CMSC 417 — Computer Networks
Fall 2024

1 General

Instructor:	Bobby Bhattacharjee
E-mail:	`bobby@cs.umd.edu`
Class meets:	12:30 p.m. – 1:45 p.m. Tuesdays and Thursdays, CSI 3117
Zoom:	`https://umd.zoom.us/j/94634819862?pwd=TrTmCobrNTKguyhB7sUd5YxGYZxXmf.1`
Office Hours:	3:00 p.m – 4:00 p.m Wednesdays (online)
Class web site:	`http://www.cs.umd.edu/class/fall2024/cmsc417`
Forum:	`https://piazza.com/class/m036spgeu1e326`
Teaching Assistants:	Yang Bai, Alex Yelovich, David Miller
E-mail:	`{yangbai8, ayelovic, dym}@umd.edu`
Office Hours:	12:30 p.m. - 2:00 p.m. Mondays
	2:00 p.m. - 3:15 p.m. Tuesdays, AVW 4132
	3:00 p.m. - 4:30 p.m. Thursdays, AVW 4132 and Zoom

2 Course Summary

This course will cover the basic principles of networking with a focus on protocols, implementations, and issues specific to the Internet. We will study how routing, transport, and internetworking protocols work using the Internet family of protocols as examples. We will selectively implement new protocols and network services. This course has a substantial programming requirement.

3 Expectations: Background

The official pre-requisites for this course are CMSC 351 and and CMSC 330: introductory courses in algorithms and programming languages. I expect you to (a) understand the basics of computer architecture and operating systems and (b) have experience in implementing non-trivial programming projects in C. You should also be familiar with probability theory and analysis of algorithms.

You must also be able to read GNU/UNIX manual pages, and be familiar with a PDF viewer. You must be able to *work in a group*. I cannot emphasize this point enough — I will allow you to (at least partially) form your own groups, and you may choose anyone in this section who you think you can work with. Use the forum as a matching service if you like. However, you will not have the option of not working in a group, and I will not be sympathetic to any complaints at the end of the semester about how your group-mates did not do any work.

To do the programming assignments well, you should be able to design, implement and test non-trivial programs in C. A fixed non-empty set of assignments *must* be coded in C (or C++); you may select your language of choice for the term project. You should be comfortable with at least one debugger. Experience with multi-threaded code may

be helpful for implementing the project, but is not required. Please note that *you* are completely responsible for these pre-requisites and they will not be covered in class or by the TA.

This is a senior-level course, I expect you to be motivated, eager to learn, willing to work hard and make up, on your own, any deficiencies you have.

Expectations: Postcondition

After attending the course, I expect you to be able to:

- Understand the fundamentals of networking protocols, including protocol layering, basic medium access including wireless protocols, routing, addressing, congestion control.

- Understand the principles behind the Internet protocols and some application layer protocols such as `http` and `DNS`, and a few peer-to-peer systems/protocols such as `BitTorrent` and `Bitcoin`.

- Understand some of the limitations of the current Internet and its service model

- Understand the causes behind network congestion, and explain the basic methods for alleviating congestion

- Design, implement, and test substantial parts of network protocols

4 Reading

The required textbook for this course is:

- *Computer Networks: A Systems Approach* by Larry Peterson and Bruce Davie. Morgan Kaufman, 5th Edition, 2011. ISBN 978-0123850591.

The authors have released and updated the book online under a CC-BY 4.0 license, located at `https://book.systemsapproach.org`. The sixth or fourth editions should also be fine. However, you are responsible for reconciling differences (e.g., if I give you an exercise from the fifth edition, you need to find the details yourself.)

Unless you are already intimately familiar with network programming, please obtain the supplementary text. It is quite possibly the best(-written) small book in systems programming.

- *TCP/IP Sockets in C: A Practical Guide for Programmers* by Jeff Donahoo and Ken Calvert. Morgan Kaufmann, 2nd edition, 2009. ISBN 978-0123745408.

I also recommend the following books for reference:

- *Computer Networking: A Top Down Approach* by Jim Kurose and Keith Ross. Pearson, 7th edition, 2016. ISBN: 978-0133594140.

- *TCP/IP Illustrated volume 1* by W. Richard Stevens. Addison-Wesley, 2nd edition, 2011. ISBN: 0-321-33631-6.

I will also ask you to look at classic RFCs and current Internet-Drafts. These are available from `http://www.rfc-editor.org` and `http://www.ietf.org` respectively.

5   Getting Help

Get to know your TAs, and avail of their knowledge of networking during the TA office hours.

Don't forget the forum. Please use the forum to post questions and answers that may be useful to others. Questions of the form "How do I link *foo*?", "Does *bar* option in compiler *bazz* work for you?", or "How do I get to the fifth bit on the third byte of an int under Linux/ARM running on a raspberry pi?", "What is the precise interpretation of homework question III, part b?" should be posted on the forum first. Keep up with the course web page, as it will include updates over the semester.

> Note well: You will be responsible for checking all announcements on the forum and the web site.

My office hours are 3:00 p.m. – 4:00 p.m. on Wednesdays, and will be conducted online using the zoom link posted above. You are welcome to schedule other times after making an e-mail appointment. Please remember to put the string `CMSC 417:` in the subject line of your e-mail. (Messages that have that string in their subject line are certain to be processed quickly).

6   Grading

The grading allocation is given below and *is subject to change.*

Final	20%
In-term exams	30%
Final Project	20%
Assignments	25%
Class Participation	5%

Bonus points on assignments, projects, and exams apply only towards their own category. As an example, bonus points on assignments cannot be used to better your exam grades.

Attendance is not mandatory. You will, however, be responsible for all material covered and assigned in class. The class participation points are to motivate you to speak up in class and to post to the forum.

7   Policy and Academic Honesty

- Turn assignments in *on time.* Unless previously negotiated, you will receive no credit for work that is not turned in on the *day and time* it is due. The only exception is for excused absences as defined by the university (Section V-1.00(G) of the Consolidated USMH & UMCP Policies and Procedures Manual).

- Do not miss exams. Unless previously negotiated, you will receive zero credit for missed exams. Once again, the only exception is for excused absences as defined by the university.

- The punt box rule: on any problem you turn in (homework or exam), if you clearly mark a rectangular box with an "X", we will not grade the problem and you will receive $\frac{1}{10}$th the points for that problem. All punt points are added, and rounded up. This is to stop you from guessing on exams.

- Please read and understand the UMCP code on academic integrity (Section III-1.00(A) of the Consolidated USMH & UMCP Policies and Procedures Manual `http://www.president.umd.edu/policies/iii100a.html`.)

Do not violate it. It is not worth your time (or mine) to be here if you do.

In general, I do not mind your working together on projects. In fact, I encourage it. (Do not work together on in-class exams. If you have trouble with homework questions, post to the forum and ask for hints.) Unless otherwise instructed, homeworks/programs you turn in must be your own, i.e. written completely by you.

> An example for clarification: Suppose Alice and Bob are working on a programming project/homework. It is fine for Alice and Bob to discuss their proposed solutions, work on a whiteboard together, and even ask questions on the forum. Once they figure something out, they can also answer specific questions on the forum. However, they should not post complete solutions (or code snippets), unless specified by the instructor or the TAs.
>
> After discussing their solutions, Alice and Bob go off and write up their work/code up their project. This level of cooperation is allowed and encouraged.
>
> However, if Alice or Bob had simply copied code or text from each other, their effort would be deemed dishonest. They should not use "old versions" of the other's code, or steal throwaway code from a temporary directory or a dustbin, or "look at the other's screens" while typing in their solution.

When you are writing/coding something you will turn in, you should be working by yourself. Even mundane pieces of code that you turn in, and was not provided to you by us, should be written entirely by you. When all is said and done, *you* are the final authority on whether you did something wrong or not, and I'm certain you will know as well as any professor when you do something wrong. Please try not to put yourself in those situations where you have to ask yourself whether something you're about to do is correct or not. If there is any debate about the authenticity of your work, I will have no recourse but to recommend a hearing before the honor board. I sincerely hope this is the last I have to say on this subject for this semester.