EE/PEP 345

Modeling and Simulation

Spring 2004

Class 12

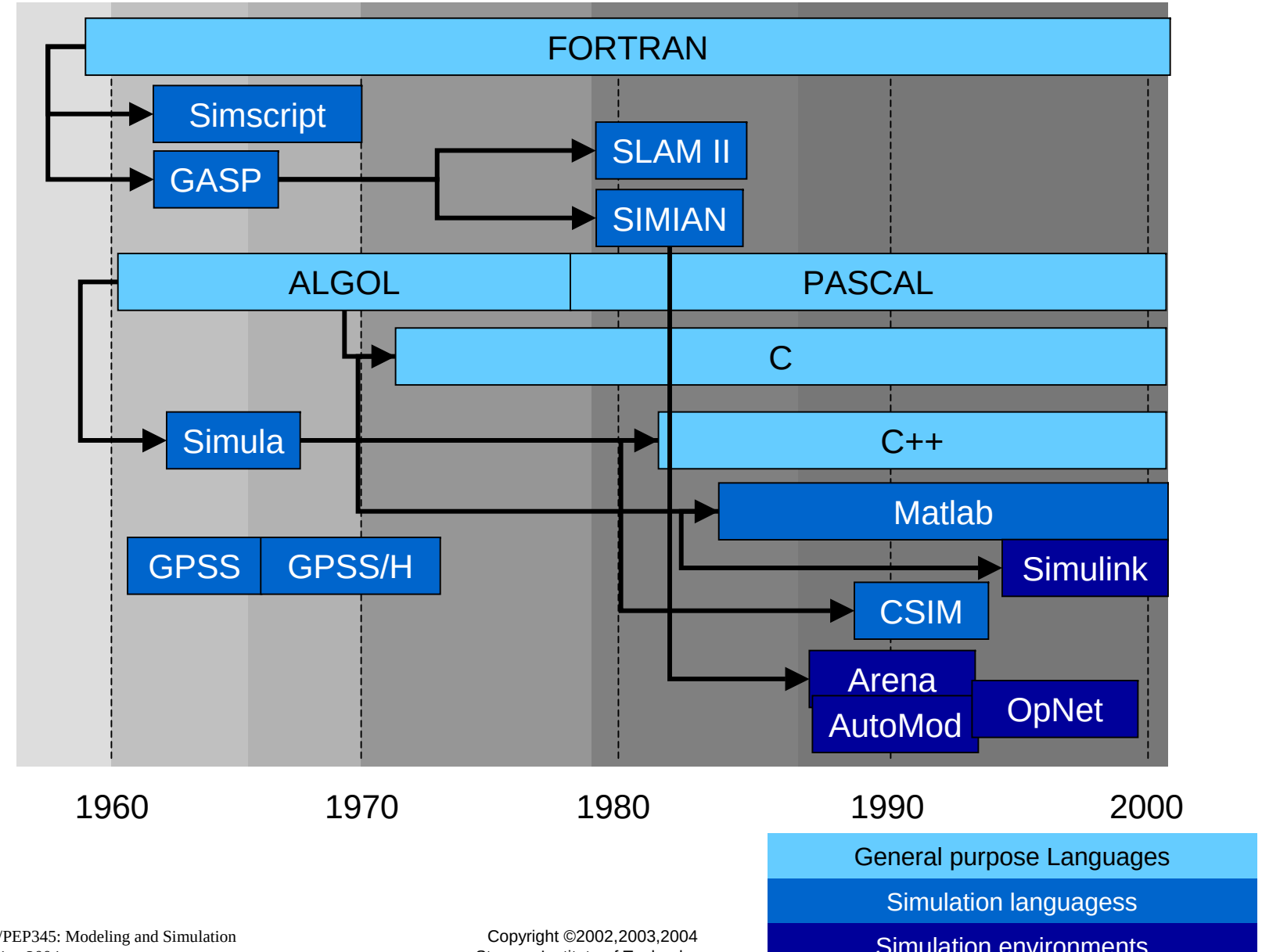
This week

- Simulation software (beyond Chapter 4 in Banks, et. al.)
- System simulation considerations

Simulation Software

- General purpose programming languages
  - FORTRAN
  - C/C++
- Simulation programming languages
  - GPSS/H™
  - SIMIAN V®
- Simulation environments
  - OpNet
  - Matlab Simulink™
  - Arena

Evolution of simulation software



Some Guidelines on Simulation Software Selection

- Consider multiple broad issues: Ease of use, accuracy, support, training, applicability to problem
- Execution speed is important: slow speed during debugging can impede development
- Caveat Emptor – vendors sometimes lie
- Try before you buy – run on sample problems. Evaluation downloads are extremely important (e.g., MathWorks offers complete 30 day evaluation copy)
- Beware of simple feature lists
- Extending package with your own C (et. al.) routines is extremely important
- Graphical interface is easiest for describing process flow. Programming language-like routines are best for describing procedural steps. You'll need both.
- Sophisticated graphical display of complex output is essential (e.g., 3-D graphs)

Model-building features

Feature	Function
Model world view	Process interaction, events
Input data analysis	Estimate distributions from raw data
Graphical model building	Block diagrams to illustrate process flow, functional blocks, network
Conditional routing	Route entities thru simulation based on state of system
Simulation programming	Interface to allow programmatic control of simulation, preferably in standard HLL
Syntax	Easily understood, consistent, unambiguous, similar to a language user already understands
Input flexibility	Accepts data from external sources, e.g., sensors, files, user
Conciseness	Powerful actions, blocks, or nodes
Randomness	Random process generators for common distributions. Ability to randomize or repeat same sequence
Specialized components	Off-the-shelf components for your particular problem
User-built custom objects	Reusable objects, templates, and sub-models. Ideally, they should be indistinguishable from built-in features
Interface with GPL	E.g., model is C-callable, Model can include C routines

Runtime Environment

Feature	Function
Execution speed	Many short runs are needed during debugging. Long runs are needed during experimentation. Fast startup as well as efficient execution are required. Ability to compile/optimize/profile model needed
Model size	There should be no hard limits, other than those imposed by platform you are running on (e.g., available RAM, hard disk space)
Interactive debugger	Breakpoints, trap event, run until condition, single step, error recovery
Model status and statistics	Display at any time during simulation
Runtime license	Ability to design a model and then run it with same structure but different parameters on multiple platforms simultaneously. At $30k per license for some packages, you don't want to have to buy 10 licenses to run 10 instances of same simulation

Simulation Packages

- Common characteristics:
  - GUI based,
  - Animated graphics
  - Automatic data collection

- **Arena**
  - modeling of business processes
  - built on SIMIAN
- AutoMod
  - focus on material handling and manufacturing processes
  - generated AVI movies of 3D animations, with pan/zoom
- Deneb/QUEST
  - robotic simulation
- OpNet Modeler/IT Guru
  - graphical modeling of complex networks
- **Matlab/SIMULINK**
  - block diagram focus
  - focus on scientific/technical applications
  - rich set of Blocksets/Toolboxes
- **MathCAD**
  - equation-based worksheets
  - includes symbolic programming (e.g., simplification/expansion of equations)

Trends in Simulation Packages

- High-fidelity simulation
  - High-accuracy simulation of complex systems
- Data exchange standards
  - Simulation input/output can be interfaced to other packages
- Distributed (client/server) computing support
  - Large organization/wide-area collaboration (e.g., across LAN, Internet)
- General purpose simulations vs. specialized simulations
  - Do it once, make it reusable
- Richer object libraries/reusable blocksets
- Multiple computer simulations to accelerate simulations

Implementation Directions

- Top Down
  - Define high level structure first, fill in details
  - Nothing is working until the details are done
- Bottom Up
  - Define the details first, stitch them together
  - Interfaces will change as more details are defined
- Straight through
  - Start at system input, progress through to final output (or vice versa)
- Outside In
  - Front and back interfaces are defined first, interior details later, meet in middle
  - Pieces may not join at the center properly
- Inside Out
  - Inner connections are completed, outer pieces are added
  - There is something to test from the beginning

The Problem

- Your (intentionally vague) hypothetical task:
  - Investigate alternatives for wireless LANs (802.11)

The Problem

- Your (intentionally vague) hypothetical task:
  - Investigate alternatives for wireless LANs (802.11)
  - This might involve
    - Research: extensions to current standards (802.11a/b/g)
      - Range, throughput, mobility, interference avoidance, other operating conditions
    - Systems engineering: does existing technology meet customer requirements for planned system?
    - Development: implementation choices, performance optimization
    - Deployment: site considerations, interference

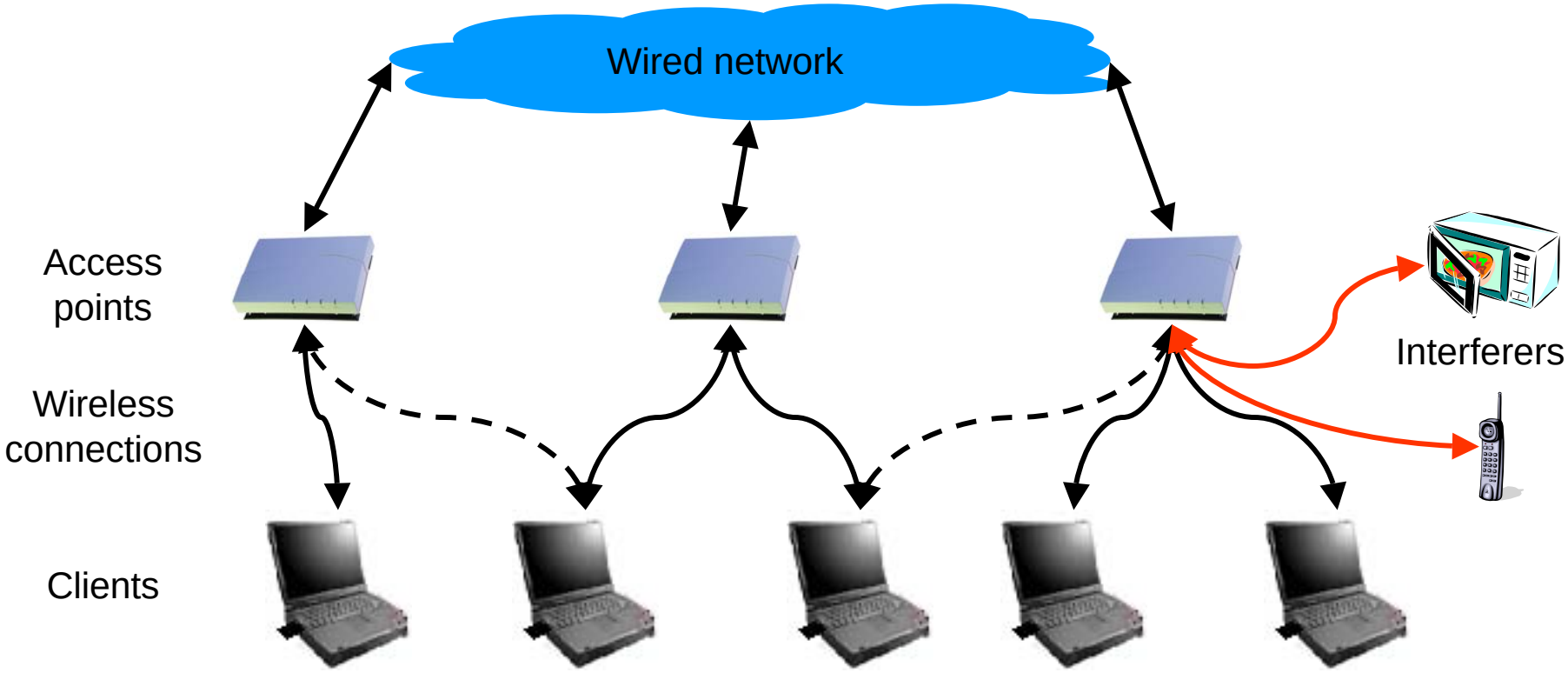
The Problem

- Your (intentionally vague) hypothetical task:
  - Investigate alternatives for wireless LANs (802.11)
  - This might involve
    - Research: extensions to current standards (802.11a/b/g)
      - Range, throughput, mobility, interference avoidance, other operating conditions
    - Systems engineering: does existing technology meet customer requirements for planned system?
    - Development: implementation choices, performance optimization
    - Deployment: site considerations, interference
- **Modeling and simulation to be used as design tool** (obviously)

The Problem

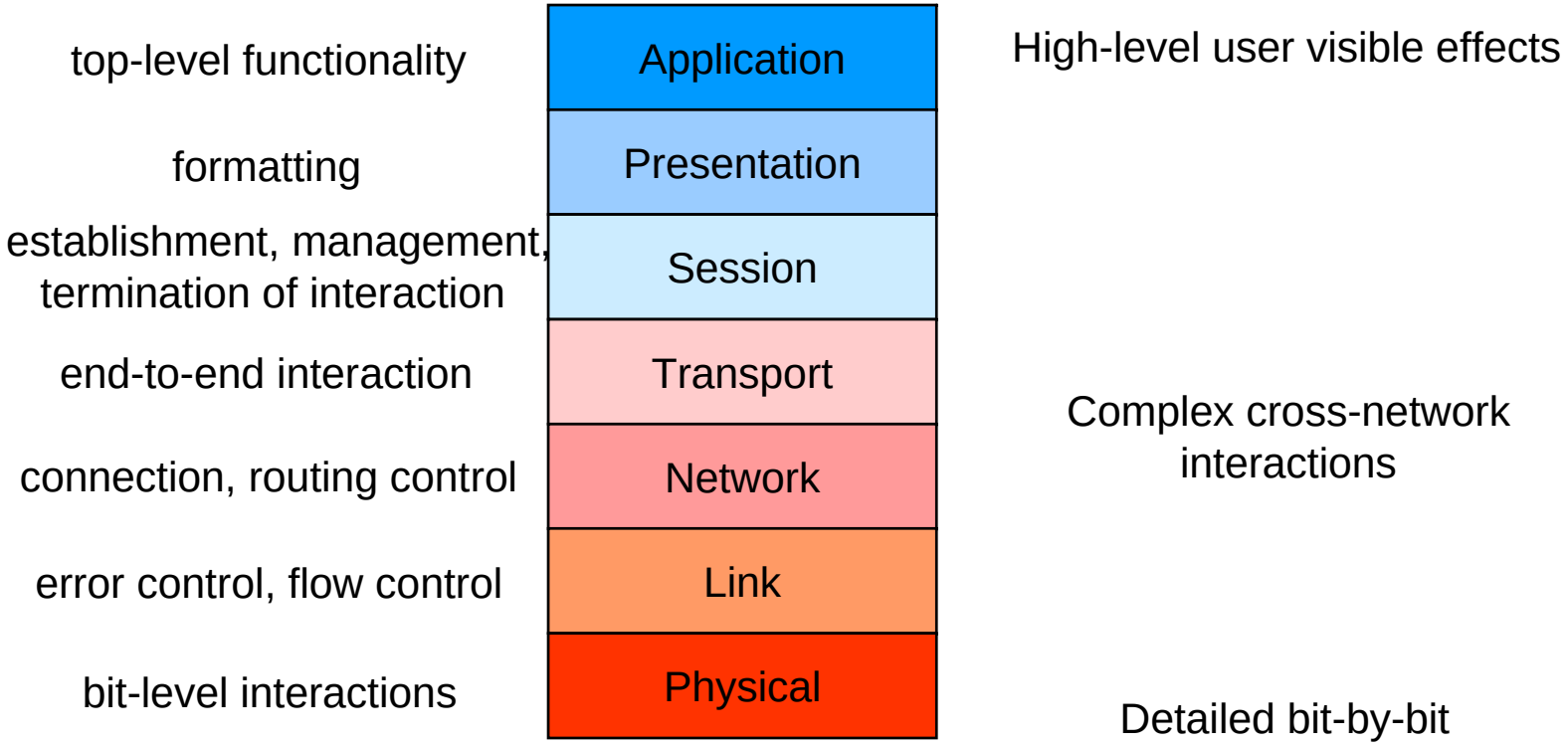
- Your (intentionally vague) hypothetical task:
  - Investigate alternatives for wireless LANs (802.11)
  - This might involve
    - Research: extensions to current standards (802.11a/b/g)
      - Range, throughput, mobility, interference avoidance, other operating conditions
    - Systems engineering: does existing technology meet customer requirements for planned system?
    - Development: implementation choices, performance optimization
    - Deployment: site considerations, interference
- **Modeling and simulation to be used as design tool** (obviously)
- Questions:
  - What are the right tools to use?
  - Where do you start to define the problem?

The Network

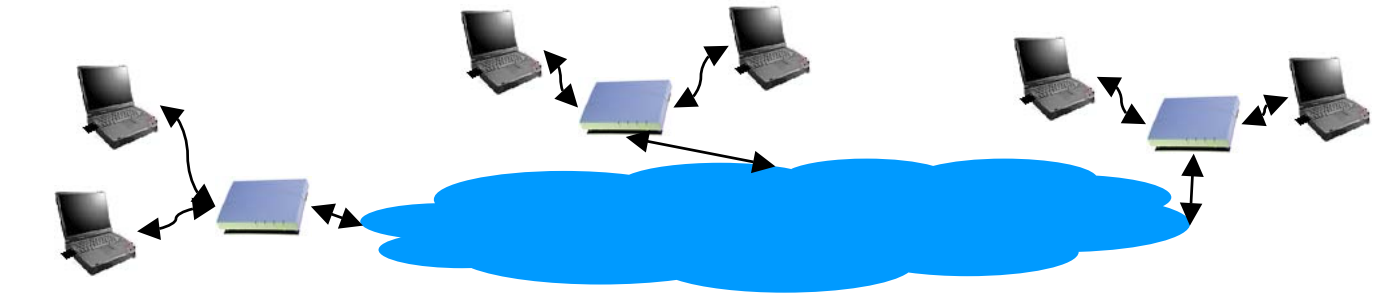


- Do you model one link in the presence of many?
- Do you model the entire network - at what level of detail?

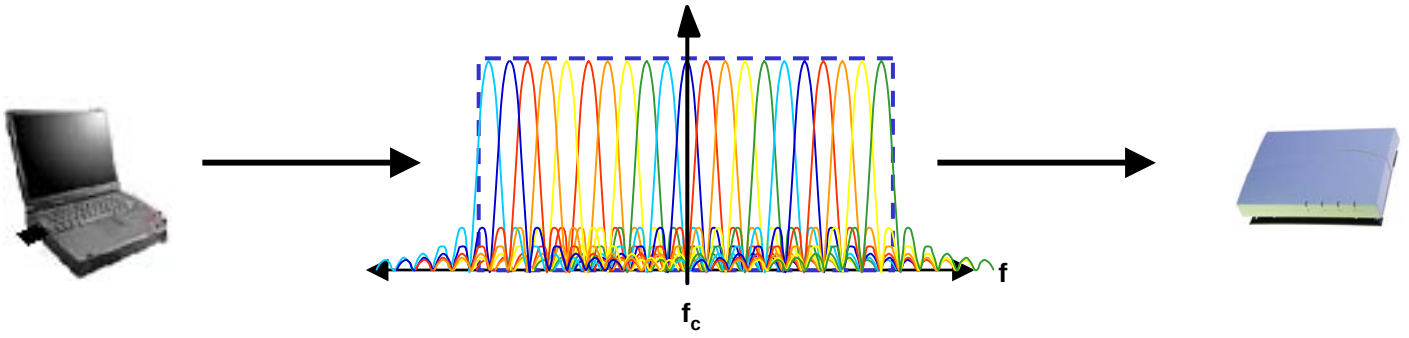
What level simulation?



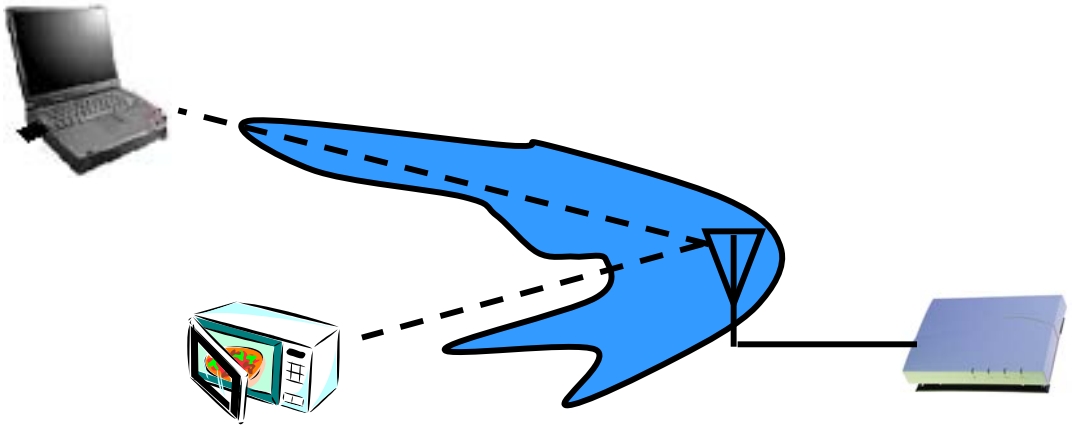
ISO 7 layer protocol stack



Arena.lnk



IEEE80211a.mdl.lnk



maybe_mrc.lnk

beam field strength.lnk

References

- *Simulation with Arena*, (with student edition software) W.D. Kelton, R.P. Sadowski & D.A. Sadowski, McGraw-Hill, 2002, ISBN 0-07-250739-X

Homework

- Study for final: In class next week
  - format similar to midterm
  - ~60% of final will be on material since midterm
  - today's material will be included, but not emphasized