

CSEE W4119 - Computer Networks

(Call # 73333)

Course Information

Professor Dan Rubenstein

Spring 2010

Contact Information			
	Dan Rubenstein (Instructor)	Tarun Sharma (TA)	Berk Birand (TA)
Office	CEPSR 816	CEPSR 8LE5	CEPSR 910
Phone	(212) 854-0050	TBD	TBD
e-mail	danr@cs.columbia.edu	ts2579@columbia.edu	berk@ee.columbia.edu
Mailbox	CS Main Office	8LE5	CEPSR 910
Office Hours	Tu, Th 11-12 or by appt.	M 4-5, F 4:30-5:30	Tu, Th 3-4

Course URL: <http://www.cs.columbia.edu/~danr/4119>.

Also see Courseworks (<https://courseworks.columbia.edu/>) for additional handouts, etc.

Course meeting time / location: 1:10pm - 2:25 pm on M,W in 644 Mudd

Pre-Requisites

C or Java programming, Course in algorithms, Course in probability

Description

Topics: Introduction to computer networks and the technical foundations of the Internet, including applications, protocols, local area networks, algorithms for routing and congestion control, security, elementary performance evaluation. Several programming assignments required.

This course is a joint EE/CS course. It requires both skill at programming and math/algorithms/probability.

Grading

Your grade consists of:

20% **Homework:** Unless otherwise specified, homework will be due one week after it is assigned and should be turned directly to the TAs by 5pm on the day that it is due.

You may discuss and work on questions with other students in the class. However, you should write your solutions on your own. In other words, if I were to later ask you to re-derive one of your homework solutions or to solve a similar problem when you were without your friends, you should be able to do so or have a clear understanding of how to approach the problem. This can only be learned by doing, so you should do your homework.

35% **Programming Assignments:** same rules as homework. You should write your own code. Note our ability to check the compiled code for similarities. **Students caught copying code or providing code for copying are subject to disciplinary action, including the possibility of failure or expulsion.**

25% **Mid-term:** March 10 in-class, open book, open note

30% **Final:** TBD by registrar (probably May 10), open book, open note

Reading / Texts

The textbooks provide a guide for the course, but the course is not entirely contained within the required text.

- **Required:** James F. Kurose and Keith W. Ross, *Computer Networking: A Top-Down Approach Featuring the Internet*, 5th ed. Addison-Wesley, 2009, ISBN 0-13-607967-9 (ISBN-13: 978-013-6079675)
- **Optional:** Dimitri Bertsekas and Robert Gallager *Data Networks (2nd ed.)*, Prentice Hall, 1992. ISBN 0-13-200916-1. Significant mathematical treatment (graduate 6000 level - practical stuff a bit out of date).
- **Optional:** Andrew S. Tanenbaum, *Computer Networks (4th ed.)*, Prentice Hall, 2003. ISBN 0-13-066102-3. A lot like Kurose/Ross
- **Optional:** Alberto Leon-Garcia and Indra Widjaja, *Communication Networks: Fundamental Concepts and Key Architectures, 2nd ed.*, McGraw-Hill, 2004. ISBN 0-07-246352X. A bit more mathematical than Kurose-Ross, but less than B&G.

Computing Accounts

You need access to a computer with Berkeley Sockets or the Java equivalent

Cheating

We will follow the “Policies and Procedures regarding academic honesty” laid out by the Computer Science Department at <http://www.cs.columbia.edu/education/honesty>.

Student Feedback

I’m always looking for ways to improve the course. If you have any comments or criticism about the course, or find any mistakes or misleading facts / comments in the lecture, please feel free to contact me. This includes comments on the material being covered, teaching style, pace of the class, workload, etc. I will try and accommodate, but I can’t make any promises...

Things to know about Professor Rubenstein

- I teach what I believe is fundamental material. Often, this means I gravitate toward the theoretical side and de-emphasize practical details, which I assume Columbia students are more than capable of picking up on their own. I expect students to have decent mathematical sophistication (i.e., know probability and algorithms). The way I teach does **not** prepare you directly for a job as a network programmer, or give you the know-how to work at CUIT, or hack Skype, build your own P2P software, etc. It’s not *what* is taught, so much as the thought process behind the evaluation and understanding.
Not everyone agrees with me that this is what should be taught in a networking class. If you disagree with me, the smart thing to do is to drop the course.
- I often write on the board so if you want to know what is going on in class, come to class, or get a friend to take notes. Strangely enough, if you’re paying attention, I believe you learn a lot more when taking notes yourself. For some reason, when you just read off of notes, your brain tricks itself into thinking it’s seen everything when it hasn’t.

- I use the book as a rough guide, but I don't follow it verbatim. I will leave out lots of material that is covered in the book, and will interject material that I think is relevant (i.e., a more theoretical/mathematical treatment than what is provided in the book). **Some students hate this.** Sorry, one book is too hard and outdated for this course (Bertsekas and Gallager), the others (including Kurose and Ross) are too lightweight.
- I respond to e-mail in batches. I get between 50-100 e-mails a day that require a response. I read everything as it comes in (Unless traveling, I check e-mail several times during the day, before I go to bed, when I wake up, etc.) but if an e-mail takes more than a minute to craft a response, I usually wait to answer it. Roughly once or twice a week, I do a sweep of my inbox and respond to e-mails in a batch. So if you have questions on the homework and cannot come to office hours, my advice is to not wait until the last minute if you want me to answer questions via e-mail.

Tentative Course Schedule: CSEE 4119 Spring'09

Date	#	Topics/chapters covered	Reading	Assigned	Due
1/20	1	Course Overview; Protocol Layers & Encapsulation	1		
1/25	2	IP, Connection-oriented and connectionless flows; Socket Programming;	2.1, 2.7, 2.8	PA #1	
1/27	3	APP: DNS/http	2.2, 2.5		
2/1	4	APP: P2P: Search & DHTs	2.6		
2/3	5	APP: P2P: BitTorrent		HW #2	PA #1
2/8	6	*** Catchup ***			
2/10	7	TRA: MUX/DeMux; Reliable Data Transfer: Alternating-Bit Protocol	3.1-3.4		HW #2
2/15	8	TRA: Pipelined Reliable Data Transfer: Selective Repeat / Go-Back-N / Parity & Network Coding Techniques PA #3			
2/17	9	TRA: Flow and Congestion Control	3.6		
2/22	10	TRA: Connection Setup & Teardown; TCP case study	3.5, 3.7		
2/24	11	TRA: Inter-flow fairness (max-min, proportional, TCP)		HW #4	PA #3
3/1	12	*** Catchup ***			
3/3	13	*** Catchup ***			HW #4
3/8	14	*** Catchup and/or Midterm review ***			
3/10	15	MIDTERM (in class)			
3/15	–	Spring Break - no class			
3/17	–	Spring Break - no class			
3/22	16	NET: Switching / Fast Lookups / Flow Identification	4.1-4.4		
3/26	17	NET: Routing I	4.5		
3/29	18	NET: Routing II			
3/31	19	NET: Case Studies (BGP, etc.)	4.6		
4/5	20	NET: Multicast and Anycast	4.7		
4/7	21	LINK: Bit error Detection/Correction techniques	5.1, 5.2		
4/12	22	LINK: Bit error cont'd			
4/14	23	LINK: MAC	5.3		
4/19	24	LINK: MAC II			
4/21	25	*** Catchup and/or review ***			
4/26	26	*** Catchup and/or review ***			
4/28	27	*** Catchup and/or review ***			
5/3	28	*** Catchup and/or review ***			
5/10 at 1:10pm		FINAL EXAM: Location Mudd 644			