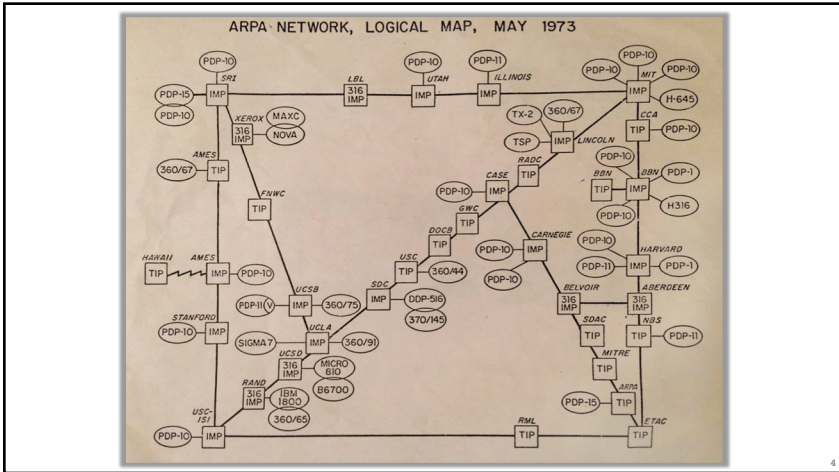


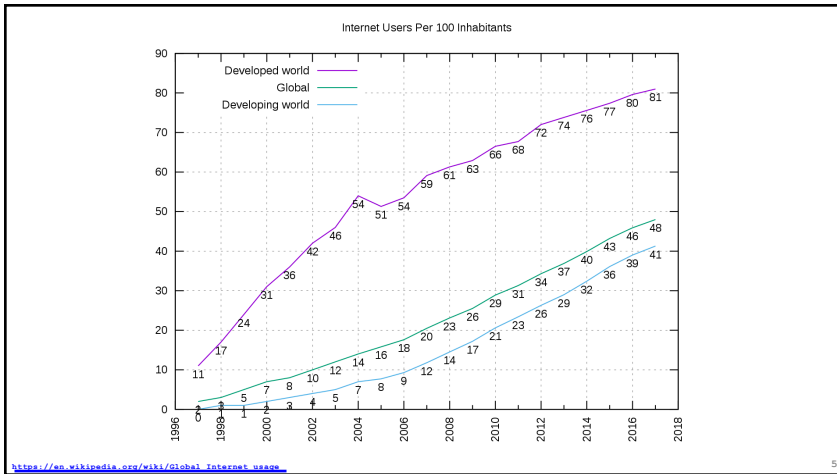
# COS 461: Computer Networks

Mike Freedman  
Spring 2020  
Lectures: MW 1:30-2:50pm in Richardson 002  
<http://www.cs.princeton.edu/courses/archive/spring20/cos461/>

The Internet is an Exciting Place

29 Oct 69	2100	LOADED OP. PROGRAM CSK FOR BEN BARTER BBV	
	22:30	Talked to SRF Host to Host	CSK
		Left op. program CSK running after sending a host lead message to imp.	



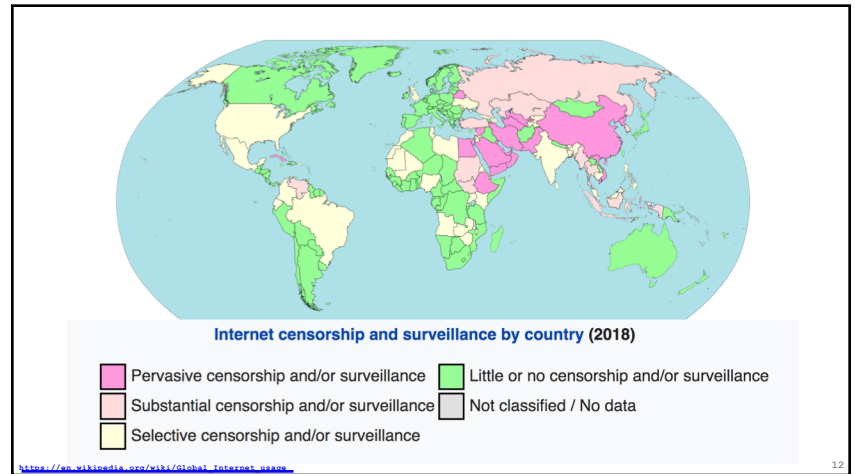
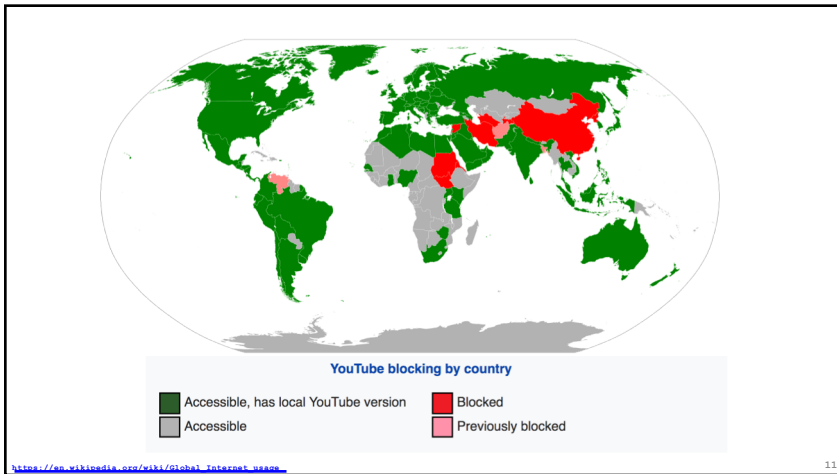


How does the design of the Internet support **growth** and foster **innovation**?

The Internet is a Tense Place

The collage features several news headlines on a blue background with binary code patterns:

- Los Angeles Times**: "\$17,000 bitcoin ransom paid by hospital to hackers sparks outrage"
- in p r**: "Malware Attacks On Hospitals Put Patients At Risk"
- Newsweek**: "RANSOMWARE WREAKING HAVOC IN AMERICAN AND CANADIAN HOSPITALS"
- CBS THIS MORNING** logo is visible in the bottom left corner.



How does the design of the Internet **create** or **exacerbate** these tensions?

13

What *is* the Internet?

14

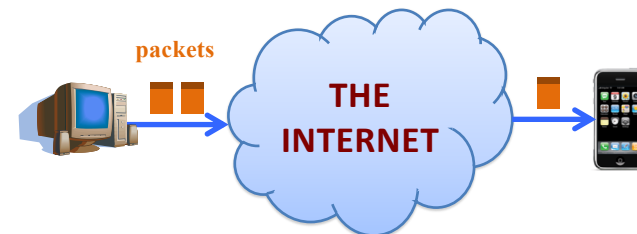
## I Can Haz Wikipedia

The Internet is the worldwide, **publicly accessible** network of interconnected computer networks that transmit data by **packet switching** using the **standard** Internet Protocol (IP).

It is a "**network of networks**" that consists of millions of smaller domestic, academic, business, and government networks, which together carry **various information and services**.

<http://en.wikipedia.org/wiki/Internet>

## "Best-Effort Packet Delivery Service"



15



## Power at the Edge

### End-to-End Principle

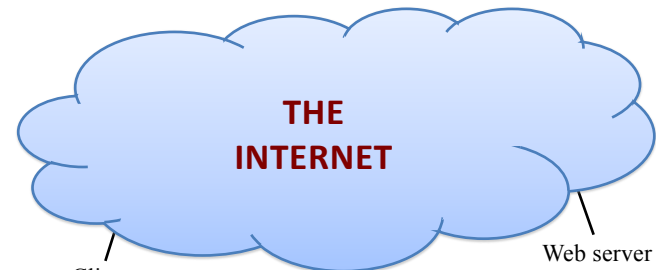
Whenever possible, communications protocol operations should be defined to occur at the **end-points** of a communications system.

### Programmability

With programmable end hosts, new network services can be added at **any time, by anyone**.

And end hosts became powerful and ubiquitous....

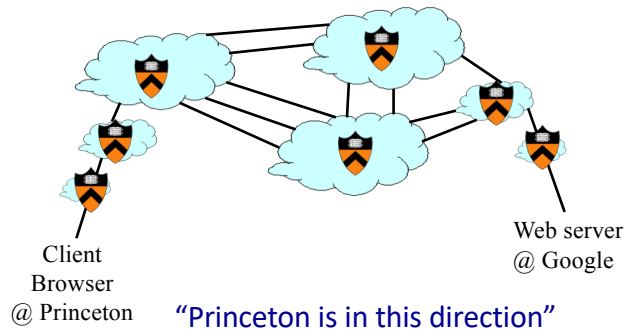
## "A Network of Networks"



- How do you name?
- How do you find a name?

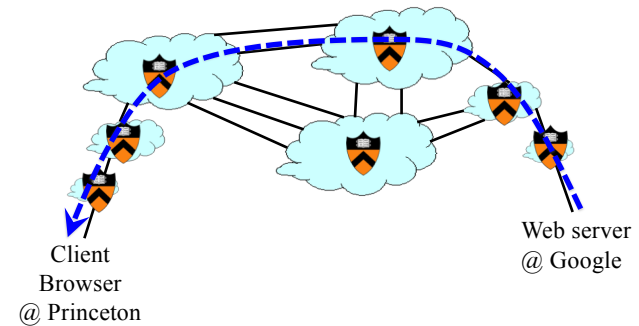
18

## Announcing a Route



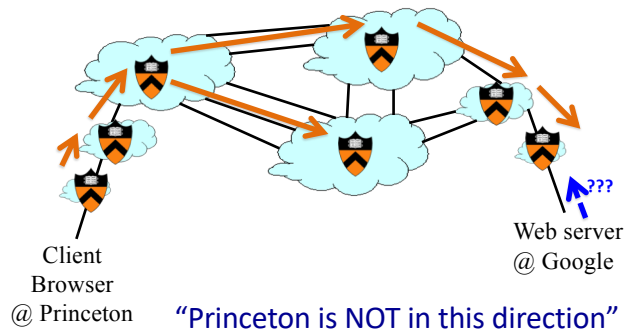
19

## Forwarding Traffic



20

## Withdrawing a traffic route



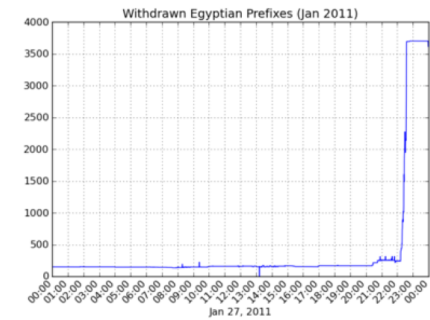
21



## Egypt Leaves the Internet

By James Cowie on January 27, 2011 7:56 PM

At 22:34 UTC (00:34am local time), Renesys observed the virtually simultaneous withdrawal of all routes to Egyptian networks in the Internet's global routing table. Approximately 3,500 individual BGP routes were withdrawn, leaving no valid paths by which the rest of the world could continue to exchange Internet traffic with Egypt's service providers. Virtually all of Egypt's Internet addresses are now unreachable, worldwide.

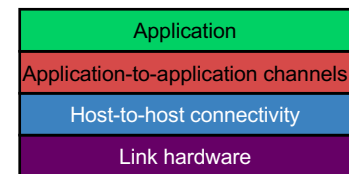


## Central concepts in networking

23

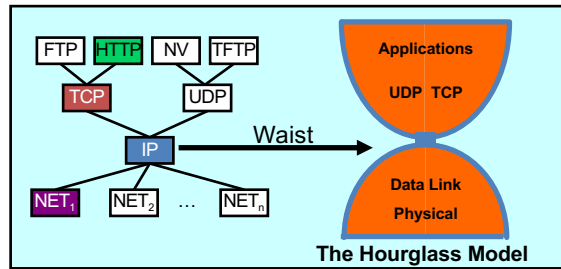
## Abstraction through Protocol Layering

- Layers partition the system
  - Each layer **solely** relies on services from layer below
  - Each layer **solely** exports services to layer above
- Interface between layers defines interaction
  - Hides implementation details
  - Layers can change without disturbing other layers



24

## The Internet Protocol Suite



The "narrow waist" facilitates interoperability

25

## Application: HyperText Transfer Protocol

```
GET /courses/archive/spr20/cos461/ HTTP/1.1
Host: www.cs.princeton.edu
User-Agent: Mozilla/4.03
CRLF
```

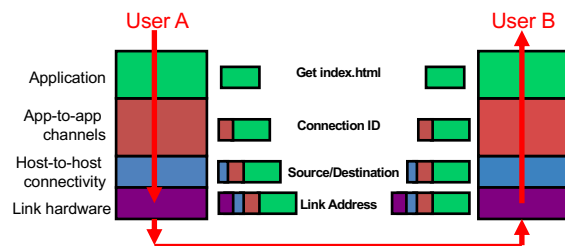
Request

```
HTTP/1.1 200 OK
Date: Mon, 4 Feb 2013 11:09:03 GMT
Server: Netscape-Enterprise/3.5.1
Last-Modified: Mon, 2 Feb 2013 19:12:23 GMT
Content-Length: 21
CRLF
Site under construction
```

Response

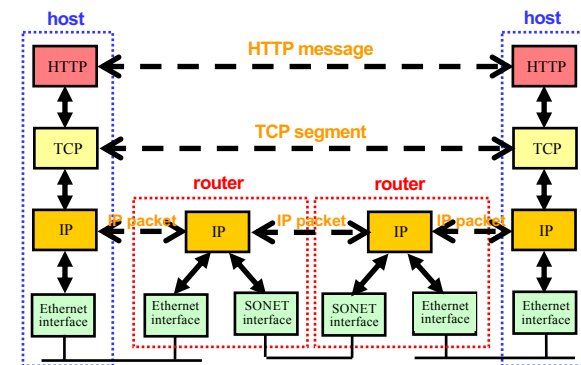
26

## Layer Encapsulation in HTTP



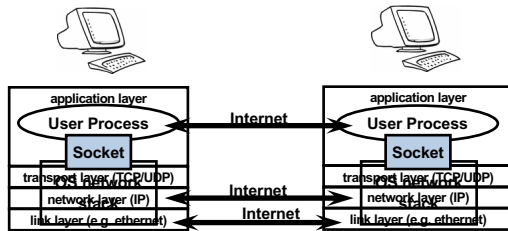
27

## End Hosts vs. Routers



28

## Socket and Process Communication

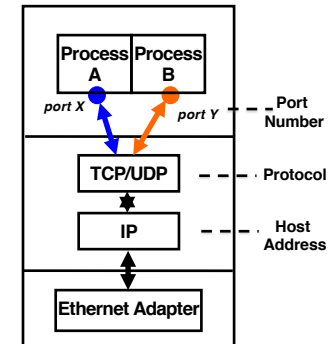


The interface that the OS provides to its networking subsystem

29

## Socket and Process Communication

- **Receiving host**
  - Destination **address** that uniquely identifies host
  - **IP address**: 32-bit quantity ("1.2.3.4")
- **Receiving socket**
  - Host may be running many different processes
  - Destination **port** that uniquely identifies socket
  - **Port number**: 16-bits ("80")



30

## Key Concepts in Networking

- **Naming**
  - What to call computers, services, protocols, ...
- **Layering**
  - Abstraction is the key to managing complexity
- **Protocols**
  - Speaking the same language
  - Syntax and semantics
- **Resource allocation**
  - Dividing scarce resources among competing parties
  - Memory, link bandwidth, wireless spectrum, paths

31

## Course Summary

32

## What You Learn in This Course

- **Knowledge:** how the Internet works, and why
  - Protocol stack: link, network, transport, application
  - Resource allocation: congestion control, routing
  - Applications: Web, P2P, ...
  - Networks: enterprise, cloud, backbone, wireless, ...
- **Insight:** key concepts in networking
  - Naming, layering, protocols, resource allocation, ...
- **Skill:** network programming
  - Many nodes are general-purpose computers
  - Can innovate and develop new uses of networks

33

## meClickers: Quick Surveys

Growth/innovation vs. create/exacerbate tensions

- Does Internet design prevent misuse?
  1. Individual endpoints can only use addresses given to them when connect to the network
  2. Individual end-points can “spoof” any IP address

34

## meClickers: Quick Surveys

Growth/innovation vs. create/exacerbate tensions

- Does Internet design prevent misuse?

Networks are assigned unique IP address blocks from a central authority (“IANA”): Princeton has 128.112.\*

  1. Network can only announce assigned addresses
  2. Networks can spoof any address

35

## meClickers: Quick Surveys

Growth/innovation vs. create/exacerbate tensions

- Does “Internet” provide reliable packet delivery?
  1. Yes, that’s necessary for protocols like HTTP that require in-order streams
  2. No, packets may be arbitrary dropped or reordered

36



## Course Organization

37

## Learning the Material: People

- **Instructor: Prof. Mike Freedman**
  - Slides available online at course Web site
- **TAs: David Liu, Ashwini Raina, Srikar Kasi**
- **Main Q&A forum: [www.piazza.com](http://www.piazza.com)**
  - No anonymous posts or questions, can DM instructors
  - Setting expectation: TAs will monitor/respond to Piazza 1-2 times per day in a burst of activity

38

## Learning the Material: Lectures! (primary)

- **Lectures: MW 1:30 – 2:50, Robertson 002**
- **Attend lectures and take notes!**
  - Lecture slides posted day/night before
  - Recommendation: Print slides & take notes
  - Not everything covered in class is on slides
  - You are responsible for everything covered in class

39

## Learning the Material: Books (secondary)

- **Main textbook**
  - *Computer Networks: A Systems Approach*, by Peterson and Davie
  - Also online: <https://book.systemsapproach.org/>
- **Additional books (may be of interest)**
  - Networking textbooks
    - *Computer Networking: A Top-Down Approach Featuring the Internet*, by Kurose and Ross
    - *Computer Networks*, by Tanenbaum
  - Network programming references
    - *TCP/IP Illustrated, Volume 1: The Protocols*, by Stevens
    - *Unix Network Programming, Vol 1: Sockets Networking API*, by Stevens, Fenner, & Rudolf

40

## Grading

- **Six assignments (9% each)**
  - 90% 24 hours late, 80% 2 days late, 50% >5 days late
  - **Three** free late days (we'll figure which one is best)
  - Only failing grades I've given are for students who don't / try to do assignments
- **Two exams (46% total)**
  - Midterm exam before spring break (20%)
  - Final exam during exam period (26%)

41

## Policy: Write Your Own Code

Programming is an individual creative process. At first, discussions with friends is fine. When writing code, unless stated otherwise, the program must be your own work.

Do not copy another person's programs, comments, or any part of submitted assignment. This includes character-by-character transliteration but also derivative works. Cannot use another's code, etc. even while "citing" them.

Writing code for use by another or using another's code is academic fraud in context of coursework.

Do not publish your code e.g., on github, during/after course!

42

## Policy: Write Your Own Code

Programming is an individual creative process. At first, discussions with friends is fine. When writing code, unless stated otherwise, the program must be your own work.

Do not copy another person's programs, comments, or any part of submitted assignment. This includes character-by-character transliteration but also derivative works. Cannot use another's code, etc. even while "citing" them.

Writing code for use by another or using another's code is academic fraud in context of coursework.

Do not publish your code e.g., on github, during/after course!

**Don't Plagiarize!**

43

## Setting Expectations: Don't expect 24x7 answers



- **Try to figure out yourself**
- **Piazza not designed for debugging**
  - Utilize right venue: Go to TA office hours
  - Send detailed Q's / bug reports, not "no idea what's wrong"
- **Instructors are not on pager duty 24 x 7**
  - Don't expect response before next business day
  - Questions Friday night @ 11pm should not expect fast responses. Be happy with something before Monday.
- **Implications**
  - Students should answer each other; start your assignments early!

44

## Assignment 1: Socket Programming

45

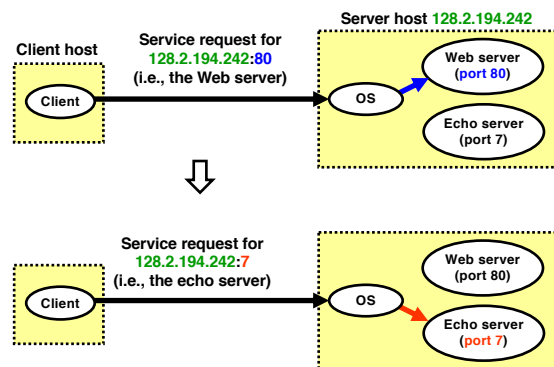
## Client-Server Communication

- Client "sometimes on"
  - Initiates a request to the server when interested
  - E.g., Web browser on your laptop or cell phone
  - Doesn't communicate directly with other clients
  - Needs to know server's address
- Server is "always on"
  - Handles services requests from many client hosts
  - E.g., Web server for the [www.cnn.com](http://www.cnn.com) Web site
  - Doesn't initiate contact with the clients
  - Needs fixed, known address



46

## Using IP + Ports to Identify Services



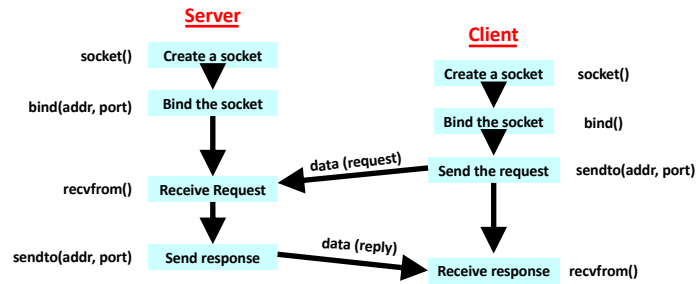
47

## Two Types of Application Processes Communication

- Datagram Socket (UDP)
  - Collection of messages
  - Best effort
  - Connectionless
- Stream Socket (TCP)
  - Stream of bytes
  - Reliable
  - Connection-oriented

48

## Client-Server Communication Datagram Sockets (UDP): Connectionless



49

## Client-Server Communication Stream Sockets (TCP): Connection-oriented

