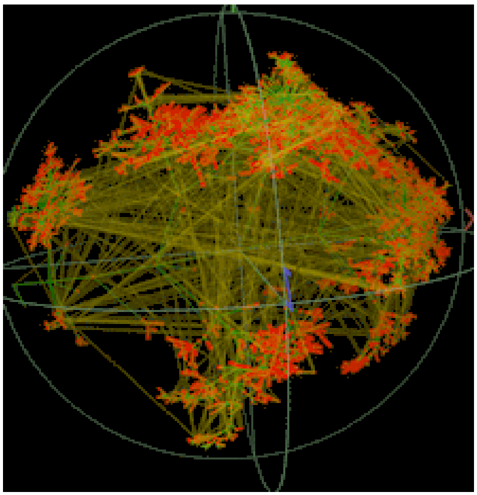


Computer Science 60
Computer Networks -- want to program the Internet?

schedule and notes lab assignments group project resources

Course

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.

The ORC: 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.

Prerequisite: CS 50 and CS 51. You need strong C and Unix (shell programming, gdb, cvs, process, threads) skills (taught in CS 50) to do this course. All programming assignments and project are in C and Unix.

For a summary of the course see syllabus.

Topics covered in this course include network applications (e.g., peer-to-peer applications such as BitTorrent), TCP and UDP transport protocols, flow control, congestion control, the IP protocol, routing algorithms, medium access layer (e.g., IEEE 802.11 wireless MAC) and Internet infrastructure such as domain name system (DNS) - these comprise the Internet's core components.

There is a significant amount of programming in this course requiring a significant time commitment on the part of the student.

The course breaks down into lectures, labs, and a course project - note, there are no exams for this class, i.e., 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 week of the course so students can fully focus on their group projects. The projects will be organized around a design review, code review and demo or die event.

The course will include seven stepwise programming assignments based around building your own

P2P IM system, implementing the Simple Reliable Transport protocol (SRT), Simple Network Protocol (SNP) (including packet forwarding and routing) and Overlay Network (ON). Collectively, we call the transport, network, and overlay software system **DartNet**.

Students will implement DartNet over the course of the programming assignments and then "put it all together" so that their IM application uses their transport and network layer running over an IP overlay network which students also code. Development of DartNet gives students a knowledge base and skill set that mirrors what computer protocol developers execute in industry. So each student will run their IM app over their transport and routed network that they build - huge insights will be gained in doing this. And, believe me when you tell people that you did this as an undergrad, even at Cisco, they'll be way impressed! Having worked in the software networking industry for a decade before falling into academia I'm a big believer in implementing networking software as the essential part of fully understanding complex computer networks - such as, the Internet - else, things remain too abstract. You will write approximately 3000 lines of C code for the assignments and 3500 lines for the group project (shared between the group members). Note, there is some code reuse for the DartNet set of labs. As a result you will tangible programming experience in building a network app, and coding transport and network software. If you code it and you'll understand it. Or as Internet pionneer David Clark famously put it: *"We reject kings, presidents, and voting. We believe in rough consensus and running code."*. David was refering to the working practice of how the Internet Engineering Task Force (IETF) went about developing the core Internet standards.

When and Where

1.45 pm-2.50 pm Mon, Wed, Fri (Life Sciences Center, Room 105)

1.00pm-1.50pm X-hour Thursday (Life Sciences Center, Room 105)

Team

Andrew T. Campbell (Instructor)
campbell AT cs.dartmouth.edu
260 Sudikoff
Office Hours: Monday and Friday 3-4 PM

TA:
Melo Qiao
longquan.qiao.gr@dartmouth.edu
Office hours in Lab 001 or 108 (to be confirmed): Thursday, Friday and Sunday 5-8 PM

Section Leaders:
Graham Findlay
Melissa Queen

The network programmers.

Team

Mail cs60-help@cs.dartmouth.edu

Grading

5% Class and lab participation - jump in, get involved, be a good CS60 citizen.

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

70% - Laboratory exercises (10% for each of the 7 labs)

There are 7 weekly laboratory assignments over the first 8 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 - plan a head. You need to be organized to get through the labs 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 code solutions we provide you with (see honor code).**

Important: all labs 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.

25% - Team project

The project is made up of a small team (approx four 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 overal 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 assesment.

Policy for Late Assignments

Unless prior arrangements have been made, or in case of medical or family emergencies, or specific disabilities, all assignments are subject to the following policy regarding late submissions:

Programming assignments must be submitted according to the schedule; if late, the following grade penalties are assessed:

Late < 24 hours
5% will be deducted from the final score.
Late ≥ 24 hours and < 48 hours
15% will be deducted from the final score.
Late 48 or more hours
Speak to the lecturer.

Please talk to me if you are running behind. The most important thing for you to do is to finish the programming assignments and not worry
about being late.

In that spirit: *Two free pass for 48 hour extension with no penalty*. You really don't want to use these because they mean you are behind and
need to catch up. But they are there if you need then - pressure value of sorts.

Lab Access

In order to obtain access to Sudikoff after hours, and to get into Sudikoff's Lab 001, you will need to have your Dartmouth ID card activated for the appropriate access. To do this, stop by the front desk 101 Sudikoff on a weekday between 8:30am-12:00pm, or 1:00-4:00pm, and bring your Dartmouth ID card -- and let them know you are taking CS60, and require access to Lab 001. You will have to fill out and sign a form stating that you understand the various policies about access to the labs in Sudikoff.

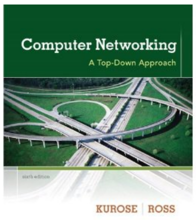
Keep in mind that it may take 24 hours for access to be activated, so please plan ahead!

You will also require a computer account on the CS Linux machines. Contact Andrew Campbell, via email, informing him of your preferred (new) account name.

Please note that the exterior doors of Sudikoff are automatically locked after 6:00pm weekdays, and also every weekend and holiday. In addition, the laboratory doors are locked at all times. You will need your access card to pass through locked doors.

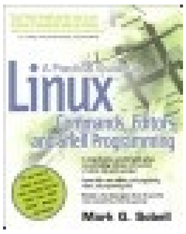
Books

This is the course book (6th Edition now available at the bookstore):



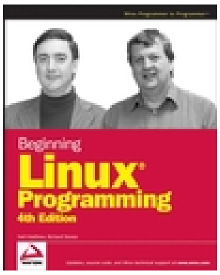
Computer Networking: A Top-Down Approach (6th Edition) by James F. Kurose and Keith W. Ross

Not requireed but if I where to recommend a hands on book on Linux and shell programming it would be this one (lots of good stuff in this):



A Practical Guide to Linux Commands, Editors, and Shell Programming by Mark G. Sobell

Another really good book covering, debugging, processes, threads, and socket programming in clear and easy manner to grasp:



Beginning Linux Programming, 4th Edition by Neil Matthew, Richard Stones

Just to make things clear. Buy the Computer Network book for the course. But if you are interested in hacking C, shell scripts, Linux you might like the other books for your library. They are all really nice books - I love them.

Policy on Joint Work and the 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 CS23:

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 essense 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 we will be more than happy to answer your questions.