

UNIVERSITY OF CALIFORNIA, LOS ANGELES  
DEPARTMENT OF COMPUTER SCIENCE

**CS118 Computer Network Fundamentals**

Fall 2022

**MEETING TIMES**

Monday Wednesday 4:00-5:50 PM

**INSTRUCTOR**

George Varghese  
Engineering VI 477  
Tel: (office) 825-7649  
email: varghese@cs.ucla.edu

**LECTURE**

Monday Wednesday 4:00-5:50 PM, Dodd 121  
Office hours: Monday 6:00-7:00 PM, Engineering VI 477

**COURSE PHILOSOPHY**

CS118 assumes no previous background in networking. The focus of the course will be on learning various principles and techniques so you can learn networking *from a designer's point of view*.

UCLA was the first Internet node and Len Kleinrock and Mario Gerla were early Internet pioneers. Vint Cerf who wrote much of TCP and IP was a graduate student at UCLA. Later, Deborah Estrin and Greg Pottie helped start the sensor network revolution.

But we are not done yet at UCLA as the networking faculty are working on new innovations in networking. I hope our UCLA students will help pioneer the next revolution in networks, and this course will get you to start thinking about what you can do.

**FEATURES FOR FALL 2018**

- **Interviews with Internet Inventors:** In the past two years, we have had interviews with Internet inventors by Zoom which students participated in. This year we will not have that in a physical class but I hope to show clips of these interviews by students in CS 118 in Fall 2020 interviewing Internet pioneers like Len Kleinrock (packet switching), Bob Metcalfe (Ethernet), Radia Perlman (Bridge Spanning Tree, and OSPF Routing), Vint Cerf (TCP/IP) and Paul Mockapetris (DNS). We hope these interviews will inspire you to be creators yourself.
- **Mini Flip:** In a classic flipped classroom, the class is recorded and the students watch the video ahead of class, and class is reserved for discussion and problem solving. For all classes

except a few, I hope to post a mini-flip: a short video (10 minutes) of the main idea of the class that you watch before class so you can come prepared for the full lecture.

- **Email:** I am fairly slow with email except where there is an emergency. So please try the TAs first and they can get me if there is need for my input. This is especially so for HWs but also for other clarifications on the material.

## COURSE OBJECTIVES

- A complete understanding of 1 network system (TCP/IP).
- The unifying systems ideas (multiplexing, resource allocation, naming and addressing, security) that are common to Operating Systems, Databases, and Computer Architecture.
- Insight into designing a protocol.
- Insight into real world constraints: what will life look like in industry?
- Relevant Homework.
- Knowledge of what's going on currently in the networking world and where networking jobs are to be found.

## WORK INVOLVED

- **Laboratory and Programming Exercises**

There will be two programming assignments. The first will be a CRC programming assignment and the second will be a moderately difficult assignment to program routers or a variant of TCP. You will do both programs in C or C++; programs will be done individually by each student.

Students are assumed to be competent in C or C++. Students not familiar will have to learn those outside the class room.

- **Exams**

There will be one midterm exam and a final, all open book. I allow students to keep a cheatsheet of 1 page to the class not because I do not want them to look at their full notes during the exam. This is because it would waste time as most of the exam requires thinking. Remember this class is not about memorizing networking facts but about understanding networking *ideas*.

- **Homework**

There will be 4 homework problems handed out at least every two weeks. They will be due (unless otherwise stated) at the start of class on the due date, sometimes in a week and sometimes a bit longer. Because of the large class size, I will only randomly pick 1 or 2 of the questions to grade and base your grade on those. We will not accept homework after the class starts. Feel free to discuss the homework with friends after you have thought about it by yourself. Also DO write up the homework by yourself in your own style.

At times, I will put in an interesting no-grade problem for the people who are really interested. Since the focus is not on programming (which I assume you will learn from other courses) but on concepts, trying the homeworks yourself is important for the tests.

- **Reading**

I post the slides I use on BruinLearn on the day of class. I will also supplement the lecture with my own notes posted on BruinLearn before the exam to encourage you to come to lecture to solve the homework problems (the slides are hard to decipher without coming to class, and for most people who have taken my class in the past, lectures have been fun).

**GRADING** Grades will be assigned on a curve only if the average test and homework scores are very low. If everyone does well, everyone will get good grades. However, there will be a special *A\** grade (reported in your grade sheet as an *A+*) that I will keep track of, generally for the highest score, but not always. If you are applying for graduate school or for a job, and have an *A\**, I would be happy to write a very strong recommendation letter for you. I will do my best to help such students with job and graduate school recommendations.

The breakup of points for grading is:

|                       |      |
|-----------------------|------|
| Homeworks             | 20 % |
| Programming Project 1 | 15 % |
| Programming Project 2 | 25%  |
| 1 Midterm             | 15 % |
| Final                 | 25 % |

- LECTURE PLAN (subject to change)**
1. Sept 26nd: Intro Lecture 1: Layering, Abstractions, All you need to know about Networking, a full overview
  2. September 28th: Physical Layer 1: The basic limits, noise, signal strength and interference, **HW 1 Handed out**
  3. October 3rd: Physical Layer 2: Clock recovery, coding
  4. October 5th: Media
  5. October 10th: Data Link 1: Framing: grouping bits into units, **HW 1 Due**,
  6. October 12th: Data Link 2: Error Detection and Checksums, **Project 1 Handed out, HW 2 Handed out**
  7. October 17th: Data Link 3: Error Recovery: the Protocol Plays,
  8. October 19th: Data Link 4: Multiaccess protocols and the Historical Ethernet, **HW 2 Due**
  9. October 24th: Data Link 5: Bridging: the class will reinvent a classic design, **Project 1 Due**
  10. October 26th: **Midterm**, Sample midterm and prior midterms will be posted a week before, and I hope to do an midterm review the previous week. Midterm will be in class during class hours.
  11. October 31st: Routing 1: Bridges versus routers, Addressing, **Programming Assignment 2 handed out, due Dec 3.**
  12. November 2: Routing 2: IP Forwarding, fast lookups, packet scheduling, **HW 3 handed out**
  13. November 7: Routing 3: Route Computation: Distance Vector and Link State,

14. November 9: Routing 4: BGP: or how competing ISPs work together, **HW 3 Due, HW 4 handed out**
15. November 14: Domain Name Service and how folks like Akamai hack it,
16. November 16: TCP 1: The socket abstraction of two connected queues, **HW 4 due**
17. Nov 21: TCP 2: TCP 2: Reliability
18. Nov 23: TCP 3: Congestion control, reacting to Internet overloads (November 24, 25th: Thanksgiving Holiday, does not affect class )
19. Spare lecture in case we fall behind schedule: otherwise Final exam review in class.
20. Nov 30: Last Lecture: Overview of class, **Project 2 due**
21. Monday, Dec 2nd, **Final Exam** (will try to do an exam review the previous week)

**TEXT** 22. Dec 9th: our goal to post final grades

*Computer Networking: A Top Down Approach Featuring the Internet*, Jim Kurose and Keith Ross. (Nice book with lots of stuff on multimedia and security that's in no other book.) (Optional: only if you really, really want a textbook on all of networking: it's the one most universities use. I will give you notes that are all you need for doing the exams and HWs.)

*Interconnections*, Radia Perlman, Addison-Wesley, 1992. As with the Kurose book it's completely optional and only if you want a classic reference on network protocols.)