

# CMSC 23300/33300

## Networks and Distributed Systems

Department of Computer Science  
University of Chicago

Last updated: January 6, 2015

*Winter 2015 Quarter*

**Dates:** January 6 – March 10, 2015

**Lectures:** TuTh 12:00-1:20 in Ryerson 251

**Discussion:** Tu 3:00-4:20 in Harper 140

**Websites:**

<http://www.classes.cs.uchicago.edu/archive/2015/winter/23300-1/>

<https://github.com/uchicago-cs/cmsc23300/wiki>

**Lecturer:** Borja Sotomayor

**E-mail:** [borja@cs.uchicago.edu](mailto:borja@cs.uchicago.edu)

**Office:** Ryerson 165-A

**Office hours:** After class and by appointment.

### Teaching Assistants

Nathan Bartley	Tiratat “Knot” Patana-anake
<a href="mailto:bartleyn@uchicago.edu">bartleyn@uchicago.edu</a>	<a href="mailto:tiratatp@uchicago.edu">tiratatp@uchicago.edu</a>

TA office hours will be posted on the Piazza site

### Contents of this Document

Course description . . . . .	2
Course organization . . . . .	2
Books . . . . .	5
Grading . . . . .	5
Policy on academic honesty . . . . .	6
Asking questions . . . . .	6
Acknowledgements . . . . .	7

---

## Course description

This course focuses on the principles and techniques used in the development of networked and distributed software. Topics include programming with sockets; concurrent programming; data link layer (Ethernet, packet switching, 802.11, etc.); internet and routing protocols (IP, IPv6, ARP, intra-domain and inter-domain routing, etc.); end-to-end protocols (UDP, TCP); and other commonly used network protocols and techniques.

In this course, students will learn how to...

- ...implement multithreaded client/server applications using sockets.
- ...interpret existing specifications of network protocols, and translate them into code.
- ...design and combine network protocols that form the foundation of the Internet.
- ...develop software collaboratively through the use of version control tools, code reviews, and project management.

In a nutshell, students will learn how the Internet works. By the end of this course, students should understand *everything* that happens “under the hood” when (for example) a web page is requested, from the moment you click on a link in your browser to the moment you get the requested page back.

CMSC 15400 and a working knowledge of the C programming language are strict prerequisites of this course. Students who have not taken CMSC 15400 must speak with the instructor to verify that they meet the prerequisites for this course.

---

## Course organization

The development of three programming projects accounts for the majority of the grade in this course. To successfully complete these projects, students must master fundamental concepts and skills in networking. The class meets two times a week for lectures that provide a conceptual scaffolding for the projects, but will also cover material that is not directly applied in the projects. There is also an optional weekly discussion session primarily intended for in-depth discussion of the projects. Besides the projects, students will also be evaluated with a midterm and a final exam. The course calendar, including the contents of each lecture and project deadlines, is shown in Table 1.

## Projects

Throughout the quarter, students will develop three projects:

1.  $\chi$ **IRC**: Implementing an Internet Relay Protocol (IRC) server (partially compliant with RFC 2810, 2811, 2812, and 2813) using POSIX sockets and pthreads.
2.  $\chi$ **TCP**: Implementing the Transmission Control Protocol.

3. **Simple Router:** Implementing an Internet router using the Mininet network emulator (<https://github.com/mininet/mininet>)

Each project is independent from the others. Students will develop these projects in pairs. Project groups must be formed by Friday, January 9. Groups can be changed from one project to another, but you must inform the instructor that you intend to do so. If your partner drops out of the course or you feel he/she is not contributing to the group's effort, you should make the instructor aware of this.

### **Discussion Session**

The course includes an optional discussion session which will focus primarily on the projects. During these discussion sessions, we will (1) provide suggestions on how to approach the projects, (2) provide general feedback on submitted projects, and (3) provide a general forum to answer questions about the projects. For avoidance of doubt, the discussion sessions are not intended to provide individual feedback or help on the projects; please use Piazza and office hours for that purpose.

Even though these discussion sessions are optional and have no direct impact on your grade, we encourage you to attend them, as they will provide you with very valuable pointers on how to do the projects. At the very least, one of the two members in each group should plan to attend the discussion session each week.

Please note that the first two discussion sessions will be used to provide a refresher on socket programming and concurrent programming, respectively. Students who did not take CMSC 15400 in the previous school year, or who are not comfortable working with sockets or threads in C, are *strongly* encouraged to attend these first two discussion sessions.

### **Graduate Discussion Session**

Graduate students enrolled under the CMSC 33300 code will also participate in four discussion sessions throughout the quarter, where we will discuss seminal papers in the field of computer networks. The date and time of these discussion sessions will be scheduled in consultation with the students registered for the CMSC 33300 code.

Undergraduate students *cannot* enroll in the CMSC 33300 code, but are welcome to attend the graduate discussion sessions.

### **Exams**

There will be a midterm on Tuesday, February 3. This midterm will take place in class, and will only occupy the first 50 minutes of the lecture. The final exam will take place during Finals Week, on Thursday, March 19 from 10:30 to 12:30. An early final exam will be scheduled during 10th week for graduating students.

Questions and exercises related to the projects will make up a substantial part of both exams. Students who have worked on the projects (which requires understanding the material presented in class) should be able to answer these questions with relative ease. However, the exams will also include questions that are not related to the projects, but will be in line with the learning goals outlined at the beginning of this syllabus.

Table 1: CMSC 23300/33300 Winter 2015 Calendar

Week	Date	Lecture	CNESA	TIV1	Project Due	Discussion Session
1	Tu 6 January	Introduction	1	1		Sockets review
	Th 8 January	Ethernet, Switching	2, 3	3		
2	M 12 January				Project 1a	Concurrent Programming review
	Tu 13 January	Switching	2, 3	3		
3	Th 15 January	IP, Routing	3	2, 5, 7		
	Tu 20 January	Routing	3	5, 7		χIRC
4	Th 22 January	TCP	5	12–15	Project 1b	
	Tu 27 January	TCP	5	12–15		χIRC
5	Th 29 January	TCP	5	12–15		
	M 2 February				Project 1c	No discussion session
6	Tu 3 February	Midterm	—	—		
	Th 5 February	DNS	9.3.1	11		
7	Tu 11 February	Application Layer Protocols	9	—		χTCP
	Th 12 February	IPv6	4.1.3	5		
8	Tu 17 February	Internet-level routing	4, 6	5		χTCP
	W 18 February				Project 2a	
9	Th 19 February	Congestion Control	4, 6	16		
	Tu 24 February	Security	8	18		Simple Router
10	W 25 February				Project 2b	
	Th 26 February	IPSec, DNSSec	8	18		
9	Tu 3 March	Wireless Networks	2.7	3.5		Simple Router
	Th 5 March	Wireless Networks	2.7	3.5		
10	Tu 10 March	Review	—	—		No discussion session
	W 11 March				Project 3	

All project deadlines are at 8pm (Chicago local time) on the specified date.

**CNESA:** Computer Networks: A Systems Approach **TIV1:** TCP/IP Illustrated, Volume 1

---

## Books

This course has two *suggested* texts:

- *Computer Networks: A Systems Approach*, 5th edition, Larry L. Peterson and Bruce S. Davie, Morgan Kaufmann 2012.
- *TCP/IP Illustrated, Volume 1: The Protocols*, 2nd Edition, Kevin Fall and W. Richard Stevens, Addison-Wesley 2011.

Both are available for purchase from the Seminary Co-op Bookstore. These texts are *not* required, and most students are able to complete the class successfully without them. However, they can be a good complement to the lectures and projects.

---

## Grading

For undergraduates, the final grade will be based on the projects (60%, each project worth 20%), midterm (15%), and final exam (25%).

For graduate students, the final grade will be based on the projects (45%, each project worth 15%), participation in graduate discussion (15%), midterm (15%), and final exam (25%).

Grades are not curved in this class or, at least, not in the traditional sense. We use a standard set of grade boundaries:

- 95-100: A
- 90-95: A-
- 85-90: B+
- 80-85: B
- 75-80: B-
- 70-75: C+
- < 70: Dealt on a case-by-case basis

We curve only to the extent that we may (and often do) *lower* the boundaries for one or more letter grades, depending on the distribution of the raw scores. We will never raise the boundaries in response to the distribution.

Before the end of the quarter, we will provide students with an estimated grade based on all the work they have done up to that point, including a preliminary and non-binding list of grade boundaries.

## Types of grades

Students may take this course for a quality grade (a “letter” grade) or a pass/fail grade. By default, we assume students are taking the class for a quality grade. We will honor all requests to withdraw or take the class pass/fail *before* the final exam.

Note: *Students taking this course to meet general education requirements must take the course for a letter grade.*

## Late submissions

For the projects, the instructors will collect the latest revision each group commits to their GitHub repository before the deadline. Any work committed after the deadline is ignored and not collected. Each group is allowed four 24-hour extensions during the quarter. More than one extension can be applied to a single submission. i.e., a single 24-hour extension on four submissions, or a 96-hour extension on a single submission.

If extraordinary circumstances (illness, family emergency, etc.) prevent a student from meeting a deadline, the student must inform the instructor *before* the deadline.

---

## Policy on academic honesty

The University of Chicago has a formal policy on academic honesty that you are expected to adhere to:

http:  
`//college.uchicago.edu/policies-regulations/academic-integrity-student-conduct`

In brief, academic dishonesty (handing in someone else’s work as your own, taking existing code and not citing its origin, etc.) will *not* be tolerated in this course. Depending on the severity of the offense, you risk getting a hefty point penalty or being dismissed altogether from the course. All occurrences of academic dishonesty will furthermore be referred to the Dean of Students office, which may impose further penalties, including suspension and expulsion.

Even so, discussing the concepts necessary to complete the projects is certainly allowed (and encouraged). Under *no circumstances* should you show (or email) another student your code or post your solution to a web-page or social media site. If you have discussed parts of an assignment with someone else, then make sure to say so in your submission (e.g., in a README file or as a comment at the top of your source code file). If you consulted other sources, please make sure you cite these sources.

If you have any questions regarding what would or would not be considered academic dishonesty in this course, please don’t hesitate to ask the instructor.

---

## Asking questions

The preferred form of support for this course is through *Piazza* (<http://www.piazza.com/>), an on-line discussion service which can be used to ask questions and share useful information with your classmates. Students will be enrolled in Piazza at the start of the quarter.

All questions regarding assignments or material covered in class must be sent to Piazza, and not directly to the instructors or TAs, as this allows your classmates to join in the discussion and benefit from the replies to your question. If you send a message directly to the instructor or the TAs, you will get a gentle reminder that your question should be asked on Piazza.

Piazza has a mechanism that allows you to ask a private question, which will be seen only by the instructors and teaching assistants. This mechanism should be used *only* for questions that require revealing part of your solution to a project.

Piazza also allows students to post anonymously. *Anonymous posts will be ignored* (you will also get a gentle reminder asking you to not post anonymously). This is a majors-level course: you are expected to feel comfortable sharing your questions and thoughts with your classmates without hiding behind a veil of anonymity.

Additionally, all course announcements will be made through Piazza. It is your responsibility to check Piazza often to see if there are any announcements. Please note that you can configure your Piazza account to send you e-mail notifications every time there is a new post on Piazza. Just go to your Account Settings, then to Class Settings, click on “Edit Notifications” under CMSC 23300. We encourage you to select either the “Real Time” option (get a notification as soon as there are new posts) or the “Smart Digest” option (get a summary of all the posts sent over the last 1-6 hours – you can select the frequency).

---

## Acknowledgements

This syllabus is based on previous CMSC 23300/33300 syllabi developed by Prof. Anne Rogers and Prof. Ian Foster from the University of Chicago.