Course Information

- Instructor
  - Prof. Robin Kravets
    3114 SC
    217-244-6026
    rhk@illinois.edu

- TAs
  - Federico Cifuentes-Urtubey, Rohan Tabish, Chien-Ying Chen

- Class Webpage
  - http://courses.engr.illinois.edu/cs438/
Course Information

- Use Piazza for all class related communication
  - Announcements and discussions
    - http://www.piazza.com/illinois/cs438
      - All class questions
      - This is your one-stop help-line!
      - Will get answer < 24 hours
  - For personal communications, do not send email
    - Use the private message function on Piazza
Course Information

- **Text book**

- **Supplemental Text books**
  - UNIX Network Programming, by Stevens
Prerequisites

- Operating Systems Concepts
  - CS 241 or ECE 391 or equivalent
    - Threads, memory management, sockets

- C or C++ Programming
  - Preferably Unix

- Probability and Statistics
Grading Policy

- **Homework** 14%
  - 7 homework assignments

- **Programming Projects** 46%
  - MP0 3%, MP1 11%, MP2 16%, MP3 16%

- **Midterm Exam** 15%
  - March 4, 7 - 9PM

- **Final Exam** 25%
  - TBA
Homework and Projects

- Homework
  - 7 homeworks each worth 2%
  - Due Wednesdays at start of class.
  - General extension to Fridays start of class (hard deadline).
    - Solutions handed out on Fridays
  - No questions to Professor, TAs or on Piazza after class on Wednesday.

- Projects
  - Late policy for projects - 2% off per hour late
  - MP0 and MP1 are solo
  - MP2 and MP3 are 2 person teams
Regrades

- Within one week of posting of grades for a homework, MP or exam
- Regrades must be submitted in writing on a separate piece of paper
  - Please do not write on your homework, MP or Exam
Academic Honesty

- Your work in this class **must** be your own.
- If students are found to have cheated (e.g., by copying or sharing answers during an examination or sharing code for a project), **all** involved will at a minimum receive grades of 0 for the first infraction.
  - We will run a similarity-checking system on code and binaries
- Further infractions will result in failure in the course and/or recommendation for dismissal from the university.
- Department honor code: [https://cs.illinois.edu/academics/honor-code](https://cs.illinois.edu/academics/honor-code)
What is cheating in a programming class?

- At a minimum
  - Copying code
  - Copying pseudo-code
  - Copying flow charts

- Consider
  - Did some one else tell you how to do it?

- Does this mean I can’t help my friend?
  - No, but don’t solve their problems for them
Graduate Students

- Graduate students MAY take an extra one hour project in conjunction with this class
  - Graduate students
    - Write a survey paper in a networking research area of your choice
    - Project proposal with list of 10+ academic references (no URL’s) due February 20th
    - Paper due last day of class
  - Undergraduates may not take this project course
    - However, if you are interested in networking research, please contact me
Goal: foundational view of computer networks

- Fundamental challenges of computer networking
- Design principles of computer networks
- From principles to practical protocols
- Build real network applications
Course Contents

- Introduction to UNIX Network Programming
- Direct Link Networks
- Packet Switched Networks
- Routing
- Internetworking
- End-to-End Protocols
- Congestion Control
- Mobile Networks
- Network Security
- … more if there is time
Complete Schedule

- See class webpage
- http://www.cs.illinois.edu/class/cs438
  - Schedule is dynamic
  - Check regularly for updates

Content
- Slides will be posted by the night before class
  - Some class material may not be in slides
    - Examples may be worked out in class
What do these two things have in common?

- First printing press
- The Internet

Both lowered the cost of distributing information and changed human society
A Brief History of the Internet
Visionaries

- Vannevar Bush, “As we may think” (1945):
  - memex - an adjustable microfilm viewer

- J. C. R. Licklider (1962): “Galactic Network”
  - Concept of a global network of computers connecting people with data and programs
  - First head of DARPA computer research, October 1962
  - Funded Arpanet
Circuit switching

1920s

1967
1961-64: Packet switching

- Leonard Kleinrock

- Paul Baran (RAND), Donald Davies
  - Concurrent work from (National Physical Laboratories, UK)
1961-64: Packet switching

<table>
<thead>
<tr>
<th>Circuit Switching</th>
<th>Datagram packet switching</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### 1961-64: Packet switching

<table>
<thead>
<tr>
<th>Circuit Switching</th>
<th>Datagram packet switching</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical channel carrying stream of data from source to destination</td>
<td></td>
</tr>
<tr>
<td>Three phase: setup, data transfer, tear-down</td>
<td></td>
</tr>
<tr>
<td>Data transfer involves no routing</td>
<td></td>
</tr>
</tbody>
</table>
1961-64: Packet switching

<table>
<thead>
<tr>
<th>Circuit Switching</th>
<th>Datagram packet switching</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical channel carrying stream of data from source to destination</td>
<td>Message broken into short packets, each handled separately</td>
</tr>
<tr>
<td>Three phase: setup, data transfer, tear-down</td>
<td>One operation: send packet</td>
</tr>
<tr>
<td>Data transfer involves no routing</td>
<td>Packets stored (queued) in each router, forwarded to appropriate neighbor</td>
</tr>
</tbody>
</table>
1965: First computer network

- Lawrence Roberts and Thomas Merrill connect a TX-2 at MIT to a Q-32 in Santa Monica, CA
- ARPA-funded project
- Connected with telephone line – it works, but it’s inefficient and expensive – confirming motivation for packet switching
The ARPANET begins

- Roberts joins DARPA (1966), publishes plan for the ARPANET computer network (1967)
- December 1968: Bolt, Beranek, and Newman (BBN) wins bid to build packet switch, the Interface Message Processor (IMP)
- September 1969: BBN delivers first IMP to Kleinrock’s lab at UCLA
ARPANET comes alive

Stanford Research Institute (SRI)  

“LO”  
Oct 29, 1969  

UCLA
ARPANET grows

- Dec 1970: ARPANET Network Control Protocol (NCP)
- 1971: Telnet, FTP
- 1972: Email (Ray Tomlinson, BBN)
- 1979: USENET
And grows …

ARPA NETWORK, LOGICAL MAP, SEPTEMBER 1973

77 nodes

How many do we have today?
Meanwhile, other networks such as PRnet, SATNET developed

May 1973: Vinton G. Cerf and Robert E. Kahn present first paper on interconnecting networks

Concept of connecting diverse networks, unreliable datagrams, global addressing, ...

Became TCP/IP

2004 Turing Award!
TCP/IP deployment

- TCP/IP implemented on mainframes by groups at Stanford, BBN, UCL
- David Clark implements it on Xerox Alto and IBM PC
  - Design by committee didn’t win out
- January 1, 1983: “Flag Day” NCP to TCP/IP transition on ARPANET
OSI Protocol Stack

- **Application:** Application specific protocols
- **Presentation:** Format of exchanged data
- **Session:** Name space for connection mgmt
- **Transport:** Process-to-process channel
- **Network:** Host-to-host packet delivery
- **Data Link:** Framing of data bits
- **Physical:** Transmission of raw bits
Growth from Ethernet

- **Ethernet**
  - R. Metcalfe and D. Boggs, July 1976

- **Spanning Tree protocol**
  - Radia Perlman, 1985

- **Made local area networking easy**
Growth spurs organic change

- **Early 1980s**
  - Many new networks: CSNET, BITNET, MFENet, SPAN (NASA), ...

- **Nov 1983**
  - DNS developed by Jon Postel, Paul Mockapetris (USC/ISI), Craig Partridge (BBN)

- **1984**
  - Hierarchical routing: EGP and IGP (later to become eBGP and iBGP)
NSFNET

- 1984: NSFNET for US higher education
  - Serve many users, not just one field
  - Encourage development of private infrastructure (e.g., initially, backbone required to be used for Research and Education)
  - Stimulated investment in commercial long-haul networks
- 1990: ARPANET ends
- 1995: NSFNET decommissioned
Explosive growth!

In users

<table>
<thead>
<tr>
<th>World Regions</th>
<th>Population (2018 Est.)</th>
<th>Population % of World</th>
<th>Internet Users 30 June 2018</th>
<th>Penetration Rate (% Pop.)</th>
<th>Growth 2000-2018</th>
<th>Internet Users %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Africa</td>
<td>1,287,914,329</td>
<td>16.9 %</td>
<td>464,923,169</td>
<td>36.1 %</td>
<td>10,199 %</td>
<td>11.0 %</td>
</tr>
<tr>
<td>Asia</td>
<td>4,207,588,157</td>
<td>55.1 %</td>
<td>2,062,197,366</td>
<td>49.0 %</td>
<td>1,704 %</td>
<td>49.0 %</td>
</tr>
<tr>
<td>Europe</td>
<td>827,650,849</td>
<td>10.8 %</td>
<td>705,064,923</td>
<td>85.2 %</td>
<td>570 %</td>
<td>16.8 %</td>
</tr>
<tr>
<td>Latin America / Caribbean</td>
<td>652,047,996</td>
<td>8.5 %</td>
<td>438,248,446</td>
<td>67.2 %</td>
<td>2,325 %</td>
<td>10.4 %</td>
</tr>
<tr>
<td>Middle East</td>
<td>254,438,981</td>
<td>3.3 %</td>
<td>164,037,259</td>
<td>64.5 %</td>
<td>4,894 %</td>
<td>3.9 %</td>
</tr>
<tr>
<td>North America</td>
<td>363,844,662</td>
<td>4.8 %</td>
<td>345,660,847</td>
<td>95.0 %</td>
<td>219 %</td>
<td>8.2 %</td>
</tr>
<tr>
<td>Oceania / Australia</td>
<td>41,273,454</td>
<td>0.6 %</td>
<td>28,439,277</td>
<td>68.9 %</td>
<td>273 %</td>
<td>0.7 %</td>
</tr>
<tr>
<td>WORLD TOTAL</td>
<td>7,634,758,428</td>
<td>100.0 %</td>
<td>4,208,571,287</td>
<td>55.1 %</td>
<td>1,066 %</td>
<td>100.0 %</td>
</tr>
</tbody>
</table>
Explosive growth!

In users

<table>
<thead>
<tr>
<th>World Regions</th>
<th>Population (2018 Est.)</th>
<th>Population % of World</th>
<th>Internet Users 30 June 2018</th>
<th>Penetration Rate (% Pop.)</th>
<th>Growth 2000-2018</th>
<th>Internet Users %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Africa</td>
<td>1,287,914,329</td>
<td>16.9 %</td>
<td>464,923,169</td>
<td>36.1 %</td>
<td>10,199 %</td>
<td>11.0 %</td>
</tr>
<tr>
<td>Asia</td>
<td>4,207,588,157</td>
<td>55.1 %</td>
<td>2,062,197,366</td>
<td>49.0 %</td>
<td>1,704 %</td>
<td>49.0 %</td>
</tr>
<tr>
<td>Europe</td>
<td>827,650,849</td>
<td>10.8 %</td>
<td>705,064,923</td>
<td>85.2 %</td>
<td>570 %</td>
<td>16.8 %</td>
</tr>
<tr>
<td>Latin America / Caribbean</td>
<td>652,047,996</td>
<td>8.5 %</td>
<td>438,248,446</td>
<td>67.2 %</td>
<td>2,325 %</td>
<td>10.4 %</td>
</tr>
<tr>
<td>Middle East</td>
<td>254,438,981</td>
<td>3.3 %</td>
<td>164,037,259</td>
<td>64.5 %</td>
<td>4,894 %</td>
<td>3.9 %</td>
</tr>
<tr>
<td>North America</td>
<td>363,844,662</td>
<td>4.8 %</td>
<td>345,660,847</td>
<td>95.0 %</td>
<td>219 %</td>
<td>8.2 %</td>
</tr>
<tr>
<td>Oceania / Australia</td>
<td>41,273,454</td>
<td>0.6 %</td>
<td>28,439,277</td>
<td>68.9 %</td>
<td>273 %</td>
<td>0.7 %</td>
</tr>
<tr>
<td>WORLD TOTAL</td>
<td>7,634,758,428</td>
<td>100.0 %</td>
<td>4,208,571,287</td>
<td>55.1 %</td>
<td>1,066 %</td>
<td>100.0 %</td>
</tr>
</tbody>
</table>
Explosive growth!

In hosts

Juniper estimates 38 Billion Devices in 2020!
Explosive growth!

In networks
Explosive growth!

In complexity

Autonomous System

BGP router

IP router

Routing protocols

- eBGP, iBGP
- MPLS, CSPF, OSPF, RIP, ...

spanning tree + learning broadcast

LAN

ethernet segment

hub

switch
Explosive growth!

In technologies
- Link speeds 200,000x faster
- NATs and firewalls
- Wireless everywhere
- Mobile everywhere
- Tiny devices (smart phones)
- Giant devices (data centers)

In applications
- Morris Internet Worm (1988)
- World wide web (1989)
- MOSAIC browser (1992)
- Search engines
- Peer-to-peer
- Voice
- Radio
- Botnets
- Social networking
- Streaming video
- Data centers
- Cloud computing
- IoT
Explosive growth!
Top 30 inventions of the last 30 years

Compiled by the Wharton School @ U Penn, 2009

1. Internet/Broadband/World Wide Web
2. PC/Laptop Computers
3. Mobile Phones
4. E-Mail
5. DNA Testing and Sequencing/Human Genome Mapping
6. Magnetic Resonance Imaging (MRI)
7. Microprocessors
8. Fiber Optics
9. Office Software
10. Non-Invasive Laser/Robotic Surgery
11. Open Source Software and Services
12. Light Emitting Diodes (LEDs)
13. Liquid Crystal Displays (LCDs)
14. GPS
15. Online Shopping/E-Commerce/Auctions
16. Media File Compression
17. Microfinance
18. Photovoltaic Solar Energy
19. Large Scale Wind Turbines
20. Social Networking via Internet
21. Graphic User Interface (GUI)
22. Digital Photography/Videography
23. RFID
24. Genetically Modified Plants
25. Biofuels
26. Bar Codes and Scanners
27. ATMs
28. Stents
29. SRAM/Flash Memory
30. Anti-Retroviral Treatment for AIDS
Top 30 inventions of the last 30 years

Compiled by the Wharton School @ U Penn, 2009

1. Internet/Broadband/World Wide Web
2. PC/Laptop Computers
3. Mobile Phones
4. E-Mail
5. DNA Testing and Sequencing/Human Genome Mapping
6. Magnetic Resonance Imaging (MRI)
7. Microprocessors
8. Fiber Optics
9. Office Software
10. Non-Invasive Laser/Robotic Surgery
11. Open Source Software and Services
12. Light Emitting Diodes (LEDs)
13. Liquid Crystal Displays (LCDs)
14. GPS
15. Online Shopping/E-Commerce/Auctions
16. Media File Compression
17. Microfinance
18. Photovoltaic Solar Energy
19. Large Scale Wind Turbines
20. Social Networking via Internet
21. Graphic User Interface (GUI)
22. Digital Photography/Videography
23. RFID
24. Genetically Modified Plants
25. Biofuels
26. Bar Codes and Scanners
27. ATMs
28. Stents
29. SRAM/Flash Memory
30. Anti-Retroviral Treatment for AIDS
Why is Networking Challenging

That’s it! …right?
Fundamental Challenge: Speed of Light

- How long does it take light to travel from UIUC to Mountain View, CA (Google Headquarters)?

**Answer:**
- Distance UIUC → Mountain View is 2,935 km
- Traveling 300,000 km/s: 9.78ms

**Note: Dependent on transmission medium**
- $3.0 \times 10^8$ meters/second in a vacuum
- $2.3 \times 10^8$ meters/second in a cable
- $2.0 \times 10^8$ meters/second in a fiber
Fundamental Challenge: Speed of Light

- How long does it take an Internet “packet” to travel from UIUC to Mountain View?

Answer:
- For sure ≥ 9.78 ms
- But also depends on:
  - The route the packet takes (could be circuitous!)
  - The propagation speed of the links the packet traverses
    - e.g. in optical fiber light propagates only at 2/3 C
  - The transmission rate (bandwidth) of the links (bits/sec)
    - And also the size of the packet
  - Number of hops traversed (“store and forward” delay)
  - The “competition” for bandwidth the packet encounters (congestion). It may have to wait in router queues.
- In practice this boils down to ≥ 40 ms (and likely more)
  - With variance (can be hard to predict!)
Performance

- **Bandwidth/throughput**
  - Data transmitted per unit time
  - Example: 10 Mbps
  - Link bandwidth vs. end-to-end bandwidth

- **Latency/delay**
  - Time from A to B
  - Example: 30 msec
  - Many applications depend on round-trip time (RTT)

- **Notation**
  - KB = $2^{10}$ bytes
  - Mbps = $10^6$ bits per second

Why?
You will mess this up at least once on a HW or exam!
Delay x Bandwidth Product

- Amount of data in “pipe”
  - channel = pipe
  - delay = length
  - bandwidth = area of a cross section
  - bandwidth x delay product = volume
Delay x Bandwidth Product

- Bandwidth x delay product
  - How many bits the sender must transmit before the first bit arrives at the receiver if the sender keeps the pipe full
  - Takes another one-way latency to receive a response from the receiver
Delay x Bandwidth Product

- Bandwidth x delay product
  - How many bits the sender must transmit before the first bit arrives at the receiver if the sender keeps the pipe full
  - Takes another one-way latency to receive a response from the receiver
Delay x Bandwidth Product

- Bandwidth x delay product
  - How many bits the sender must transmit before the first bit arrives at the receiver if the sender keeps the pipe full
  - Takes another one-way latency to receive a response from the receiver
Delay x Bandwidth Product

- Bandwidth x delay product
  - How many bits the sender must transmit before the first bit arrives at the receiver if the sender keeps the pipe full
  - Takes another one-way latency to receive a response from the receiver
Delay x Bandwidth Product

- Bandwidth x delay product
  - How many bits the sender must transmit before the first bit arrives at the receiver if the sender keeps the pipe full
  - Takes another one-way latency to receive a response from the receiver
Delay x Bandwidth Product

- Bandwidth x delay product
  - How many bits the sender must transmit before the first bit arrives at the receiver if the sender keeps the pipe full
  - Takes another one-way latency to receive a response from the receiver
Delay x Bandwidth Product

- Bandwidth x delay product
  - How many bits the sender must transmit before the first bit arrives at the receiver if the sender keeps the pipe full
  - Takes another one-way latency to receive a response from the receiver
**Delay x Bandwidth Product**

- **Bandwidth x delay product**
  - How many bits the sender must transmit before the first bit arrives at the receiver if the sender keeps the pipe full
  - Takes another one-way latency to receive a response from the receiver (round trip BxD)
Delay x Bandwidth Product

- Example: Transcontinental Channel
  - BW = 45 Mbps
  - delay = 50ms
  - bandwidth x delay product
    \[= (50 \times 10^{-3} \text{ sec}) \times (45 \times 10^6 \text{ bits/sec})\]
    \[= 2.25 \times 10^6 \text{ bits}\]
Bandwidth vs. Latency

- Relative importance
  - 1-byte: Latency bound
    - 1ms vs 100ms latency dominates 1Mbps vs 100Mbps BW
  - 25MB: Bandwidth bound
    - 1Mbps vs 100Mbps BW dominates 1ms vs 100ms latency
Bandwidth vs. Latency

- Infinite bandwidth
  - RTT dominates
    - Throughput = TransferSize / TransferTime
    - TransferTime = RTT + 1/Bandwidth x TransferSize

- It's all relative
  - 1-MB file on a 1-Gbps link looks like a 1-KB packet on a 1-Mbps link
Fundamental Challenge: Speed of Light

- How many cycles does your PC execute before it can possibly get a reply to a message it sent to a Mountain View web server?

Answer
- Round trip takes \( \geq 80\text{ms} \)
- PC runs at (say) 3 GHz
- \( 3,000,000,000 \text{ cycles/sec} \times 0.08 \text{ sec} = 240,000,000 \text{ cycles} \)

Thus
- Communication feedback is always dated
- Communication fundamentally asynchronous
Fundamental Challenge: Speed of Light

- What about machines directly connected (via a local area network or LAN)?

Answer:

```
% ping www.cs.uiuc.edu
PING dcs-www.cs.uiuc.edu (128.174.252.83) 56(84) bytes of data.
64 bytes from 128.174.252.83: icmp_seq=1 ttl=63 time=0.263 ms
64 bytes from 128.174.252.83: icmp_seq=2 ttl=63 time=0.595 ms
64 bytes from 128.174.252.83: icmp_seq=3 ttl=63 time=0.588 ms
64 bytes from 128.174.252.83: icmp_seq=4 ttl=63 time=0.554 ms
...
```

- 500us = 1,500,000 cycles
  - Still a looooonng time…
Fundamental Challenge: Shared infrastructure

- Different parties must work together
  - Multiple parties with different agendas must agree how to divide the task between them

- Working together requires
  - Protocols (defining who does what)
    - These generally need to be standardized
  - Agreements regarding how different types of activity are treated (policy)

- Different parties very well might try to “game” the network’s mechanisms to their advantage
Fundamental Challenge: Shared infrastructure

- Physical links and switches must be shared among many users

- Common multiplexing strategies
  - (Synchronous) time-division multiplexing (TDM)
  - Frequency-division multiplexing (FDM)
Fundamental Challenge: Shared infrastructure

- Statistical Multiplexing (SM)
  - On-demand time-division multiplexing
  - Scheduled on a per-packet basis
  - Packets from different sources are interleaved
  - Uses upper bounds to limit transmission
    - Queue size determines capacity per source
Fundamental Challenge: Shared infrastructure

- Packets buffered in switch until forwarded
- Selection of next packet depends on policy
  - How do we make these decisions in a fair manner? Round Robin? FIFO?
  - How should the switch handle congestion?
Fundamental Challenge: Things break

- Communication involves a chain of interfaces, links, routers, and switches...
  ...stitched together with many layers of software...
  ...all of which must function correctly!
Suppose a communication involves 50 components that work correctly (independently) 99% of the time.

What’s the likelihood the communication fails at a given point in time?

- Answer: success requires that they all function, so failure probability = \(1 - 0.99^{50} = 39.5\%\)

So we have a lot of components, which tend to fail…

- … and we may not find out for a looooong time
Fundamental Challenge: Enormous dynamic range

- **Challenge:** enormous dynamic range
  - Round trip times (**latency**) 10 us’s to sec’s ($10^5$)
  - Data rates (**bandwidth**) kbps to 10 Gbps ($10^7$)
  - Queuing delays in the network 0 to sec’s
  - Packet loss 0 to 90+% 
  - End system (**host**) capabilities cell phones to clusters
  - Application needs: size of transfers, bidirectionality, reliability, tolerance of **jitter**
Fundamental Challenge: Enormous dynamic range

- Challenge: enormous dynamic range

- Related challenge: very often, there is no such thing as “typical”
  - Beware of your “mental models”!
  - Must think in terms of design ranges, not points
  - Mechanisms need to be adaptive
Fundamental Challenge: Security

- Challenge: there are Bad Guys out there!
- Early days
  - Vandals
  - Hackers
  - Crazies
  - Researchers
- As network population grows, it becomes more and more attractive to crooks
- As size of and dependence on the network grows, becomes more attractive to spies, governments, and militaries
Fundamental Challenge: Security

- Attackers seek ways to misuse the network towards their gain
  - Carefully crafted “bogus” traffic to manipulate the network’s operation
  - Torrents of traffic to overwhelm a service (denial-of-service) for purposes of extortion/competition
  - Passively recording network traffic in transit (sniffing)
  - Exploit flaws in clients and servers using the network to trick into executing the attacker’s code (compromise)

- They all do this energetically because there is significant $$$ to be made
The Ultimate Challenge

- Cannot reboot the Internet
  - Everyone depends on the Internet
    - Businesses
    - Hospitals
    - Education institutions
    - Financial sector
    - ...

- Fixing the Internet akin to changing the engine while you are flying the plane!
Why Networking is Challenging

- Tubes: not entirely wrong, but simplistic
- How do we build a communication infrastructure for all of humanity?
- Must design for extreme heterogeneity across technology, applications, users
What’s next

- MP 0
  - Available Friday
  - Sockets refresher

- HW 1
  - Available Friday

- Next topic
  - UNIX network programming

- Next week
  - Technical overview of Internet architecture
  - Data link technologies