COMP 431 — INTERNET SERVICES & PROTOCOLS

Spring 2019

Course Syllabus

READ THIS CAREFULLY. YOU ARE RESPONSIBLE FOR KNOWING THE GROUND-RULES FOR THIS COURSE!

Bulletin Description

Application-level protocols HTTP, SMTP, FTP, content delivery networks and media streaming; the Domain Name System; transport protocols TCP and UDP; Internet architecture, naming, addressing, and routing; wired and wireless Ethernet. Sockets programming.

General Course Info

Term: Spring 2019
Department: COMP
Course Number: 431
Section Number: 001
Time: TueThu, 11:00 am – 12:15 pm
Location: MU 0116
Website: https://www.cs.unc.edu/~jasleen/Courses/COMP431

Instructor Info

Name: Dr. Jasleen Kaur
Office: FB 136
Email: jasleen@cs.unc.edu
Office Hours: Tue, 12:15 – 1:15 pm (tentative)

Teaching Assistants

Name: Hasan Alan
Office: TBA
Email: alan@cs.unc.edu
Office Hours: TBA

Name: Junhua Yan
Office: TBA
Email: junhuayan@cs.unc.edu
Office Hours: TBA

Course Description

The Internet has revolutionized the way large portions of the population communicate, learn about the world around them, shop, entertain themselves, and conduct business. But what exactly is the Internet and how exactly does it work? How is it that a request for content finds its way from a web browser to a web server and the reply makes it back? How does this process work for sites like Google when there is no one computer named www.google.com? How do the thousands of web requests and responses that transit the UNC campus network every second share the capacity of the network such that every byte transmitted is received and received in the order sent? More importantly, can one control or even improve the performance of their network connections?
This course provides a technical introduction to the architecture and operation of the Internet. The goal is to understand how such basic services as e-mail, file transfer, the World-Wide Web, and streaming media work and the principles involved in the design of such distributed services. We will study the myriad protocols and non-user-visible services involved in realizing these applications including consideration of the physical networks and switching elements that transmit 1s and 0s. Some of the specific topics we’ll study include:

- Application-level protocols HTTP, FTP, SMTP (e-mail), IMAP, POP, Dash, Bit Torrent, and the Domain Name System.
- Socket programming and client/server computing.
- Transport protocols TCP and UDP.
- Congestion control principles and algorithms.
- The Internet Protocol IP and Internet routing architecture, protocols, and algorithms (DHCP, NAT, OSPF, RIP, BGP).
- Physical-layer transmission technologies such as Ethernet, Wireless LANs, and modems, and interconnection technologies such as bridges, switches, and routers. (If time permits.)

Target Audience

The course is geared towards advanced undergraduate computer science majors and first year graduate students in computer science. Computer networking is a classic topic in a computer science curriculum, all the more so today as virtually no computing task is performed without some computer having to communicate with another computer over a network. You can’t understand how a computer operates today unless you understand networks, the Internet, and communication protocols.

Prerequisites

Prerequisites for this course are:

- COMP 411 (Computer Organization),
- COMP 410 (Data Structures),
- A working knowledge of the Linux program development environment, and
- A working knowledge of the Python programming language.

These are hard prerequisites — you will not survive in this course if you do not have these (or do not develop these within the first week of the course)! Students not meeting these requirements must receive the explicit permission of the instructor to remain in this course.

A “working knowledge of the Linux program development environment” means a general familiarity with the Linux operating system (e.g., basic shell commands for file manipulation, I/O redirection, a file editor, compiler, and debugger) and basic operating system concepts such as processes, I/O, and buffering. The amount of Linux knowledge needed for this course can be picked up within a week – a tutorial is being provided to help students build this, if they already do not have it.

A “working knowledge of the Python programming language” means an ability to write a non-trivial program in Python and run on Linux. (For an example of a non-trivial program, see Homework 1.) See under “Textbooks and Resources” below for Python resources.

Goals and Key Learning Objectives

The goal for this course is for students to understand from a detailed technical perspective, the operation and performance of the Internet and the applications that have come to define the Internet.
Students will learn the classical layered protocol architecture of the Internet. We will start with the “application layer” and proceed “down” the “protocol stack” though the “transport layer,” the “network layer,” and the “data link layer.”

At the end of the course students will understand the key concepts of computer networking including the socket programming interface, the HTTP, SMTP, IMAP/POP, FTP, Dash, as well as the design and operation of content delivery networks CDNs). Students will further understand the vital role of the DNS and the TCP/IP protocol suite (including protocols TCP, UDP, ICMP, IP, DHCP, and NAT) as well as routing protocols such as RIP, OSPF, and BGP, and the wired and wireless Ethernet protocols. Students will further gain an in-depth, technical understanding of the operation of key Internet services such as the Web, email, and file transfer, and how the operation of network protocols effect the performance of these applications.

Textbooks and Resources

The textbook for this course is the 7th edition of Computer Networking: A Top-Down Approach Featuring the Internet, by Kurose and Ross and published by Pearson (2016). We will attempt to cover chapters 1-6 of the text.

This course has a “laboratory” component. You will be asked write several programs that illustrate the main concepts of network protocol design and operation. These programs will be written in Python. The code will not be terribly complex hence fear not if you don’t already know Python. If you’ve made it this far in the computer science program you can easily and quickly pick up Python. A good reference for Java programmers is the website:

http://stackoverflow.com/questions/1052435/moving-from-java-to-python


In addition to these texts, lecture notes will be posted on the course web-page before the start of class. If you miss class, copies of lecture notes, as well as copies of all class handouts, homeworks, assignments, etc, can be found on the course web page.

Online Forums

Sakai will be used for posting grades.

Gradescope will be used for written homeworks.

Piazza.com will be used as a class forum, where you can post and answer questions related to the course material. The course participation credit (see under “Grading”) will consider both in-lecture participation as well as participation (in the form of posting substantial answers) on piazza. If you’re not already registered on piazza.com, please do so. Anonymous posts are allowed (other students do not see identity, but I do).

Grading

Your grade in this class will be based on:¹

- Written homework, laboratory and programming assignments (~50%),
- Midterm examination (~20%),
- Final examination (~30%), and
- Class participation.

¹ The percentages listed are only approximate and are subject to change (by no more than 10%).
Students’ class participation may be used to adjust the final course grade up by a “+” (e.g., from a B to a B+, or from an A- to an A).

Homework Assignments

Homework will be assigned on a nearly-weekly basis and you will have an outstanding homework assignment (and sometimes, two outstanding assignments) through most of this course. Assignments will be either laboratory (programming) exercises or written word problems. The written homework assignments will have questions and problems similar to those given on the midterm and final exams. It is expected that you would have 4-6 laboratory (programming) assignments and 4-6 written assignments throughout the semester.

All written assignments will be due by 10:30 am on the day on which they are due. All programming assignments will be submitted electronically and must be submitted by 10:30 am on the day on which they are due. Written homework will be collected using Gradescope; programming assignments will be submitted by copying files to a special directory on a departmental Linux virtual machine. (The exact procedure will be described on the first homework handout.)

All written homeworks MUST be submitted by their respective submission deadlines. Late submissions for written homeworks will not be accepted at all. Under rare and exceptional cases, if you have a documented, approved excuse (medical or family emergency) for missing the homework, you may receive a pro-rated grade.

All programming homeworks submitted after their deadline are considered late. Assignments that are submitted within 24 hours after the original deadline and must be submitted by 10:30 am on the day on which they are due. Written homework will be collected using Gradescope; programming assignments will be submitted by copying files to a special directory on a departmental Linux virtual machine. (The exact procedure will be described on the first homework handout.)

Exams

There will be one midterm exam. It is tentatively scheduled to be given on March 7, 2019. The final exam for this course is scheduled for Monday, April 29, 2019 at 12:00 PM. The final exam will be comprehensive and covers the entire course.

Office Hours

Two hours of office hours have been scheduled for each working day of the week. In addition, the instructor has office hours on Tuesdays. These office hours can be changed during the semester – students can find the current office hours on the course website.

Students are encouraged to post clarification questions on lecture topics or homework questions on piazza. Students are also encouraged to post queries related to Linux issues or Python syntax or semantics on piazza. Please do not email the TAs or the instructor with such questions – instead, post them on piazza.
Students coming to the office hours would be expected to have read the slides and textbook before-hand, or have made good effort in debugging their programs. If they haven’t done these, they will be turned away and asked to return after doing so.

Course Conduct and the Honor Code

Students are encouraged to work together on homework assignments. Acceptable collaboration on written and programming homework assignments includes:

- Discussing the assigned problems to understand their meaning,
- Discussing possible approaches to assigned problems,
- Obtaining help with issues in using operating system functions, Python syntax or semantics, or programs used for laboratory assignments.

In all cases you must explicitly acknowledge any and all substantive help received from other individuals during the course of the preparation of your homework solution. That is, if you collaborate with other individuals then you must include an explicit acknowledgment in your homework solution of the persons from whom you received aid. You should include the acknowledgement with your Honor Code pledge. Acknowledging others, if done properly, will not adversely affect your grade.

Unacceptable collaboration on written homework includes:

- Copying (verbatim use) of physical papers or computer files,\(^2\)
- Working with individuals who are not presently members of the course,
- Submission of solutions that are jointly authored, or authored either wholly or in part by other individuals,
- Having someone else debug your program,
- Submission or use of code or text from an Internet sources without attribution.

The general rule to be followed is that the strategy and approach of solutions may be developed jointly but all actual solutions (i.e., the final solution) must be constructed and written up individually. Work done jointly should not be done in sufficient detail as to make it a final solution. For example, solutions may sketched out jointly, however each student must construct the final form of their solution individually and write-up or program their own solution.

For programming assignments absolutely no code may be shared between students. If you copy or transcribe one or more lines of code from any other student (including students or persons not in this course) you are in violation of the Honor Code.

Any form of unacceptable collaboration will be considered a violation of the UNC Honor Code and will be reported to the Student Attorney General.

Should questions arise the course of working on a problem please feel free to immediately contact the instructor either by telephone, electronic mail, or by an office visit. In principle, if you work with others in good faith and are honest and generous with your attributions of credit you will have no problems.

\(^2\) This includes computer files that are copied and then edited and/or reformatted.
Topic Outline

The course will closely follow the Kurose and Ross text. The goal is to complete at least most of Chapters 1-4 and as much of Chapter 5 as time permits. The following outline is taken from the text.

Chapter 1: Computer Networks and the Internet
   1.1: What Is the Internet?
   1.2: The Network Edge
   1.3: The Network Core
   1.4: Delay, Loss, and Throughput in Packet-Switched Networks
   1.5: Protocol Layers and Their Service Models
   1.6: Networks Under Attack
   1.8: History of Computer Networking and the Internet

Chapter 2: Application Layer
   2.1: Principles of Network Application
   2.2: The Web and HTTP
   2.3: Electronic Mail in the Internet
   2.4: DNS--The Internet's Directory Service
   2.5: Peer-to-Peer Applications
   2.6: Video Streaming and Content Distribution Networks
   2.7: Socket Programming: Creating Network Applications

Chapter 3: Transport Layer
   3.1: Introduction and Transport-Layer Services
   3.2: Multiplexing and Demultiplexing
   3.3: Connectionless Transport: UDP
   3.4: Principles of Reliable Data Transfer
   3.5: Connection-Oriented Transport: TCP
   3.6: Principles of Congestion Control
   3.7: TCP Congestion Control

Chapter 4: The Network Layer: Data Plane
   4.1: Overview of Network Layer
   4.2: What's Inside a Router?
   4.3: The Internet Protocol (IP): IPv4, Addressing, IPv6, and More
   4.4: Generalized Forwarding and SDN

Chapter 5: The Network Layer: Control Plane
   5.1: Introduction
   5.2: Routing Algorithms
   5.3: Intra-AS Routing in the Internet: OSPF
   5.4: Routing Along the ISPs: BGP
   5.5: The SDN Control Plane
Chapter 6: The Link Layer and LANs

6.1: Introduction to the Link Layer
6.2: Error-Detection and -Correction Techniques
6.3: Multiple Access Links and Protocols
6.4: Switched Local Area Networks
6.5: Link Virtualization: A Network as a Link Layer
6.6: Data Center Networking
6.7: Retrospective: A Day in the Life of a Web Page Request

Disclaimer

“The professor reserves to right to make changes to the syllabus, including due dates, test dates, and grade breakdown. These changes will be announced as early as possible.”