

## **Syllabus**

### **Couse Objectives**

Introduction to computer networking theory, applications, and programming, focusing on large heterogeneous networks. Broad topdown introductions to computer networking concepts including distributed applications, socket programming, operating systems, router support, router algorithms, and sending bits over congested, noisy and unreliable communication links.

This is a hands-on course. The lectures will have tooling and programming demonstration and you will be expected to do similar things for homework and project assignments.

### **Student Expectations / Prerequisites**

This is a hands-on computer science systems class. As such student must be comfortable working with a systems programming language such as C for the programming component of this course. The student should also have access and plan to use a Linux environment to complete required coursework. Acceptable options are Tux – CCI's Linux cluster, a VM on MacOS/Windows, the Linux Subsystem for Windows, or OrbStack (similar to Linux Subsystem for Windows for MacOS). There are also ways to virtualize into a Linux environment using docker containers. One good example is ".devcontainers" in MS Code. I can do a one-off tutorial on how to configure your machine to support development on Tux or in a container using a modern IDE installed on your computer.

### **Office Hours**

I conduct office hours both in person and virtual. My office hours for the term, both in-person and virtual over zoom will be posted on Blackboard under the Instructor Information tab.

### **Discord**

I do not like using emails to manage classes, its not designed for that. Blackboard announcements are also a drag. As such, we have a class Discord that is my preferred venue for communicating with the class. You can DM me over Discord or better yet, use the channels that are open to the entire class to ask questions. If you have a question, share with the class, somebody else also likely has that question.

### **Topics**

1. Introduction – OSI Reference Model, TCP/IP Intro, RFCs, Socket Review
2. Network Definition (and evolution)
3. Network Concepts (packet and network switching, latency, resiliency, packets, encapsulation, etc)
4. Network Standards (e.g., RFCs)
5. Network Protocols and PDUs

6. OSI Conceptual Reference Architecture
7. Concreate IP Network Architecture: Ethernet, TCP/IP, UDP/IP, IPv4, IPv6
8. IP Addressing (IP Address Classes, CIDR, ports, NAT, etc)
9. Socket Programming (Systems Programming)
10. Higher Level Network Programming (HTTP Frameworks, GRPC)
11. Application Layer – HTTP, FTP, DHCP, DNS
12. Transport Layer – TCP & UDP – Deep Dive into TCP
13. Network Layer – deep dive into IP
14. Link Layer (frames, error detection, ethernet)
15. Network Security - TLS
16. Routing
17. Modern Network Considerations (WiFi and Mobile)
18. Software Defined Networks (including Cloud Networking)

### **Expected Learning**

By the end of this course the student should be expected to have a solid grasp on:

- Various Networking Protocols, and how these protocols work together
- The OSI and TCP/IP Reference Model
- Key network addressing concepts, IP addresses, CIDR, MAC addresses
- The difference between connectionless and connection-oriented protocols
- How protocols are designed and support each other, both horizontally and vertically.
- Modern network architecture concerns – security, reliability, resiliency, congestion control.
- How data gets routed around networks and different types of routing protocols
- Key aspects of virtual and physical network hardware – routers, switches, WiFi, etc
- Hands on building of several different types of network protocols via the socket API.

### **Textbook**

Starting in 2024, I will no longer be requiring a course textbook. That said, a highly recommended one that I will reserve that will complement learners that benefit from having a textbook is below.

Computer Networking, A Top Down Approach, Kurose & Ross, 8<sup>th</sup> Edition.

*Note that if you can find the 7<sup>th</sup> edition cheaper you should be fine.*

We will be starting off with a broad overview of computer networks and then traverse topics starting at the high level (network concerns closest to the application), then work down the stack to the data link layer covering how bits flow over the network.

The free online resources listed below will also be useful or helpful in this course – they are also linked under the “Useful Resources” tab on blackboard:

1. Computer Networks: A Systems Approach - <https://book.systemsapproach.org/>
2. Computer Networks, Principals, Protocols & Practice - <https://github.com/cnp3/ebook/releases/download/draft-3rd/CNP3-2021.pdf>
3. Beej's Guide to Network Programming Using Internet Sockets - <https://beej.us/guide/bgnet/>

### **Tools (get these installed in your OS)**

1. "Unix" command line tools, e.g., netstat, arp, traceroute, ifconfig, etc
2. Wireshark
3. C Programming environment (I use VSCode with gcc on my machine)
4. During the end of the term we will be using GNS3 for a course routing lab. You do not need to install this on your computer, CCI provided virtual machines will be used.

### **Programming Expectations**

This full title of this course is Computer Networks: Theory, Applications, and Programming. As such, this course will require you to use a systems programming language to get practical hands-on experience working with network protocols. I will be using C in this course and will accept homework in either C or C++. Many of the assignments will prove you with scaffolded code to get you started, my code will be provided in 'C'. We will be reviewing C programming concepts in this class around socket programming and buffer manipulation in case you need to dust off your C skills.

Throughout this course I will also be demonstrating some newer systems programming languages that have strong networking support such as Go and Rust.

### **Assignments and Labs – 5 in total.**

This course will have 4 assignments that will be spread out over the course (due approx. every 2-3 weeks). Each assignment will have both a programming and a non-programming deliverable.

Once we get through aspects of the network and transport levels, we will have a network design lab where you will design a simple network and try out some routing protocols to manage traffic across the network that you will be designing.

### **Assignments, Due Date and your Late Day Bank**

This course will have assignments that will be spread out over the course as mentioned above. Given my previous experience, students tend to have different levels of readiness, especially with background in programming. I will adjust deadlines for the whole class as necessary, based on my observation of student effort and questions being asked over Discord, and attending office hours (mine or the TAs). One off deadline extensions are not possible, but I do realize that things come up from time to time. If you go into grade center, you will notice that you have a bank of 5 late days to use over the term. Late submissions will be accepted without penalty by deducting from your late day bank. Once you exhaust your bank, late assignments

will take a 50% penalty on day 1, and will not be accepted on day 2+. Assignments are typically due at the end of the week (midnight on Friday).

Note that the late day bank is to accommodate students “when things come up” for whatever reason. This includes work deadlines, needs of other courses, sickness, unexpected travel, and so on. They should be considered as “insurance”, because once they are gone, they are gone. You do not require any permission to submit assignments late – they will be accepted without penalty, without any reason for lateness so long as late days exist in your bank.

### **Expectations of Timely Communication for Hardships**

If, for whatever reason, you find yourself struggling with keeping up with course deliverables, it is important that you let me know right away, including your plan to get back on track. I am here to help you and am willing to be as flexible as possible. However, my flexibility becomes very limited if you come to me weeks after one or more assignment deadlines have been missed with a hardship story expecting me to accept missing work without significant penalty, if even at all.

### **Grading Breakdown**

60% homework, programming assignments, and labs

40% exams (midterm/final)

### **Rough Schedule**

The following is a rough schedule for this course, by week:

1. Introduction to Computer Networks, Socket Programming Tutorial
2. Introduction to Computer Networks
3. Application Layer
4. Application Layer (Deep Dive into HTTP 1.0/1.1/2.0/3.0)
5. Network Layer (UDP/TCP)
6. **Midterm Exam**, Network Layer (UDP/TCP) – Deep Dive into TCP
7. Transport Layer (IP)
8. Transport Layer (Routing) – Deep Dive into Routing Protocols
9. Data Link Layer
10. Software Defined Networking
11. **Final Exam**

### **Grades**

A+ (98-100); A (93-97); A- (90-92)

B+ (87-89); B (83-86); B- (80-82)

C+ (77-79); C (73-76); C- (70-72)

D (60-69)

F (< 60)

### **University Policies:**

This course follows university, college, and department policies, including but not limited to:

- Academic Integrity, Plagiarism, Dishonesty and Cheating  
Policy: [http://www.drexel.edu/provost/policies/academic\\_dishonesty.asp](http://www.drexel.edu/provost/policies/academic_dishonesty.asp)
- Student Life Honesty Policy from Judicial  
Affairs: <http://www.drexel.edu/provost/policies/academic-integrity>
- Students with Disability  
Statement: <http://drexel.edu/oed/disabilityResources/students/>
- Course Add/Drop Policy: <http://www.drexel.edu/provost/policies/course-add-drop>
- Course Withdrawal Policy: <http://drexel.edu/provost/policies/course-withdrawal>
- Department Academic Integrity Policy: <http://drexel.edu/ci/resources/current-students/undergraduate/policies/cs-academic-integrity/>
- Drexel Student Learning  
Priorities: <http://drexel.edu/provost/assessment/outcomes/dslp/>
- Office of Disability Resources: [http://www.drexel.edu/ods/student\\_reg.html](http://www.drexel.edu/ods/student_reg.html)

Students [requesting accommodations](#) due to a disability at Drexel University need to request a current Accommodations Verification Letter (AVL) in the [ClockWork database](#) before accommodations can be made. These requests are received by Disability Resources (DR), who then issues the AVL to the appropriate contacts. For additional information, visit the DR website at [drexel.edu/oed/disabilityResources/overview/](http://drexel.edu/oed/disabilityResources/overview/), or contact DR for more information by phone at 215.895.1401, or by email at [disability@drexel.edu](mailto:disability@drexel.edu).

### **Couse Change Policy**

The instructor may, at their discretion, change any part of the course during the term, including assignments, grade brakdowns, due-dates, and the schedule. Such changes will be communicated to students via the course web site Announcements page. This page should be checked regularly and frequently for such changes and announcements. Other announcements, although rare, may include class cancellations and other urgent announcements will be communicated via the course discord channel.