
🏠 Home

📅 Schedule

📁 Assignments

🔬 Project



CS 60: Computer Networks

Lecture: MWF, 1:45pm - 2:50pm, Life Sciences Center 200

X-hour: Thur., 1:00pm - 1:50pm, Life Sciences Center 200

Sign up on Piazza!

Lecturer: [Xia Zhou](mailto:xia@cs.dartmouth.edu) (xia@cs.dartmouth.edu)

Office Hour: 4pm - 5pm on Monday and Friday, Sudikoff 208

TA: [Chuankai An](#), [Yinan Zhang](#)

Section Leaders: [Melo Qiao](#), [Qiuhan Wang](#), [Xianan Hao](#)

Lab Hour: 4pm - 6pm on Wednesday, Friday, and Sunday. Sudikoff
Lab 001

Ever wondered what makes the Internet tick and social networks buzz? Want to gain programming skills that would allow you to implement the Internet? If so this is the right course for you.

This course focuses on the communications protocols used in computer networks: their functionality, specification, verification, implementation, and performance; and how protocols work together to provide more complex services. Aspects of network architectures are also considered. Laboratory projects are an integral part of the course in which networking concepts are explored in depth.

We will cover topics including network applications (e.g., Web, Email, peer-to-peer applications such as BitTorrent, cloud storage services such as Dropbox), TCP and UDP transport protocols, the IP protocol, routing algorithms, medium access layer (e.g., IEEE 802.11 wireless MAC), physical layer over wired and wireless medium, and Internet infrastructure such as

link layer protocols (GSM) and network layer protocols (IP).

domain name system (DNS) - these comprise the Internet's core components.

The course breaks down into lectures, assignments, and a course project. There are no exams for this class: no mid term and no final. We focus on network programming. The projects will run for the last two weeks of the course. There will be no lectures during the last two weeks of the course, so that students can fully focus on their group projects. The projects will be organized around a design review, code review and demo or die event.

Prerequisites

CS 50. We expect you have strong C and UNIX (shell programming, gdb, cvs, process, threads) skills (taught in CS 50) to take this course. All programming assignments and project are in C and Unix.

Textbooks

James Kurose and Keith Ross. [Computer Networking: A Top-Down Approach](#). The sixth edition.

References:

- Larry Peterson and Bruce Davie. [Computer Networks: A Systems Approach](#).
- Andrew S. Tanenbaum. Computer Networks.

Grading Policy

Class and lab participation: 10%

There will be a discussion in class regarding the course reading material. Active involvement in the discussion, the lectures, and the lab will help toward the class contribution part of the grade.

Assignments: 60%

There are six weekly laboratory assignments over the first seven weeks. 10% is given for each lab. Some labs are harder than others but we have a flat grading scheme across all labs. These assignments are to be done individually. The schedule is online, so plan ahead and stay on schedule.

We will provide source code solutions to all labs. The TA will grade your solutions and write up a grade sheet on the correctness, simplicity, and clarity of your code. The instructor will review the TA's grading and grade sheets. Your grade and grade sheet and our source code solution will be mailed to you one week after the lab assignment is turned in. If you have questions about the grade or grade sheet please talk to the TA first. If you are still have concerns see the instructor. Please do not distribute the source

are still have concerns see the instructor. Please do not distribute the source code solutions we provide you with (see honor code).

Important: All assignments will be submitted using svn. Each submission must have a README and makefile. Submitted code that segfaults will not be graded. Students will have to fix the code and resubmit for grading. Students can either take an extension (see below) or pay the late submission tax (see policy for late assignments). If segfaulted code is not fixed the student will receive a zero grade for the lab. This is a senior class and we expect good working code.

Project: 30%

The project is made up of a small team (4 - 5 people) and requires strong collaboration and a problem solving mindset to get the job done. The instructor will put the teams together with each member being responsible to deliver against a part of the overall system design, implementation, testing and integration. The goals of this activity are to help you develop the confidence, skills, and habits necessary to write large computer programs while part of a multi-person team. You will become conversant in software engineering paradigms, and be exposed to various public-domain and open source tools that make the software development process easier. In addition, you will develop vital skills in self-directed learning, problem solving, and communication. The project will have a design and code review as well as the demo. A project report that captures the design and implementation will be submitted as part of the assessment.

Dartmouth Honor Code

The assignments and project are all about writing great code - shell scripts and C. Here is the policy for joint work in CS 60:

First, you may discuss and help each other (e.g., help in debugging, sharing knowledge, giving moral support, getting coffee, etc.) - I promote that as the type of team spirit and joint problem solving skills that is the essence of the course and necessary to do a great project. However, you cannot work jointly on coding up (i.e., writing) your programming assignments. You can talk, discuss solutions, even show snippets of code to solve a problem but you cannot jointly write or share code.

The project phase is different. You can work jointly on writing code. But you cannot take code from anywhere (e.g., the web or any other source). It has to be the joint product of the team. One caveat: no sharing of code between teams. As above, teams can discuss code, show each other snippets on the white board, but not share source code. The project phase of the course is a friendly competition so there isn't a lot of incentive to share code

We hate to state this since you all know it but it is necessary to be explicit here:

You should not under any circumstance look at or use code from previous terms' assignments and projects. We have smart shell scripts that can find similar code. The message is - please don't make life unpleasant for all of us by breaking these rules. The penalties for cheating at Dartmouth are severe, starting with suspension and including expulsion. If you are unsure about anything, please ask.

We take the Honor Code very seriously, so please, if you are unclear on any matter regarding this policy, do not hesitate to see me in office hours, and I will be more than happy to answer your questions.

* This course is based upon the previous course designed by Prof. Andrew Campbell. I deeply appreciate his efforts on the previous course design.