

CS 340 Introduction to Computer Networking

Instructor

[Yan Chen](#), Assistant Professor
Room 330, 1890 Maple Ave., 491-4946. ychen@cs.northwestern.edu
Office Hours: Thursday. 2-4pm or by appointment, Rm 330, 1890 Maple Ave.

Teaching Assistant

Ashish Gupta
ashish@cs.northwestern.edu
Office Hours: Tu. and Th. 11am - noon(12pm), Rm 246, 1890 Maple Ave.

Location and Time

- Lectures: Tuesday and Thursday 9:30-10:50am, Room 381, 1890 Maple.
- Recitation: Wednesday, 4-5pm, Room 381, 1890 Maple.

Course Description

This course provides an introduction to fundamental concepts in the design and implementation of computer communication networks, their protocols, and applications. Topics to be covered include: overview of network architectures, applications (HTTP, FTP), network programming interfaces (e.g., sockets), transport (TCP, UDP), flow control, congestion control, IP, routing, IPv6, multicast, data link protocols, error-detection/correction, multiple access, LAN, Ethernet, wireless networks, and network security. Examples will be drawn primarily from the Internet (e.g., TCP, UDP, and IP) protocol suite. Over the course of the quarter, students program in C++ on UNIX systems to build web clients and servers, and a fully compatible TCP/IP stack that can run them.

Course Prerequisites

- Required: CS 311 or equivalent data structure course.
- Required: Knowledge of C and/or C++.
- Highly recommended: CS 213 or equivalent computer systems course.
- Highly recommended: CS 343 or equivalent operating systems course.
- Highly recommended: UNIX programming experience (gcc, gdb, make, etc.)

Course Materials

- Required textbooks:

- [Computer Networking: A Top-Down Approach Featuring the Internet](#), [KR], Second Edition, James Kurose and Keith Ross, Addison Wesley, 2002 (Note: buy the hardcover second edition, not the softcover “preliminary edition” or the first edition).
- [TCP/IP Illustrated, Volume I: The Protocols](#), Richard Stevens, Addison Wesley, 1994. Essential for lectures on TCP/IP and projects.
- A set of papers will be made available.
- Recommended books
 - [Computer Networks - A Systems Approach](#), Third Edition, Larry L. Peterson and Bruce S. Davie, Morgan Kaufmann, 2003. Good supplementary material to the required textbooks.
 - Unix Network Programming ([Volume 1, Third Edition, 2003](#), and [Volume 2, Second Edition, 1999](#)) Richard Stevens, Prentice Hall. Describes the details of socket programming and IPC on Unix.
 - [Advanced Programming in the Unix Environment](#), Richard. Stevens, Addison-Wesley, 1992. A basic book for anyone writing programs that run under Unix.
 - [The C++ Programming Language](#), Special Edition, Bjarne Stroustrup, Addison-Wesley, 2000.
- Other references
 - [UNIX for the Impatient](#), 2nd Edition, P. W. Abrahams and B. Larsonk, Addison-Wesley, 1996. A nice introductory book to Unix.
 - [Learning the Unix Operating System \(Nutsell Handbook\)](#), 5th Edition, J. D. Peek, G. Todino-Gonguet, J. Strang, O'Reilly, 2001. Good for UNIX beginner.
 - [Programming with GNU Software](#), M. Loukides and A. Oram, O'Reilly, 1995. Introduction on the tools on UNIX.
 - On coding style: [Code Complete](#), by S. McConnell, Microsoft Press, 1995.
 - [Some other pointers on C/Unix](#) collected by Prof. [Fabián E. Bustamante](#).

Grading

There will be a midterm and a final exam. Exams will be in-class, closed-book, and will cover materials from lectures, required readings and projects. The final exam will not be cumulative.

- Homeworks 10%
- Projects 50%
- Midterm 20%
- Final 20%

Communication

- Course web site: <http://www.cs.northwestern.edu/~ychen/classes/cs340/>. Check it out regularly for schedule changes, clarifications and corrections to assignments, and other course-related announcements.

- Recitation. TA will lecture on complementary materials of the lectures, address questions for homework and projects, and help to prepare the exams. We hope it can help students more efficiently than the one-on-one Q&A in office hours.
- Newsgroup (cs.340.announce and cs.340.discuss) are available for announcement, and posting questions and answers.

Homework

There will be two kinds of homework assignments: reading assignments and textbook-style questions. You should have finished the assigned reading before coming to lecture. In addition, there will be about four sets of written homeworks.

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, I will peel 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.

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

The implementation language will be C++ and the platform will be Red Hat Linux 9.0. We hope that you will use g++2.9.6 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. You'll be using the PCs in the TLAB, which will be specially configured for this class. 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 of such testcases.

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. All the projects are to be done by teams of two students. Each team should work cooperatively on the design, implementation, and testing of their solutions. And the work should be shared in a fair manner. The submissions need to include a simple specification of work undertaken by each party.

Policies

- **Late policy:**
Unless otherwise indicated, homeworks and projects are due by the end of lecture on their due date. If you hand in an assignment late, we will take off 10% for each day (or portion thereof) it is late.
- **Cheating:**
It's OK to ask someone about the concepts, algorithms, or approaches needed to do the project assignments, I encourage you to do so; both giving and taking advice will help you to learn. However, what you turn in must be your own, or for projects, your group's own work; copying other people's code, solution sets, or from any other sources is strictly prohibited. We will punish transgressors severely.

Tentative Schedules

Notes: Reading refers to textbook KR unless denoted otherwise.

Date	Lectures Topics	Reading	Assignment
Tu 1/6	Class overview, Introduction to Networking (Internet, protocol, circuit- and packet-switching)	Ch. 1.1-1.3, 1.8, Handout for project 1	Project 1 out. Find partner (groups of up to 2).
Th 1/8	Introduction to Networking (access networks, physical media, delay and loss, layered architecture)	Ch. 1.4 – 1.7	Homework 1 out.
Tu 1/13	Application Layer (Web, HTTP, FTP, Email)	Ch. 2.1 – 2.4	
Th 1/15	Application Layer (DNS, socket programming)	Ch. 2.5 – 2.8	
Tu 1/20	Socket Programming (cont'd)		
Th 1/22	Transport Layer (intro, UDP)	Ch. 3.1 -3.3	Homework 1 in. Homework 2 out.
Tu 1/27	Transport Layer (reliable data transfer, TCP)	Ch. 3.4 -3.5	Project 1 in. Project 2 out.
Th 1/29	TCP (connection)	Ch. 3.5	
Tu 2/3	TCP (congestion control)	Ch. 3.6 -3.7	Homework 2 in.
Th 2/5	Review for Midterm		
Tu 2/10	Midterm		

Th 2/12	Network Layer (intro, routing principles)	Ch. 4.1 – 4.2	Homework 3 out.
Tu 2/17	Network Layer (IP)	Ch. 4.4	
Th 2/19	Network Layer (IP cont'd, routing: OSPF, BGP)	Ch. 4.4 – 4.5	Project 2 in. Project 3 out.
Tu 2/24	Network Layer (routing cont'd, IPv6, multicast)	Ch. 4.5, 4.7 - 4.8	
Th 2/26	Data Link Layer (intro, error detection/correction)	Ch. 5.1 – 5.2	Homework 3 in. Homework 4 out.
Tu 3/ 2	Data Link Layer (multiple access protocols, LAN)	Ch. 5.1-5.4	
Th 3/4	Data Link Layer (LAN, Ethernet, wireless)	Ch. 5.4 – 5.7	
Tu 3/9	Networking Security	Ch. 7	Project 3 in.
Th 3/11	Review for final/Research in networking		Homework 4 in.

Notes:

1. RS = Richard Stevens