

# Lecture 1:

# Course Introduction

---

CSE 123: Computer Networks  
Aaron Schulman (aka Aaron Shalev)





# 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: **Aaron Schulman (aka Aaron Shalev)**
  - ◆ Office hours Tuesdays 1pm-2pm or by appointment
  - ◆ EBU3B 3120
- Project TAs:
  - ◆ **Alex Gamero-Garrido**
    - » Office hours Thursdays 2-3pm
    - » EBU3B B275
  - ◆ **Spoorti Joshi**
    - » Office hours Tuesday 3-4pm
    - » EBU3B B270A
- Project/Homework Tutor: **Sneh Shah**
  - ◆ Office hours Friday 12-1pm
  - ◆ EBU3B B275

# Prereqs

---



- CSE30, CSE101, and CSE110 – **not** CSE120
  - ◆ 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/C++
  - ◆ This course will not teach you C. The TAs 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
- 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.
  - ◆ You *will be able to* pick this up on your own with Google

# CSE 123 Class Overview

---

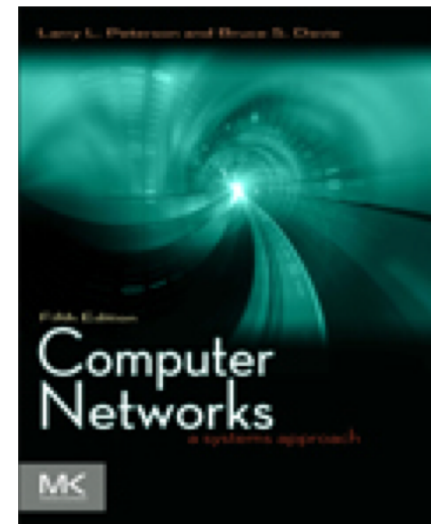


- Course material taught through class lectures, textbook readings, and discussion sections
- Course assignments are
  - ◆ Homework questions (based on lecture)
  - ◆ Two substantial programming projects
- Discussion section: ([Friday 3pm Mandeville Center 210](#))
  - ◆ Help you get started on the projects
  - ◆ Lecture material and homework
  - ◆ Additional networking topics
- Discussion board ([Piazza.com](#))
  - ◆ The place to ask questions about lecture, hw, projects, etc.

# Textbook



- Peterson and Davie, *Computer Networks: A systems Approach*, Morgan Kaufmann, 5th Edition, ISBN 978-0-12-385059-1



**PDF, eBook, HTML, and source! available at:**  
<https://github.com/SystemsApproach/book>

# Homeworks

---



- There will be 4 homeworks throughout the quarter
  - ◆ Reinforce lecture material...no better practice
- Collaboration vs. cheating
  - ◆ You *should* discuss homework problems with others
    - » You can learn a lot from each other
  - ◆ But there is a distinction between collaboration and cheating
  - ◆ Rule of thumb: Discuss together in library, walk home, and write up answers independently
  - ◆ Cheating is copying from other student's homeworks or solution sets, searching for answers on the Web, etc.
  - ◆ Suspicious homeworks will be flagged for review

# Projects

---



- There will be two programming projects
  - ◆ You will have three/five weeks to complete each
  - ◆ The first will be assigned THIS WEEK
- The projects must be completed in C/C++
  - ◆ We will provide skeleton code for you to use
  - ◆ Your job is to fill in the interesting/hard parts
  - ◆ The TAs will be available to help with coding
- The projects are **INDIVIDUAL** assignments
  - ◆ All code must be your own (not copied from github!)
  - ◆ OK to discuss design ideas, NOT OK to share/look at code
  - ◆ Projects assigned AND SUBMITTED via private GitHub repo

# Espresso Prize

---



# Computer Labs

---



- You are welcome to use any Linux machine in the labs in the basement of the CSE/EBU3B building
  - ◆ Linux running on Intel machines
- You can also use your home machine
  - ◆ The project source will work on Windows/OS X (with caveats)
  - ◆ Graders will test on ieng6 machines
  - ◆ **Be sure to test your projects there as well**

# Exams

---



- Midterm
  - ◆ Friday, May 3rd
  - ◆ Covers first third to half of class
- Final
  - ◆ Thursday, June 13<sup>th</sup> (11:30am-2:30pm)
  - ◆ Covers second half of class + selected material from first part
    - » I will be explicit about the material covered
- No makeup exams
  - ◆ Unless dire circumstances
- Closed book with crib sheet
  - ◆ You can bring one double-sided 8.5x11" page of notes to each exam to assist you in answering the questions
  - ◆ Not a substitute for thinking

# Grading

---



- Homeworks: 12%
  - ◆ Think of these collectively as a take-home midterm
- Midterm: 21%
- Final: 25%
- Projects: 42%
  - ◆ Divided evenly among the projects



# How *Not* To Pass CSE 123

---

- Do not come to lecture / discussion
  - ◆ It's nice out, class is after lunch, and the material is in the book anyway...
  - ◆ Lecture material is the primary basis for exams and directly relates to the projects
  - ◆ Besides, the professor thinks he's entertaining
- Do not do the homework
  - ◆ It's only 12% of the grade
  - ◆ Excellent practice for the exams, and some homework problems are exercises for helping with the project
  - ◆ 12% is actually a significant fraction of your grade (easily the difference between an A and a C)



# How *Not* To Pass (2)

---

- Do not ask questions in lecture, office hours, or email
  - ◆ 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
  - ◆ Bring your concerns to the professor early (avoid snowball)
  - ◆ Office hours and email will help with 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 in the last couple of days
  - ◆ Repeat: The projects cannot be done in the last couple of days

# Class Web Page

---



<http://www.cs.ucsd.edu/classes/sp19/cse123-a/>

- Serves many roles...
  - ◆ Course syllabus and schedule (updated as quarter progresses)
    - » Lecture slides
  - ◆ Announcements
  - ◆ Homework handouts
  - ◆ Project information

# 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
- **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.



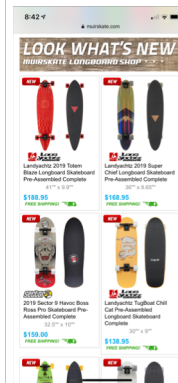
# A “Simple” Task

- Send information from one computer to another

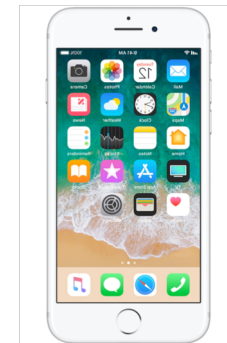


Host

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



Link



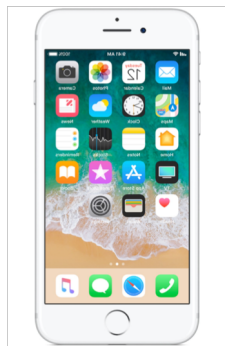
Host

# Actually Quite Complicated

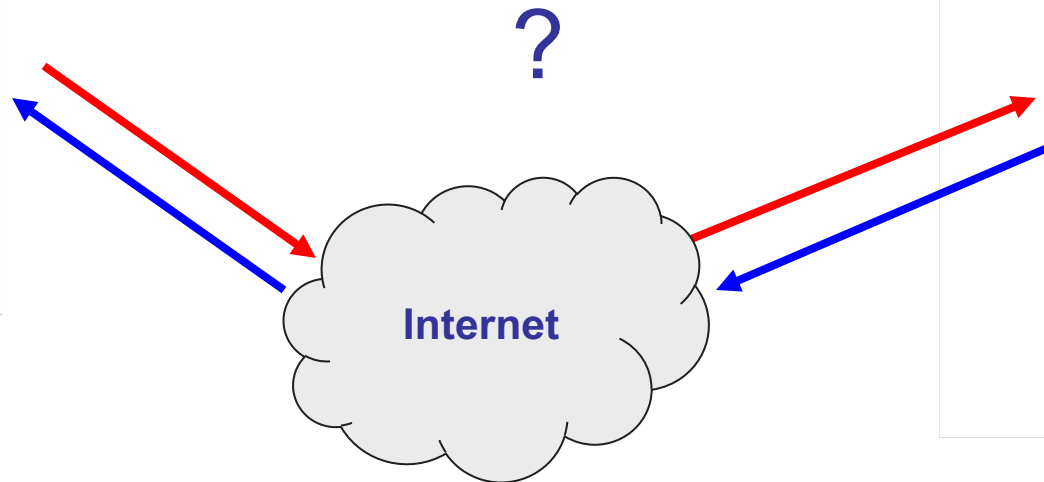


- **ROUGHLY**, what happens when I click on a Web page from UCSD?

**My iPhone**



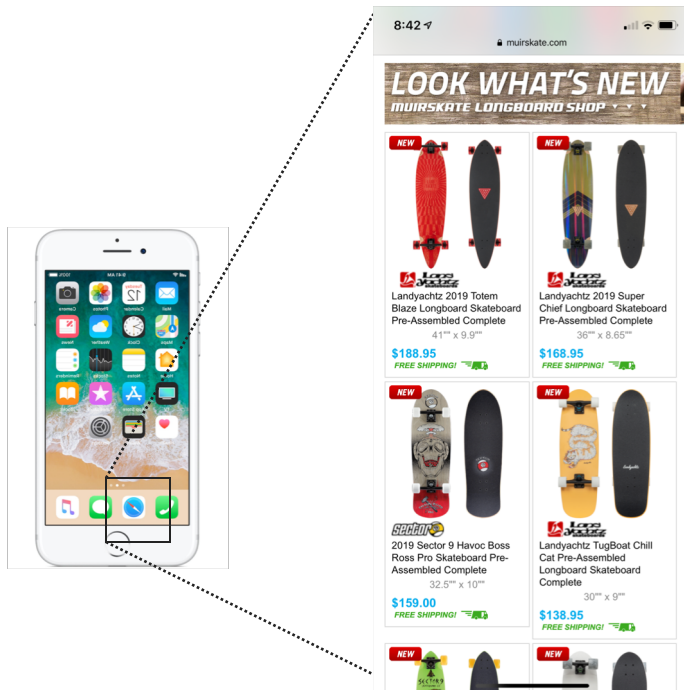
**[www.muirskate.com](http://www.muirskate.com)**





# Web request (HTTP)

- Turn click into HTTP request



**GET https://www.muirskate.com/ HTTP/1.1**  
**Host: www.muirskate.com**  
**Connection:keep-alive**  
...

# Name resolution (DNS)



- Where is `www.muirskate.com`?

**My computer**  
(132.239.9.64)



*Where is `www.muirskate.com`?*



*Many hosts have it, one is `104.20.86.126`*



**Local DNS server**  
(132.239.51.18)

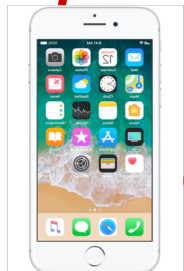




# Data transport (TCP)

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

```
GET http://www.muirskate.com HTTP/1.1
Host: www.muirskate.com
Connection:keep-alive
...
```



3 google.c

2 p://www.

1 GET htt

**“and let me know when they got there”**

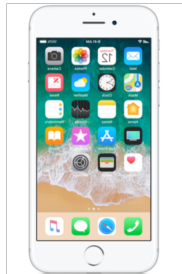


# Global Network Addressing

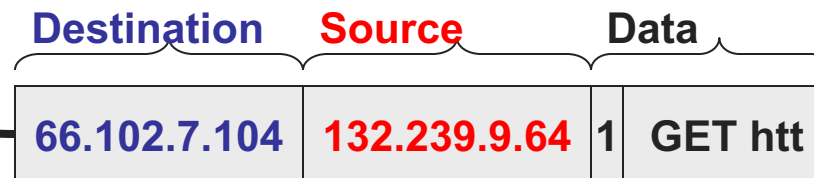


- Address each packet so it can traverse network and arrive at host

**My computer**  
**(132.239.9.64)**



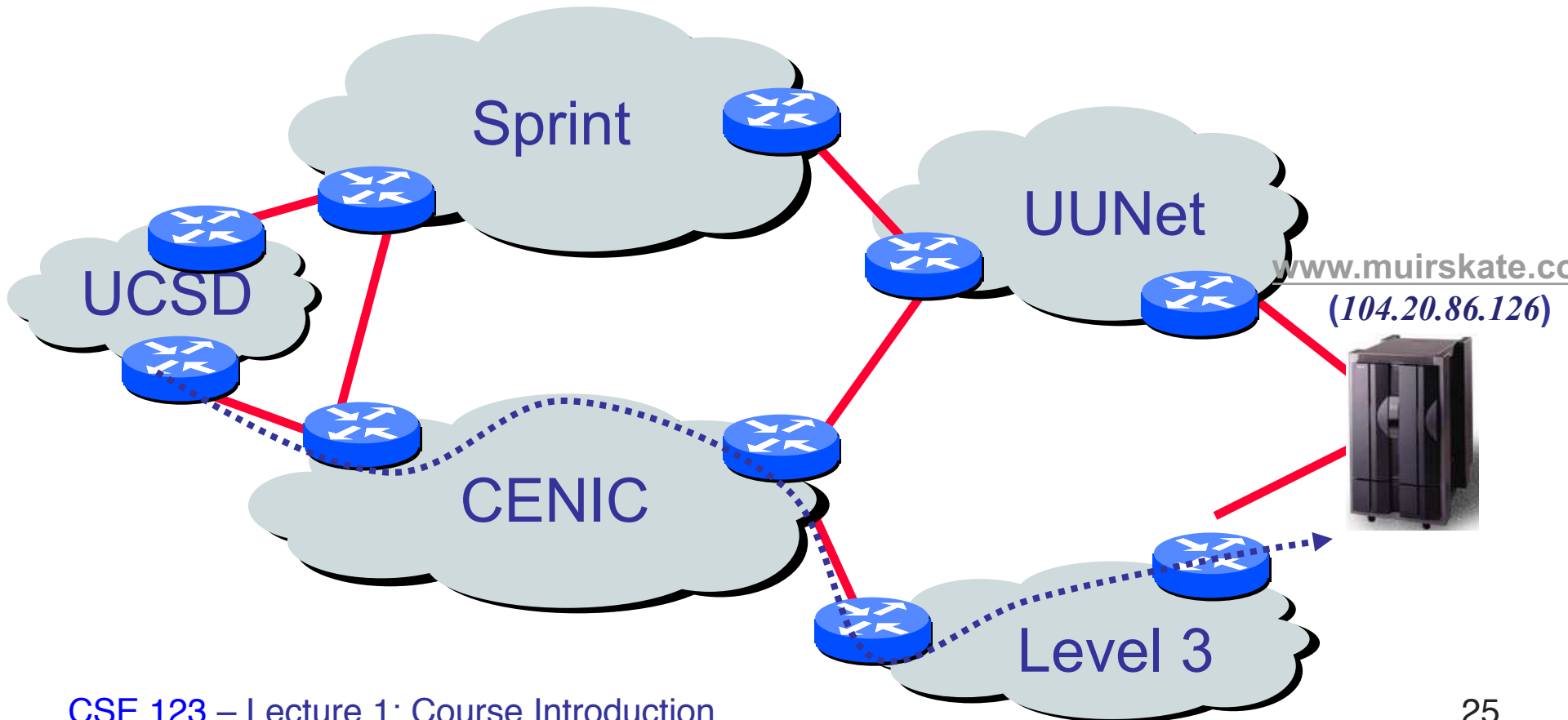
**www.muirkate.com**  
**(104.20.86.126)**





# Network Routing

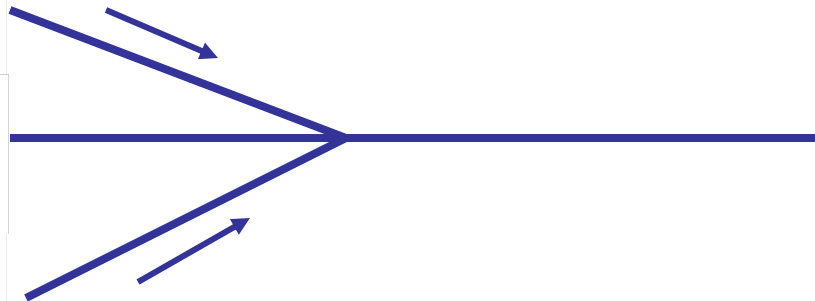
- Each router forwards packet towards destination





# Link management (WiFi)

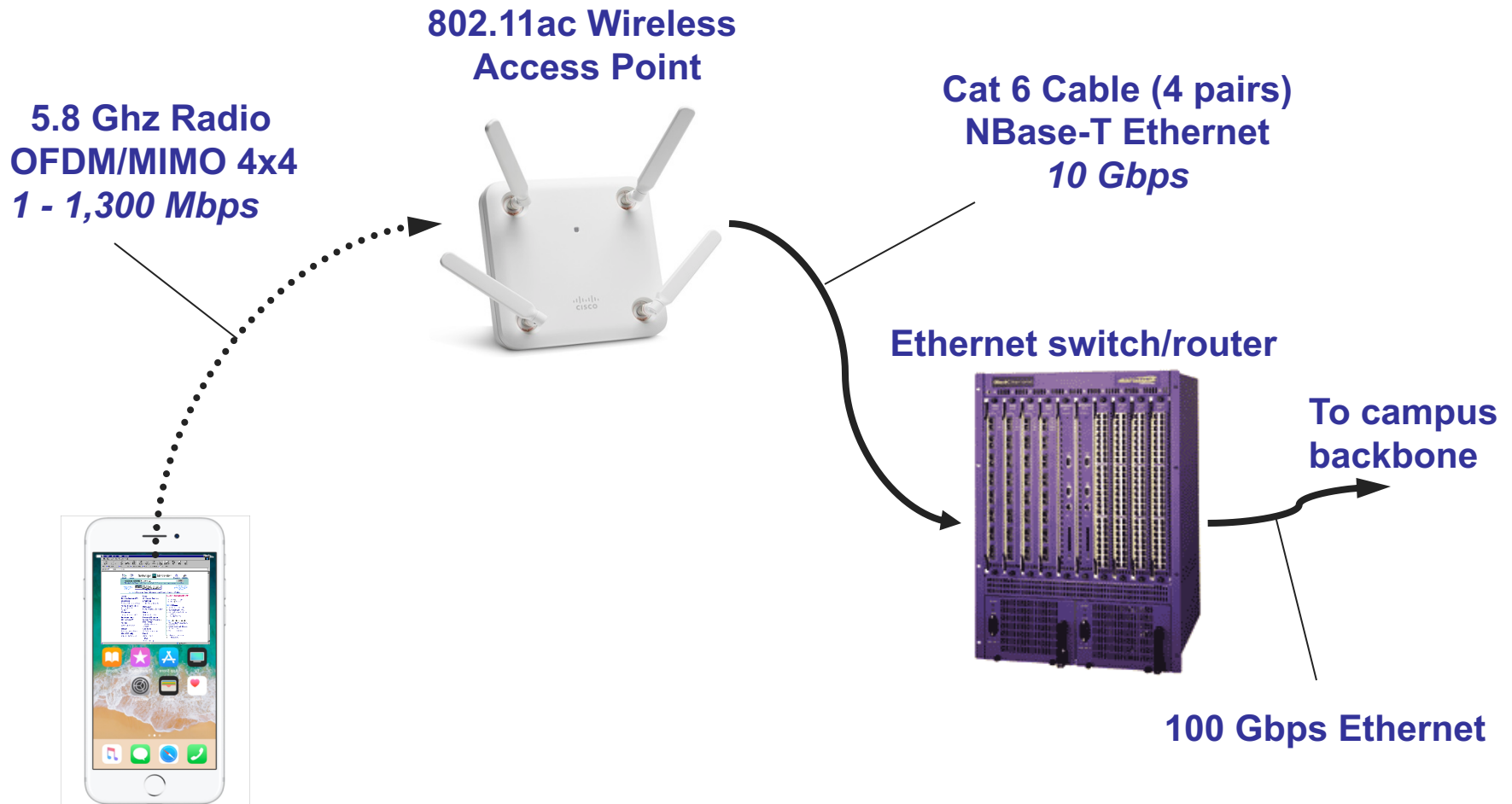
- Break message into frames
- Media Access Control (MAC)
  - ◆ Can I send now? Can I send now?



- Send frame



# Physical layer





# For Next Class...

---

- Browse the course web
  - ◆ <http://www.cs.ucsd.edu/classes/sp19/cse123-a/>
- Read Chapter 1.3 and 2.3
- Next class: Layers and Framing
- Drop now or plan to stick it out!