdays and Thursdays or by appointment

Location and Time

1890 Maple Avenue, CS Department classroom, MWF 10-11am

Prerequisites

Required	CS 311 or equivalent data structures course
Required	Knowledge of C and C++
Highly recommended	CS 343 or equivalent operating systems course
Highly recommended	Unix development experience (gcc, gdb, make, etc)
Recommended	Unix systems programming experience

Textbook and other readings (on reserve in library)

James Kurose and Keith Ross, *Computer Networking: A Top-Down Approach Featuring the Internet*, Addison Wesley, 2000 (ISBN 0-201-47711-4) (Textbook)

- Buy the hardcover first edition, not the softcover "preliminary edition." The book includes access to the on-line version at http://www.awl.com/kurose-ross. Alternatively, you can buy on-line access only for $25

Internet Requests For Comments (RFCs) (Required)

- Official specifications for the Internet which are available from http://www.ietf.org/rfc.html

Richard Stevens, Unix Network Programming (volumes 1 and 2), Prentice Hall, 1997, 1998 (ISBN 0-134-90012-X and 0-130-81081-9) (Recommended)

- Describes the nitty-gritty details of socket programming and IPC on Unix

Richard Stevens, Advanced Programming in the Unix Environment, Addison-Wesley, 1992 (ISBN 0-201-56317-7) (Recommended)

- Describes how to think like a Unix systems programmer

Bjarne Stroustrup, The C++ Programming Language, Special Edition, 2000, Addison-Wesley, (ISBN 0201700735) (Recommended)

- Definitive reference to C++

Objectives, framework, philosophy, and caveats

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

We will cover the first five chapters of Kurose, 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 is a learn-by-doing kind of class. You will get your hands dirty by examining parts of our Internet infrastructure and building other parts. 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. We don't want to have to spend too much time measuring your performance. If you care about what we're teaching, you'll do a better job of that yourself, and if you don't care, then you should take some course that you do care about.

You should know that this is the first iteration of this course. We believe that there is sufficient flexibility in the design to dynamically adjust the class's difficulty and work levels as appropriate. We promise that if you make a good faith effort, you will do well.

The goal of the course is to enable you 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

Project

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, I will peal away the layers of the stack, leaving you to implement your own versions. Each layer will have well-defined interfaces that you will fill out.

Here are the layers, as well as each one's percentage of the project grade. Note that the layers in italics will be supplied to you and are included only for completeness.

20 %	**Web server (a)**
	Sockets
50 %	**TCP (b)** *UDP*
30 %	**IP (c)** *ARP*
	Ethernet

The implementation language will be C++ and the platform will be Red Hat Linux 6.2. We hope that you will use g++ 2.95 or later as your compiler, make as your build tool, and CVS as your version control system. You may also find that the C++ standard template library will make your life easier. We have established a set of machines on a private network for use. You are 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.

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

Problem sets will be periodically assigned to help you improve your understanding of the material

Exams

There will be a midterm exam and a final exam

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 Northwestern guidelines.

Schedule

Lecture	Date	Topics	Readings	Homework/Project
	9/18 M	(end of new student week)		
1	9/20 W	Mechanics Introduction: Stacks (OSI and Internet), packet versus circuit switching, history	Chapter 1	Project Part (a) out
2	9/22 F	Fundamental concepts: streams, messages, packets, end-to-end reliability, routing, layering, naming and addressing, services	Chapter 1	Homework 1 out
3	9/25 M	Applications: HTTP and email	Chapter 2, RFC TBD	
4	9/27 W	Applications: CORBA, MPI	TBD	
5	9/29 F	Unix Network Programming	Chapter 2	
6	10/2 M	Unix Network Programming	Chapter 2	
7	10/4 W	(project help day)		Homework 1 in Homework 2 out
8	10/6 F	Transport layer	Chapter 3	
9	10/9 M	Transport layer	Chapter 3	
10	10/11 W	Transport layer	Chapter 3 TCP RFCs	Project Part (a) in Project Part (b) out
11	10/13 F	Transport layer	Chapter 3 TCP RFCs	
12	10/16 M	Transport layer	Chapter 3 TCP RFCs	
13	10/18 W	(project help day)		
14	10/20 F	Transport layer	Chapter 3 TCP RFCs	Homework 2 in Homework 3 out
15	10/23 M	Transport layer	Chapter 3 TCP RFCs	
16	10/25 W	Network layer	Chapter 4	
17	10/27 F	Network layer	Chapter 4	
18	10/30 M	(project help day)		
19	11/1 W	Network layer	Chapter 4 IP RFCs	

20	11/3 F	Network layer	Chapter 4 IP RFCs	Homework 3 in Homework 4 out
21	11/6 M	Network layer	Chapter 4 IP RFCs	Project Part (b) in Project Part (c) out
22	11/8 W	Data-link layer	Chapter 5	
23	11/10 F	Data-link layer	Chapter 5	
24	11/13 M	Ethernet	Chapter 5	
25	11/15 W	Ethernet	Chapter 5 ARP RFC IP over Ethernet RFC	
26	11/17 F	(project help day)		
27	11/20 M	Slack day		Homework 4 in
	11/22-11/26	Thanksgiving vacation		
28	11/27 M	Slack day		
29	11/29 W	Networking Research topics	TBD	Project Part (c) in
30	12/1 F	Networking Research topics	TBD	
	12/4-12/8	Final Exam period		