NETWORKS
SPRING 2020
COURSE INFORMATION
PROF. KAPLAN

Last updated: 2020-Jan-17

Be sure to read **all** of this document!

1 Introduction

All computing devices—servers, desktops, laptops, tablets, phones (smart and dumb), watches, fitness trackers, televisions, household appliances, cameras, home security systems, medical devices, speakers, cars—communicate with one another. This ubiquitous communication has only occurred within the last twenty years. Networked communication has a great deal to do with the broad and pervasive use of computing in nearly every aspect of our lives; computers would not be nearly as useful or used without networking.

How do two computing devices communicate? How is information signaled from one to another? What if the signal encounters interference? What if the receiver cannot keep up with the sender?

How do three, four, or a few dozen computing devices communicate? How do the signals from one not clobber the signals from another? How does a message get addressed to one, but not to all? How does a new device join the group?

How do hundred, thousands, or billions of devices communicate? How does any one address another, and how do the signals make their way from one to the other, anywhere in the world? How do signals from any one device find their way to the other device? What if too many devices try to contact the same device at once? How are devices added and removed from such a massive collection?

In this course, we will develop answers to these questions within an organized framework that creates a worldwide network via a *layerd stack* of abstracted capabilities. Each layer will use one group of capabilities to build the next, ultimately providing the ability for any program to communicate with another at any location, all through use of a simple interface.

2 The topics

The basic topics are given below, roughly in the order that will we cover them. If you have no idea what some (or many) of them are, don't be alarmed—that is why you're taking this course, after all.

- **Layered network stack:** How is networked communication organized?

- **Physical layer**: How can a medium be modulated to send signals from one place to another? What if the medium is shared (e.g., radio broadcast)?

- **Data link layer**: How do sender and receiver synchronize? How are errors detected and/or corrected?

- **Network layer**: How is data divided into *packets*, and how are the packets routed to their destination? What about routing of packets between two connected local networks?

- **Transport layer:** Across multiple local networks that provide different characteristics, how do two abitrary devices establish a single connection with known characteristics?

- **Socket layer:** The interface by which connections between devices are established and bytes read and written just as they are with files.

- **Application layer:** The protocols (e.g., `http, ftp, rtp`) via which different types of applications communicate.

- **Encryption:** How data can be shared securely between any two devices and their applications.

- **Compression:** How data can be sent using fewer bits.

- **Security:** How do you trust data from devices you've never seen? How do you know the data is really coming from those devices? How do you know that your data is going where you intend it?

These topics are what we directly will be covering, but underlying it all will be the concepts of *interface*, *abstraction*, and *implementation*. The building of capabilities upon capabilities, yielding qualitatively new ones, is a central theme.

This course will be project-intensive. Much of the material will seem easy enough to comprehend when presented in class, but the only way to understand this material thoroughly is to use it. In this case, *using* these ideas requires that you understand an existing implementation of a layer, and then modify or enhance it. Your projects will require you to understand existing code before you then write your own.

3 Lectures, labs, and help

Lectures and labs: This class will meet on **MWF** of each week, from **10:00 am to 10:50 am** in **SCCE A131**. We will occassionally use our class time as a lab to work on projects; be sure to bring your laptop for such days (which will be announced).

You are expected to be present for **all of the lectures and labs**. I will not teach material twice, so if you miss a class meeting, then you're on your own for whatever material was covered that day. If you must miss lecture or lab due to an illness, a curricular conflict (e.g., a Geology field trip), or an emergency situation, contact me and I will arrange to handle the problem. **If you have a extra-curricular conflict** with a lecture or lab—for an athletic event, for a (non-curricular) musical or theatrical performance, to depart early for or arrive late from a vacation, or for any other non-emergency—then **the choice is yours to miss or to attend**. If you choose to miss the class meeting, I do **not** need to know **why** nor even **that** you will be absent. You have elected, voluntarily, not to attend, and you must be prepared to obtain and learn on your own the material that you missed. I recommend that you choose to attend the class meeting when these conflicts arise. Do not underestimate the willingness of those who run extra-curricular programs to support and to accommodate your academic priorities.

Office hours and meetings: If you seek assistance, reinforcement, review, or other opportunities to discuss the course material or assignments, you should see me. There is a link on the course web page for scheduling a time to meet. Please use it; talking with me outside of class is one of the reasons you came to a small college.

Email: Many questions simply do not need an in-person meeting, at least not initially. You should certainly feel free to send email to me with your questions or concerns. Be forewarned, however, that I do not typically respond to email quickly, so do not expect a quick turnaround.

4 Texts and materials

The textbook for this course is *Computer Networks,* 5 sup *th edition*, by Andrew Tanenbaum and David Wetherall.[1]. All other tools for this course—all of the software and documentation—will be provided online.

[1] ISBN-13: 978-9332518742

5 Assignments, deadlines, and extensions

There will be a number of programming projects. The deadline for each will be stated clearly on the assignment. **Late submissions may receive failing grades**. Turn in what you have, and do so on time.

An extension for any assignment **must be requested, in writing** (email counts as *writing*), **at least 48 hours prior to the deadline**. The determination as to whether or not a particular situation merits an extension will be made on a case-by-case basis. Scheduled events are **not** sufficient reason to warrant an extension. Rather, extensions are intended for unusual circumstances that prevent you from planning your time well in order to meet the deadline. Note that a sudden onset of illness or other emergency situation that occurs less than 48 hours before a deadline will be treated as a special case.

6 Exams

There will **comprehensive final exam** given during the final exam period at the semester's end. The time and location of the final exam will be announced when the Registrar's office posts the final exam schedule.

7 Grading

Your final grade will be chosen by my evaluation of how well you have mastered the course material at the semester's end. All of the work that you submit, as well as your participation in class, contributes to my impression of that mastery.

8 Academic dishonesty

You will be expected to do your own work on all assignments and exams in this course. While I encourage you to interact with your classmates and discuss the material and assignments, there is a limit to the specificity of such discussions. I seek to make that limit clear here.

It is acceptable to discuss any assignment for the class with a classmate. You may even discuss your approach to a particular problem, or review relevant material for a problem with another person. However, you **may not show another student your work, nor see another student's work. If in doubt, *ask me***. If you are unsure whether or not a particular kind of communication would rise to the level of academic dishonesty, then you should contact me immediately and find out.