CAS CS455/655: Computer Networking
2018.Fall.Syllabus.1.0

Lecturer

Perry Donham / perryd@bu.edu / 202-567-7362
Mailslot: 111 Cummington / MCS138
Blog: http://sites.bu.edu/perryd Twitter: @perrydBUCS
Office: PSY-228C (64 Cummington Mall)

About the course

The course introduces the underlying concepts and principles of computer networks. It presents the building blocks of a network and how these blocks fit together. The course emphasizes the design and implementation of network software (protocols) that transforms raw hardware into a richly functional communication system. Real networks (such as the Internet, Ethernet, Wi-Fi) will be used as examples to reinforce the concepts and demonstrate various protocols. The course also covers applications such as electronic mail, the World Wide Web, and P2P file sharing.

CS455/655 is the first course in a two-semester networking sequence. The second course in the sequence, CS556, covers advanced topics in greater depth (including multimedia networking), and provides more analytical and experimental experience with the design and implementation of network systems.

Note that security is covered in CS558 but foundations of networking and protocol design discussed in CS455/655 provide the basis for understanding some of the vulnerabilities of existing networks and also how to design networks that are inherently more secure.

Learning outcomes

In this course, students learn concepts, principles, and protocols used in computer networks, with the Internet as a case study. They learn how to design and implement protocols at many levels of the network architecture and across different timescales. Topics include: distributed inter-process communication; performance measurements; multiplexing; error and flow control; routing; media access control; etc. They understand protocol correctness and performance evaluation via statistically reliable measurements and discrete-event simulations. They also become familiar with the Wireshark tool, a network traffic sniffer, and use it to deeply understand the operation of several Internet protocols (HTTP, DNS, TCP, NAT, IP, Ethernet, ARP, etc.).

Prerequisites

You are expected to already have the background to read and understand code, write and debug reasonably large (1000-line) programs, and learn new syntax and apply it without much difficulty. You are also expected to learn new tools/programs and run them to test and analyze network protocols. If you are in doubt of your background, please talk to the instructor.

Text

Our text this semester is *Computer Networking: A Top-Down Approach, 7*th edition, Kurose & Ross, Pearson. It should be available on Amazon and at the BU bookstore.

Blackboard
We use Blackboard (http://learn.bu.edu) as a repository for the slide sets for each class, copies of homework assignments, sample code, and announcements. You should be enrolled already, so that when you log on to the site you'll see the course listed.

Piazza is our tool of choice for discussions, including group project team discussions. The Piazza discussion will combine all the sections. We'll use github for course and project code.

Programming
Many of the protocols that we'll look at will be further explored by implementing portions of them in code. Most of the formal labs are in Python, though I might fall back on JavaScript for examples during lectures. If you need a development environment, Pycharm is my go-to for Python, and Webstorm for JavaScript. Both are free (educational license) from JetBrains.com.

Tools
Wireshark is our analysis tool of choice, and many of our labs are built around it. We'll also be using git on github; you are welcome to use whatever client (including their web interface) you prefer. We'll go over versioning systems and git specifically in class.

You are of course welcome to use other tools that you might be comfortable with, and we'll from time to time highlight a tool in class.

Prerequisites

- Solid programming skills in a high-level language (such as C, Java, Python) are required. During the lectures I will mostly use C or C-like code, which is more suitable for exposing low-level system details and achieving higher performance in real implementations. Also, if we provide a skeleton of code in some language as part of an assignment, you will be expected to incorporate your code in that language to implement specific networking functions. In this case, the code is mostly straightforward and does not need knowledge of advanced features of the language.

- A rudimentary understanding of algorithms and their mathematical foundations (CS 112, CS 131) is required.

- A rudimentary understanding of computer architecture and operating systems (CS 210) is required.

- A basic understanding of queuing and probabilistic models, discrete-event simulation and how to collect statistically reliable performance metrics (CS 350) is also helpful.

- You are expected to already have the background to read and understand code, write and debug reasonably large (1000-line) programs, and learn new syntax and apply it without much difficulty. You are also expected to learn new tools/programs and run them to test and analyze network protocols. If you are in doubt of your background, please talk to the instructor.

Grad students

The course is lectured to both senior undergraduate and graduate students. Graduate CS-655 students are expected to complete additional readings/assignments – should we need to discuss this additional material face-to-face, we will do so during a few of the CS 655 discussion sections. CS-655 students are also encouraged to participate in the Network Reading Group (NRG) meetings regularly held on Mondays in MCS 148– please subscribe to NRG and follow announcements from the NRG webpage.

Grades

There are several grading inputs in the course. We'll have a written exam in the first and second half, and you'll also be working on lab exercises during your discussion sections. Most weeks there will be a short quiz covering the previous lecture's material; quizzes are administered during discussion sections. Case studies will be assigned during the semester, and you'll be expected to write a short summary of each in addition to participating in in-class discussions.

The allocation for your final grade looks like this:

Midterm exam	25%
Final exam	30%
Quizzes / Cases	15%
Labs	25%
Discussion attendance	5%

The grading scale is numerical:

96-100	A	80-84	B	65-69	C
90-95	A-	75-79	B-	60-64	C-
85-89	B+	70-74	C+		

If your course grade happens to be close to a boundary, such as an 89.5, I'll bump you up unless your overall course work for some reason doesn't justify it. For quizzes, I'll drop your lowest-scoring quiz when doing final grades. There are no make-ups for quizzes.

Grading is done by class graders and by our TFs; if you think an assignment or quiz was mis-graded, please bring it up with one of them (contact info is posted on Blackboard).

We will use the CS Linux machines (csa1, csa2 and csa3) to grade your lab/programming assignments. Although you may use your own machine, it is your responsibility to ultimately port your assignment to our CS machines to make sure they are graded correctly.

Each assignment will have a due date. If late submissions are allowed for an assignment, there will be a 10% penalty per day for late submissions. But, no late assignments will be accepted after one week from due date, and the last day to submit any late assignments is Thursday 12/7. Extensions may be granted only for religious holidays and certified medical reasons.

No incompletes will be given, except for reasons of dire illness shortly before the end of the course, and only if a significant amount of work has been completed (e.g., attending lectures, handing in most assignments, and attending the midterm).

Attendance is important: I will depart from the textbook and its flow on occasions, and I will not provide backup lecture notes on certain additional details that I will cover in class, so it is imperative that you attend all lectures and take careful notes.

Lab / discussion sections
In addition to lecture time, each of you has an assigned discussion section. We'll use that time in a few different ways. Early in the course, it will be a place to read cases, do quizzes, and any other individual assignments that come along. You'll also be working on lab exercises during this time with feedback and guidance from your TF or me.

Contacting me and office hours
The best way to contact me is by email. You may also IM me at perryd@bu.edu (iMessage). My office is in the Psychology building at 64 Cummington Mall, room PSY-228C. Office hours will be posted on Blackboard. No appointment needed, just drop by if you have a question or want to hang out a bit. If you need to drop something off, my mail slot is in the CS office in MCS-138.

Computing
Generally speaking, you can use your own laptop for he majority of assignments. There might be an instance or two in which I'll ask you to port code to one of the CS cluster machines (csa1, csa2, or csa3) for grading or demonstration. Your operating system shouldn't matter, though I work exclusively on Macs or Linux, and any demos or sample code will be based on those.

Academic Conduct Code
The University and the College take cheating very seriously. Cheating and plagiarism will not be tolerated in any course. Cases will be referred to the Dean's office and may result in loss of credit for an exam or assignment or other disciplinary action.

Assignments must be completed individually. Discussion of issues in network systems is encouraged, but representing the work of another person as your own is expressly forbidden. This includes "borrowing", "stealing" or “buying” programs/solutions or parts of them (whether in printed or electronic form) from others. We may use an automated plagiarism checker. Cheating will not be tolerated under any circumstances. Handing in your own work a day or two late will affect your grade far less than turning in a copy of someone else's work on time!

See the CAS Academic Conduct Code, in particular regarding plagiarism and cheating on exams. A student suspected of violating this code will be reported to the Academic Conduct Committee, and if found culpable, the student will receive a grade of "F" for the course.