

Lecture 1:

Course Introduction

CSE 123: Computer Networks
Alex C. Snoeren





Lecture 1 Overview

- Class overview
 - ◆ Expected outcomes
 - ◆ Structure of the course
 - ◆ Policies and procedures
- A brief overview of Computer Networking
 - ◆ High-level concepts
 - ◆ An end-to-end example



Personnel

- Instructor: [Alex C. Snoeren](#)
 - ◆ Office hours Tuesdays 2-3pm (CSE3114/Zoom) or by appointment
- Project 1 TA: [Bennie Xie](#)
- Project 2 TA: [Prince Modi](#)
- Homework TA: [Patryk Tomalak](#)
- Project Tutor: [Andrew Nguyen](#)
 - ◆ Office hours Wed 3:00-4:00pm; Friday 3:00-5:00pm
 - ◆ In person in the basement

Office Hours Spreadsheet



	Monday	Tuesday	Wednesday	Thursday	Friday
11:00					
11:30					
12:00					
12:30					
1:00			Benny (CSE B250)		
1:30			Benny (CSE B250)		
2:00		Alex	Benny (CSE B250)		
2:30		Alex	Benny (CSE B250)		
3:00			Andrew (CSE B270A)	Prince (CSE B240A)	Andrew (CSE Basement)
3:30			Andrew (CSE B270A)	Prince (CSE B240A), Benny (Remote)	Andrew (CSE Basement)
4:00				Benny (Remote)	Andrew (CSE Basement)
4:30				Benny (Remote)	Andrew (CSE Basement)
5:00		Patryk (CSE B240A)		Benny (Remote)	
5:30		Patryk (CSE B240A)			
6:00					
6:30					
7:00					
7:30					



Prereqs

- CSE29/30, CSE101, and CSE110
 - ◆ Undergrads can't enroll without them
 - ◆ We expect it (or equivalent) even for grad students
- Programming experience
 - ◆ We will be assigning programming projects in C
 - ◆ This course will not teach you C. The TA/Tutors will help, but you need to learn it on your own if you don't already know it.



Expected Outcomes

- This course *will* teach you the *fundamentals* of computer networks:
 - ◆ Layering, signaling, framing, MAC, switching, routing, naming, Internetworking, congestion control, router design, etc.
 - ◆ At the end of this course you should completely understand what's actually happening when you view a Web page or fire up Zoom
- This course *will not* teach you signals and coding
 - ◆ Take an EE course to learn about modulation, encoding, etc. on different hardware technologies
- Similarly, we will not cover Internet apps/services
 - ◆ CSE124 covers application layer protocols, Web, etc



CSE 123 Class Overview

- Course material taught through class lectures, textbook readings, and discussion sections
- Course assignments are
 - ◆ Homework questions (based on lecture)
 - ◆ Two substantial programming projects (in four parts)
- Discussion section (**Monday 4-4:50pm WLH 2005**)
 - ◆ Help you get started on the projects
 - ◆ Lecture material and homework
- Discussion board (**Piazza**)
 - ◆ The place to ask questions about lecture, HW, projects, etc.

Online (Open Source) Textbook



- Peterson and Davie, *Computer Networks: A Systems Approach*, Morgan Kaufmann, **7th Edition**



PDF, eBook, HTML, and source! available at:
<https://7e.systemsapproach.org/>



Homeworks



- There will be 4 homeworks throughout the quarter
 - ◆ Reinforce lecture material...no better practice for exams
 - ◆ Submitted via upload to Canvas
 - ◆ Grading is *completion only*; solutions available after deadline
- Collaboration is encouraged on homeworks!
 - ◆ You *should* discuss homework problems with others
 - » You can learn a lot from each other
 - ◆ You can ask AI for help, but you(r parents) are paying tuition so you can ask a real person
 - » The answer is not the point, *learning how to arrive at the answer* is

Projects



- There will be four programming projects (really two, each split in half)
 - ◆ You will have approximately two weeks to complete each
 - ◆ The first is assigned next MONDAY, introduced in discussion
- The projects must be completed in C
 - ◆ We will provide skeleton code for you to use
 - ◆ Your job is to fill in the interesting/hard parts
 - ◆ The TA and tutors will be available to help with coding
- The projects are INDIVIDUAL assignments
- All code must be your own ([not copied from others or the Web](#))
 - ◆ OK to discuss design ideas (even w/AI), NOT OK to share/copy & paste code
 - ◆ Projects assigned AND SUBMITTED via private GitHub repo



Development environment

- We expect you to use GitHub Codespaces
 - ◆ Automatically sets up Cloud VMs using your GitHub account
 - ◆ Web-based VS Code interface with integrated Git
- You can also use your own machines
 - ◆ The project source will work on Linux/Windows/OS X
 - » Windows users need Windows Subsystem for Linux (WSL)
 - ◆ Code executes in Containers, just like on GradeScope
 - ◆ TAs will (only) address any discrepancies between GradeScope and GitHub codespace execution environments

Project 1a



- Will be available (via Canvas) on Monday immediately after class
 - ◆ The skeleton code is distributed via a private GitHub repo
- We need your GitHub ID in order to create your repo
 - ◆ Fill out the Google Form in the Assignments tab on Canvas ASAP
- The Project TA will provide an overview in discussion—please attend!
 - ◆ It will be best if you can follow along in your own GitHub repo, which means
 - ◆ You need to fill out the Google Form before discussion so we can make your repo



Assessments

- Midterm
 - ◆ Wednesday, October 28th in class
 - ◆ Covers the first half of class
- Final exam
 - ◆ Friday, December 11th (8:00-11:00am)
 - ◆ Covers last half of class + selected material from earlier parts
 - » I will be explicit about the material covered
- All exams are in person; No makeup exams
 - ◆ Unless dire circumstances; [contact me NOW if you have a conflict](#)
 - ◆ [OSD will use TTC; book your appointment ASAP](#)

Grading



- Homeworks: 5% *completion only*
- Midterm: 35%
- Final: 40%
- Projects: 20%
 - ◆ Divided evenly among the four projects



A Few Class Policies

- Regardless of modality, this class is "live"
 - ◆ In-person lectures podcast, but intended for review
- Discussion attendance is strongly encouraged
 - ◆ The projects are involved; the TAs will use the time to get you going
- Zoom Office hours are **NOT RECORDED**
 - ◆ Please turn your camera on when interacting with TAs/Tutors
- No late assignments
 - ◆ HWs submitted via GradeScope, Projects via GitHub/GradeScope



A Note on GenAI

- Yes, you can use AI tools to help with projects, but...
- **ALL CODE YOU SUBMIT MUST BE YOUR OWN**
 - ◆ We will run it through plagiarism detection tools
- It is NOT OK to copy someone else's code, no matter who/what found it
 - ◆ This applies even if the code was regurgitated by GenAI
 - ◆ And I assure you that GenAI tools have seen solutions to these labs
- So, feel free to ask AI to help you, but just as it is not OK to delegate the assignment to another human, it is not OK to delegate it entirely to AI
 - ◆ You are responsible for the provenance of everything you turn in
 - ◆ Cut & pasting GenAI responses/code is likely to result in an AI violation



How *Not* To Pass CSE 123

- Do not attend lecture / discussion
 - ◆ Podcast is available, and the material is in the book anyway
 - ◆ Lecture material is the basis for exams and directly relates to the projects
 - ◆ Besides, the professor thinks he's funny
- Do not do the homework
 - ◆ It's only 5% of the grade, and you can just submit random answers
 - ◆ Excellent practice for the exams, and some homework problems are exercises for helping with the project



How *Not* To Pass (2)

- Do not ask questions in lecture, office hours, or Piazza
 - ◆ Professor is scary, I don't want to embarrass myself
 - ◆ Asking questions is the best way to clarify lecture material at the time it is being presented
 - ◆ Office hours and Piazza will be invaluable for homeworks, projects
- Wait until the last couple of days to start a project
 - ◆ We'll have to do the crunch anyways, why do it early?
 - ◆ The projects cannot be done (by a human) in the last couple of days
 - ◆ **Repeat:** The projects cannot be done in the last couple of days



Class Web Page

- Everything on the Web (wrapped in Canvas)
<http://www.cs.ucsd.edu/classes/fa26/cse123-001/>
 - ◆ Course syllabus and schedule (updated as quarter progresses)
 - ◆ Announcements
 - ◆ Homework handouts
 - ◆ Project information
- Class will be podcast
 - ◆ Lecture slides posted to website immediately after class
 - ◆ Podcast is for review, not intended as a substitute for lecture

Questions



- Before we start the material, any questions about the class structure, contents, etc.?



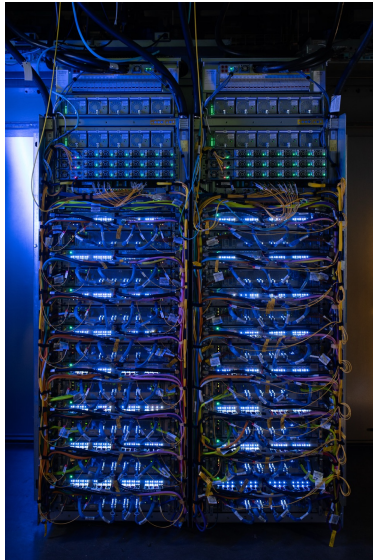
This Class in One Slide

- **Protocols & Layering**
 - ◆ Manage complexity by decomposing the tasks
 - ◆ Standardizing syntax and semantics to support interoperability
- **Addressing & Naming**
 - ◆ Agreeing on how to describe a host, application, network, etc.
- **Switching & Routing**
 - ◆ Deciding how to get from here to there
 - ◆ Forwarding messages across multiple physical components
- **Resource Allocation**
 - ◆ Figuring out how to share finite bandwidth, memory, etc.



A “Simple” Task

- Send information from one computer to another



A rack full of Hosts

- Endpoints are called **hosts**
 - ◆ Could be server, iPhone, laptop, etc.
- The plumbing is called a **link**
 - ◆ We don’t care what the physical technology is: Ethernet, WiFi, cellular, etc.



Host

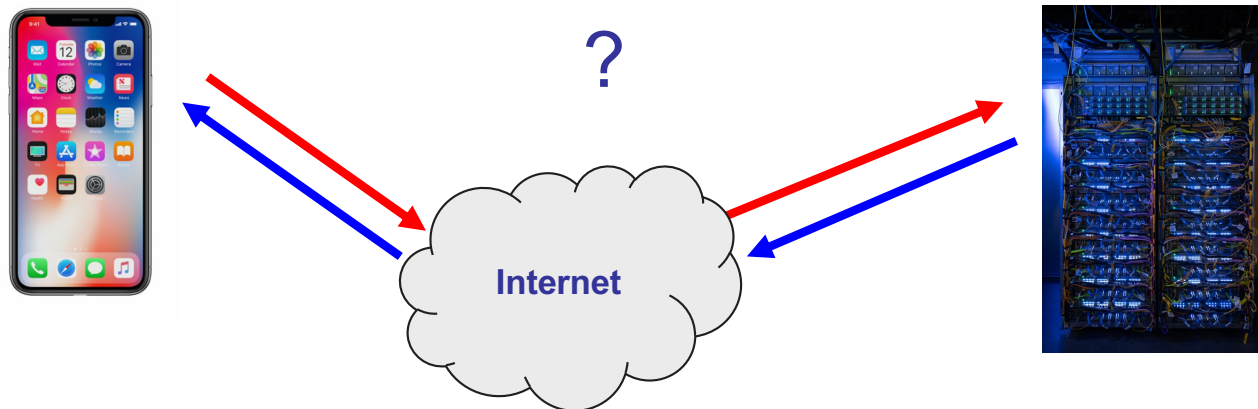
Actually Quite Complicated



- **ROUGHLY**, what happens when I use an iPhone App at UCSD?

My device

www.google.com





Apps generate Web requests (HTTP)

- Local App turns your click into an HTTP request
 - ◆ The functionality is provided by a server running on a remote host



Name resolution (DNS)



- Where is `www.google.com`?

My device
(132.239.9.64)



What's the address for `www.google.com`



Oh, you can find it at 66.102.7.104



Local DNS server
(132.239.51.18)





Data transport (TCP)

- Break message into packets (TCP segments)
- Should be delivered reliably & in-order





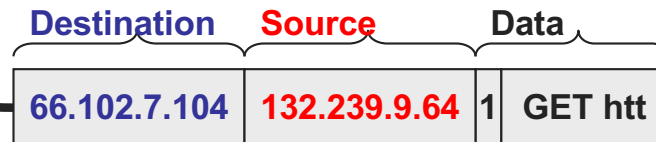
Global Network Addressing (IP)

- Address each packet so it can traverse network and arrive at host

My device
(132.239.9.64)



www.google.com
(66.102.7.104)





Channel access (WiFi)

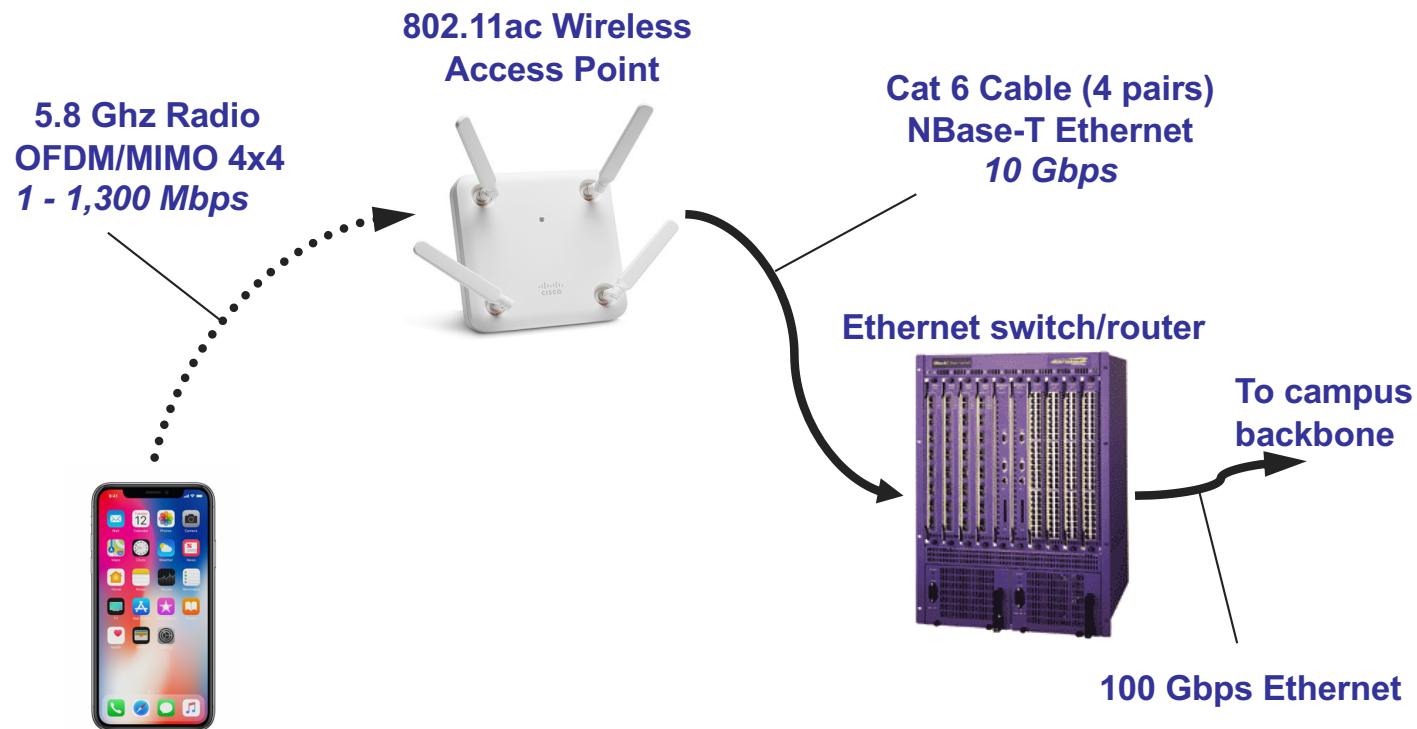
- Break message into frames
- Media Access Control (MAC)
 - ♦ Can I send now? Can I send now?



- Send frame



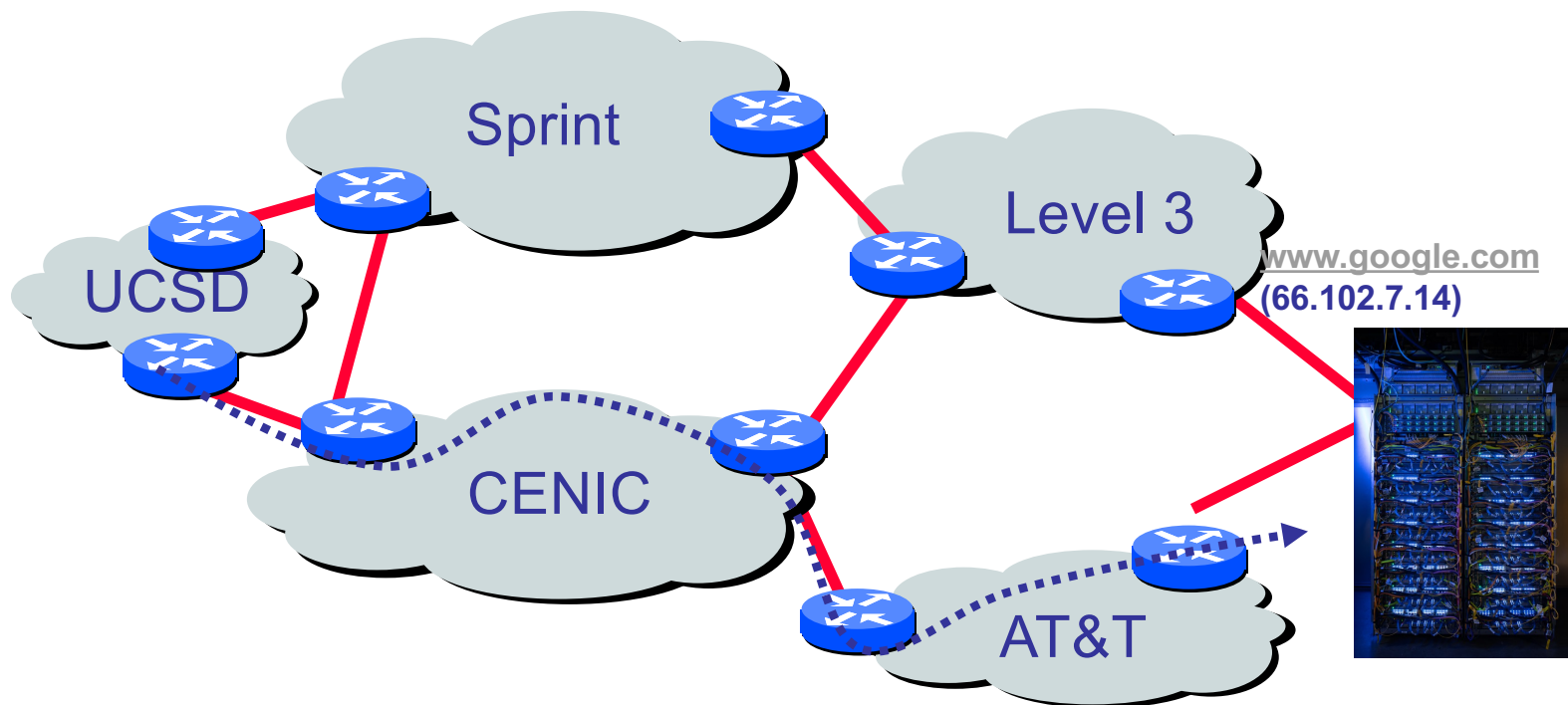
Forwarding across links





Network Routing

- Each router forwards packet towards destination





For Next Class...

- Browse the course website (also on Canvas)
 - ♦ <http://www.cs.ucsd.edu/classes/fa26/cse123-001/>
- Read Chapter 1 up to (and including) 1.3
- Monday: Network Architecture