GRS CS 655: Graduate Introduction to Computer Networks

Last updated Wednesday, September 7, 2022

Fall 2022
Prof. Abraham Matta
office: MCS 124A. Tel: (617) 353-8919
e-mail: matta AT bu DOT edu

Computer Science Department
College of Arts and Sciences
Boston University

Important Dates: Last Day to DROP Classes without a 'W' grade is October 11, 2022. Last Day to DROP Classes with a 'W' grade is November 14, 2022.

TIME & PLACE Tuesdays and Thursdays 11:00am-12:15pm. LSE B01, Life Science and Engineering building, 24 Cummington Mall.

COURSE DESCRIPTION The course introduces the underlying concepts and principles of computer networks. It presents the building blocks of a network and how these blocks fit together. The course emphasizes the design and implementation of network software (protocols) that transforms raw hardware into a richly functional communication system. Real networks (such as the Internet, Ethernet, Wi-Fi) will be used as examples to reinforce the concepts and demonstrate various protocols. The course also covers applications such as electronic mail, the World Wide Web, and P2P file sharing.

Note that security is exclusively covered in CS 558, but foundations of networking and protocol design discussed in CS 655 provide the basis for understanding some of the vulnerabilities of existing networks and how to design networks that are inherently more secure. These foundations are also helpful for other courses, including distributed systems, cloud computing, and of course, advanced networking.

LEARNING OUTCOMES In this course, students learn concepts, principles, and protocols used in computer networks, with the Internet as a case study. They learn how to design and implement protocols at many levels of the network architecture and across different timescales. Topics include distributed inter-process communication; performance measurements; multiplexing; error and flow control; routing; media access control; etc. They understand protocol correctness and performance evaluation via statistically reliable measurements, discrete-event simulations, and experiments on an open-source testbed. They also become familiar with the Wireshark tool, a network traffic sniffer, and use it to deeply understand the operation of many Internet protocols (HTTP, DNS, TCP, NAT, IP, Ethernet, ARP, etc.)

PREREQUISITES

1. Solid programming skills in a high-level language (such as C and Java) are required. During the lectures we will mostly use C or C-like code, which is more suitable for exposing low-level system details and achieving higher performance in real implementations. Also, if we provide a skeleton of code in some language as part of an assignment (likely C or Java), you will be expected to incorporate your code in that language to implement specific networking functions. In this case, the code is mostly straightforward and does not need knowledge of advanced features of the language.
2. A rudimentary understanding of algorithms and their mathematical foundations (CS 112, CS 131) is required.
3. A rudimentary understanding of computer architecture and operating systems (CS 210) is required.
4. A basic understanding of queuing and probabilistic models, discrete-event simulation, and how to collect statistically reliable performance metrics (CS 350) is also helpful.

You are expected to already have the background to read and understand code, write and debug reasonably large (1000-line) programs, and learn new syntax and apply it without much difficulty. You are also

expected to learn new tools/programs and run them to test and analyze network protocols. If you are in doubt of your background, please talk to the instructor.

CLASS SCHEDULE (tentative)

This is a tentative syllabus and subject to change. Dates are approximate. Speed and level of coverage will depend to some extent on the maturity and background of the class. Unless noted otherwise, **readings indicated below are from the Kurose & Ross textbook;** more detailed section-by-section reading assignments are noted.

Dates Tue	Dates Thu	Topics	Readings*
9/6	9/8	Overview: introduction to communications connectivity (links and nodes, LANs, WANs, internets, multiplexing, end-to-end channels, performance), standards, protocol architectures (OSI, TCP/IP)	Chapter 1 (read 1.1-1.5)
9/13	9/15		
9/20	9/22	Applications: application programming interface (sockets), client-server, Email (SMTP, MIME, POP, IMAP), Web (HTTP, cookies, caches), DNS, P2P (file sharing, Skype)	Chapter 2 (skip 2.6)
9/27	9/29		
10/4	10/6	Transport services, protocols: UDP, TCP, basics of reliable communication (stop-and-wait, go-back-n, selective-repeat), flow control (sliding window), end-to-end challenges (TCP reliability and flow control), connection management, adaptive retransmission, congestion control (TCP, AIMD, slow start, fast retransmit and recovery), … *[[No lecture on 10/11. Substitute Monday Schedule]]*	Chapter 3 (skip 3.6.2 & 3.7.2)
	10/13		
10/18	10/20		
10/25		Midterm exam	
	10/27	Congestion control: TCP (throughput, fairness) Routing: packets vs. virtual circuits, distance-vector routing (loops), link-state routing	
11/1	11/3		
11/8	11/10	Internetworking: routers, IP datagrams, fragmentation and reassembly, global addressing, forwarding, software-defined networking (OpenFlow), address resolution (ARP), IPv6, scalability (classless routing, NAT, hierarchical routing), intra- and inter-domain routing protocols (RIP, OSPF, BGP), policy (path-vector) routing	Chapters 4 & 5 (skip 4.2, 5.3, 5.6, 5.7)
11/15	11/17		
11/22			
11/29	12/1	*[[No class on 11/24. Thanksgiving Recess 11/23-11/27]]* Point-to-point links, LANs: framing, error detection, MAC protocols (Ethernet CSMA/CD), internetworking (spanning-tree switches, forwarding) Wireless: challenges, elements, characteristics (hidden terminals, signal fading), 802.11 Wi-Fi (CSMA/CA, RTS/CTS), mobility Other Topics (as time permits): video streaming (2.6), … Wrap-up: Last lecture on 12/8	Chapter 6 (skip 6.2, 6.3.4, 6.5, 6.6)
12/6	12/8		Chapter 7 (read 7.1, 7.3.1-7.3.4)
Final Exam date TBD during 12/15-12/21, 2022			

* *Additional readings/assignments will be noted elsewhere.*

OFFICE HOURS

Mondays 2:30-3:30pm and Tuesdays 4-5pm, or by appointment. MCS 124A.

The purpose of the office hours of the Instructor and TFs is to answer specific questions or clarify specific issues. Office hours are not to be used to fill you in on a class you skipped or to explain entire topics. *Please attend class*!

TEACHING ASSISTANTS

Name: Lei Huang
Email: lei AT bu DOT edu
Office Hours: Wednesdays and Thursdays 3:30-5pm, or by appointment. BU CS Student Center.
Discussion Sections: Fridays, A2 (1:25-2:15pm) and A3 (2:30-3:20pm), PSY B39

Name: Tianyi Chen
Email: ctony AT bu DOT edu
Office Hours: Wednesdays and Thursdays 9:15-10:45am, or by appointment. BU CS Student Center.
Discussion Sections: Fridays, A4 (3:35-4:25pm) and A5 (4:40-5:30pm), MCS B37

Grading (except for the final exam) is done by a number of class graders, under the direct supervision of the TFs and the Instructor. If you have an issue with a grade (homework or exam), **please submit a regrade request on Gradescope**. **Grades must be appealed within one week of receipt**, but December 9th is the last day to request a regrade.

RECOMMENDED TEXTBOOK We will follow the top-down approach of this textbook: Computer Networking: A Top Down Approach (8th edition), James F. Kurose and Keith W. Ross, Pearson, 2021. It should be a very good reference. It is available from sites like Barnes & Noble and Amazon. It's fine to also use the earlier 7th edition since the difference between the two editions does not affect the topics that we will cover, and the 7th edition could be perhaps cheaper to buy or rent!

ONLINE TEXTBOOK Another reference is available online: Computer Networking: Principles, Protocols and Practice (aka CNP3), Olivier Bonaventure, 2nd edition.

ONLINE RESOURCES We will be using Piazza for class discussion. The system is highly catered to getting you help fast and efficiently from classmates, the TFs, and the instructor. Rather than emailing questions to the teaching staff, you are encouraged to post your questions on Piazza. If you have any problems or feedback for the developers, email team@piazza.com.

Find our class page at: https://piazza.com/bu/fall2022/grscs655/home. All course material will also be accessible through Piazza. You should regularly visit Piazza for online references and up-to-date information regarding readings, assignments, exam-related material, announcements, *etc.* You can use the discussion board for asking questions and seeking clarifications, whether from the Instructor, TFs, or classmates.

For grading and grade reporting, we will make use of Gradescope.

GRADING POLICY There will be one midterm exam and one final exam. Also, there will be about 3 homeworks that consist of both paper-and-pencil problems and hands-on exercises, 2 programming assignments, and a mini team project. You will typically have 2 weeks to complete each assignment, and 3 weeks for the mini team project. Final grades will be approximately based on the following policy:

- MIDTERM EXAM (25%)
- ASSIGNMENTS (40%): Homeworks (30%), Programming (10%)
- FINAL EXAM (25%)
- MINI TEAM PROJECT (10%)

Unless automated and otherwise specified, the grading of programming assignments will be generally based on the following policy:

- Program: works correctly (60%); in-line documentation (10%); design quality (10%)
- Design document: how it works (5%); tradeoffs and extensions (5%)
- Testing document: complete compilation instructions (5%); thoroughness of test cases (5%)

We will use the CS Linux machines (csa1, csa2 and csa3) to grade your lab/programming assignments. Although you may use your own machine, it is your responsibility to ultimately port your assignment to our CS machines to make sure they are graded correctly.

Each assignment will have a due date. If late submissions are allowed for an assignment, there will be a 10% penalty per day for late submissions. But no late assignments will be accepted after two days from due date, and the last day to submit any late assignments is December 9th. *Extensions may be granted only for religious holidays and certified medical reasons*.

No incompletes will be given, except for reasons of dire illness shortly before the end of the course, and only if a significant amount of work has been completed (*e.g.,* handing in most assignments, and taking the midterm).

Important: I will depart from the textbook and its flow on occasions, and I will not provide backup lecture notes on certain additional details that I will cover in class, so it is imperative that you attend all lectures and take careful notes.

LOGISTICS AND EXPECTATIONS

- We are all required to follow health safety protocols as posted on the COVID-19 Information website.

- **"Unauthorized downloading, uploading, sharing, and/or duplicating course materials including, but not limited to, assignments, exams, quizzes, slides, videos, and any other material created and/or provided by the instructor without the instructor's express permission" is a violation of the Academic Conduct Code.**

COLLABORATION POLICY You are strongly encouraged to collaborate with one another in studying the lecture materials and preparing for exams. You may discuss ideas and approaches to the assignments with others (provided that you acknowledge doing so in your solution), but such discussions should be kept at a high level and should not involve actual details of the code or of other types of answers. ***You must complete the actual solutions on your own***.

ACADEMIC HONESTY We will assume that you understand BU CS's Academic Conduct Code. See also the Graduate School's Academic Conduct Code. Prohibited behaviors include:

- copying all or part of someone else's work, even if you subsequently modify it; this includes cases in which someone tells you what you should write for your solution
- viewing all or part of someone else's work
- showing all or part of your work to another student
- consulting solutions from past semesters, or those found online or in books
- posting your work where others can view it (e.g., online).

Incidents of academic misconduct will be reported to the Academic Conduct Committee (ACC). The ACC may suspend/expel students found guilty of misconduct. ***At a minimum, students who engage in misconduct will have their final grade reduced by one letter grade (e.g., from a B to a C).***

We may use an automated plagiarism checker. Cheating will not be tolerated under any circumstances. Handing in your own work a day or two late will affect your grade far less than turning in a copy of someone else's work on time!

GRS CS 655: Graduate Introduction to Computer Networks

Last updated Wednesday, September 7, 2022

Fall 2022
Prof. Abraham Matta
office: MCS 124A. Tel: (617) 353-8919
e-mail: matta AT bu DOT edu

Computer Science Department
College of Arts and Sciences
Boston University

ASSIGNMENTS

Dates are tentative, until the assignment is actually posted.

Unless otherwise noted, all assignments are to be completed individually. When explicitly specified, you may have the option to work in teams of two on some hands-on or programming assignments. See Academic Honesty on the syllabus.

Tentative Assignment Schedule at a Glance

Assignment	Posted Date	Due Date
HW 1 (Getting Started, Chapter 1)	**Wed 9/7**	**Thu 9/22**
PA 1 (Sockets Programming)	**Wed 9/21**	**Thu 10/6**
HW 2 (Apps & Transport, Chapters 2 & 3) (NO LATE SUBMISSIONS WILL BE ACCEPTED)	**Wed 10/5**	**Thu 10/20**
MIDTERM on Tue 10/25		
PA 2 (Protocol Implementation & Evaluation)	**Wed 10/19**	**Thu 11/3**
HW 3 (Addressing & Routing, Chapters 4 & 5)	**Wed 11/2**	**Thu 11/17**
THANKSGIVING BREAK: Wed 11/23 – Sun 11/27		
GENI Mini Project (NO LATE SUBMISSIONS WILL BE ACCEPTED)	**Wed 11/16**	**Thu 12/8**

DISCUSSION SECTIONS

Discussion sections will be on Fridays. Each student will attend the discussion section that he/she registered for.

The main purpose of the discussion sections is to give you the chance to discuss and ask questions, in a smaller setting under the TF guidance, about specific concepts covered in the lectures or about the assignments, and to allow you to get started on the hands-on Wireshark[1] labs and GENI[2] labs.

As part of the homework assignments, you may be asked to submit short reports or answer questions on hands-on labs / exercises. **The final assignment will be a mini project that asks you to propose and experiment with an extension to some GENI exercise to explore a certain networking aspect more deeply.**

Each student attends one of the discussion sections held on Fridays. **You are expected to use your own laptop and install Wireshark (and other tools) to do the hands-on labs / exercises.**

[1] Wireshark is a "sniffer" that allows you to observe network traffic so you can understand better how application and network protocols work by observing their operation live!
[2] GENI (Global Environment for Network Innovations) is a wide-area experimental testbed.

Tentative Discussion Schedule

Date	Topic
Week 1 (Fri 9/9)	HW1 discussion / *Wireshark Introduction*
Week 2 (Fri 9/16)	HW1 discussion / *GENI Intro*
Week 3 (Fri 9/23)	PA1 discussion / sockets
Week 4 (Fri 9/30)	PA1 discussion / *Wireshark HTTP & DNS*
Week 5 (Fri 10/7)	HW2 discussion / *GENI Scheduling*
Week 6 (Fri 10/14)	HW2 discussion / *GENI Web Server*
Week 7 (Fri 10/21)	PA2 discussion / Midterm review
Week 8 (Fri 10/28)	PA2 discussion / *Wireshark TCP*
Week 9 (Fri 11/4)	HW3 discussion / *GENI Subnetting*
Week 10 (Fri 11/11)	HW3 discussion / *GENI Routing/SDN/NAT*
Week 11 (Fri 11/18)	*GENI Mini-Project (TCP CC, DASH, ...)*
Week 12 (Fri 11/25)	***No sections (Thanksgiving Recess)***
Week 13 (Fri 12/2)	*GENI Mini-Project*
Week 14 (Fri 12/9)	Final review

GUIDELINES ON SUBMISSIONS OF ASSIGNMENTS

Assignments will be available on-line.

You are required to **submit an electronic version** of your homework solutions and your programming/lab documentation/reports. We will use Gradescope to grade your submissions online. You can type and submit your write-up in PDF, or if handwritten, you can scan your write-up and submit a PDF. The timestamp on your electronic submission should indicate submission by the due date/time.

Graded assignments will be available online from Gradescope.

Check for Assignments regularly. Start early!

PROGRAMMING / LAB ASSIGNMENTS

You will be required to **submit an electronic copy of your code using the gsubmit program** (as described below). For supporting documentation of your code and any requested written report, you should submit electronic copies using Gradescope (as described above).

When explicitly specified, you may have the option to work in teams of two.

General Requirements on What to Submit

Unless otherwise specified, the program you submit should work correctly and be documented. You should submit an electronic copy of the following:

1. **Program:** a program listing containing in-line documentation (i.e., comments).
2. **Design document:** a separate file (a page or so) describing the overall program design, a verbal description of "how it works", and design tradeoffs considered and made. Also, describe possible improvements and extensions to your program (and sketch how they might be made).
3. **Testing document:** a separate file describing how to run your program. Specify the steps that must be followed to successfully run your program. Also, describe the tests you ran on your program to convince yourself that it is indeed correct. Also, describe any cases for which your program is known not to work correctly.

If a programming assignment asks you to follow a specific design, then you don't have to describe that design in your documentation, but you must still describe possible improvements and extensions.

HOW TO SUBMIT AN ELECTRONIC COPY USING GSUBMIT FROM YOUR CSA ACCOUNT (*Only plain ASCII files!*)

To access csa1/2/3 login nodes, you will need to activate a CS departmental user account if you don't already have one. If you don't have an account and enrolled in this CS course, you can activate your account at https://www.bu.edu/cs/account/

To submit your assignments, use the gsubmit program from your csa account. Submit only your text and source files (*e.g.*, README file and source .java, .c or .py files), but not compiled files (e.g. .class, .o or .pyc files). Do **not** email your assignments.

gsubmit is an electronic file submission engine, which will submit files or directories of files to the grader so they can be marked.

Every file submitted by a given student for a given assignment should have a unique file name. If a file is submitted with a duplicate name, it will either overwrite the file or generate an error message.

To make it easy for the grader to find the files relating to a specific assignment, all files for each assignment should be stored in a subdirectory called lab1, lab2, hw1, hw2, pa1, pa2, etc. and the entire directory should be submitted.

To submit an assignment:

- Create a subdirectory, say "pa#" where # is the assignment number. This is done using the mkdir command: *e.g.*, **mkdir pa5**
- Copy all files necessary for that assignment into the new subdirectory, using the cp command: *e.g.*, **cp prog1.java pa5**
  Be sure to copy only the files you need to submit into this subdirectory.
- Use **gsubmit** to submit the entire subdirectory: **gsubmit cs655 -cp pa5**
  If submission is successful, a status message will be printed.

To submit a file to an already-submitted subdirectory:

If you only submitted part of the assignment and would like to add another file:

- To submit a file README.txt to subdirectory pa5, type (at the prompt):
  gsubmit cs655 -cp README.txt pa5

To resubmit a file:

- To resubmit a file prog1.java in subdirectory pa5, first un-submit (i.e., remove) the file:
  gsubmit cs655 -rm pa5/prog1.java
- Then resubmit it:
  gsubmit cs655 -cp prog1.java pa5

List all files which you have submitted:

- To list all files that you have submitted, type: **gsubmit cs655 -ls**

Looking at a file which has already been submitted:

- To look at a file that has already been submitted, type: **gsubmit cs655 -cat pa5/prog1.java**
- You can store this in a file foo.java by typing **gsubmit cs655 -cat pa5/prog1.java > foo.java**

Where do submitted files go?

Each student who submits an assignment has a subdirectory created to hold his/her files, in a directory for the specified course. This is called the student's "submission spool directory".

How can the grader tell when a file has been submitted?

Every gsubmit command is automatically logged in a log file, along with a time stamp.

For further information:

Note - The information in this document is taken from the gsubmit man page.
For further information, type: **man gsubmit**
