CS 6114
Course Overview

Nate Foster
Cornell University
Take I
The Internet is an enormous success—it's one of the "wonders" of the modern world.
The Internet is an enormous success—it's one of the "wonders" of the modern world. With a few notable exceptions, networking remains an engineering discipline...
Although many thorny problems remain...
Although many thorny problems remain...

...transporting results from theory to practice is typically not straightforward
This started to change \( \sim 10 \) years ago with the emergence of large-scale data centers...
This started to change ~10 years ago with the emergence of large-scale data centers...

- Complexity became unmanageable
- Growing need to deploy new features
- Big players unhappy with market dynamics
Take II
Many programming languages were designed in an age when computers looked like this...
But nowadays, computers look like this...
And applications are built like this...
• Stand-alone
• Centralized
• Sequential
• Stand-alone
• Centralized
• Sequential
• Stand-alone
• Centralized
• Sequential

• Networked
• Distributed
• Concurrent
We need new kinds of abstractions and tools for programming these networked systems!

- Stand-alone
- Centralized
- Sequential
- Networked
- Distributed
- Concurrent
Take III
Network Programming
Network Programming

**Data plane:** forward packets, balances load, implements monitoring, etc.
Network Programming

Control Plane: discovers topology, computes routes, enforces policies, etc.
Network Devices: implement packet processing, buffering, queueing, etc. at line rate
Network Programming Challenges

Networks are distributed systems with thousands of interacting nodes.

Networks enforce complex security policies that span trust boundaries.

Networks are expected to offer good performance with limited resources.
Administrativa
Staff

Background
• BA Williams College
• MPhil Cambridge University
• PhD University of Pennsylvania
• Postdoc Princeton University

Office Hours
Gates 432
Thursdays 1:15-2:15pm

Contact
jnfoster@cs.cornell.edu
http://www.cs.cornell.edu/~jnfoster/
Schedule

Lecture
• Tuesdays & Thursdays
• Hollister B14
• 10:10am-11:25am

Expectations
• Attend!
• Read papers
• Contribute to discussions
Coursework and Grades

**Participation (10%)**
- Attend lectures
- Contribute to discussions

**Problem Sets (6 x 10% each)**
- Mostly programming assignments, with some pencil-and-paper problems mixed in.
- I'll supply a virtual machine, starter code, etc.

**Mini Project (30%)**
- Should be at least mildly ambitious in scope
- Design and implementation is up to you
Academic Integrity

• Cheating is a lot easier to detect than you might imagine (I use automated tools to find similarities between submissions)

• Violations are unpleasant and painful for everyone involved

• To avoid pressure, start assignments early

• A simple guideline: provide attribution for everything you obtain from an outside source

• If you run into difficulty, speak up!
Rough Outline

- Introduction
- P4 Language
- NetKAT Language
- Advanced Topics

Guest Lectures!
Themes

- **Language Design:** what abstractions should we use to program networks?
- **Compilation:** how do we map those abstractions down to effect packet-processing devices?
- **Semantics:** what mathematical objects do network programs denote?
- **Verification:** how do we check that a network behaves as expected?
- **Applications:** what innovative features can we implement with programmable networks?