CS 6114/5114 Network Programming Languages

Nate Foster
Cornell University
Spring 2013

Based on lecture notes by Jennifer Rexford and Michael Freedman

Administrivia

Instructor

Nate Foster

jnfoster@cs.cornell.edu

Background

PhD @ Penn

Postdoc @ Princeton

Office hours

Tuesdays 3-4pm

Upson 4137

Additional office hours

By request

Schedule

Lectures

Tuesdays & Thursdays
Hollister Hall 306
8:40am-9:55am

Expectations

Attend!
Read papers
Contribute to discussions

Breakfast

I will cover a number of breakfasts
Volunteers can sign up on Piazza

Coursework and Grades

Participation (25%)

- Review one paper per class, due at the start of class
- Two ~10 minute presentations during the semester
- Contribute to discussions

Problem sets (3 x 15% each)

- During first half of the semester
- Mostly programming assignments

Course project (40%)

- Teams of 2-3
- Intermediate checkpoints
- Final report and presentation

Late Policy

Reviews

- Four "misses" with no questions asked

Problem sets

- 10% late penalty per day until I start grading
- After that, no credit

Save your code…

- Submit early and often
- Use version control (svn, git, etc.)
- CMS is your friend

If you have a major emergency (e.g., medical, family) please talk to me as soon as possible.

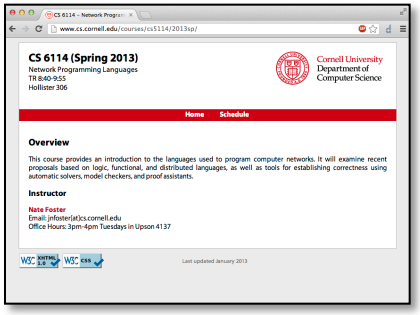
Online Resources

Website

www.cs.cornell.edu/Courses/cs5114

Readings

Lectures

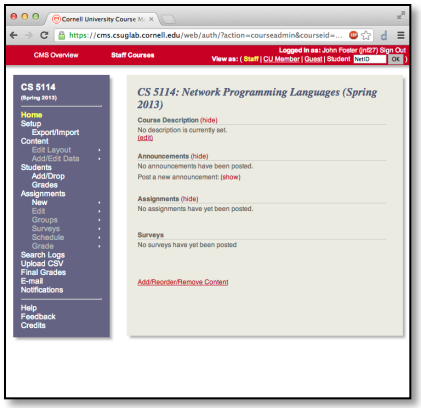


CMS

cms.csuglab.cornell.edu

Problem sets

Grades



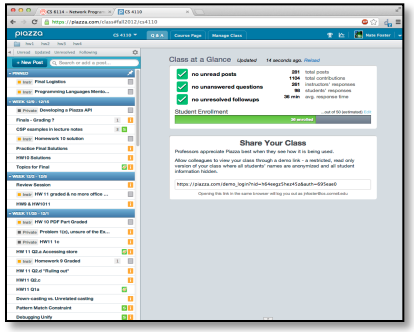
Piazza

piazza.com

Announcements

Scheduling

Discussion



Academic Integrity

Strictly enforced

Violations are easier to detect than you might think

Unpleasant and painful for everyone involved

To avoid pressure, start problem sets early

A simple guideline: provide attribution for *everything* you obtain from another source

Let me know if you run into difficulty

Networks: An Exciting Time

The Internet

- A research experiment that escaped the lab…
- Became *the* global communications infrastructure

An ever-growing reach

- Today: 1.7+ billion users
- Tomorrow: more users, computers, devices

Constant innovation

- Appications: Web, social networks, peer-to-peer, …
- Links: fiber optics, WiFi, cellular, …

Networks Are Transforming Everything

Business

- E-commerce, advertising, cloud computing, ...

Relationships

- E-mail, instant messaging, Facebook, virtual worlds

Legal system

- Interstate commerce? National boundaries?

Government

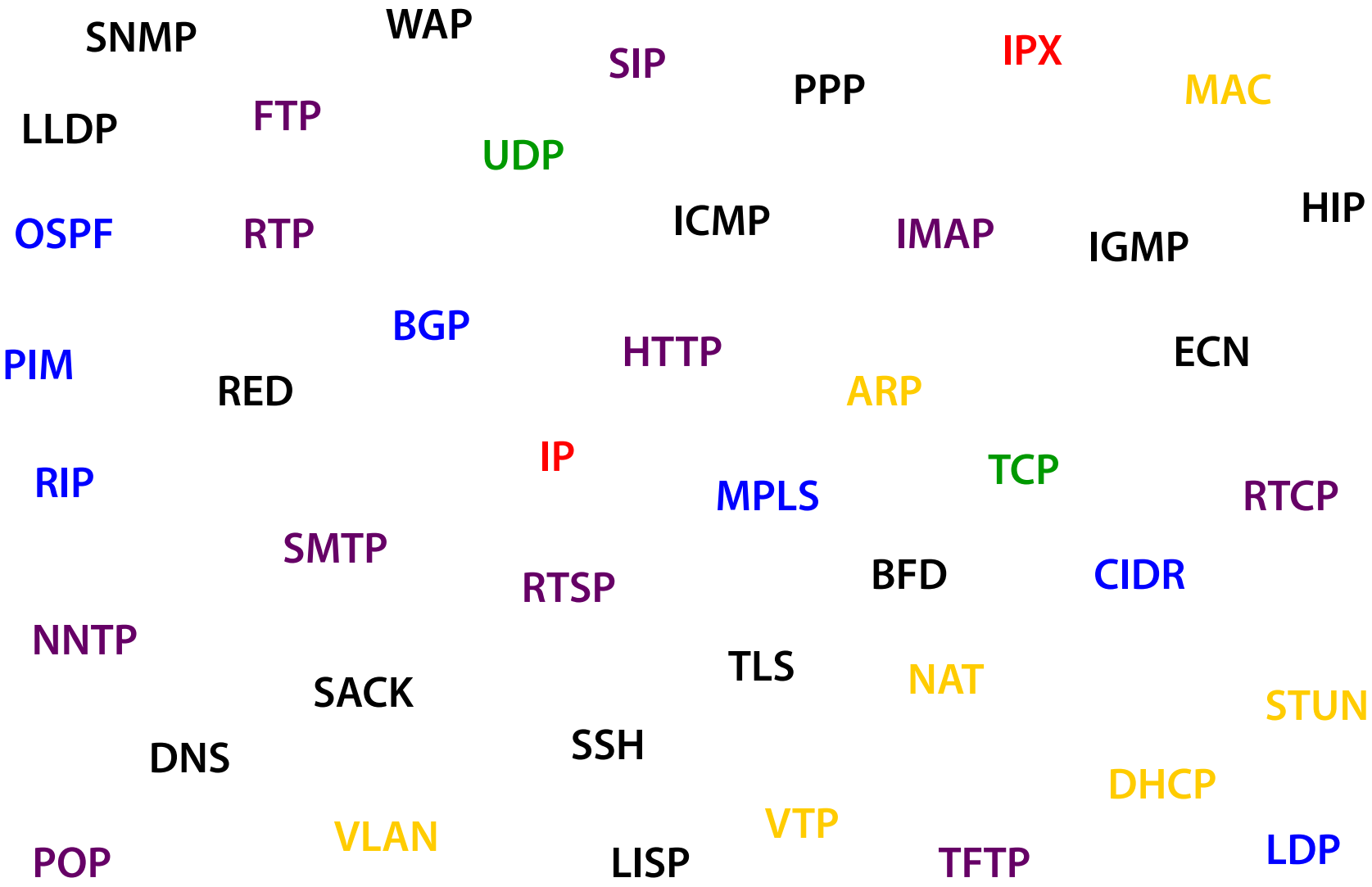
- E-voting and e-government
- Censorship and wiretapping

Warfare

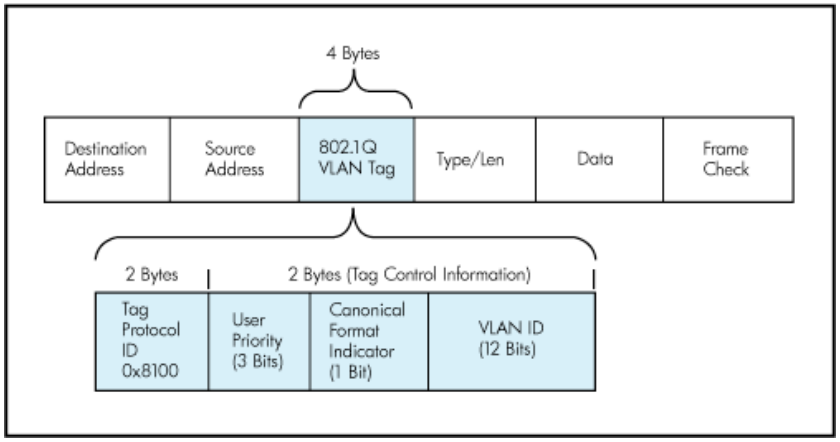
- Drones and cyber-warfare

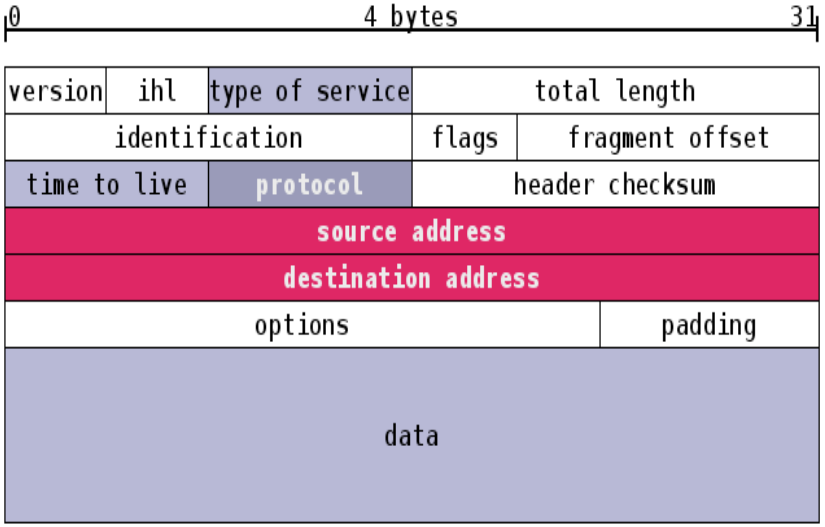
But, What *is* Networking?

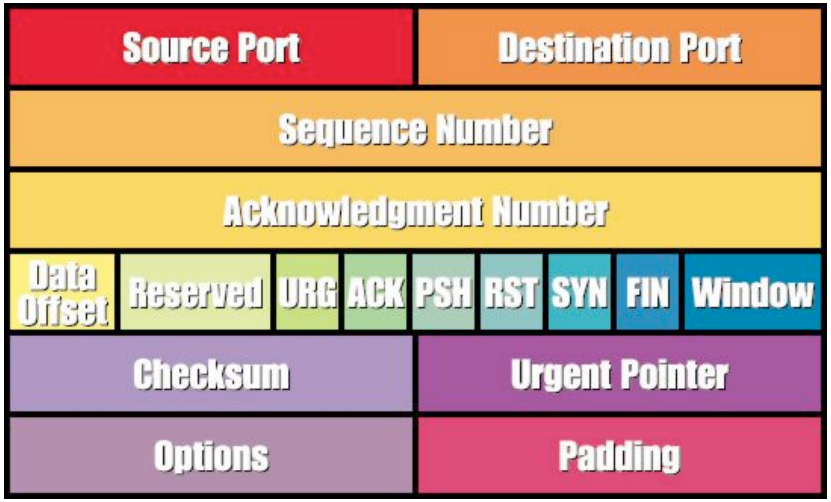
A Plethora of Protocol Acronyms?



A Heap of Header Formats?



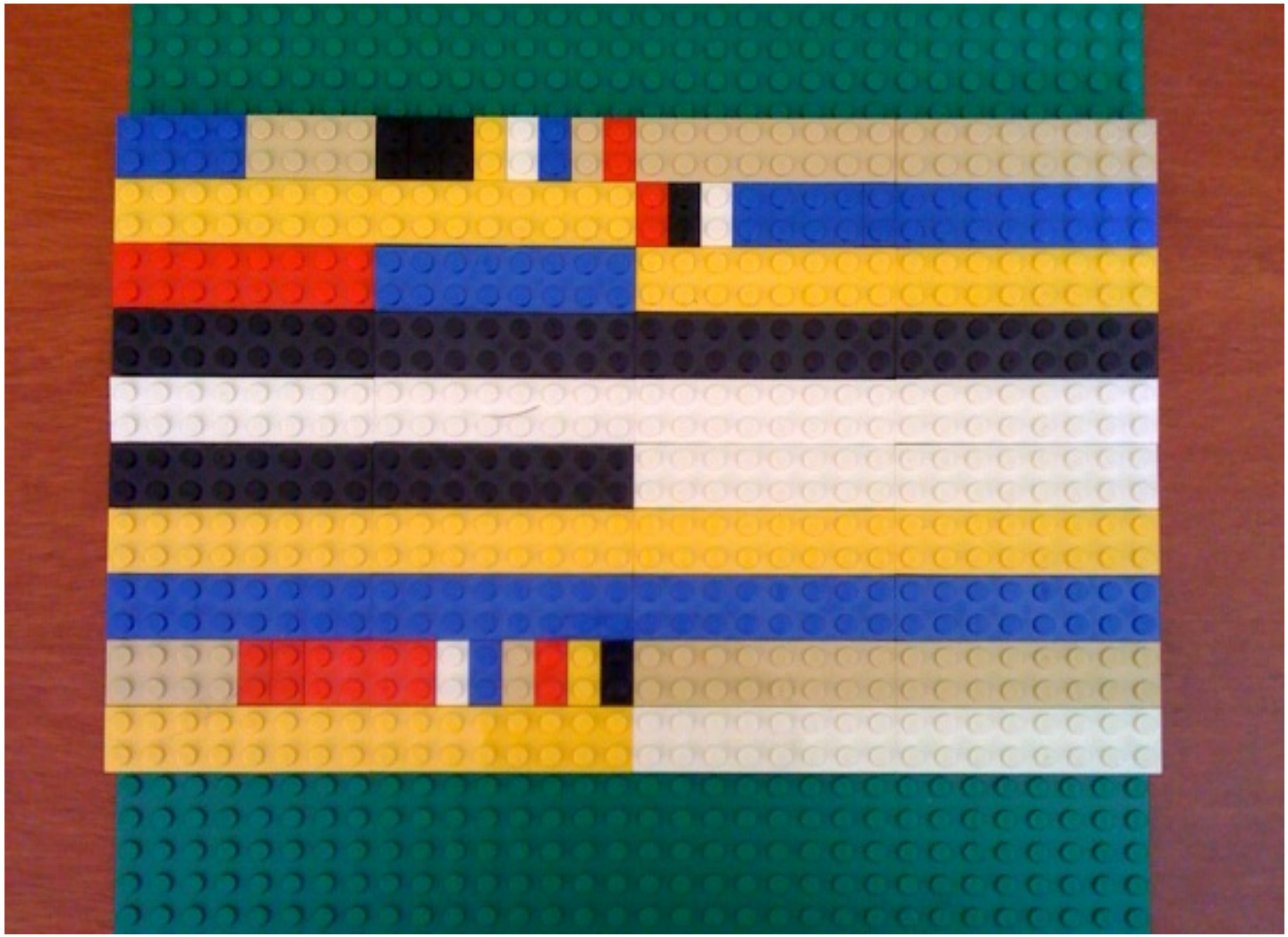




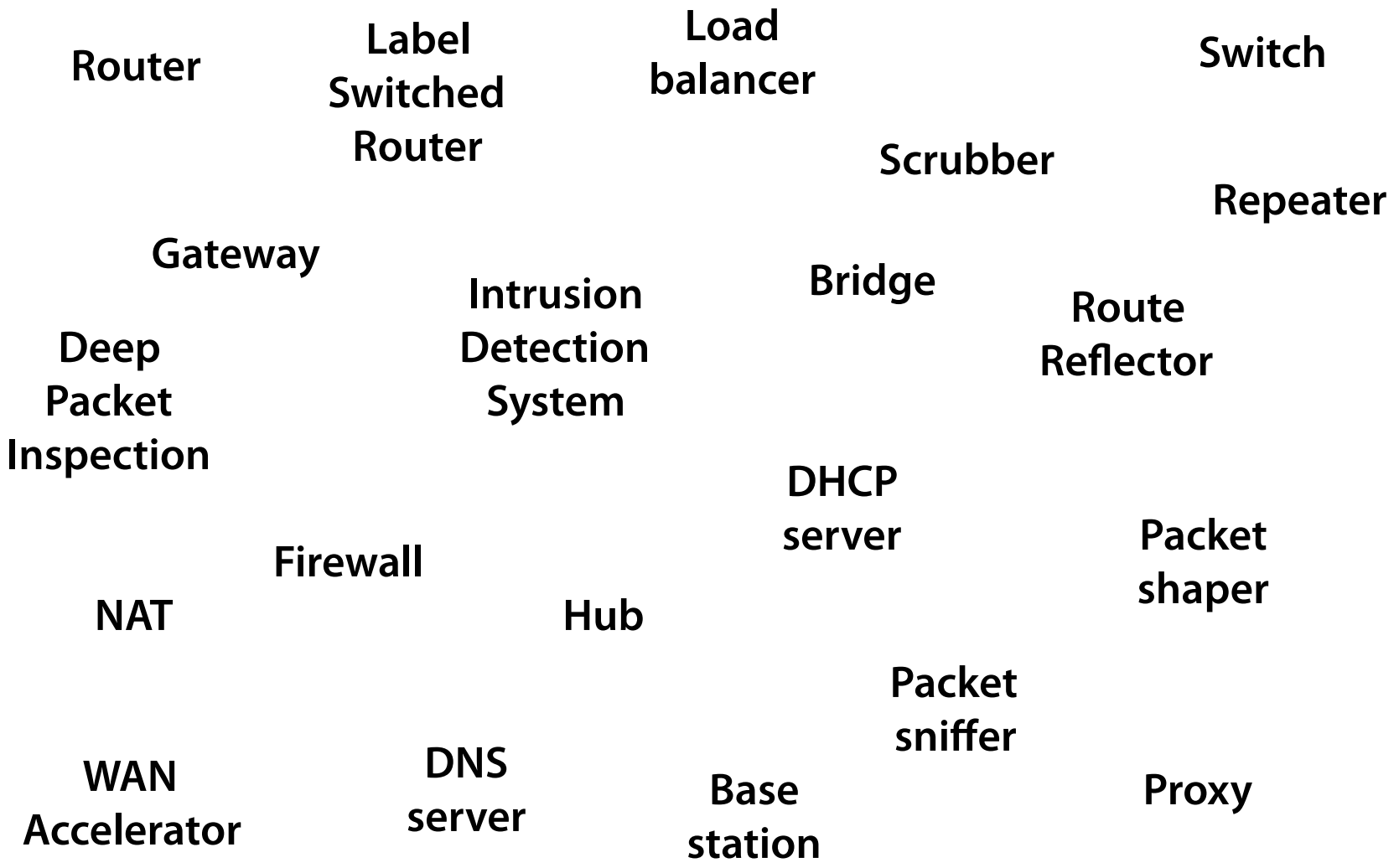
HTTP Response Header

Name	Value
HTTP Status Code: HTTP/1.1 200 OK	
Date:	Thu, 27 Mar 2008 13:37:17 GMT
Server:	Apache/2.0.55 (Ubuntu) PHP/5.1.2
Last-Modified:	Fri, 21 Mar 2008 13:57:30 GMT
ETag:	"358a4e4-56000-ddf5c680"
Accept-Ranges:	bytes
Content-Length:	352256
Connection:	close
Content-Type:	application/x-msdos-program

TCP/IP Header Formats in Lego



A Bunch of Boxes?



A Ton of Tools?

arpwatch

tcpdump

syslog

wget

nslookup

traceroute

trat

snort

nmap

whois

ipconfig

rancid

ntop

bro

net-snmp

dig

ping

iperf

NDT

wireshark

dummynet

mrtg

What Do Our Peers Say?

"You networking people are very curious. You really love your artifacts."

"I fell asleep at the start of the semester when the IP header was on the screen, and woke up at the end of the semester with the TCP header on the screen."

"Networking is all details and no principles."

"ARP, DHCP, ICMP, IGMP, IP, SONET, TCP, UDP, FML"

Is networking "just the (arti)facts"?

An Application Domain?

Application Domain for Theory?

Algorithms and data structures

Control theory

Queuing theory

Optimization theory

Game theory

Formal methods

Information theory

Cryptography

Graph theory

Application Domain for Systems?

Distributed systems

Operating systems

Computer architecture

Programming languages

Software engineering

…

An Exercise in Entrepreneurship?

Identify a need or desirable capability

- Whether previously known or not

Invent a new feature or system that provides it

Determine how it fits in the existing network

Build and/or evaluate your solution

Pitch or $ell the problem and solution to others

- Whether to investors or a program committee

Bask in glory, or lick your wounds

What Peers in Other Fields Say?

"Networking papers are strange. They have a lot of text."

"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 is an opportunistic discipline."

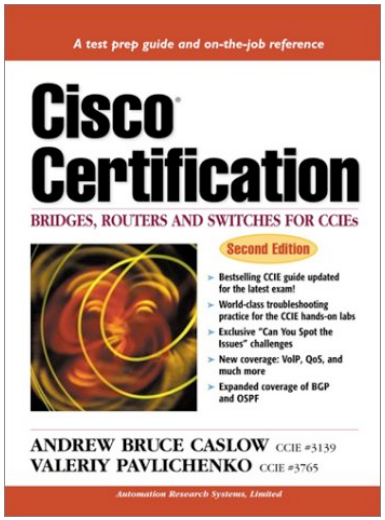
Is networking a problem domain or a scholarly discipline?

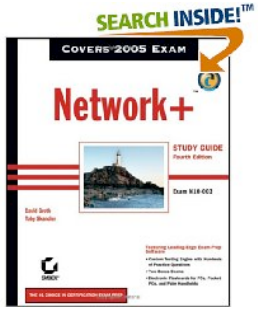
What Do We Teach Networking Students?

How Practitioners Learn Networking

Certification courses

- How to configure specific pieces of equipment





"On the job" training, AKA "trial by fire"

How Colleges Teach Networking

Undergraduates: how the Internet works

Graduates: read the 20 "best" papers

Few general principles, little "hands-on" experience

> "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

Now That I've Bummed You Out…

Or, why should you stay in this class?

Why is Networking Cool?

Tangible, direct relationship to reality

- Can measure/build things (we do "love our artifacts")
- Can truly effect far-reaching change in the real world

Inherently interdisciplinary

- Well-motivated problems + rigorous techniques
- Interplay with policy, economics, and social science

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)

Why is Networking Cool? (Continued)

Young, relatively immature field

- Great if you like to make order out of chaos
- Tremendous intellectual progress still needed
- You can help shape the field!

Defining the problem is a big part of the challenge

- Recognizing a need, formulating a well-defined problem
- … is at least as important as solving the problem…

Lots of platforms for building your ideas

- Programmability: OpenFlow, Click, NetFPGA,…
- Routing software: Quagga, XORP, Bird,…
- Testbeds: Mininet, Emulab, PlanetLab, Orbit, GENI, …
- Measurements: RouteViews, traceroute, Internet2, …

But That Doesn't Say What Networking Really Is

Or, what will this course be about?

One Take on Defining Networking

How to

- Design and operate components and protocols
- That may solve well-defined engineering problems
- That may can be used and combined in many ways

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
- What makes a good division of labor?

What Excites Me about Networking

Freedom to introduce new functionality

Designing algorithms, protocols, and data structures that offer better performance, robustness, security, …

The art of system design for cleaner abstractions and easier management…

The development of programming languages and verification tools that implement these abstractions and provide assurance

What Is This Course About?

Classic Work

- What problems were being solved?
- What were the underlying assumptions and solutions?

Modern Work

- Datacenters: the engine driving the tech boom
  - New settings means new problems and challenges
- Software-Defined Networking
  - Eliminate legacy artifacts; provides freedom to program the network directly

Focus on *languages* used to express network algorithms and new tools for verifying correctness

Course Structure

Internet Basics

Host-Network Division of Labor

Packet switching

- Divide messages into a sequence of packets
- Headers with source and destination address

Best-effort delivery

- Packets may be lost
- Packets may be corrupted
- Packets may be delivered out of order

Host-Network Interface: Why Packets?

Traffic is bursty

- Logging in to remote machines
- Exchanging e-mail messages

Don't want to waste bandwidth

- No traffic exchanged during idle periods

Allows multiplexing

- Different transfers share access to same links

Packets can be delivered by most anything

- RFC 1149: IP Datagrams over Avian Carriers

Why Best-Effort?

Means never having to say you're sorry…

- Don't reserve bandwidth and memory
- Don't do error detection and correction
- Don't remember from one packet to next

Easier to survive failures

- Transient disruptions are okay during failover

Can run on nearly any link technology

- Greater interoperability and evolution

Intermediate Transport Layer

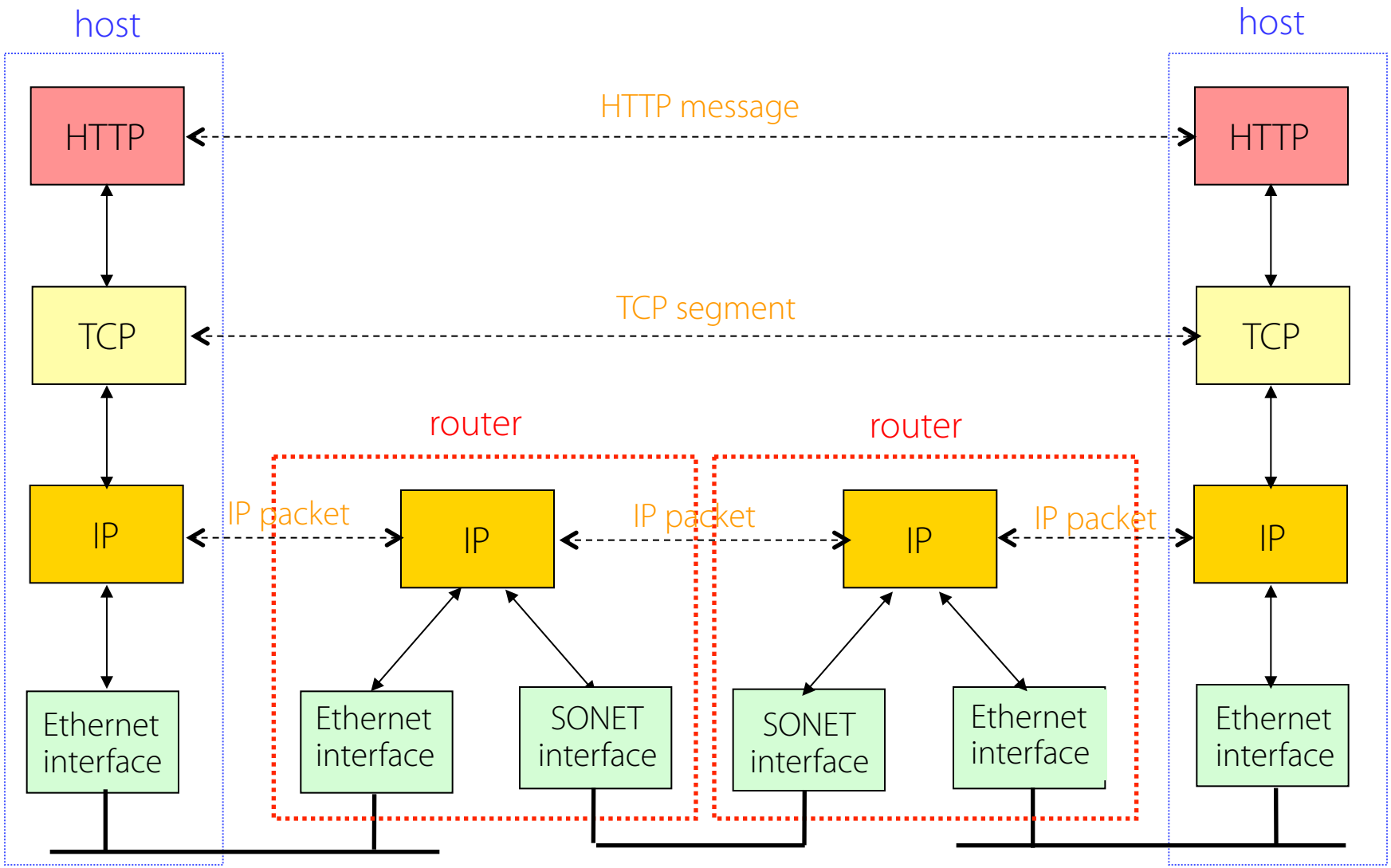
But applications want efficient, accurate transfer of data in order, in a timely fashion

- Let end hosts handle all of that!
- This is the classic "end-to-end argument"

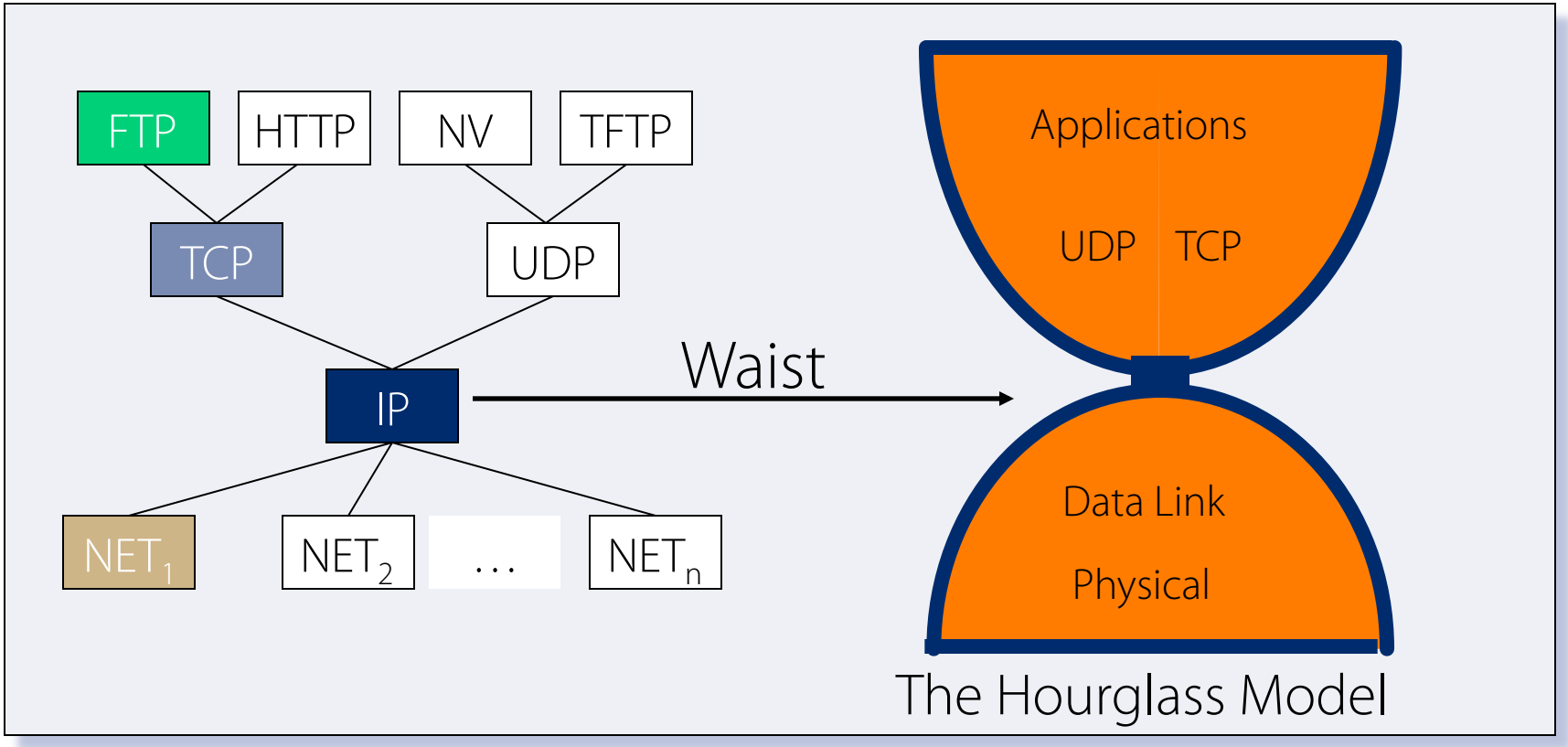
Transport layer can optionally…

- Retransmit lost packets
- Put packets back in order
- Detect and handle corrupted packets
- Avoid overloading the receiver
- <insert your favorite requirement here>

IP Suite: End Hosts vs. Routers

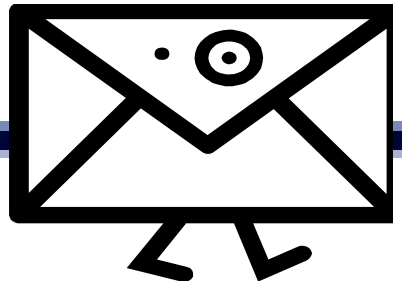


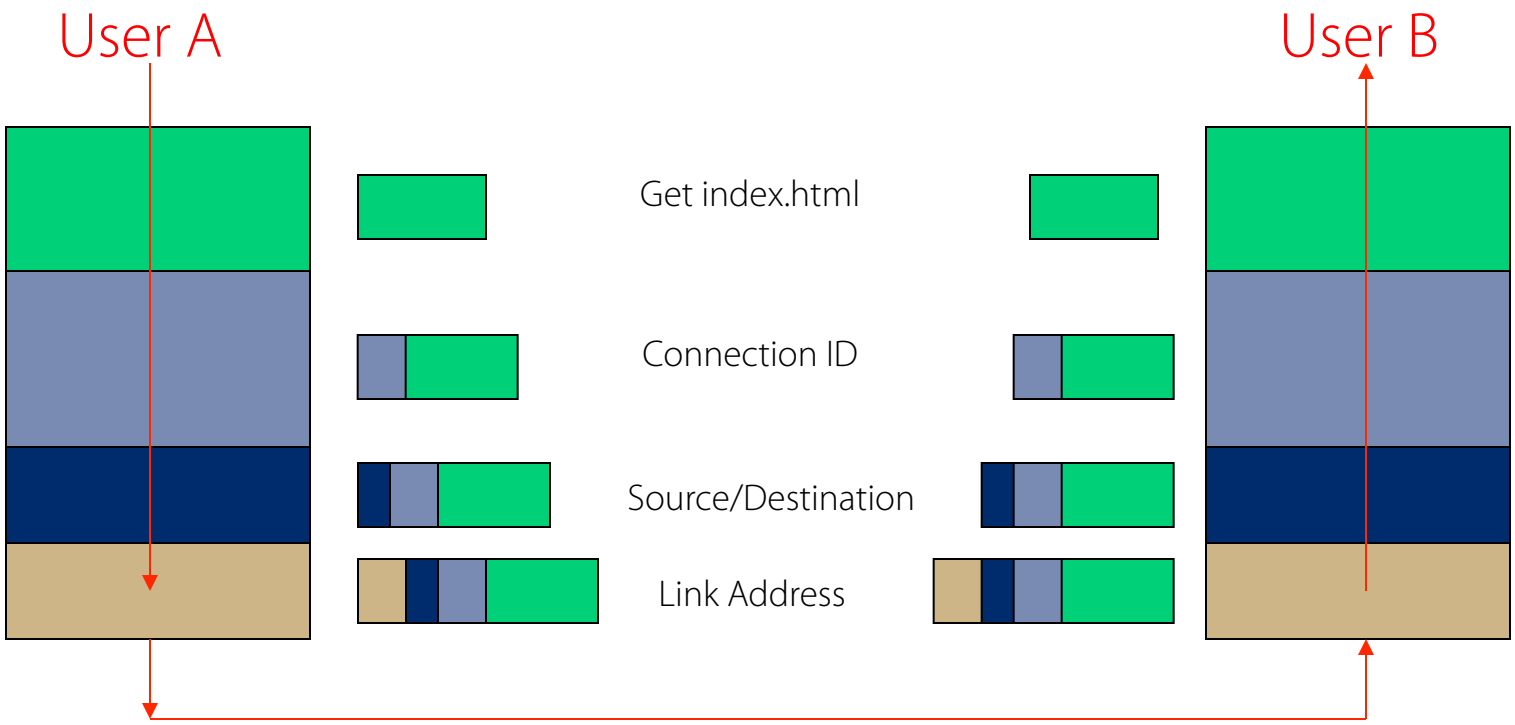
The "Narrow Waist" of the Internet



The narrow waist facilitates interoperability

Layer Encapsulation





Next Few Classes: Review

Host

- Network discovery and bootstrapping
- Resource allocation and interface to applications

Control plane

- Distributed algorithms for computing paths
- Disseminating the addresses of end hosts

Data plane

- Streaming algorithms and switch fabric
- Forward, filter, buffer, schedule, mark, monitor, …

How to Read a Paper

Keshav's Three-Pass Approach: Step 1

A ten-minute scan to get the general idea

- Title, abstract, and introduction
- Section and subsection titles
- Conclusion
- Bibliography

What to learn: the five C's

- Category: What type of paper is it?
- Context: What body of work does it relate to?
- Correctness: Do the assumptions seem valid?
- Contributions: What are the main research contributions?
- Clarity: Is the paper well-written?

Decide whether to read further…

Keshav's Three-Pass Approach: Step 2

A more careful, one-hour reading

- Read with greater care, but ignore details like proofs
- Figures, diagrams, and illustrations
- Mark relevant references for later reading

Grasp the content of the paper

- Be able to summarize the main thrust to others
- Identify whether you can (or should) fully understand

Decide whether to…

- Abandon reading the paper in greater depth
- Read background material before proceeding further
- Persevere and continue on to the third pass

Kesha's Three-Pass Approach: Step 3

Virtual re-implementation of the work

- Making the same assumptions, recreate the work
- Identify the paper's innovations and its failings
- Identify and challenge every assumption
- Think how you would present the ideas yourself
- Jot down ideas for future work

When should you read this carefully?

- Reviewing for a conference or journal
- Giving colleagues feedback on a paper
- Understanding papers closely related to your research
- Deeply understanding a classic paper in the field

http://dl.acm.org/citation.cfm?id=1273458

Other Tips for Reading Papers

Read at the right level for your needs

- "Work smarter, not harder"

Read at the right time of day

- When you are fresh, not sleepy

Read in the right place

- Where you are not distracted, and have enough time

Read actively

- With a purpose (what is your goal?)
- With a pen or computer to take notes

Read critically

- Think, question, challenge, critique, …

How to Write a Review

Four Sections

1. Summary
2. Paper strengths
3. Paper weaknesses
4. Detailed comments

Summary

- 1-2 points: What problem?
- 1-2 points: Core novel ideas or technical contributions
- 3-5 points: Summarize approach, mechanisms, findings

Strength/Weaknesses: 2-4 points each

Detailed comments: Longer exposition. Be constructive.

Imagine a conversation: what would you tell the authors?