

# ECE 435 – Network Engineering

## Lecture 1

Vince Weaver

`https://web.eece.maine.edu/~vweaver`

`vincent.weaver@maine.edu`

Barrows 125, Tues 12:30pm

16 January 2024

# Introduction

- Distribute and go over syllabus
- [https://web.eece.maine.edu/~vweaver/classes/ece435\\_2024s](https://web.eece.maine.edu/~vweaver/classes/ece435_2024s)



# ECE435 – Office Hours

- Name
- Office Location: 203 Barrows
- Office Hours: 11am-noon Monday/Wednesday, drop in



# Syllabus – Textbook

- (optional) Tanenbaum “Computer Networks”



# Syllabus – Hardware

- Will be helpful to have a device with network connection (Ethernet, wifi) that you can run tests on
- Raspberry Pi is great because if you mess things up you can just pop in a new SD card
- Laptop or desktop is fine too. It helps if it is running Linux. Although most TCP/IP stacks are similar for backwards compatibility reasons.



# ECE435 Syllabus – Homeworks

- Homeworks, 50%
- Roughly 10 total, lowest dropped
- Generally will be due on Thursday by beginning of class
- Will usually have at least a week to do them.
- Submission by e-mail, grades sent in response to that e-mail
- Will send out e-mail when posted on website



# ECE435 Syllabus – Homeworks

- Will initially involve some coding
- Will involve some C coding, and maybe Linux command line
- Lab: no dedicated lab.



# ECE435 Syllabus – Exams

- Midterm, 10%
- Final, 15%





# ECE435 Syllabus – Project

- Project, 20%
- Involves using what we learn to do a networking-related project
- Can work in groups
- Open-ended, any language you want
- Presentation last week of class
- Writeup at the end
- More details as we get closer.



# ECE435 Syllabus – Late Work

- Class participation, 5%
- Late work penalty  
please turn in work, even if late, even if incomplete.  
homework grade adds up
- Class notes will be posted on the website.



# ECE435 Syllabus – Other

- COVID policy – please don't come to class if you're super sick (and mask if possible if regular sick)
- Requesting Help – please send code
- Academic Honesty – please make sure your code is your own work
- Standard UMaine boilerplate stuff



# Notes on the Class

- Note: not a lab class.  
Programming routers with serial ports and subnetting not as key anymore
- Networking vaguely static from 1990s to ~5 years ago
- Lots has been changing recently, hard to keep up
- Used to be pretty open, now a lot happens inside of big companies, mostly google
- I often get really knowledgeable people in this class. If I make a mistake, let me know



# Networking

In this class we will cover

- Computer Networks
- Computer Security



# What is a Computer Network?

- A group of computers, connected so they can communicate
- Probably familiar with the Internet, which is a network of networks.



# How can they be connected together?

- Wire (Ethernet, telephone, powerlines)
- Fiber Optic
- Wirelessly: radio, microwave, infrared, laser
- Sound?
- Barbed Wire? Wet string? Carrier Pigeons?



# Why have networks?

- Resource sharing (printer, fileserver, etc.)
- Communication (e-mail, text messaging, video-conferencing, etc)
- Entertainment/Gaming
- Operating system/Security Updates
- Shopping
- Accessing Info





# Why have network to your home?

- Older books would actually spell this out because back in the day people needed convincing to connect to the internet



# Network Concerns that we'll Discuss

- Reliability
  - What makes a reliable network?
- Security
  - Is security a network-related problem? It makes local security issues exploitable world-wide...
- Expense
- Speed
  - Latency vs Bandwidth
- Addressing (how to find a machine)



- Error correction
- Scalability
  - Trouble that appears as networks get bigger
- Standards
  - How do two computers understand each other? Who defines the rules?
- Privacy
  - Encryption? Trust? Authority? Tracking?
- Complexity
  - A lot of networking used to be easy and hands-on
  - Things are now fast at the expense of understandability



# Some Network terms

- Client/Server
- Broadcast vs point-to-point
- Wide area network, local area network
- Bandwidth vs Latency.  
1Gbps might be fast, but what if 100ms latency?
- Connection oriented vs packet based (Switched phone vs VOIP)
- Topology (star, ring, cube, mesh, hypercube)



# OSI Reference Model

ISO/OSI Open Systems Interconnection (1984)

ISO 7498

Many thought this would be the standard, but didn't end up that way

Everyone still talks about it anyway

Various layers each a new layer of abstraction.

Layers should be independent. Layering violations



1. Physical – **bits**: the raw bits. How 0 and 1 encoded, electrons or photons, etc.  
pins, volts, timing, frequency  
topology, how wires laid out  
bandwidth
2. Data Link – **frames**: Transforms raw line to one that handles errors,  
breaks up data into frames, etc.  
Unique identity for each device on network  
Flow control, error handling



3. Network – **packets**: management of subnet. How packets routed from one network to another, addressing. (routing: what is routing?)
  
4. Transport – **end-to-end delivery**  
accepts a stream of bytes from above and make it suitable for the network layer.  
Gets back split up packets and turns it back into a total message.  
flow control, reliable delivery, error correction
  
5. Session – allows different machines to have sessions



between them. session management, synchronization.  
Lets different apps share one connection to the network.

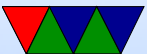
6. Presentation – syntax of data being transmitted. Char encoding, compression, encryption
7. Application – high level protocol, like webserver (http), ssh, etc.





# 8th layer

- The user?
- Political? Financial? Government?



# Summary

	OSI	TCP/IP
7	Application	Application
6	Presentation	
5	Session	
4	Transport	Transport
3	Network	Internet
2	Data Link	Host-to-network
1	Physical	Host-to-network



# Layering

- Why is it good?  
Abstraction. Easier to do one layer and do it right.  
Should the webserver be aware if it is serving over copper vs fiber?
- Counterpoint: RFC 3439: “Layering considered harmful”  
(“considered harmful” is a Dijkstra meme)



# Results

- OSI (theoretical) never caught on for various reasons
- TCP/IP (practical) did, but has its own limitations which we'll discuss later



# This year's Plan

- We'll start at the top and work our way down.  
Either way has issues



# Coding

- Have you written a network program?
- How do you write a network program?
- We'll use C

