# COS 461/561: Computer Networks
# Lecture 1: Class Introduction

Kyle Jamieson (461); Ravi Netravali (561)

Fall 2021

Class Meeting: T 10:00-10:50 AM

www.cs.princeton.edu/courses/archive/fall21/cos461

# Today

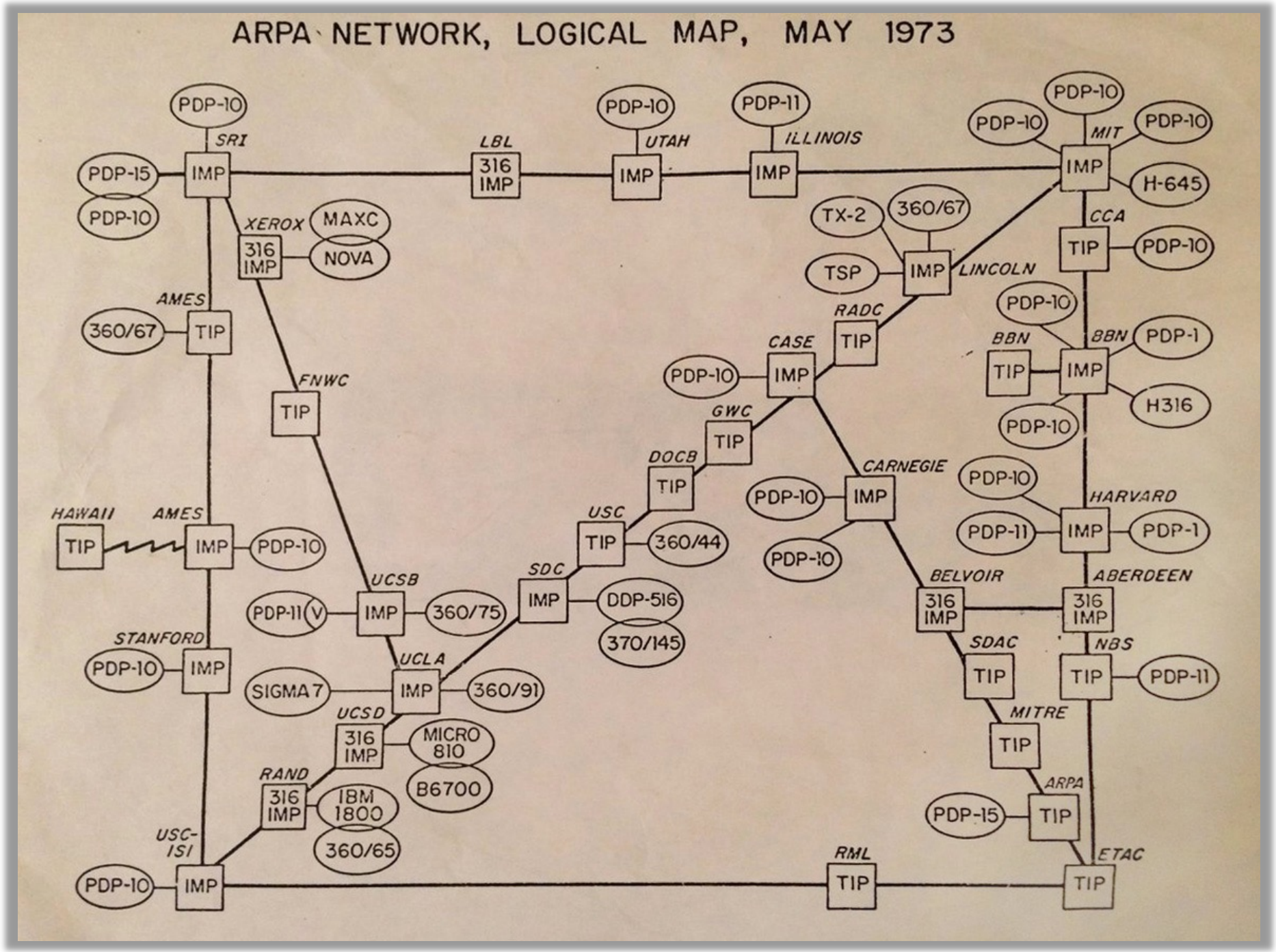
1. Origins of the Internet

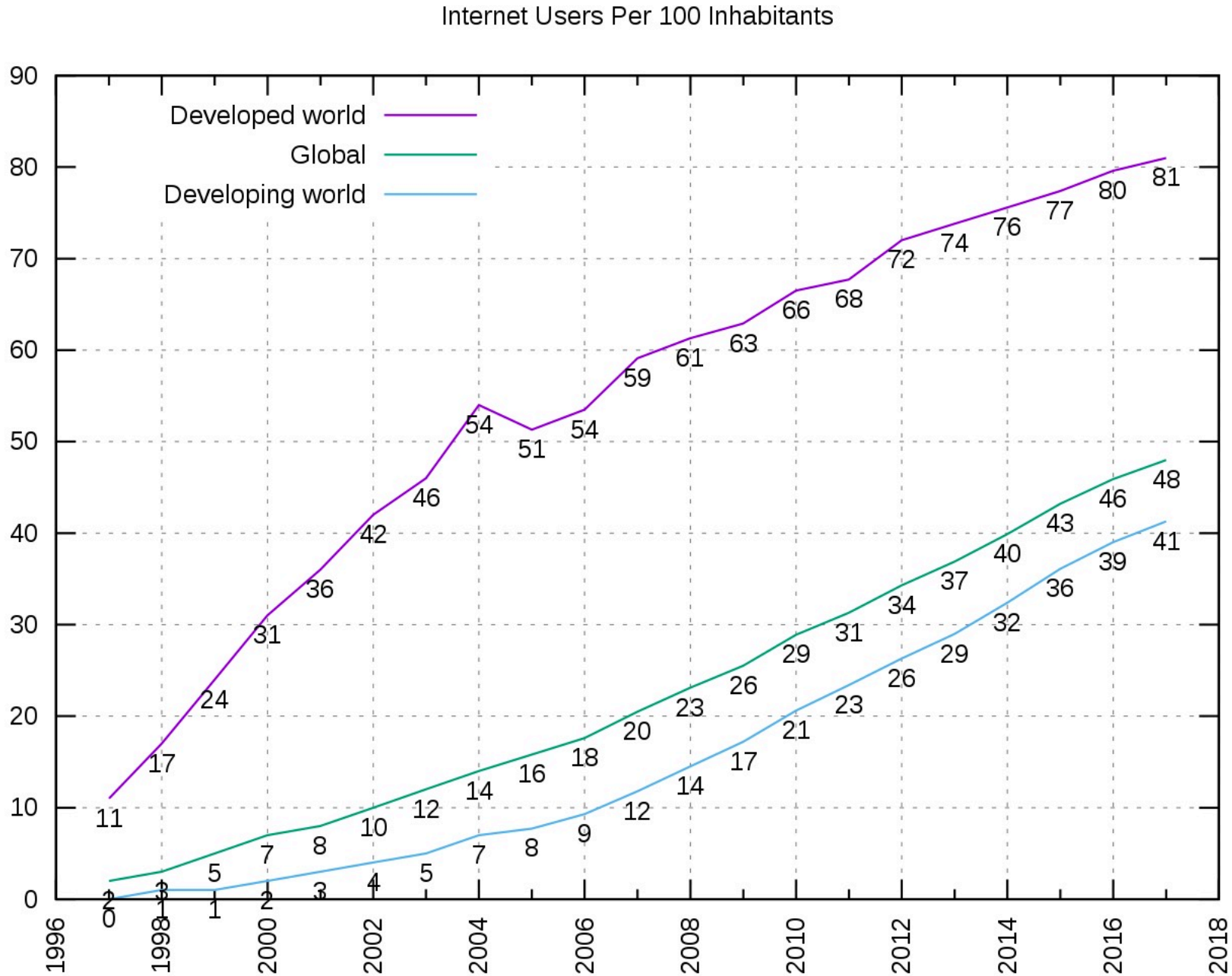
2. Course Introduction and Policies (461 & 561)

# What are the Origins of Today's Internet?

# Interface Message Processor

| | | | |
|---|---|---|---|
| 29OCT69 | 2100 | LOADED OP. PROGRAM<br>EOI2 BEN BARKER<br>BBN | CSK |
| | 22:30 | talked to SRI<br>Host to Host | CSC |
| | | Left op. imp. program<br>running after sending<br>a host dead message<br>to imp. | CSC |


ARPA NETWORK, LOGICAL MAP, MAY 1973
SRI
PDP-10
PDP-15
PDP-10
IMP
LBL
316 IMP
UTAH
PDP-10
IMP
ILLINOIS
PDP-11
IMP
MIT
PDP-10
PDP-10
PDP-10
IMP
H-645
XEROX
316 IMP
MAXC
NOVA
TX-2
360/67
TSP
IMP
LINCOLN
CCA
TIP
PDP-10
AMES
360/67
TIP
FNWC
TIP
RADC
TIP
CASE
IMP
PDP-10
BBN
TIP
BBN
IMP
PDP-1
H316
PDP-10
PDP-10
GWC
TIP
DOCB
TIP
CARNEGIE
PDP-10
IMP
PDP-10
HARVARD
PDP-10
PDP-11
IMP
PDP-1
HAWAII
TIP
AMES
IMP
PDP-10
USC
TIP
360/44
SDC
IMP
DDP-516
370/145
UCSB
PDP-11 V
IMP
360/75
BELVOIR
316 IMP
ABERDEEN
316 IMP
STANFORD
PDP-10
IMP
UCLA
SIGMA 7
IMP
360/91
SDAC
TIP
NBS
TIP
PDP-11
UCSD
316 IMP
MICRO 810
B6700
MITRE
TIP
RAND
316 IMP
IBM 1800
360/65
ARPA
PDP-15
TIP
USC-ISI
PDP-10
IMP
RML
TIP
ETAC
TIP




https://en.wikipedia.org/wiki/Global_Internet_usage

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

# The Internet is a Tense Place

Cybersecurity

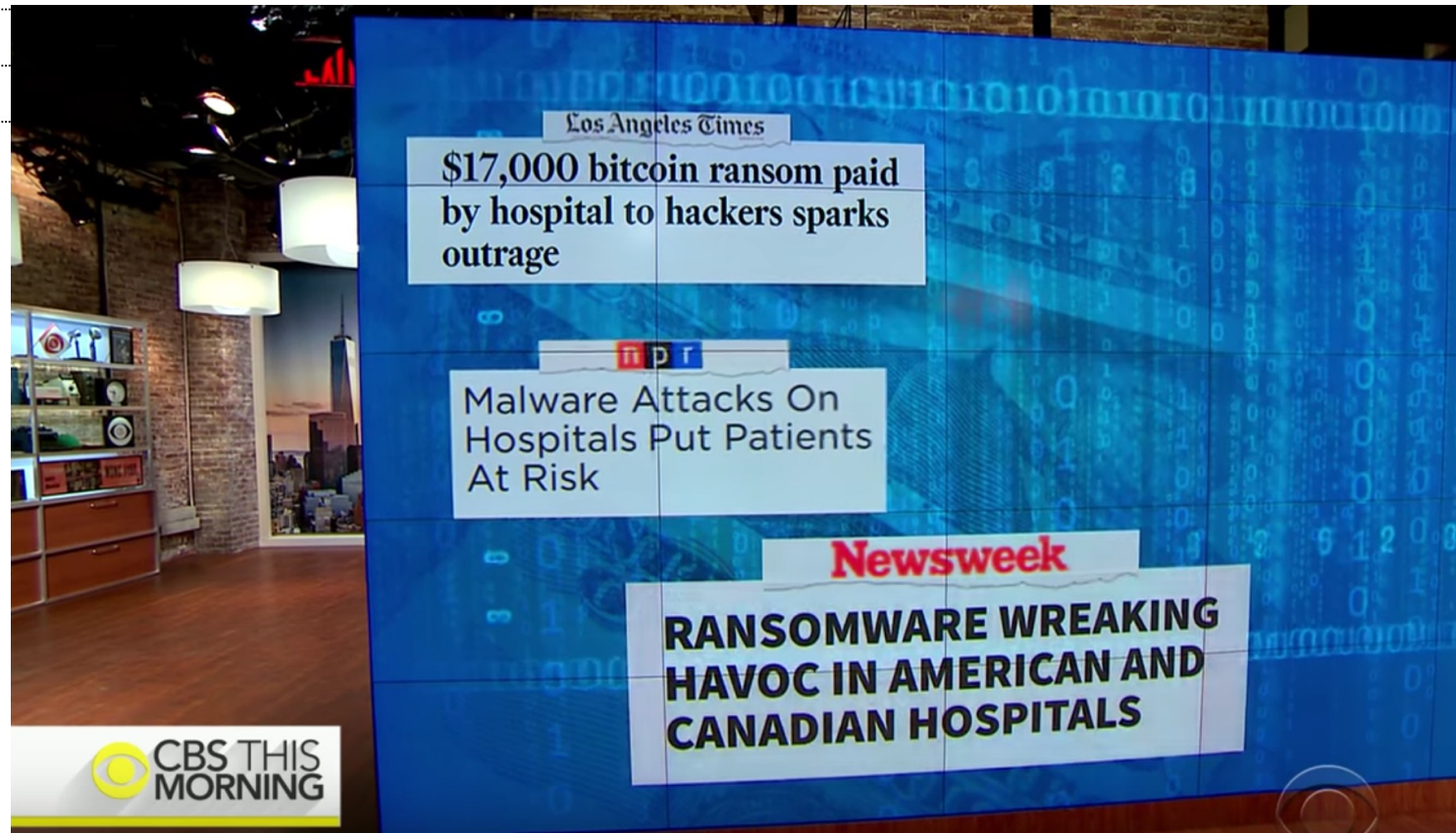
# Cyber-Attack Hits U.S. Health Agency Amid Covid-19 Outbreak

By Shira Stein and Jennifer Jacobs

March 16, 2020, 8:37 AM EDT *Updated on March 16, 2020, 4:35 PM EDT*

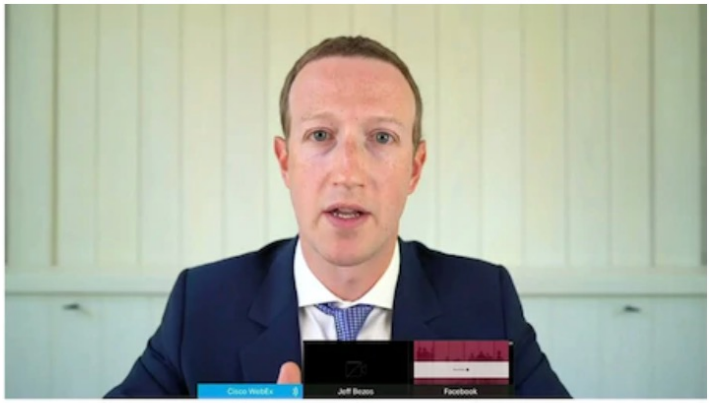
- NSC tweet on disinformation Sunday was connected to attack
- Cyber intrusion comes as U.S. battles the coronavirus pandemic

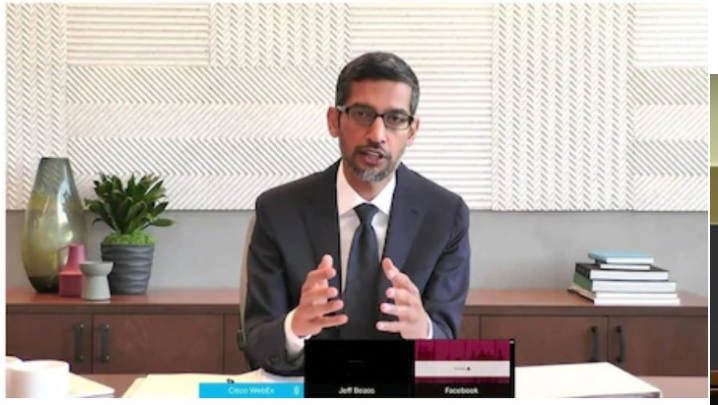
**2018:**



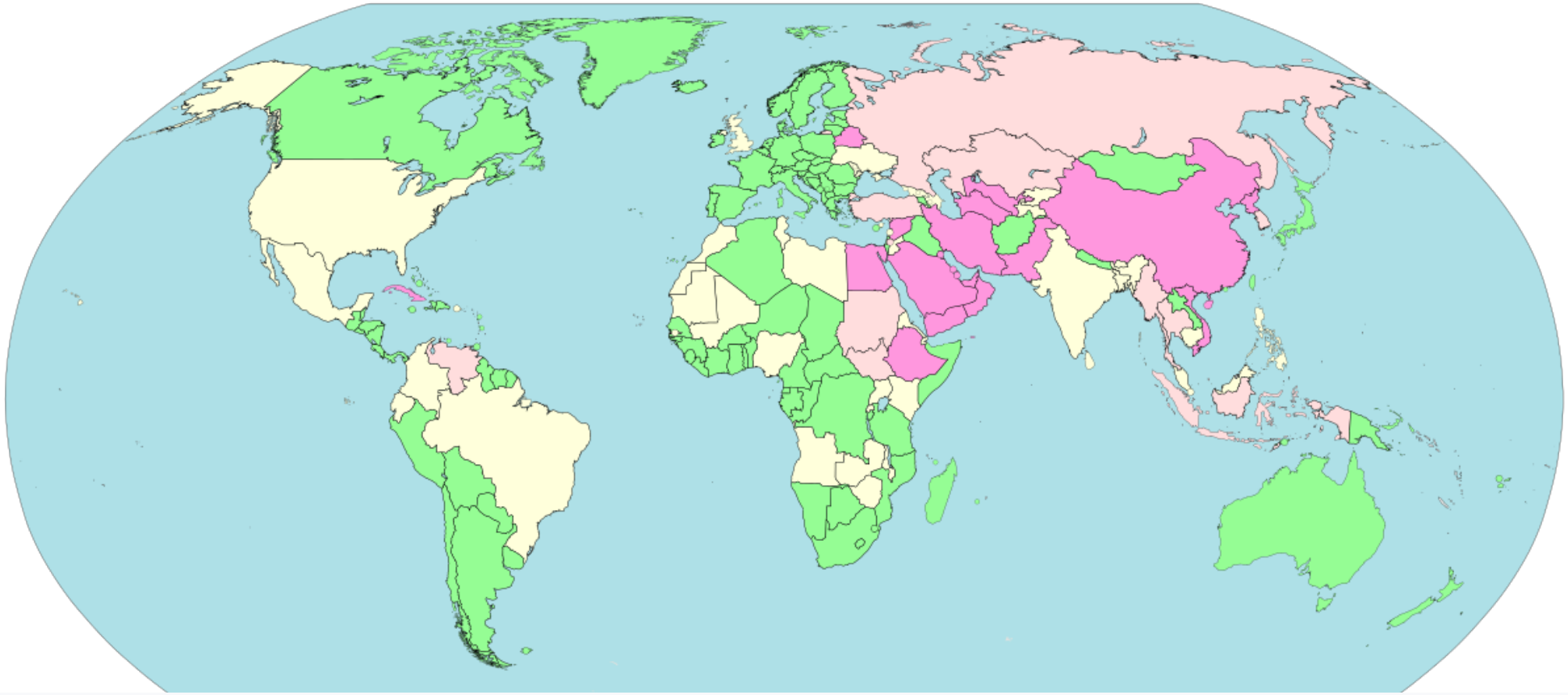

NET
NEUTRALITY

## July 2020





## October 2019



**Internet censorship and surveillance by country (2018)**

- Pervasive censorship and/or surveillance
- Substantial censorship and/or surveillance
- Selective censorship and/or surveillance
- Little or no censorship and/or surveillance
- Not classified / No data

https://en.wikipedia.org/wiki/Global_Internet_usage

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

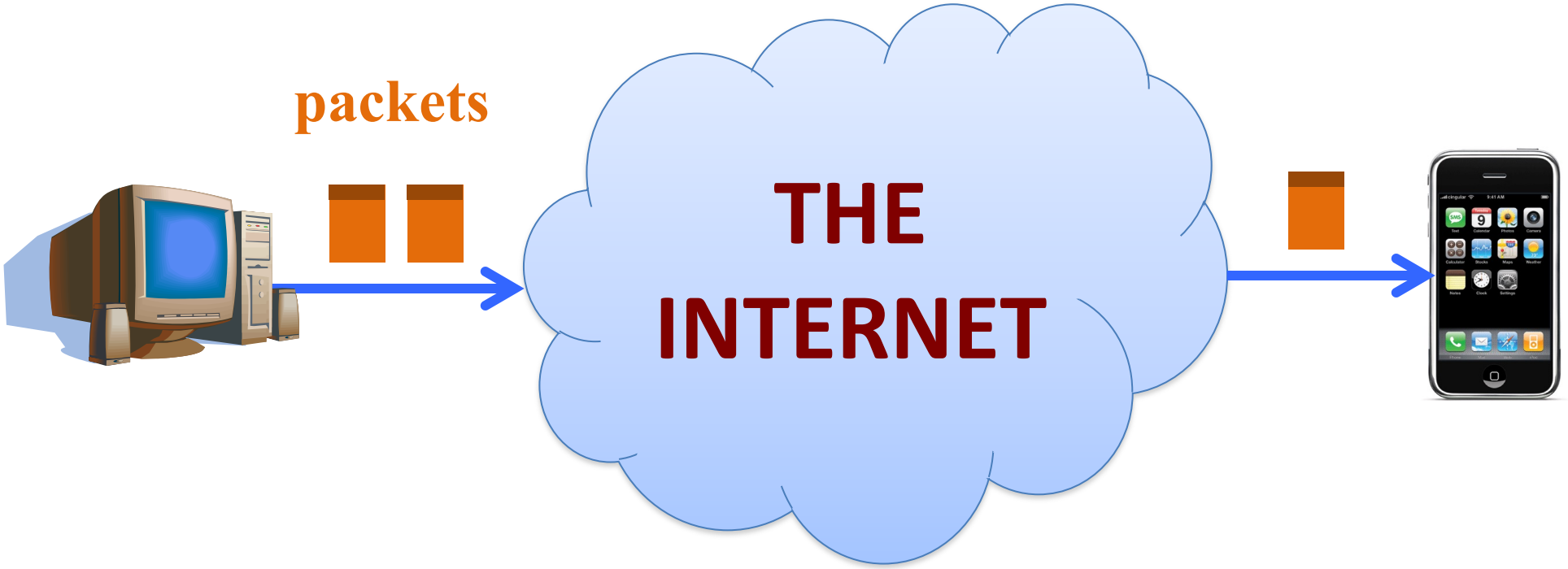
# What *is* the Internet?

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

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.

# “Best-Effort Packet Delivery Service”



# Power at the Edge

End-to-End Principle

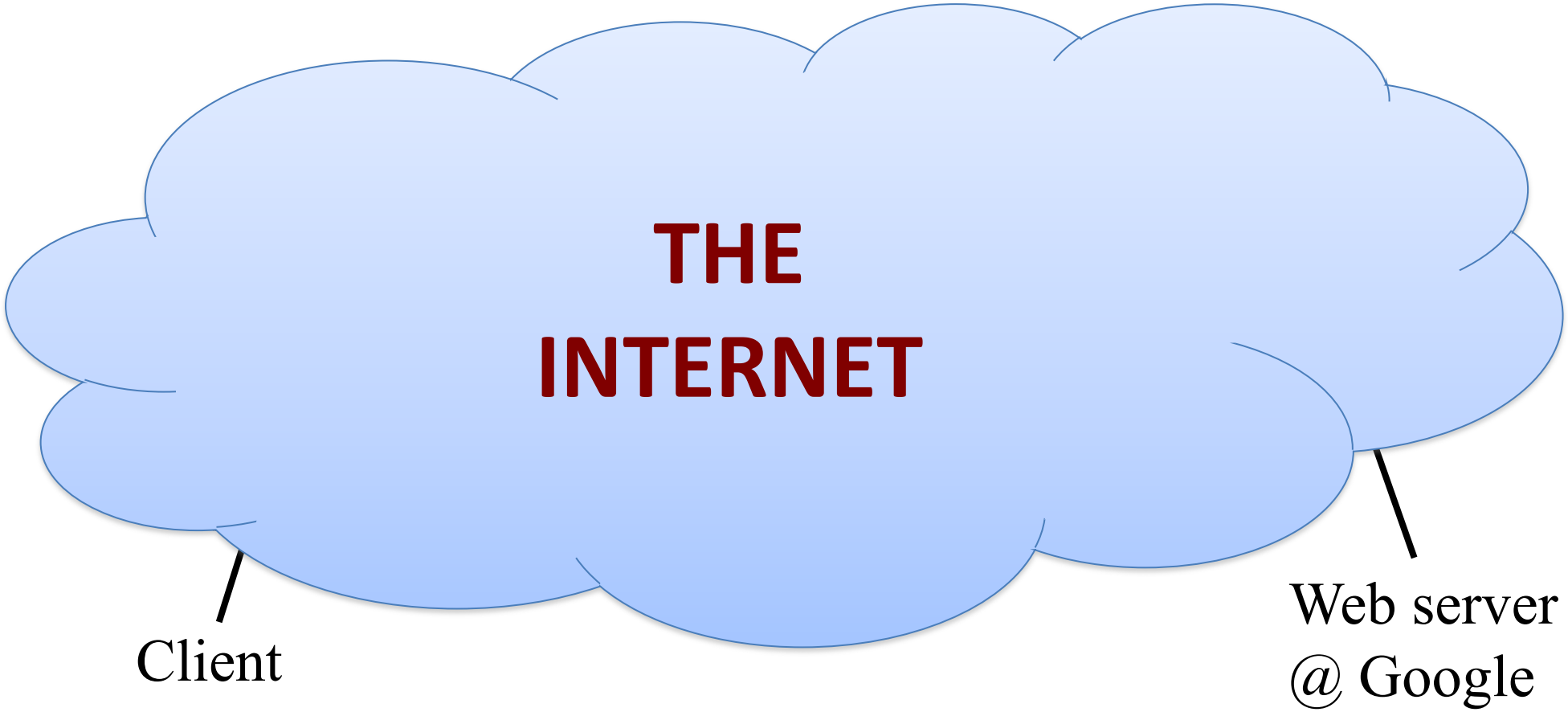
Whenever possible, communications protocol operations should be defined to occur at the end-points of a communications system.

Programmability

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

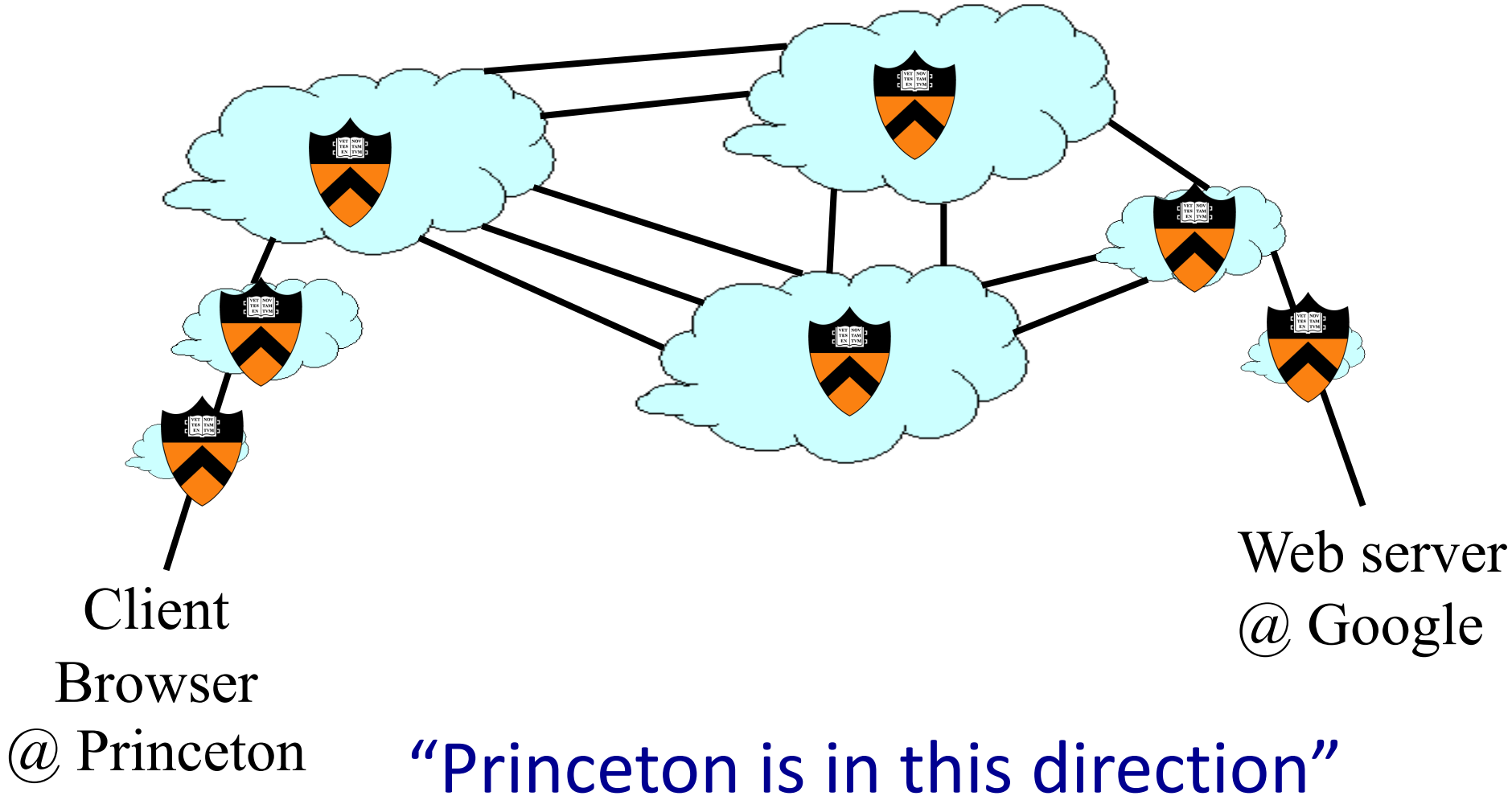
And end hosts became powerful and ubiquitous….

# “A Network of Networks”



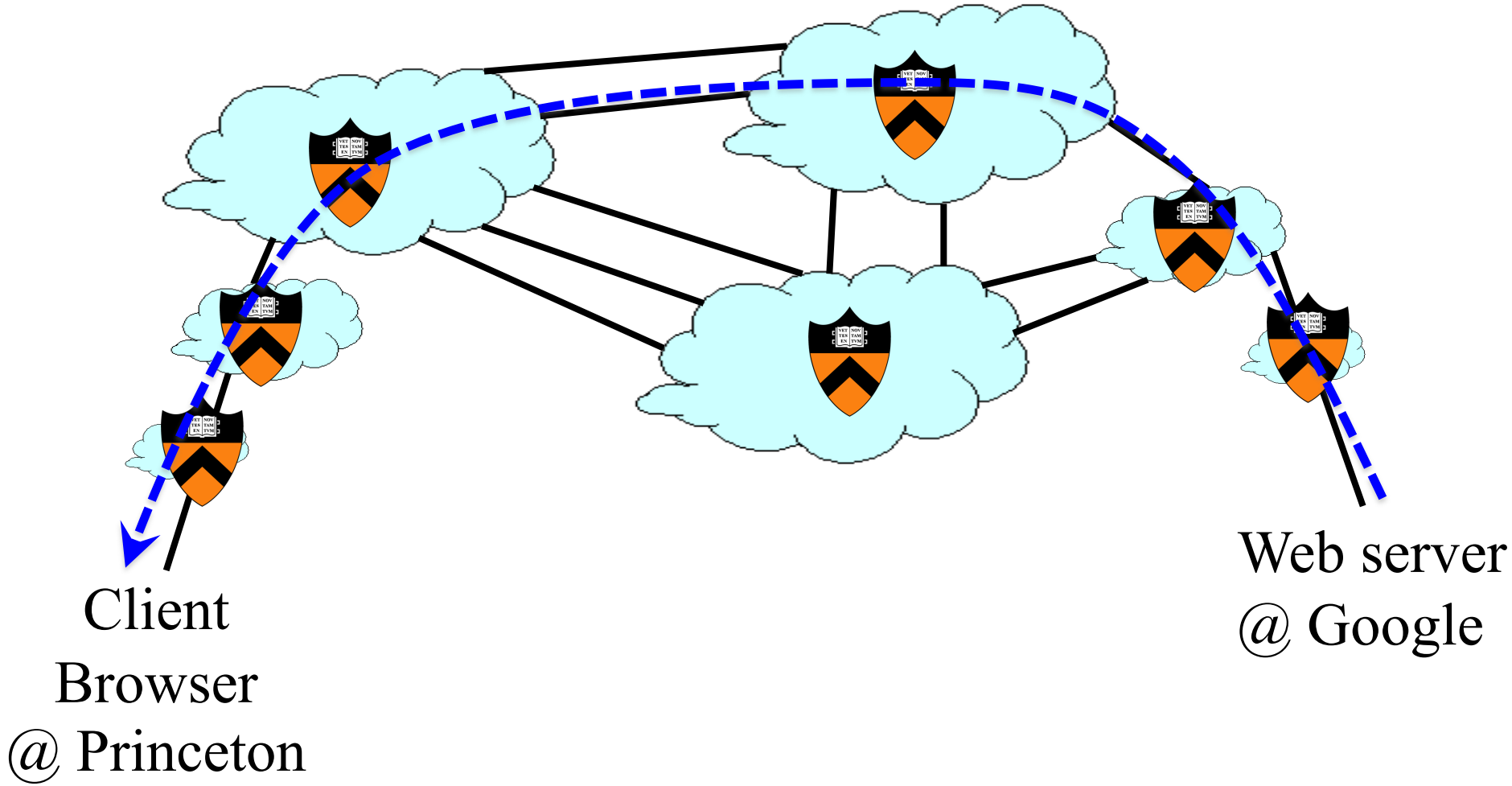

- How do you name?
- How do you find a name?

# Announcing a Route

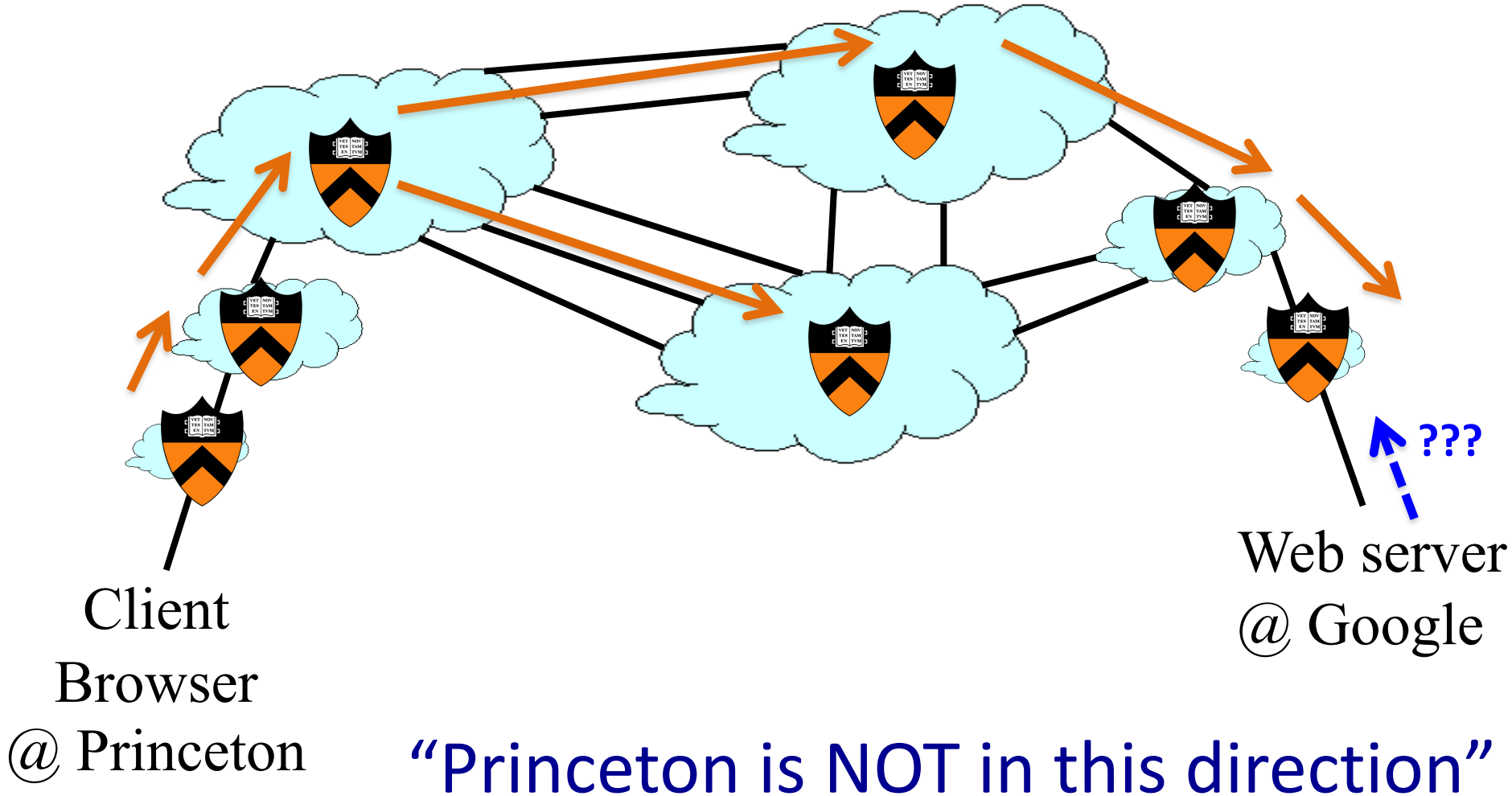



"Princeton is in this direction"

# Forwarding Traffic



# Withdrawing a traffic route



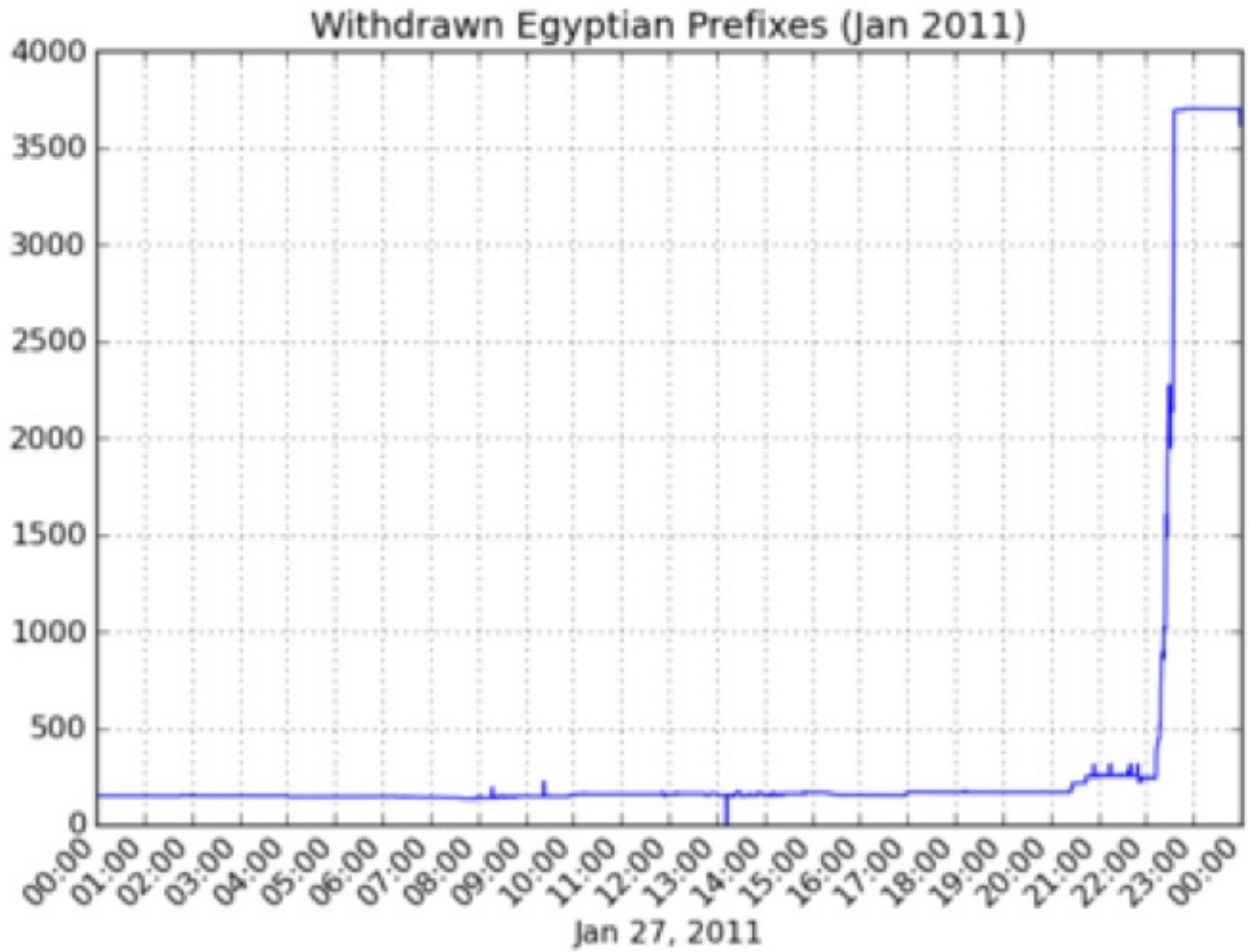

"Princeton is NOT in this direction"

renesys | blog

# Egypt Leaves the Internet

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

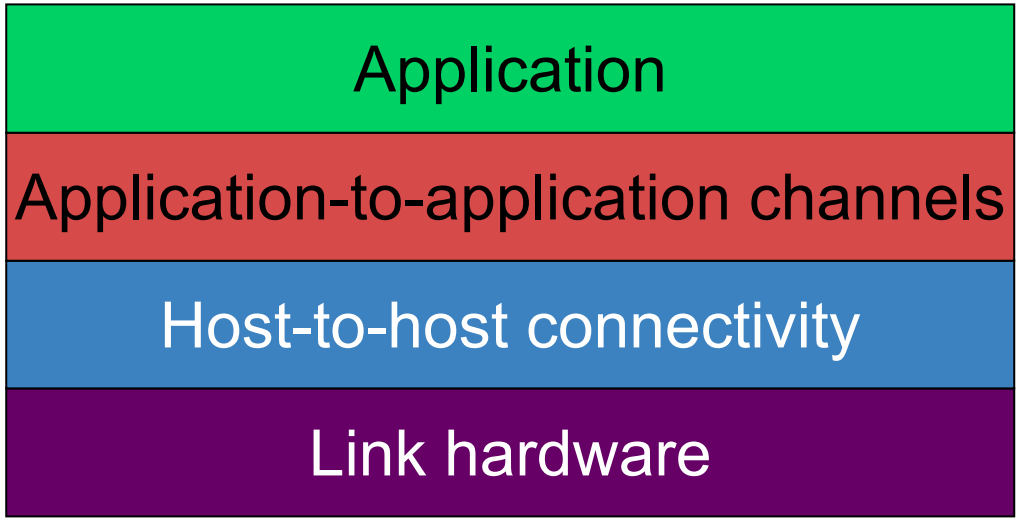
At 22:34 UTC (00:34am local time), Renesys observed the virtually simultaneous withdrawal of all routes to Egyptian networks in the Internet's global routing table. Approximately 3,500 individual BGP routes were withdrawn, leaving no valid paths by which the rest of the world could continue to exchange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



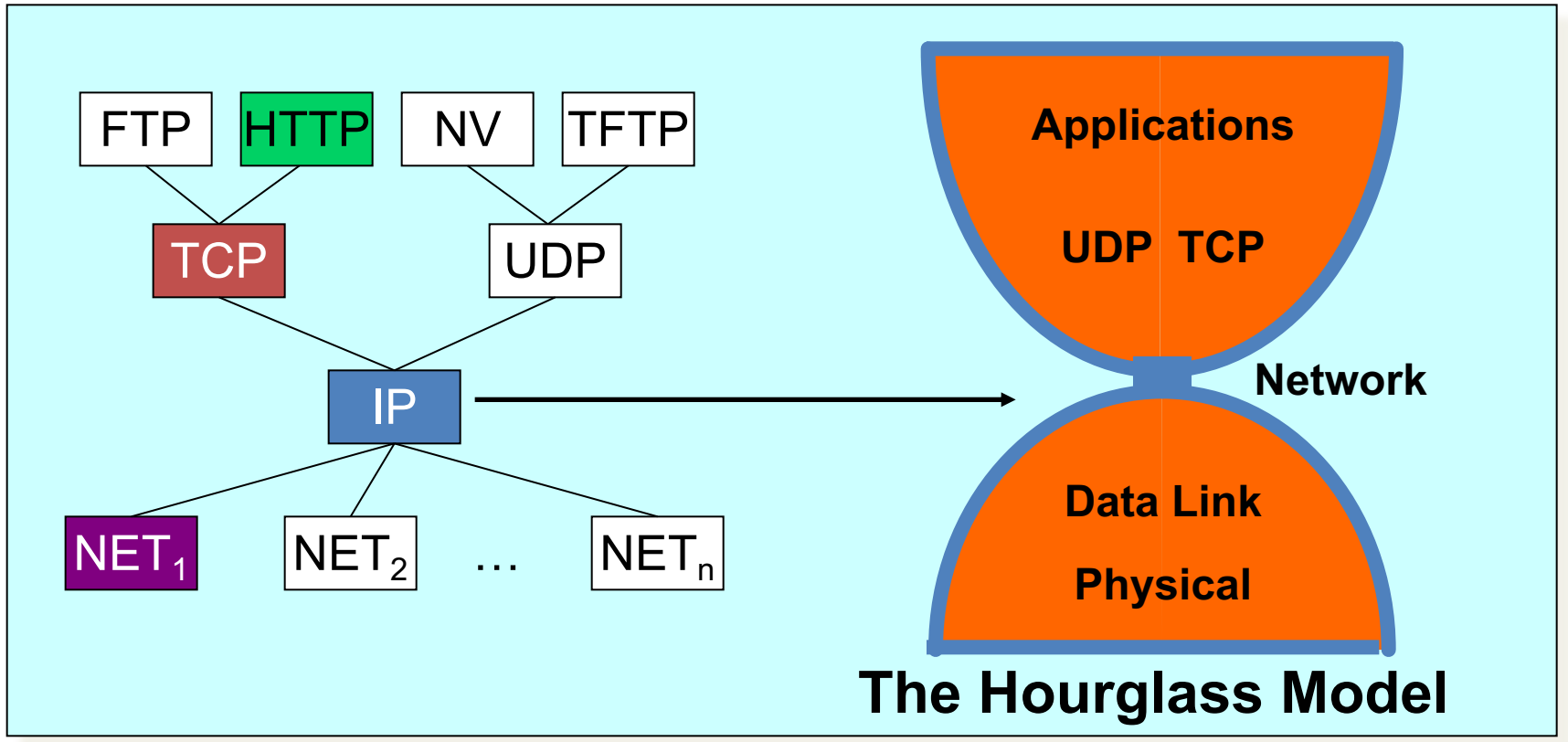
# Central concepts in networking

# 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



# The Internet Protocol Suite




The thin Network layer facilitates **interoperability**

# Application: HyperText Transfer Protocol

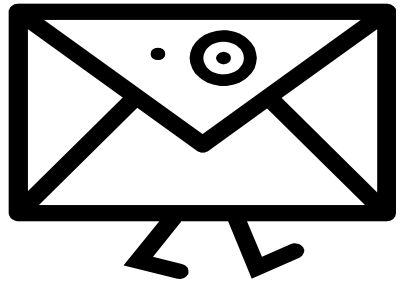
```
GET /courses/archive/spr20/cos461/ HTTP/1.1
Host: www.cs.princeton.edu
User-Agent: Mozilla/4.03
CRLF
```

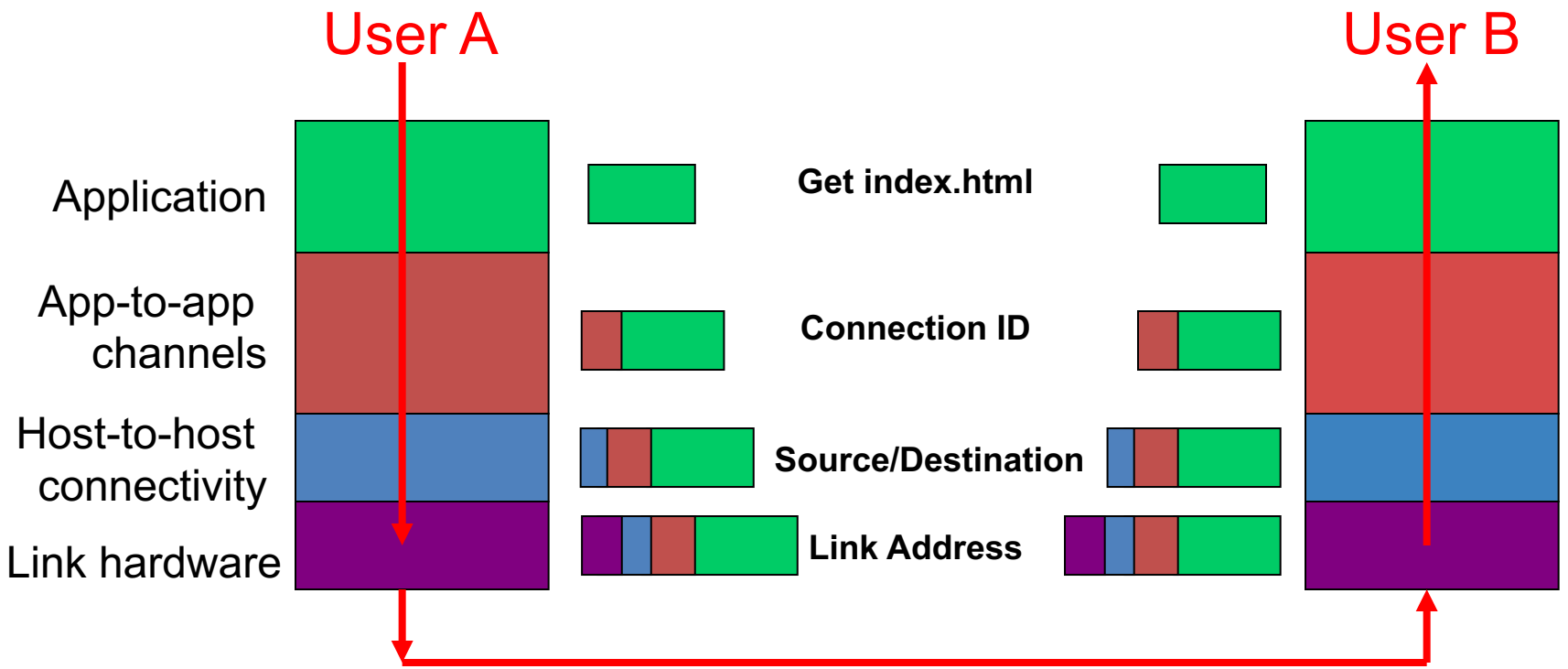
Request

Response

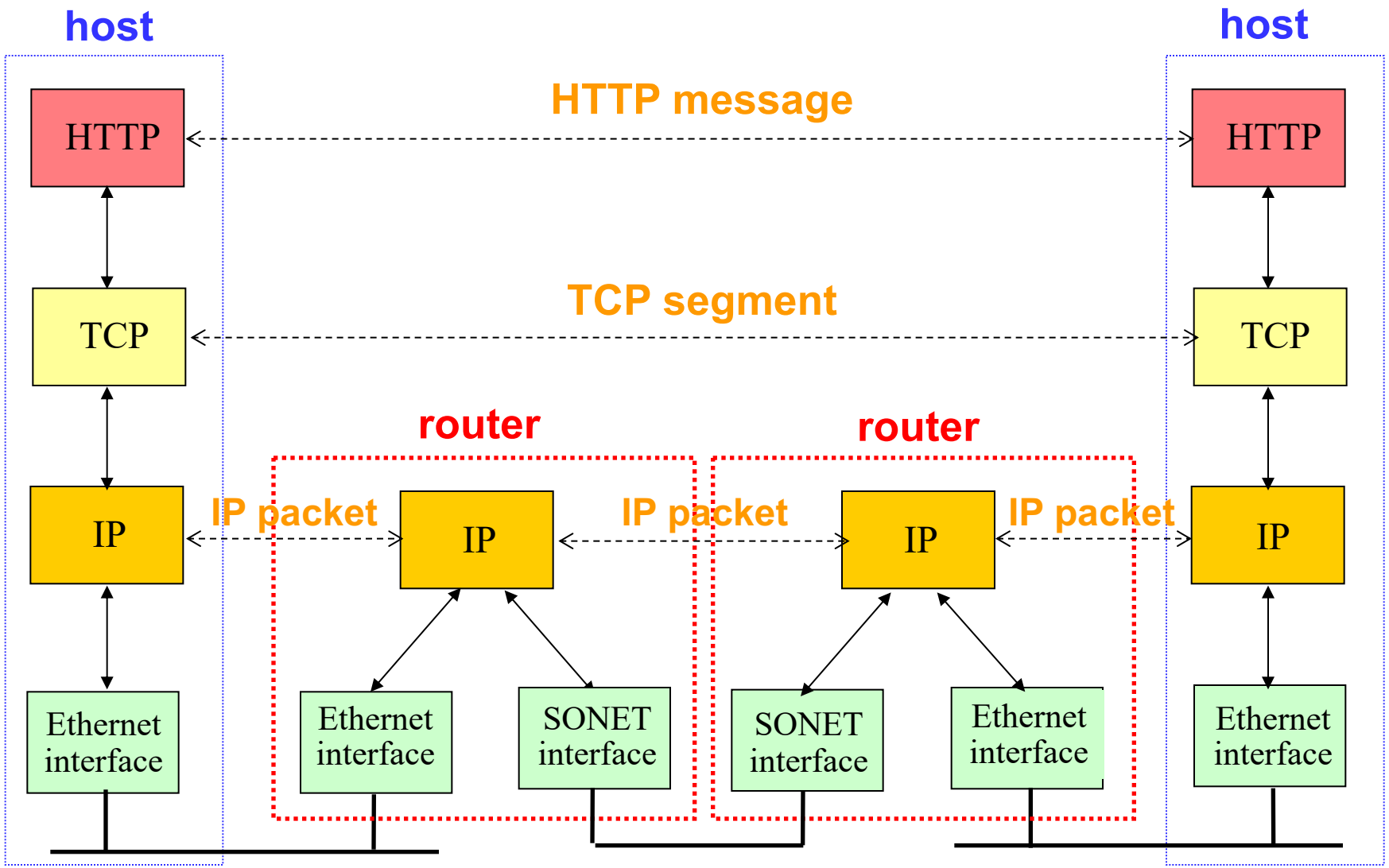
```
HTTP/1.1 200 OK
Date: Mon, 4 Feb 2013 11:09:03 GMT
Server: Netscape-Enterprise/3.5.1
Last-Modified: Mon, 2 Feb  2013 19:12:23 GMT
Content-Length: 21
CRLF
Site under construction
```

# Layer Encapsulation in HTTP

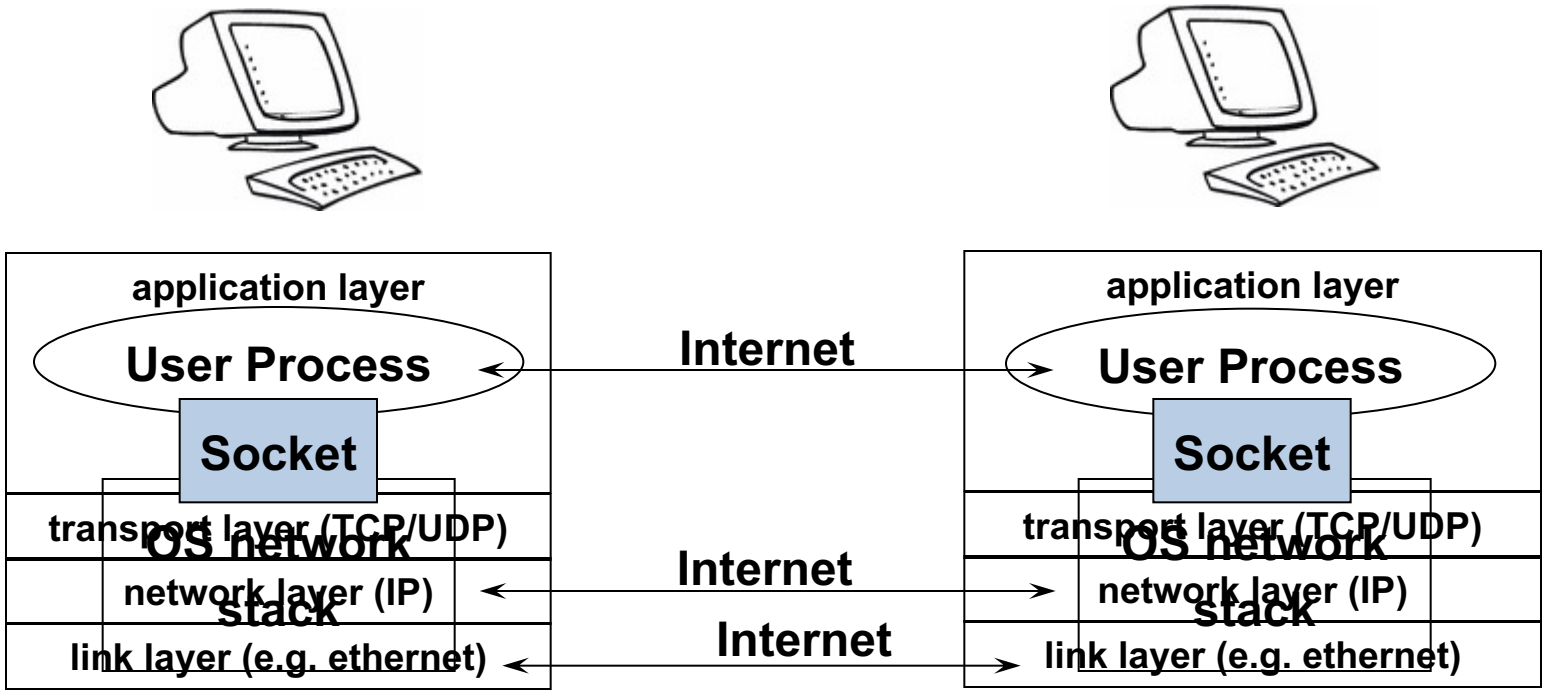




# End Hosts vs. Routers



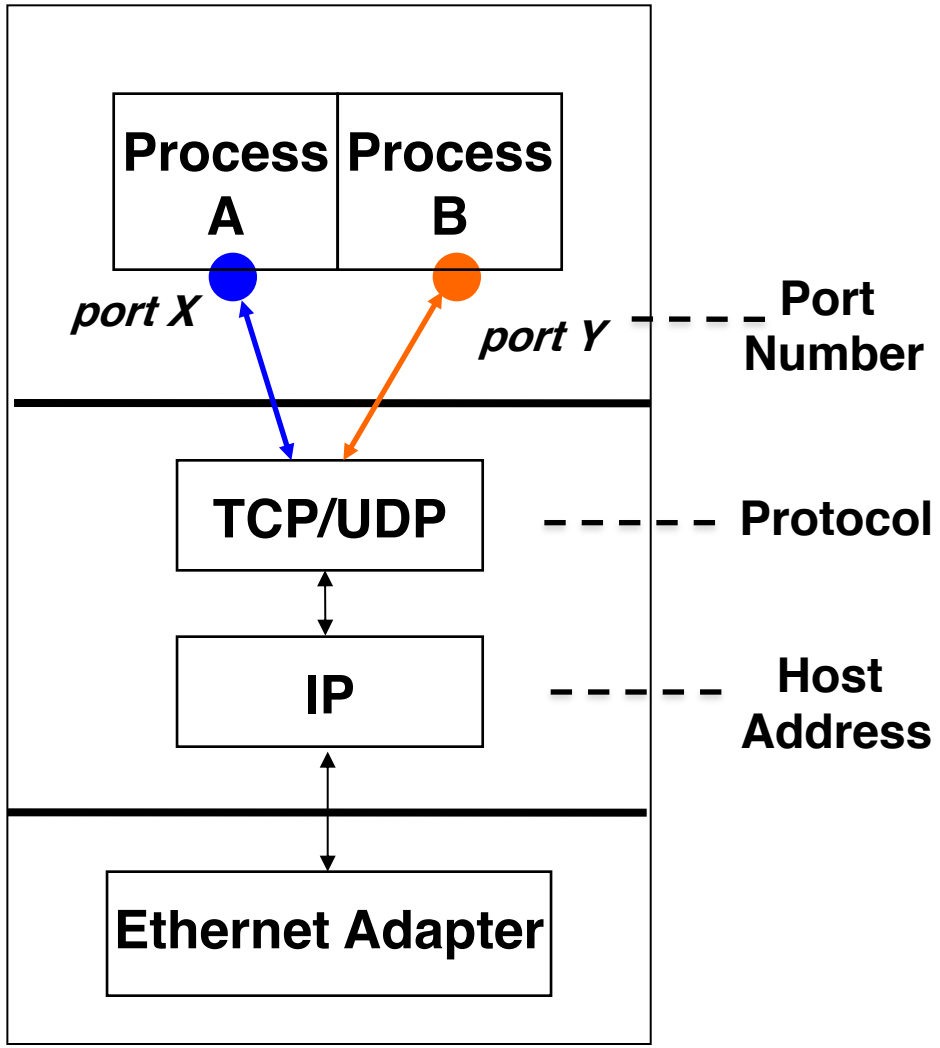
# Socket and Process Communication




**The interface that the OS provides to its networking subsystem**

# 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")



# 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 scare resources among competing parties
  - Memory, link bandwidth, wireless spectrum, paths

# Quick Surveys

Topic: 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

# Quick Surveys

Topic: Growth/innovation vs. create/exacerbate tensions

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

- Which of the following is true:

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

# Quick Surveys

Topic: Growth/innovation vs. create/exacerbate tensions

- Does the *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

# 461, 561: Class Summary

# 461: 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

# 561: What You Learn in This Course

- **Knowledge:** how the Internet works, and why
- **Insight:** key concepts and state of the art in networking
  - Naming, layering, protocols, resource allocation, …
  - Discuss classic & state of the art networking research papers, in depth. Tied to lecture topics in 461
- ~~**Skill: network programming**~~
- **Skill:** network research
  - Semester systems-building/research project, in groups
  - Reproduce a result (more common), or build a novel project

# Course Organization: 461, 561

# Learning the Material: 461 People

- Class Meeting: Prof Kyle Jamieson
  - Tue 10:00 – 10:50 AM
  - Slides and lecture videos available online at course Web site
- Precepts: TAs Nanqinqin Li, Olga Solodova
- Main Q&A forum:  Canvas
  - Setting expectation:  TAs will monitor/respond to questions 1-2 times per day in a burst of activity

# Learning the Material: 561 People

- Precepts:  Prof Ravi Netravali
  - Friday: 10:00 – 10:50 AM or 11:00-11:50 AM
  - Room: CS 105
- Precept TA: Neil Agarwal
- Main Q&A forum:  Perusall + Email

# Learning the Material: 461 & 561 Class Meetings

- Class Meetings: TR 10:00 – 10:50
- **461** attend class meeting, view lectures, participate in Q&A
  - Recommendation: print slides and take notes
  - Not everything covered in class is on slides
  - You are responsible for everything covered in class
- **561** is responsible for all 461 lecture material, but need not attend 461 class meeting or Q&A

# Learning the Material: Precepts

- 461 precepts focus on programming assignments
  - Led by TAs

- 561 precepts discuss papers in depth
  - Discuss 1 research paper in depth each week; 5 *insightful* comments due on Perusall the evening before each precept (i.e., Thursday)
  - Topic will relate to that week's 461 lectures, but assumes 461 content as background
  - Precept attendance is critical
  - Let instructors know if you must miss, accommodations made

# 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

# Grading in **UG COS 461**

- Five assignments (50% total)
  - 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
- Midterm exam (20%)
- Final exam (25%)
- Class participation (precept, 5%)

# Grading in **Graduate-Level COS 561**

- Semester-long Research Project (40% total)
  - Includes proposal, presentation, and final write-up
  - In groups of 2/3 students; must involve programming
  - Can (1) reproduce research results, or (2) conduct novel research; regardless, *must* relate to COS 561 topics
- One take-home midterm exam (30% total)
  - November 9-10
  - Open-ended questions, e.g., how solutions work/don't work, extensions to solutions for different goals/settings
- Participation (precept, 30%)
  - Includes in-precept discussion and Perusall comments

# 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!

# 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.

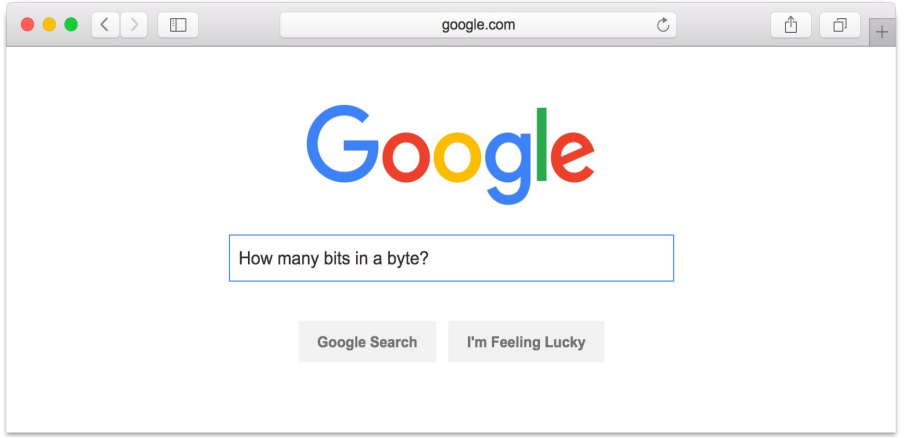
Do not copy another person's programs, comments, or any part of submitted assignment. This includes character-by-character transliteration but also derivative works. Cannot use another's code, etc. even while "citing" them.

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

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

**Don't Plagiarize!**

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

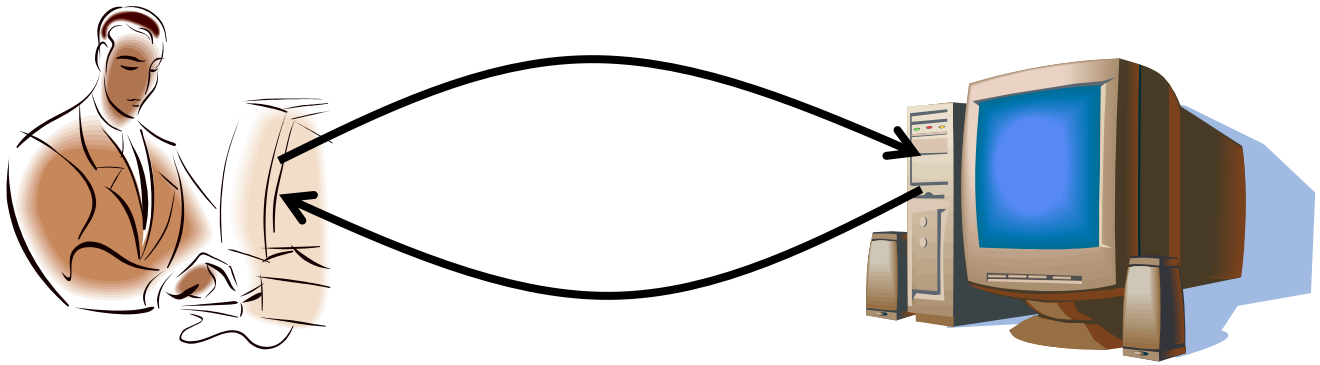



- Try to figure out yourself
- Forums are not for debugging
  - Utilize right venue:  Go to TA office hours
  - Send detailed Q’s / bug reports, not “no idea what’s wrong”
- Instructors are not on pager duty 24 x 7
  - Don’t expect response before next business day
  - Questions Friday night @ 11pm should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

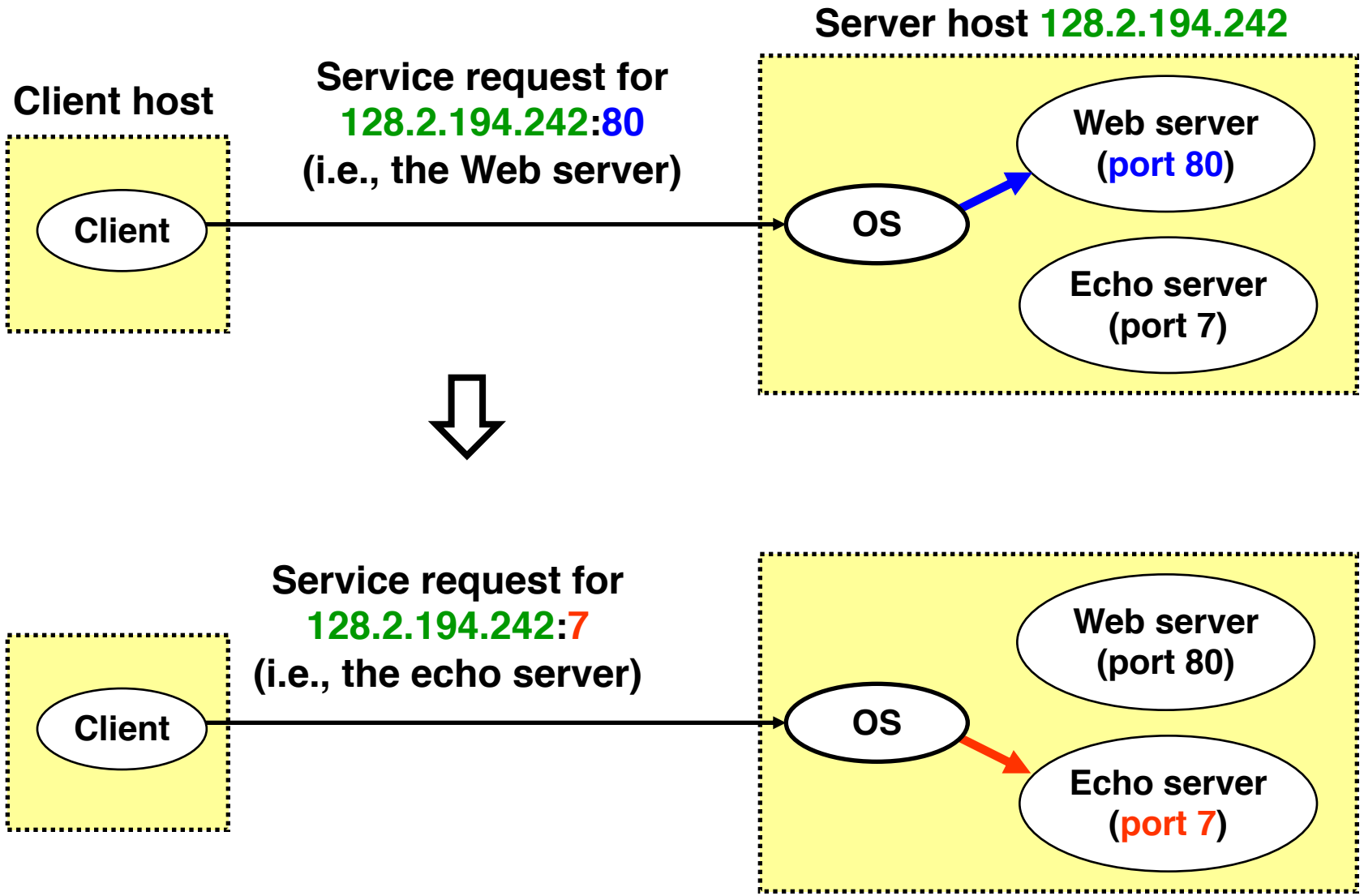
# Assignment 1:
# Socket Programming

# 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 Web site
  - Doesn't initiate contact with the clients
  - Needs fixed, known address



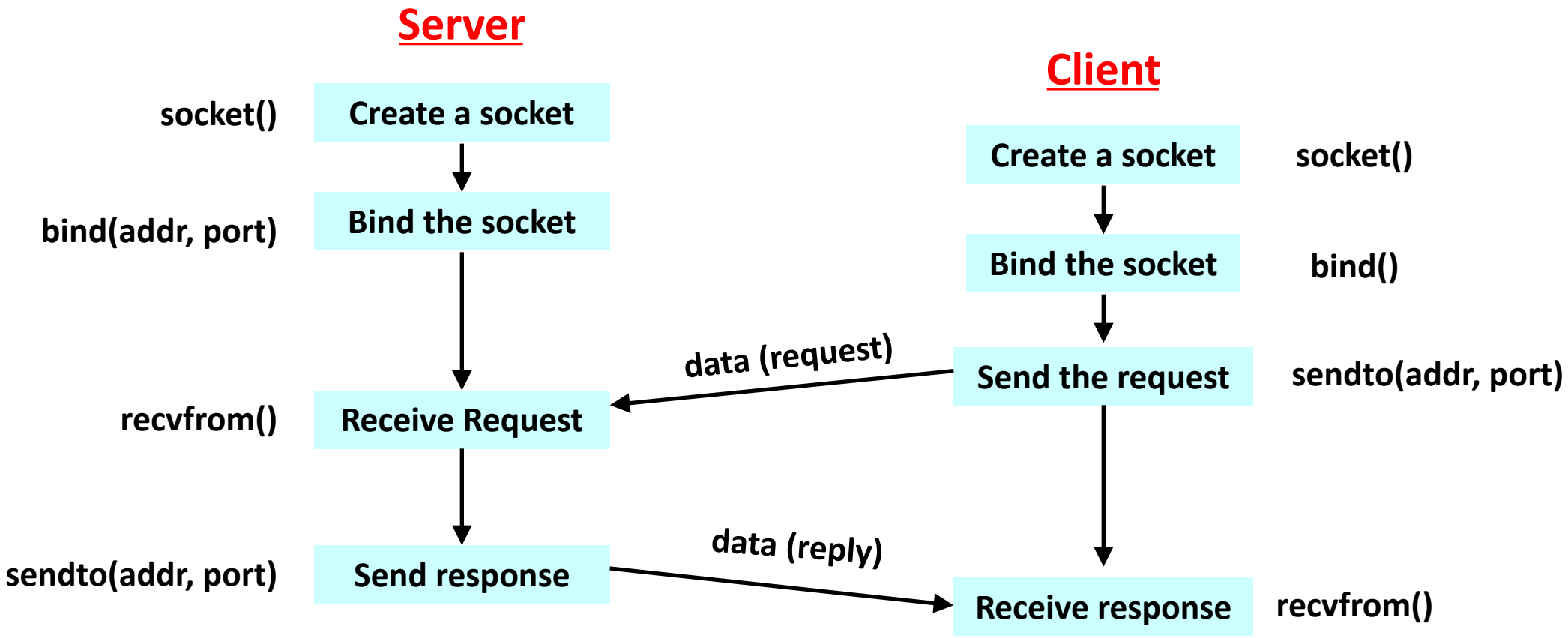
# Using IP + Ports to Identify Services



# Two Types of Application Processes Communication

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

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



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