# Introduction
# CS 118
# Computer Network Fundamentals
# Peter Reiher

# Purpose of the class

- To familiarize you with the basic concepts of computer networking
- Computer networks are increasingly key to most systems
- All educated computer scientists should have a good understanding of how they work

# Pre-requisite

- CS 111 – Operating System Principles
  - Which itself has CS 31, 32, and 35 as pre-requisites
- So you're expected to be able to program
- And to have a reasonable understanding about how computer software systems work

# Textbooks

- Shannon/Weaver,
  The Mathematical Theory of Communication
  (any edition)
- Peterson/Davie,
  Computer Networks: A systems approach
  (any edition)*
  - *readings are cited from the Sixth Edition; students are responsible for location of corresponding material if using other editions

# Assignments

- Programming projects
  - Two
  - On a schedule set by the TA
- All work is to be completed INDIVIDUALLY.

# Grading

- 30% projects
  - 15% each for 2 assignments
- 30% midterm
  - Feb. 4, in class
- 40% final exam
  - March 14, 8-11 AM

- Projects due as announced
  - Due at the start of class on date indicated
  - TA will set policy for late submissions

# Office Hours

- TTh 2-3 PM
- In 3532F Boelter Hall
- Other times possible by arrangment

# The TA

- Seungbae Kim
  - ksb2043@gmail.com
- He will handle all issues related to the projects
- Also will hold weekly recitation sections and office hours
  - Times to be announced

# A bit about style

- A bit more "abstract" than typical
  - This is an education, not merely training
  - It's for your entire life, not just your first job
- You're expected to *apply* what you learn
  - Repeating what you learn will not be enough
  - Just attending class will not be enough
- You will be challenged
- I am here to help
  - Specific questions will always be answered

# Mastering the material

- There's a lot of stuff
  - What should you focus on?
- Things to keep in mind:
  - Understanding
  - Recognizing
  - NOT memorizing
- Focus on the subject
  - Side-discussions are intended to illuminate, not dump extra stuff on you

# Let's begin…

# A Roadmap

- Introduction and history
- Performance and efficiency

# Overview

- Definitions
- What about the layers we've heard about?
- The first-principles approach
- A little history

# Why are we here?

- Computer networking
  - Really: networked computer communication
  - Information exchange between computers

- The challenge:
  - What is information?
  - What is communication?
  - What is networking?
  - How are these related?

# What is communication?

- Methods for exchanging information between:
  - a fixed set of
  - directly-connected parties
  - using a single, shared set of pre-agreed rules

# So then what's a protocol?

- A single, shared set of pre-agreed rules

- *E.g.*:
  - I call you
  - The phone rings
  - You pickup and say "Hello"
  - We start talking

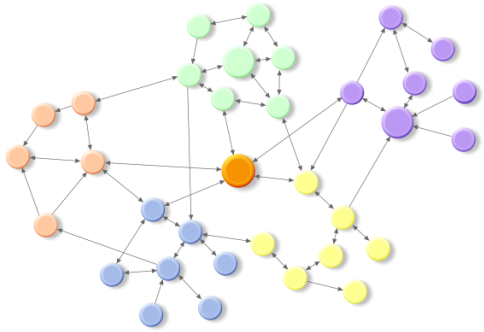
# Protocol variations

- What word (for the telephone)?
  - Bell originally proposed "Ahoy!"
- Who talks first when I call you?
  - Typically:
    - You pickup and **you** say "Hello" [callee first]
  - Alternate:
    - You pickup and **I** say "Hello" [caller first]
  - Either one works
    - Only if both sides agree in advance

There is a lot of complexity
in just two-party communication.

# What is networking?

- Methods to enable communication between:
  - varying sets of
  - indirectly connected parties
  - that don't share a single set of rules



- Networking:
  - how we get from "nothing" to being able to communicate

# Let's compare…

**Communication**

- Methods for exchanging information between:
  - a ***fixed*** set of
  - ***directly-connected*** parties
  - using a ***single, shared set*** of pre-agreed rules (a protocol)
- How you exchange info when you know *who* you're talking to and *how*

**Networking**

- Methods to enable communication between:
  - ***varying*** sets of
  - ***indirectly connected*** parties
  - that ***don't share a single set*** of rules
- How you figure out who you're talking to and how

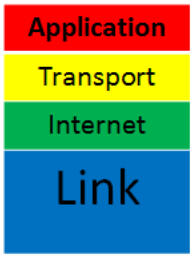
# Summary definitions

- Communication
  - Methods for exchanging information between a fixed set of directly-connected parties using a single protocol
- Networking
  - Methods to enable communication between varying sets of indirectly connected parties that don't share a single protocol
- Protocol
  - A set of rules, agreed in advance, that enable communication

# Where are the layers we've heard about?

- International Standards Organization (ISO)
  - Open Systems Interconnect (OSI)
  - Seven layers based on function/capability
  - Developed as a reference model
  - Implemented but not really used


- Internet
  - Four layers
  - More or less . . .



# Slapping Names on Layers Isn't Useful

- The name doesn't really tell you anything
  - Calling it "transport" doesn't mean much
- What's important is what happens in the network
- There can be many ways of mapping desired functionality into elements of the system

# Names – What's valuable about them

- They allow us to specify things
- To make sure the right actions happen to the right things
- In networks, to get messages to the right recipients
- In network layers, to ensure that we understand what layer we're dealing with

# Names – What's unimportant about them

- The actual name is meaningless
- Meaning is achieved by binding it to something
- The same thing can have several different names
- The same name can be applied to several different things
  - Depending on context
  - Changing over time

# The important lesson about names

- Don’t obsess about the name itself
- Concentrate on how the name relates to reality

# These Layers Aren't the Truth, Anyway

- It's not 1984 anymore
  - Both models describe early networking
- Layers aren't defined by function
  - Most layers do most functions now
- There are too many exceptions
  - In-between layers
  - Virtual layers (tunnels)

# Let's go back to the beginning…

*Two fundamental ideas of CS:*

- Abstraction
- Recursion

# Abstraction

- Represent something complex…
  - with something simpler…
  - that is easier to understand
  - AND
  - that can be used to predict the behavior of the complex

A MODEL

# Recursion

- The converse of induction
  - decompose a large problem into the combination of its components
  - declare a value for the minimal atomic component

# The goal of our approach

- To describe networked computer communication from first principles of:
  - Abstraction
  - Recursion
- We'll still have layers
  - Just recursive ones

# If layers aren't fixed things?

- Then what are they?
  - A layer is the largest set that can communicate
  - *i.e.*, a layer is the largest group that is:
  - directly connected
  - shares a single, common protocol

# A Roadmap Through the Course

- Bits
  - A very fine place to start…
- Communication
  - Two-party bit sharing
- Networking
  - Multiparty bit sharing

# Course roadmap

- Communication
  - Two-party shared state
  - Channels
  - Protocols

# Course roadmap

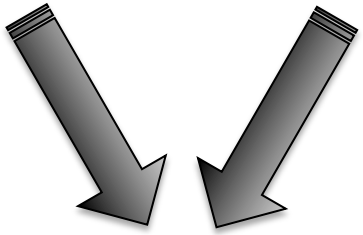
- Communication
  - Two-party shared state
  - Channels
  - Protocols

- Networking
  - Multiparty complications
  - Layers
  - Naming
  - Recursion/forwarding

# Course roadmap

- Communication
  - Two-party shared state
  - Channels
  - Protocols

- Networking
  - Multiparty complications
  - Layers
  - Naming
  - Recursion/forwarding



- Examples & mechanisms
  - Communication
  - Networking

# A little history too

- ~5000 years of networking to consider!

# Couriers

- Human-based
  - More reliable
- Slow
  - Walking, horse galloping
- Limited range
  - Tens of miles
  - Relay only where pre-deployed
- Vulnerable
  - Loss, corruption, interference
- Costly

# Carrier pigeons

- Unidirectional messaging
  - From release to "home"
- Hard to "reset"
  - Bring the pigeon back
- Fixed locations
  - Messages go only where pigeons are "homed"
- Unpredictable
  - High loss rate!

# Beacons

- Limited BW
  - One signal
  - Slow to reset
- Long distance
  - Relays over hundreds of miles
- Costly
  - Requires resident attendant
- First optical comms!
  - Works at night
  - Better than daytime
  - Worked for Paul Revere

# Heliograph

- More optical comms
  - Sunlight
- Unreliable
  - Hard to aim
- Limited use
  - Sunny days only
  - Low bitrate

# Flags

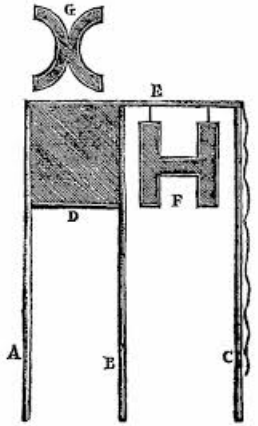
- Still in current use
  - Maritime communications
  - Public communications
    - E.g., swim safety

# Origins

- Couriers Spoken/written (30,000 BC)
- Pigeons 2900 BC, Egypt
- Beacons 1200 BC, Troy
- Heliographs 400 BC, Greece
- Flags 400 BC, Greece

# Hooke

- Yes, the microscope guy
  - 1680's
  - "On Showing a Way How to Communicate One's Mind at a Distance"
  - Telescope + semaphores



# French Telegraph

- Semaphore telegraph
  - 1790s, Claude Chappe
  - Letters, numbers
  - Time sync
  - Contention (message collision)
  - Priority
  - Flow control
  - Error recovery

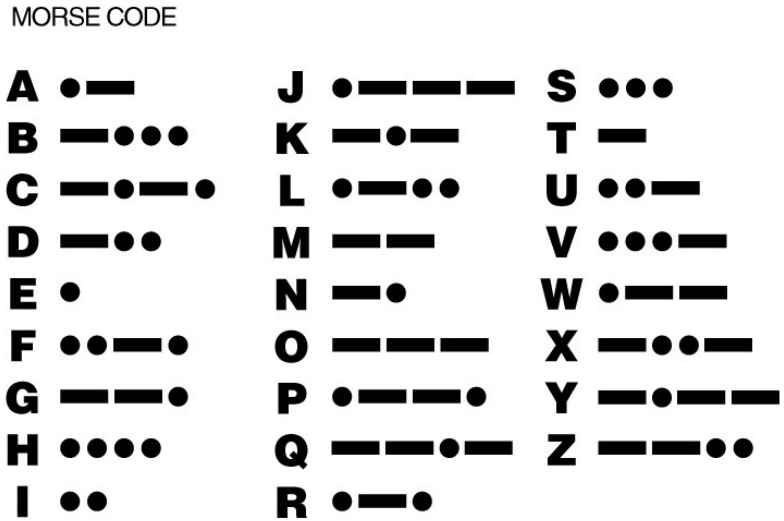
# Emergence of electricity

- Electromagnets invented 1820 (Sturgeon)
  - Electrical relays – 1835
- Cooke/Wheatstone – 1837
  - Multiple needles
  - 13 miles near London
- Morse – 1837
  - Single relay
  - Killed the Pony Express (courier) by 1861

# Cooke/Wheatstone

# Morse

- Symbols == letters
- Time encoding
  - Dot
  - 3 dots = dash
  - Intra-symbol
    - dot delay
  - Inter-symbol
    - dash delay
  - Inter-word
    - seven dot delay



# Telephone

- First patented by Alexander Graham Bell
  - In 1876

- Carried actual voice over electromagnetic media
- In wide use by early 20th century
- Still in wide use today

# Radio

- Transmission of signals without wires
  - Originally encoding sound
  - Eventually encoding many forms of data
- Theoretical possibility shown by Maxwell (1864)
- Patent of practical device by Marconi (1896)

# Computer networking

- Small, special purpose computer networks in 1950s, 1960s
- Packet switching developed in 1960s
- ARPANET went online in 1969
- Internet replaced the ARPANET in 1981
  - And became commercial in 1989
- World Wide Web introduced in 1991
  - Not a new hardware technique
  - But a revolution in what networks could do

# Characterizing Networks

- Some characterizations are based on purpose
  - "It's a network for voice"
- Others are numerical
  - "It can transmit 10 Mbytes per second"
  - Numerical characterizations tend to be more useful
- What will we measure for networks?
- Values to characterize work and power
  - Time
  - Number & size of messages

# Communications is all about time…

- Time for information transfer
  - Info at A -> info at B
- Time for a transformation
  - Info -> f(info)
- Time for a transaction

| | |
|---|---|
| I at A -> | request starts at A |
| I at B -> | request arrives at B |
| f(I) at B -> | response created at B |
| f(I) at A | response moves to A |

# Communications/Network Measures

- Frequency
  - Bandwidth
  - Processing
- Speed
  - Propagation speed
- Delay
  - Propagation latency
  - Access delay
- Loss rate

# Rate *vs*. Frequency

- Rate
  - Events per unit time
- Frequency
  - Time between events
  - Sometimes: time to complete (TTC) a given event

how often it gets sent out

- Not always related!
  - Rate = 1/TTC * #servers
  - *E.g.,* you can cook pies at a rate faster than 1 pie per hour, but each pie will still take 1 hour to cook (i.e., pie baking frequency doesn't change)

throughput: rate ; latency: freq

# The importance of being quick

- Latency is the *fundamental metric* of computing and communication
  - Performance is measured as the latency required to perform a task
  - Everything else is a means to that end
  - Exceptions aren't computing or communication (*e.g.*, I/O capabilities such as screen size, pixel depth, digitizer resolution)

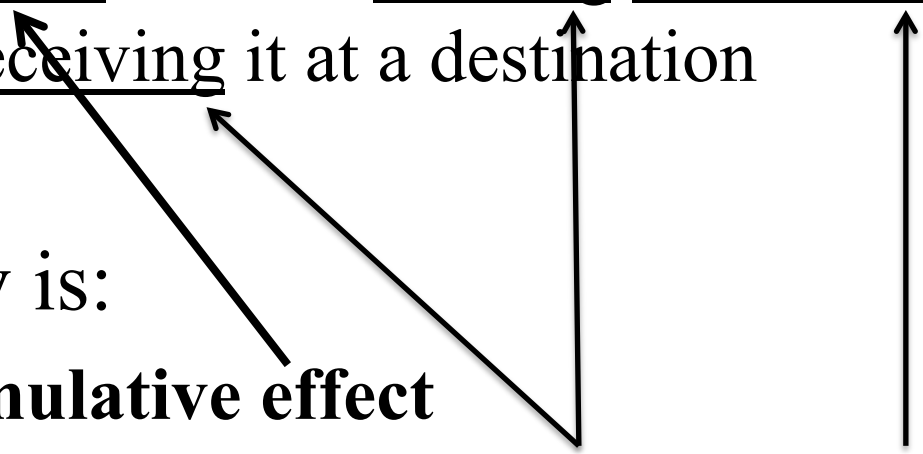
# What is latency?

- Latency is...

| (focus) | The time between: |
|---|---|
| Generic | two events |
| Interaction | asking question and receiving an answer |
| Communication | creating information at a source and receiving it at a destination |

# Defining latency

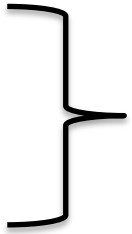
- Latency is:
  - The time between creating information at a source and receiving it at a destination



- Latency is:
  - A **cumulative effect**
  - A property of ***two events*** and a ***message*** in a ***system*** (sender/receiver/path)

# Latency isn't a single value

- The cumulative *system* impact on a *message*
  - Fixed, per-message costs
    - Header processing
    - Message house-keeping
    - Propagation delay
  - Proportional, per-bit costs
    - Message composition/interpretation
    - Transmission delay
  - Unpredictable aggregate effect
    - Not strictly additive
    - Some latencies overlap (pipeline), others don't



***Message size matters***

# Five Root Causes

of Latency

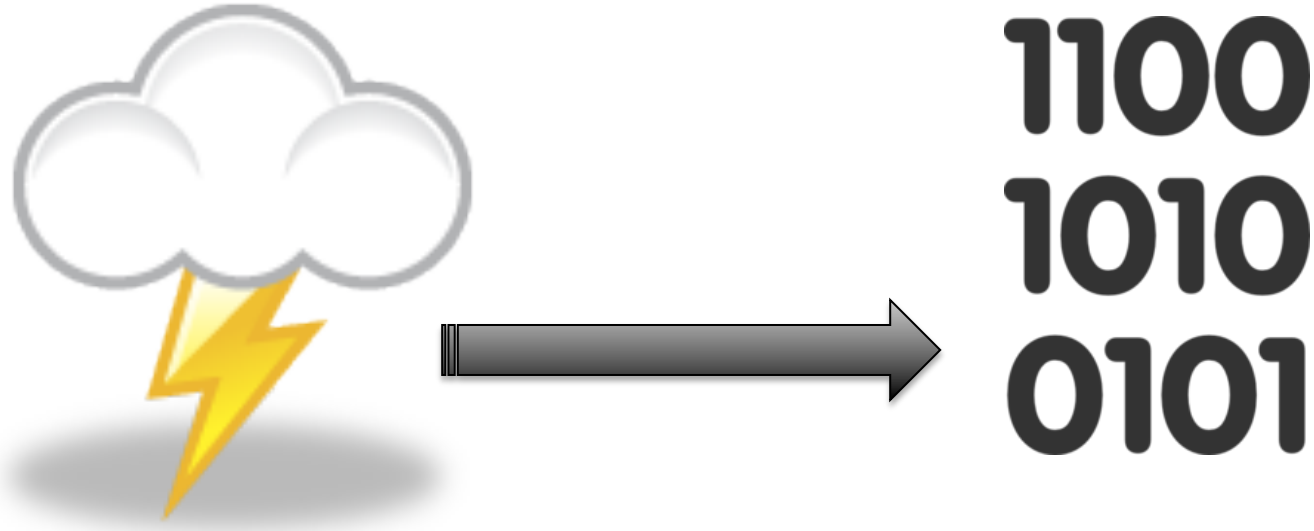
1. Generation creation of data
2. Transmission sending
3. Processing translating data
4. Multiplexing sharing resource with others
5. Grouping delay of packets

More than propagation + transmit + queue!

aggregation

# Cost #1: Generation

- Delay between occurrence of a physical event and the availability of information



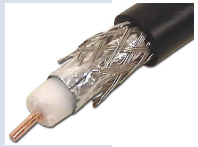
# Cost #2: Transmission

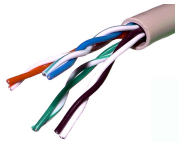
- The delay in transferring information from one location to another

# The speed of light – or less

- Constant *in each medium*:

| | |
|---|---|
| Vacuum | c (3E8 m/s) |
| Air (RF) | 0.9997 c |
| Open-ladder wire | 0.95 c |
| Twin-axial wire | 0.8 c |
| Coax wire<br>Twisted –pair wire<br>Optical fiber | 0.66 c |





# Cost #3: Processing

- The delay due to the computational translation or frequency of information

# Cost #4: Multiplexing

- The delay incurred as the result of sharing a resource

# Cost #5: Grouping

- The delay incurred to reduce the amount of control information and overhead

# Summary

- Definitions
  - Communication, networking, and protocol
- Names are just names
  - You do need to know them
  - But their meaning is just as important
- Networking didn't start with the Internet
  - There's a lot of history that's still useful
- Important characterization of network performance are time related
  - Especially latency