CS244
Lecture 1: Introduction

Nick McKeown and
Keith Winstein
The Internet: An Exciting Time

One of the most influential inventions
- A research experiment that escaped from the lab
- ... to be the global communications infrastructure

Ever wider reach
- Today: 2 billion users
- Tomorrow: more users, computers, sensors, content

Constant innovation
- Apps: Web, P2P, social networks, virtual worlds
- Links: optics, WiFi, cellular, ...
Transforming Everything

The ways we do business
  - E-commerce, advertising, cloud computing, ...

The way we have relationships
  - E-mail, IM, Facebook friends, virtual worlds

How we think about law
  - Interstate commerce? National boundaries? Wikileaks?

The way we govern
  - E-voting and E-government
  - Censorship and wiretapping

The way we fight
  - Cyber-attacks, including nation-state attacks
But what *is* networking?
A Plethora of Protocol Acronyms?

SNMP, WAP, SIP, PPP, IPX, MAC

LLDP, FTP, UDP, ICMP, IMAP, IGMP, HIP

OSPF, RTP, BGP, HTTP, ARP, ECN, SACK, SNMP, TFTP, TLS, WAP, SIP, IPX, IP, VLAN, VTP, TFTP, DHCP, MAC

PIM, RED, SNMP, BGP, HTTP, ARP, ECN, SACK, SNMP, TFTP, TLS, WAP, SIP, IPX, IP, VLAN, VTP, TFTP, DHCP, MAC

RIP, SMTP, NAT, STUN, RTP, RTSP, RTCP, MPLS, LDP, HIP, LISP, LLDP, BFD

NNTP, FTP, UDP, ICMP, IMAP, IGMP, HIP, LISP, LLDP, BFD

DNS, SMTP, SACK, SSH, TLS, NAT, STUN, DHCP, LDP

POP, VLAN, LISP, VTP, TFTP, DHCP, LDP
A Heap of Header Formats?

### Ethernet Header

- **Version**: 4 Bytes
- **IHL**: 4 Bytes
- **Type of Service**: 4 Bytes
- **Total Length**: 4 Bytes
- **Identification**: 4 Bytes
- **Flags**: 2 Bytes
- **Fragment Offset**: 2 Bytes
- **Time to Live**: 1 Byte
- **Protocol**: 1 Byte
- **Header Checksum**: 2 Bytes
- **Source Address**: 6 Bytes
- **Destination Address**: 6 Bytes
- **Options**: Variable Length
- **Padding**: Variable Length
- **Data**: Variable Length

### HTTP Response Header

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP Status Code:</td>
<td>HTTP/1.1 200 OK</td>
</tr>
<tr>
<td>Date</td>
<td>Thu, 27 Mar 2008 13:37:17 GMT</td>
</tr>
<tr>
<td>Server</td>
<td>Apache/2.0.55 (Ubuntu) PHP/5.1.2</td>
</tr>
<tr>
<td>Last-Modified:</td>
<td>Fri, 21 Mar 2008 13:57:30 GMT</td>
</tr>
<tr>
<td>ETag</td>
<td>&quot;358a4e4-56000-ddf5c680&quot;</td>
</tr>
<tr>
<td>Accept-Ranges:</td>
<td>bytes</td>
</tr>
<tr>
<td>Content-Length:</td>
<td>352256</td>
</tr>
<tr>
<td>Connection:</td>
<td>close</td>
</tr>
<tr>
<td>Content-Type:</td>
<td>application/x-msdos-program</td>
</tr>
</tbody>
</table>
TCP/IP Header Formats in Lego
A Big Bunch of Boxes?
An Application Domain?
A place to apply theory?

- Algorithms and data structures
- Control theory
- Queuing theory
- Optimization theory
- Game theory and mechanism design
- Formal methods
- Information theory
- Cryptography
- Programming languages
- Graph theory
A place to build systems?

- Distributed systems
- Operating systems
- Computer architecture
- Software engineering
- ...


“What are the top ten classic problems in networking? I would like to solve one of them and submit a paper to SIGCOMM.” After hearing that we don't have such a list: “Then how do you consider networking a discipline?”

“So, these networking research people today aren't doing theory, and yet they aren't the people who brought us the Internet. What exactly are they doing?”

“Networking papers are strange. They have a lot of text.”

Is networking a problem domain or a scholarly discipline?
“There is a tendency in our field to believe that everything we currently use is a paragon of engineering, rather than a snapshot of our understanding at the time. We build great myths of spin about how what we have done is the only way to do it to the point that our universities now teach the flaws to students (and professors and textbook authors) who don’t know better.” -- John Day (Internet pioneer)
Before you all leave …
So, Why is Networking Cool?

Relevant
- Can measure/build things
- Can impact the real world

Interdisciplinary
- Well-motivated problems + rigorous solution techniques
- Interplay with policy and economics

Widely-read papers
- Many of the most cited papers in CS are in networking
- Congestion control, distributed hash tables, resource reservation, self-similar traffic, multimedia protocols,…
- Three of top-ten CS authors (Shenker, Jacobson, Floyd)
So, Why is Networking Cool?

Young, relatively immature field
- Tremendous intellectual progress is still needed
- You can help decide what networking really is

Defining the problem is a big part of the challenge
- Recognizing a need, formulating well-defined problem
- ... is at least as important as solving the problem.

Lots of platforms for building your ideas
- Testbeds: Emulab, PlanetLab, Orbit, GENI
- Programmability: Click, NetFPGA, Mininet
- Routing software: Quagga, XORP, and Bird
- Measurements: RouteViews, traceroute, Internet2
Architectural questions tend to dominate CS networking research
Decomposition of Function

Definition and placement of function
– What to do, and where to do it

The “division of labor”
– Between the host, network, and management systems
– Across multiple concurrent protocols and mechanisms
Software Defined Network (SDN)
Network Function Virtualization (NFV)
Network Function Virtualization (NFV)
“Software will eat the world”

Marc Andreessen
Conclusion

Networking is extremely cool right now

– Real, important problems
– Opportunities for impact
– Inherently interdisciplinary

But the field is immature

– More of a “domain” than a “discipline”
About this class
Goals

1. To become familiar with the field of networking research: Network architecture, protocols and systems.

2. To get some practice in the art of reading research papers.

3. Learn the art of reproducing research results OR do an original project.

It’s a big field, so we have to focus on just a few topics.
Basics

Lecture

– Each class we will discuss 1-2 papers
– You **must** read the papers before class
– Papers should be read in depth
– Most of the lecture will be spent on discussion
– 30% of your grade comes from critiques, and in-class participation

So… read the papers, come to class, and be ready to participate
Grade

Reading and participation 35%
- Critiques before class: 20%
- In-class participation: 15%

Programming assignments 45%
- PA #1: 10% Congestion control competition
- PA #2: 10% Reproducing a particular research result
- PA #3: 25% EITHER an original research project OR reproducing a paper of your choice (not previously reproduced in CS244)

Midterm exam 20%
- Midterm: 20% (in-class, Wednesday, May 16)
- No final!
In-class participation

Come prepared to discuss the main ideas!

We will all learn from each other

Attendance is a necessary but not sufficient condition for “participation”

Let’s have no laptops in class
Critiques

What to submit?

– Short critique for each paper before the class (by midnight the night before the lecture)
– Submit online (see class webpage)

Questions to answer while writing your critique:

– What problem are the authors solving?
– What is the main idea and what do you think of it?
– What was the status quo ante before this paper, and what is the clearest way to explain this paper's contribution?
– How well is the paper written?

Grade: 0, 1, or 2 points
Three Assignments

Assignment 1 – Congestion control competition
- To be released this Wednesday, April 4
- Due Tuesday, Apr 17, 5 p.m.

Assignment 2 – “Compulsory” result reproduction
- Due Friday, May 4, 5 p.m.
- Paper TBA

Assignment 3 – EITHER
- Reproducing a new result from a networking research paper
- OR An original project in computer networking
- Proposal due Tuesday, May 1 at 5pm.
- Intermediate report due Tuesday, May 22 at 5pm.
- Final Report due Saturday, June 9 at 5pm.
- (optional) Presentation Wednesday, June 6 in-class.
Logistics

Who will lead the discussions
- Nick McKeown (nickm@stanford.edu)
- Keith Winstein (keithw@cs.stanford.edu)
- Some guest experts

TAs
- Saachi Jain (saachi@stanford.edu)
- Emre Orbay (eorbay@stanford.edu)
Contact

Whenever possible: Piazza
  – Quickest response
  – Someone else probably has same question
  – Please don’t send to class list

If private: Post a private Piazza post

All extension requests must go to Nick and Keith
PA #1

Over to Keith…
PA #1: Congestion control contest

• Project goals
  – Learn basic network programming
  – TCP effects on queue occupancy

• Assignment out April 4
  – Best run on Ubuntu Linux VM
  – Can run in VirtualBox or VMware (your choice)
Hints on reading a paper
Keshav: “How to Read a paper”, CCR 2007

Three stage approach

1. Read quickly in 5-10 minutes
2. Read with greater care; ignore proofs
3. Deconstruct paper; question all assumptions
Stage 1: 5-10 minute read

Read title, abstract, introduction, section headings, conclusion, reference list.

Look for “5 C’s”

1. **Category**: What type of paper is it?
2. **Context**: Where does it fit in?
3. **Correctness**: Do assumptions make sense?
4. **Contributions**: What are the main ones?
5. **Clarity**: Is it well-written?
Stage 2: Read with care

- Spend an hour re-reading paper in detail
- Try to understand the “story”
- Summarize the main thrust
- Identify main supporting evidence
Stage 3: Deconstruct the paper

• This can take one or more hours
• Understand every proof
• Question every assumption
• Identify missing references
• Why was the paper written this way?
• What was the author trying to say?
• How else could the paper have been written?
First papers
Read thoroughly, submit critiques

For Wednesday:
1. The Design Philosophy of the DARPA Internet Protocols - Clark, 1988

For Monday:
1. End-to-End Arguments in System Design - Saltzer, Reed and Clark, 1984
2. Flow Rate Fairness: Dismantling a Religion – Briscoe, 2007
First optional papers
Read through briefly

Optional (easy reads)

1. A Brief History of the Internet - Leiner et al., 2003
2. On Distributed Communication Networks – Paul Baran, 1963
Our first paper

The Design Philosophy of the DARPA Internet Protocols

David D. Clark*
Massachusetts Institute of Technology
Laboratory for Computer Science
Cambridge, MA. 02139


Abstract

The Internet protocol suite, TCP/IP, was first proposed fifteen years ago. It was developed by the Defense Advanced Research Projects Agency (DARPA), and has been used widely in military and commercial systems. While there have been papers and specifications that describe how the protocols work, it is sometimes difficult to deduce from these why the protocol is as it is. For example, the Internet protocol is based on a connectionless or datagram mode of service. The motivation for this has been greatly misunderstood. This paper attempts to capture some of the early reasoning which shaped the Internet protocols.

1. Introduction

For the last 15 years, the Architecture

architecture into the IP and TCP layers. This seems basic to the design, but was also not a part of the original proposal. These changes in the Internet design arose through the repeated pattern of implementation and testing that occurred before the standards were set.

The Internet architecture is still evolving. Sometimes a new extension challenges one of the design principles, but in any case an understanding of the history of the design provides a necessary context for current design extensions. The connectionless configuration of ISO protocols has also been colored by the history of the Internet suite, so an understanding of the Internet design philosophy may be helpful to those working with ISO.

This paper catalogs one view of the history of the Internet architecture.
David D. Clark (MIT)

• Chief Protocol Architect for the Internet from 1981.
• Continues to be a network visionary today.
• At the time of writing (1987)…
  – (Almost) no commercial Internet
  – Number of hosts reaches 10,000
  – NSFNET backbone 1 year old; 1.5Mb/s
  – 1 yr after Cisco’s 1st product, IETF started
The Design Philosophy of the DARPA Internet Protocols

**Goal 0**: An “effective” technique for multiplexed utilization of existing interconnected networks.

**Goal 1**: Internet communication must continue despite loss of networks or gateways.

**Goal 2**: The Internet must support multiple types of communication service.

**Goal 3**: The Internet architecture must accommodate a variety of networks [underneath].

**Goal 4**: The Internet architecture must permit distributed management of its resources.

**Goal 5**: The Internet architecture must be cost effective.

**Goal 6**: The Internet architecture must permit host attachment with a low level of effort.

**Goal 7**: The resources used in the internet architecture must be accountable.