CS/ECE 438: Communication Networks

Prof. Robin Kravets

Course Information

- Instructor
  - Prof. Robin Kravets
    3114 SC
    217-244-6026
    rhk@illinois.edu
- TAs
  - Federico Cifuentes-Urtubey, Zhanghao Chen, Bo 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
  - Computer Networks: A Systems Approach, by Peterson and Davie, 5th Ed. (minor differences from 4th edition)
- 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 10, 7 - 9PM
- Final Exam 25%
  - TBA

Homework and Projects

- Homework
  - 7 homeworks each worth 2%
  - Due Thursday at start of class (hard deadline).
    - Solutions posted on Thursdays
- 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`**

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?
  - Did you find the code on the web?
- Does this mean I can’t help my friend?
  - No, but don’t solve their problems for them

Graduate Students

- Graduate students MAY take this class for 4 credits
  - 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 25th
    - Paper due last day of class
  - Undergraduates may not take this 4 credit 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

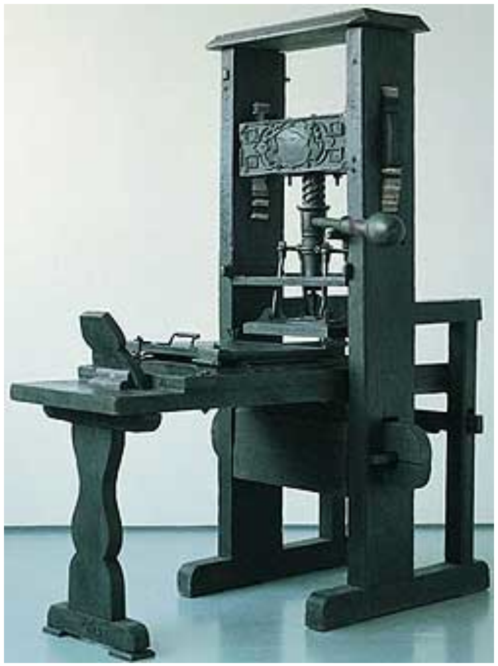
Course Contents

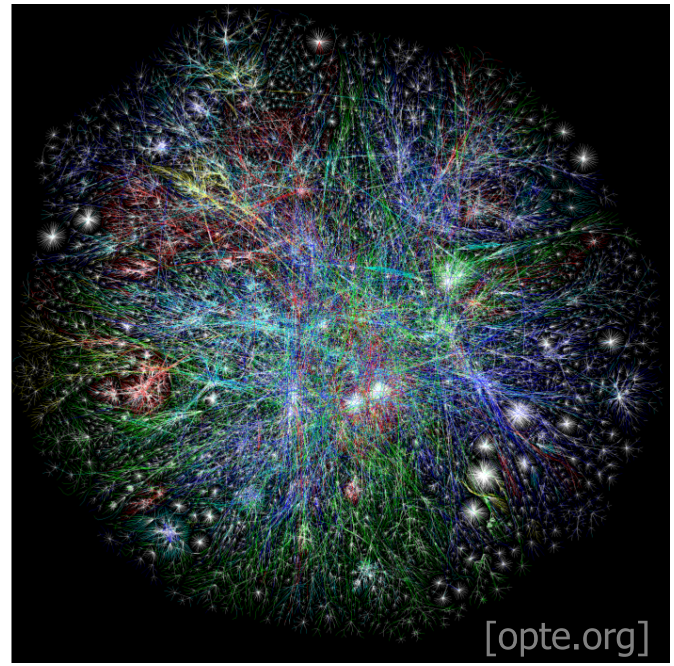
- Introduction to UNIX Network Programming
- Direct Link Networks
- Packet Switched Networks
- Routing
- Internetworking
- End-to-End Protocols
- Congestion Control
- Mobile Networks
- Network Security
- … more if there is time

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

Bush

Licklider

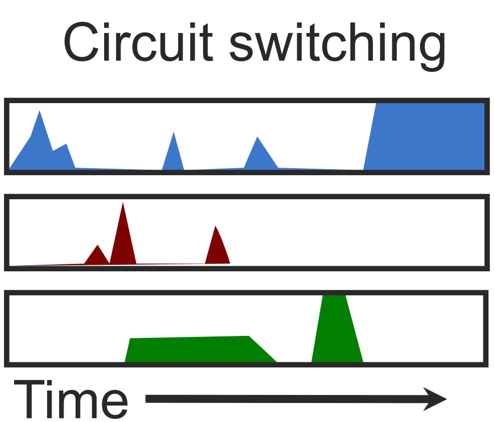
Circuit switching

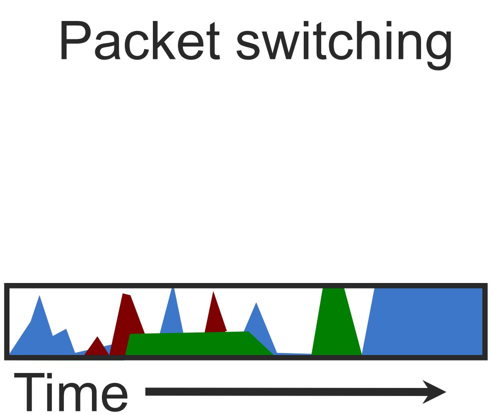
1920s

1967

1961-64: Packet switching

- Leonard Kleinrock
  - Queueing-theoretic analysis of packet switching in MIT Ph.D. thesis (1961-63) demonstrated value of statistical multiplexing
- Paul Baran (RAND), Donald Davies
  - Concurrent work from (National Physical Labratories, UK)





Kleinrock

Baran

1961-64: Packet switching

Circuit Switching	Datagram packet switching

1961-64: Packet switching

Circuit Switching	Datagram packet switching
Physical channel carrying stream of data from source to destination	
Three phase: setup, data transfer, tear-down	
Data transfer involves no routing	

1961-64: Packet switching

Circuit Switching	Datagram packet switching
Physical channel carrying stream of data from source to destination	Message broken into short packets, each handled separately
Three phase: setup, data transfer, tear-down	One operation: send packet
Data transfer involves no routing	Packets stored (queued) in each router, forwarded to appropriate neighbor

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

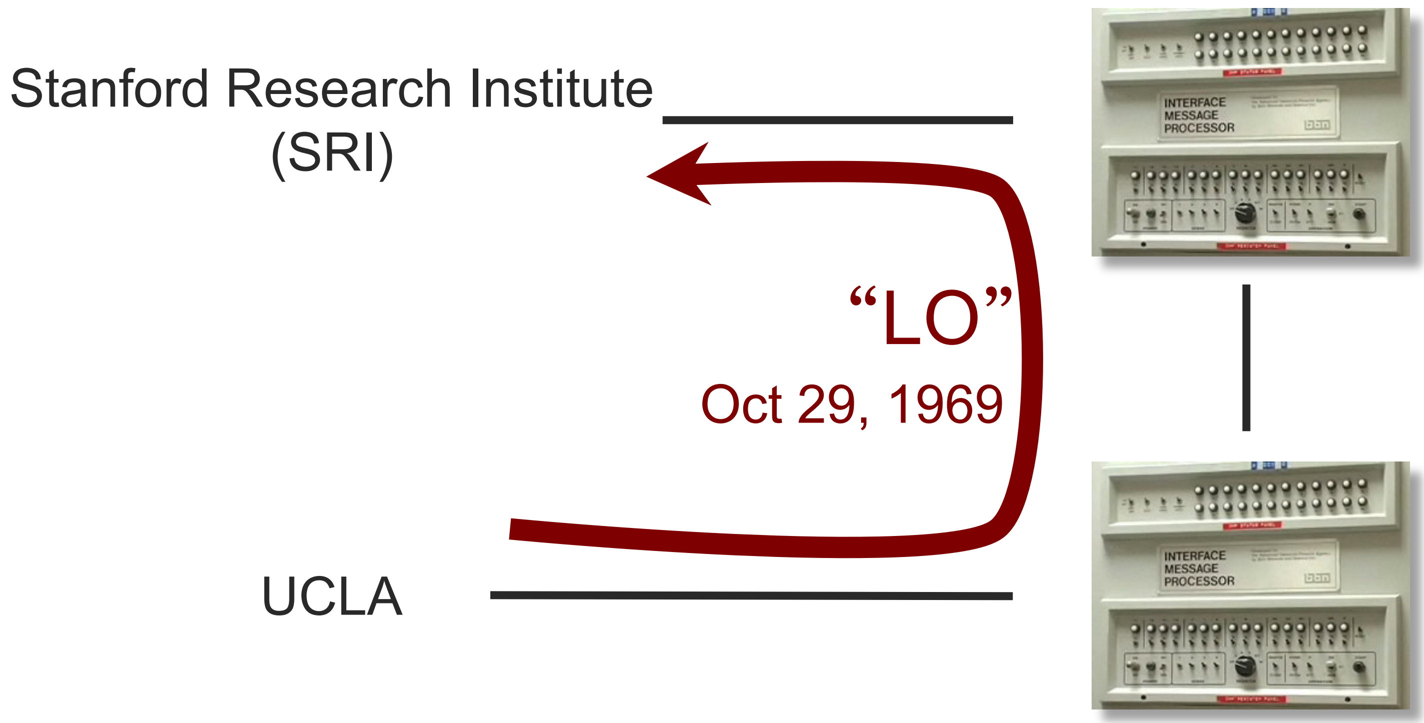
Roberts

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

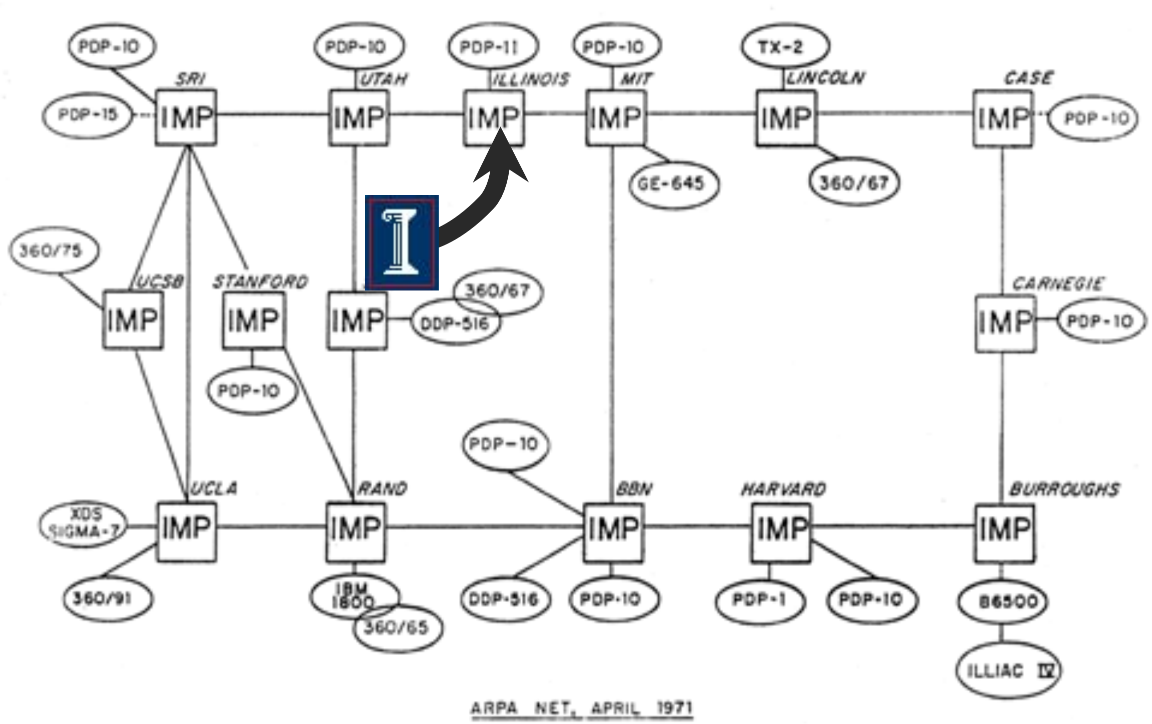
An older Kleinrock with the first IMP

ARPANET comes alive



ARPANET grows

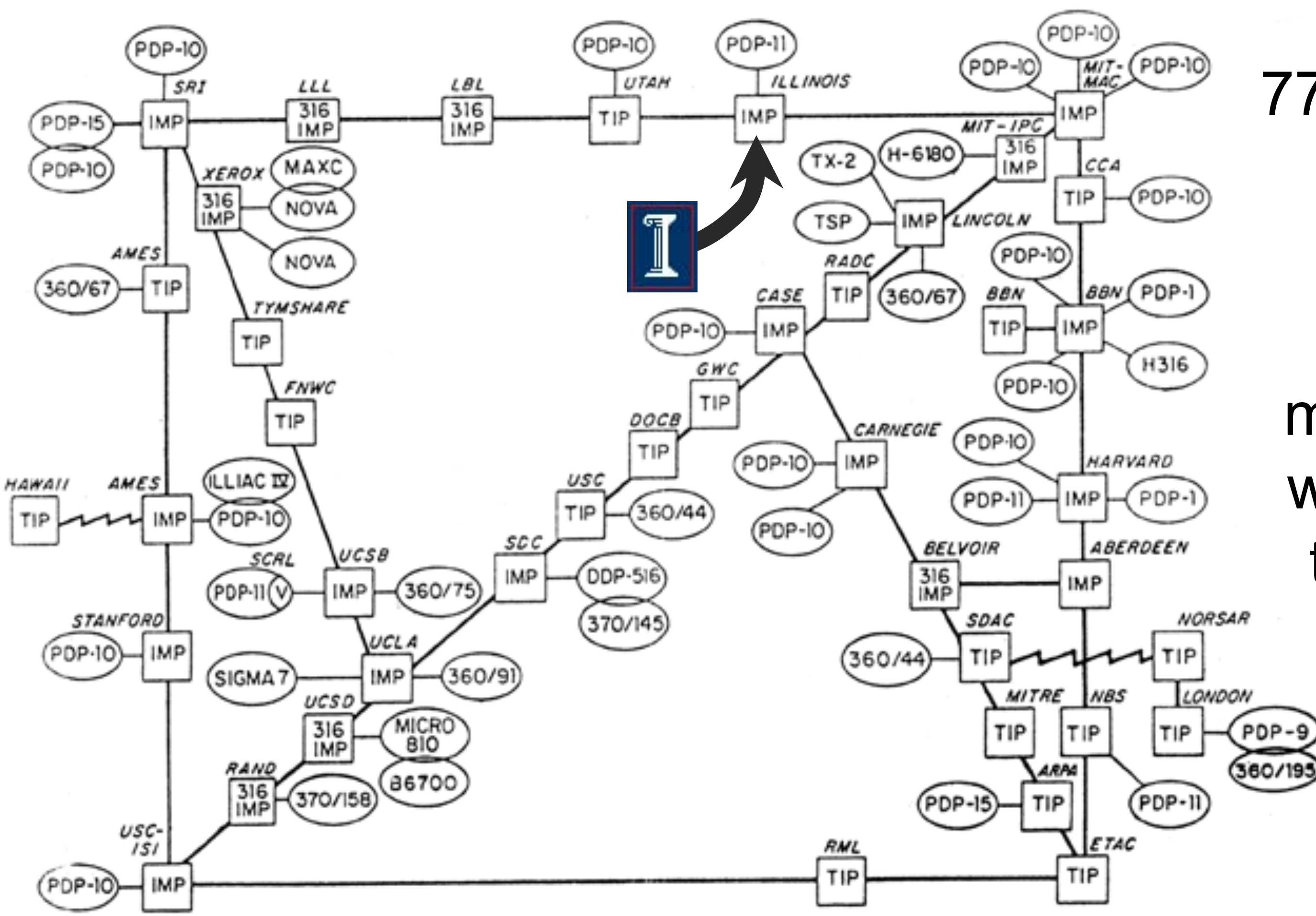
- Dec 1970: ARPANET Network Control Protocol (NCP)
- 1971: Telnet, FTP
- 1972: Email (Ray Tomlinson, BBN)
- 1979: USENET



ARPANET, April 1971

And grows …

ARPA NETWORK, LOGICAL MAP, SEPTEMBER 1973



77 nodes

How many do we have today?

~ 1B

ARPANET to Internet

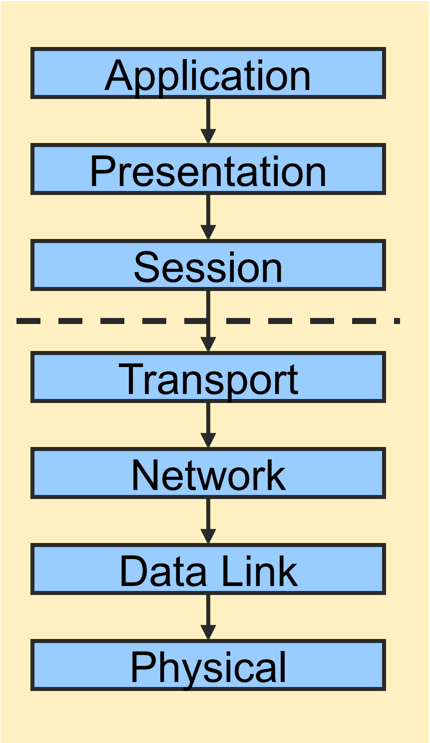
- Meanwhile, other networks such as PRnet, SATNET deveoped
- May 1973: Vinton G. Cerf and Robert E. Kahn present first paper on interconnecting networks
- Concept of connecting diverse networks, unreliable datagrams, global addressing, ...
- Became TCP/IP

2004 Turing Award!

Cerf

Kahn

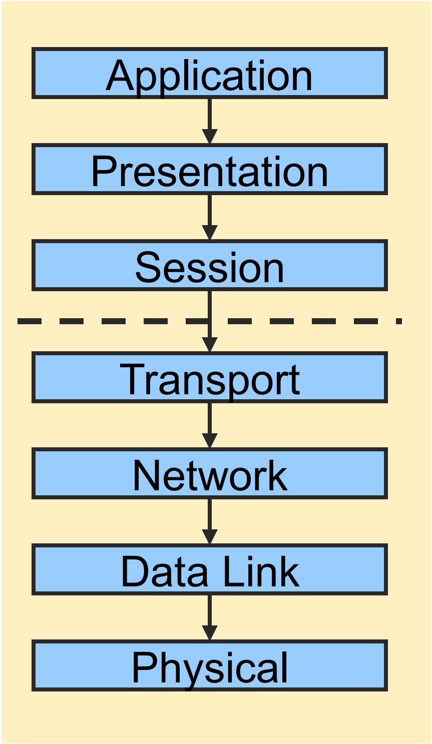
TCP/IP deployment



OSI Reference Model's layers

- TCP/IP implemented on mainframes by groups at Stanford, BBN, UCL
- David Clark implements it on Xerox Alto and IBM PC
- 1982: International Organization for Standards (ISO) releases Open Systems Interconnection (OSI) reference model
  - Design by committee didn't win out
- January 1, 1983: "Flag Day" NCP to TCP/IP transition on ARPANET

OSI Protocol Stack

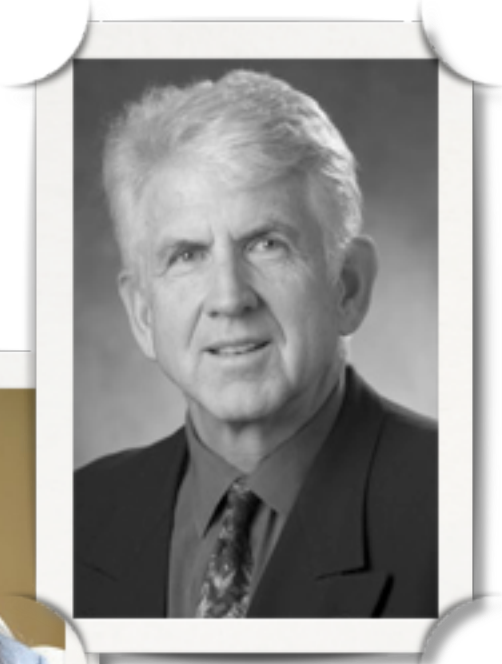


- Application: Application specific protocols
- Presentation: Format of exchanged data
- Session: Name space for connection mgmt

- 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

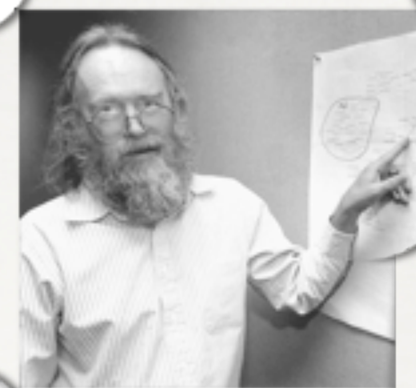


Metcalfe

Perlman

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)



Postel

Mockapetris

Partridge

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

NSFNET backbone, 1992

Explosive growth!

In users

WORLD INTERNET USAGE AND POPULATION STATISTICS 2019 Mid-Year Estimates						
World Regions	Population (2019 Est.)	Population % of World	Internet Users 30 June 2019	Penetration Rate (% Pop.)	Growth 2000-2019	Internet World %
Africa	1,320,038,716	17.1 %	522,809,480	39.6 %	11,481 %	11.5 %
Asia	4,241,972,790	55.0 %	2,300,469,859	54.2 %	1,913 %	50.7 %
Europe	829,173,007	10.7 %	727,559,682	87.7 %	592 %	16.0 %
Latin America / Caribbean	658,345,826	8.5 %	453,702,292	68.9 %	2,411 %	10.0 %
Middle East	258,356,867	3.3 %	175,502,589	67.9 %	5,243 %	3.9 %
North America	366,496,802	4.7 %	327,568,628	89.4 %	203 %	7.2 %
Oceania / Australia	41,839,201	0.5 %	28,636,278	68.4 %	276 %	0.6 %
WORLD TOTAL	7,716,223,209	100.0 %	4,536,248,808	58.8 %	1,157 %	100.0 %

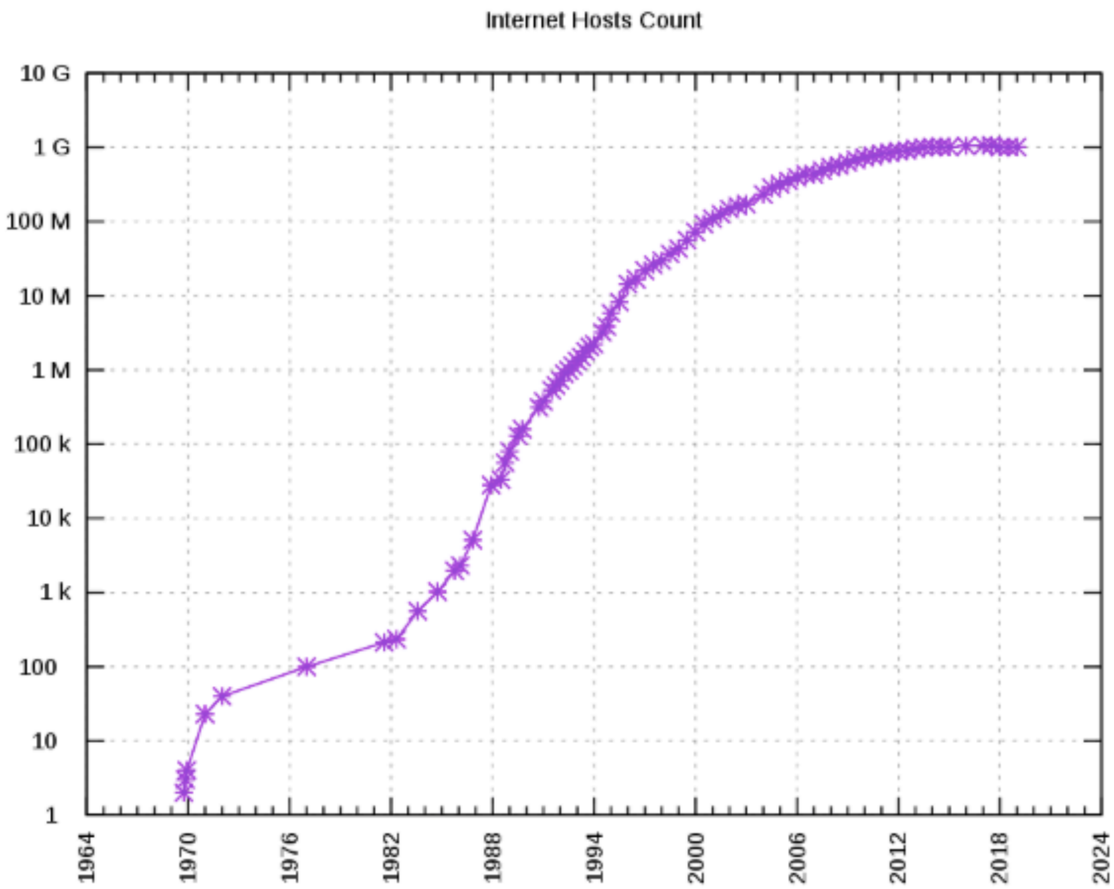
Explosive growth!

In users

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

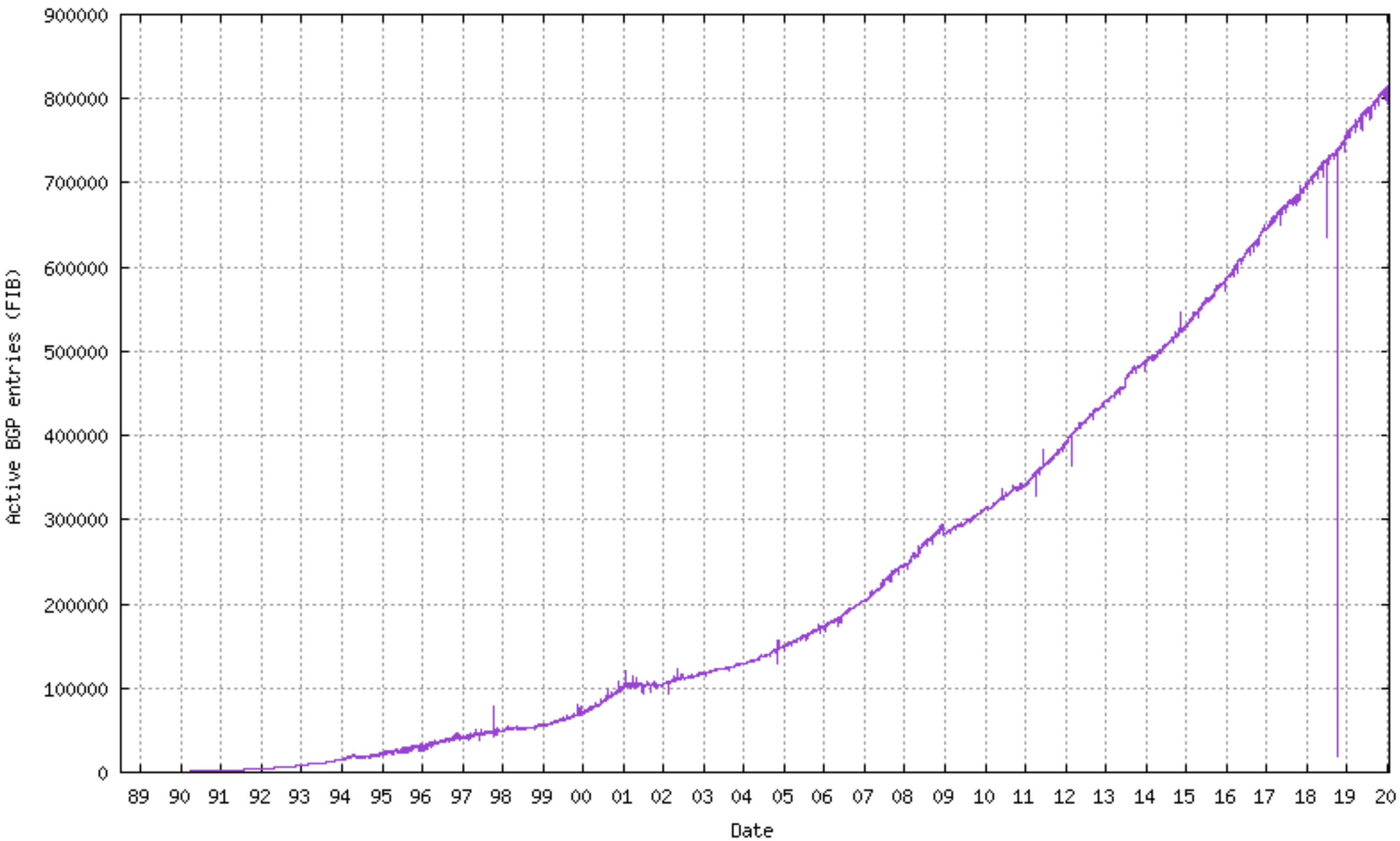
Explosive growth!

In hosts



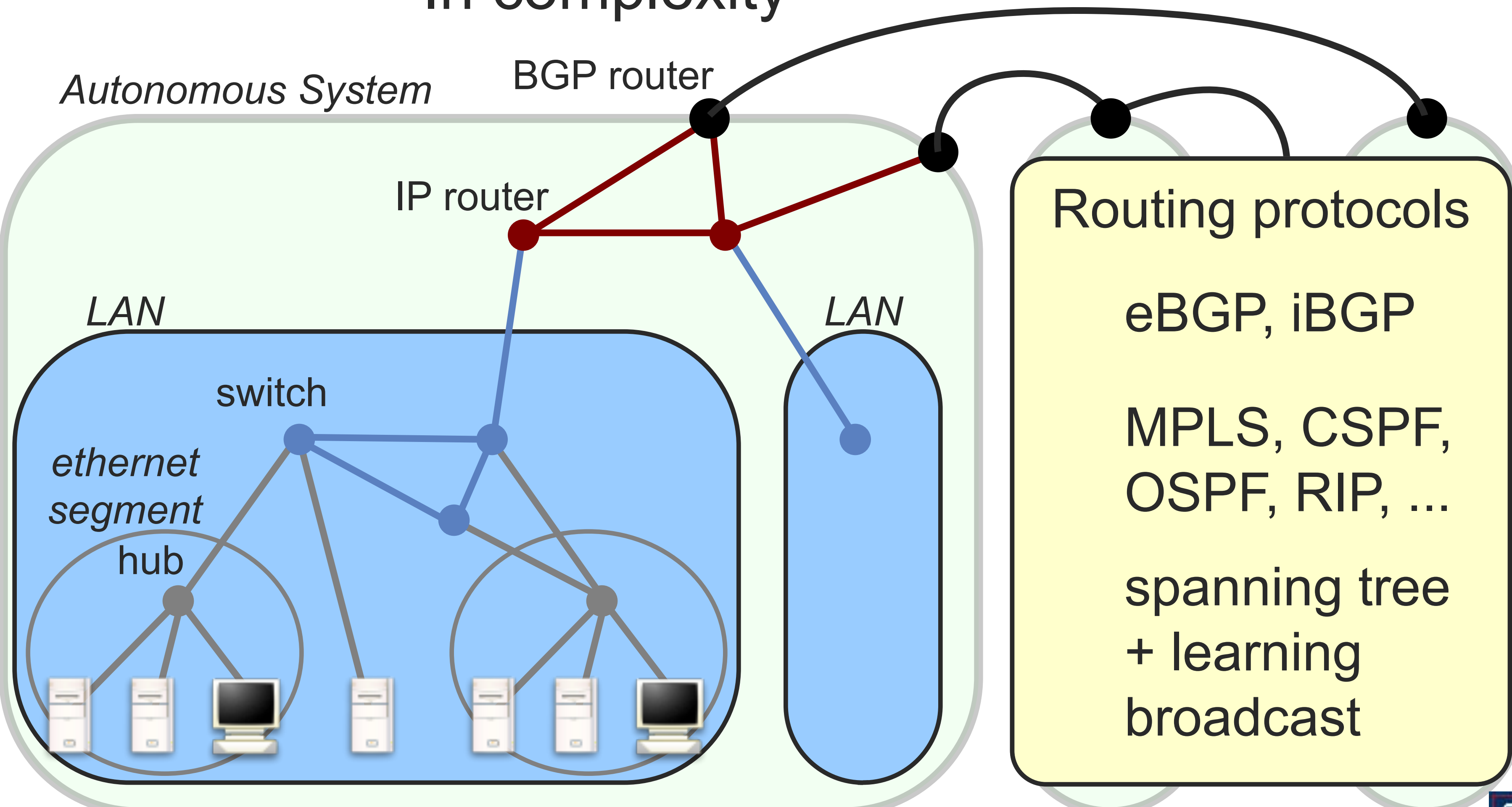
Explosive growth!

In networks



Explosive growth!

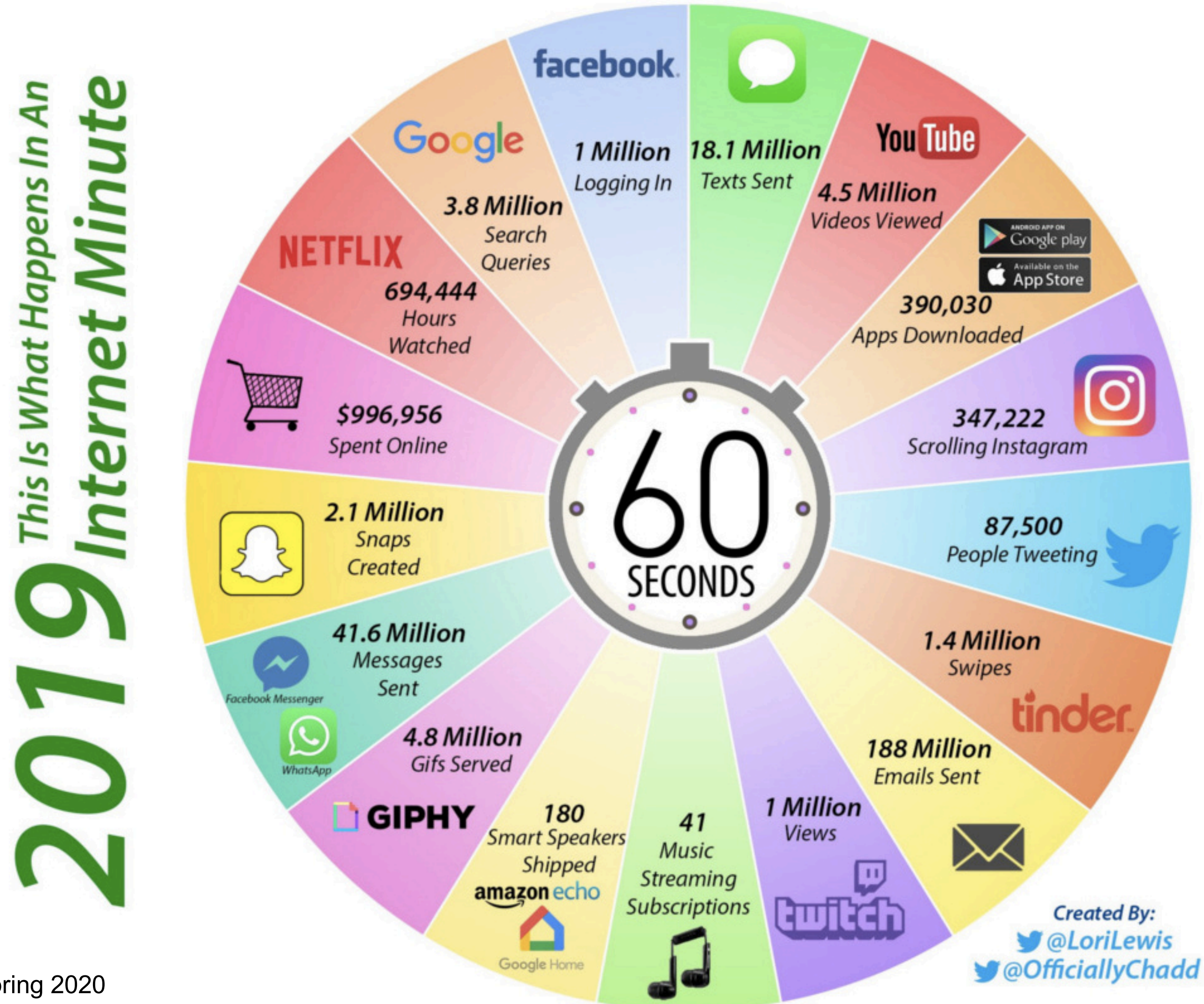
In complexity



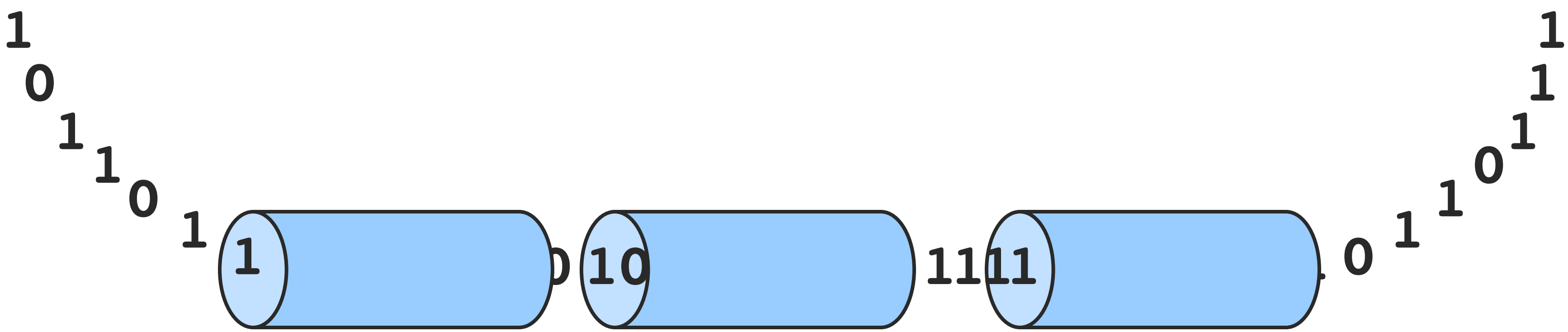
Explosive growth!

- 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!



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
  - 3.0×10^8 meters/second in a vacuum
  - 2.3×10^8 meters/second in a cable
  - 2.0×10^8 meters/second in a fiber

nanosec

Fundamental Challenge: Speed of Light

- How long does it take an Internet "packet" to travel from UIUC to Mountain View?
- Answer:
  - For sure ≥ 9.78ms
  - 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 ≥ 40ms (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
  - Notation
    - KB = 2^{10} bytes
    - Mbps = 10^{6} bits per second
- Latency/delay
  - Time from A to B
  - Example: 30 msec
  - Many applications depend on round-trip time (RTT)

Why?

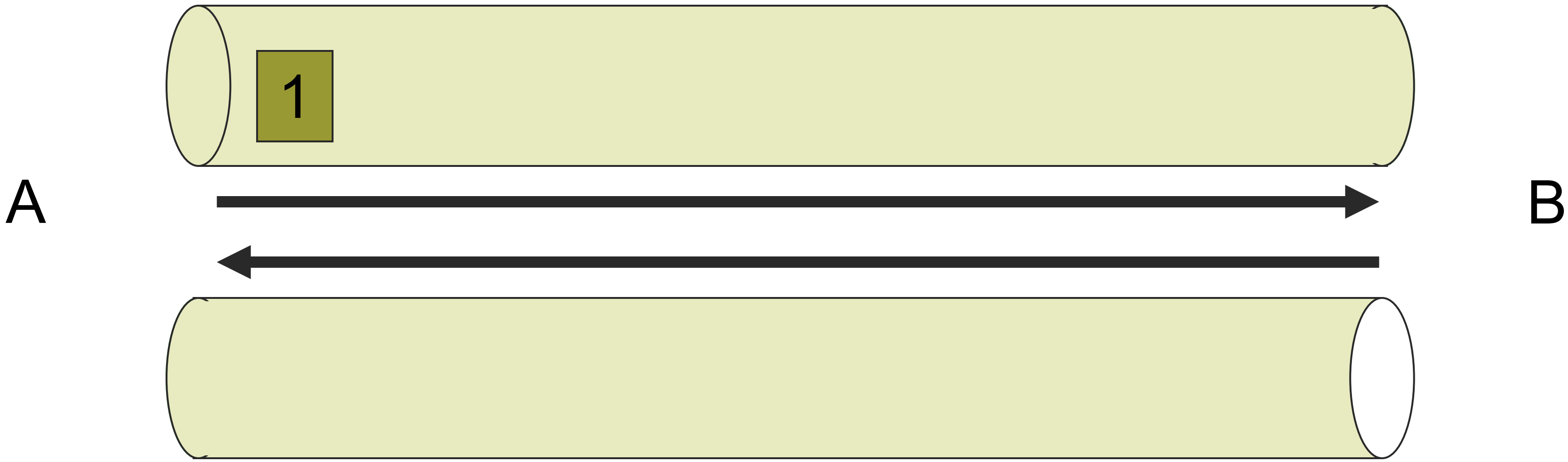
You will mess this up at least once on a HW or exam!

Delay x Bandwidth Product

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

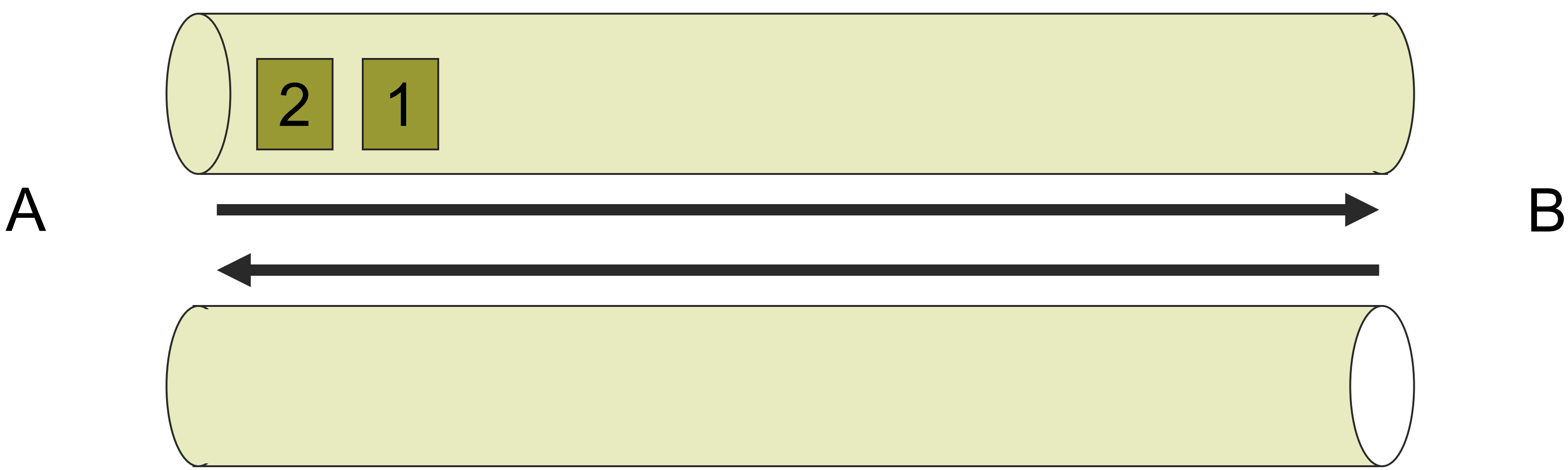
Delay x Bandwidth Product

- 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



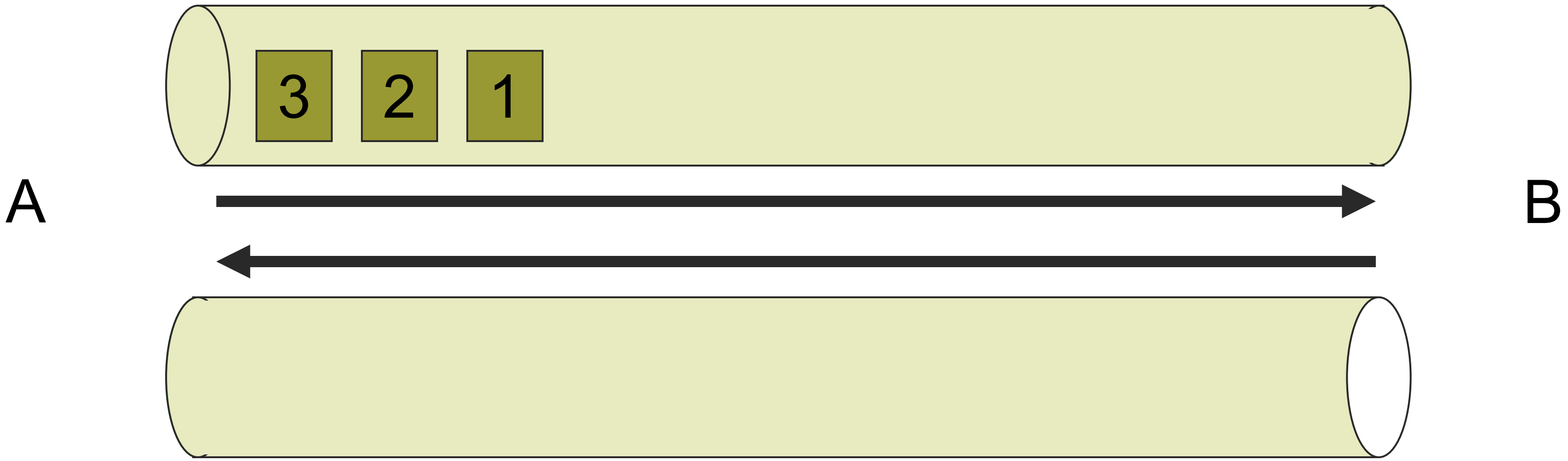
Delay x Bandwidth Product

- 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



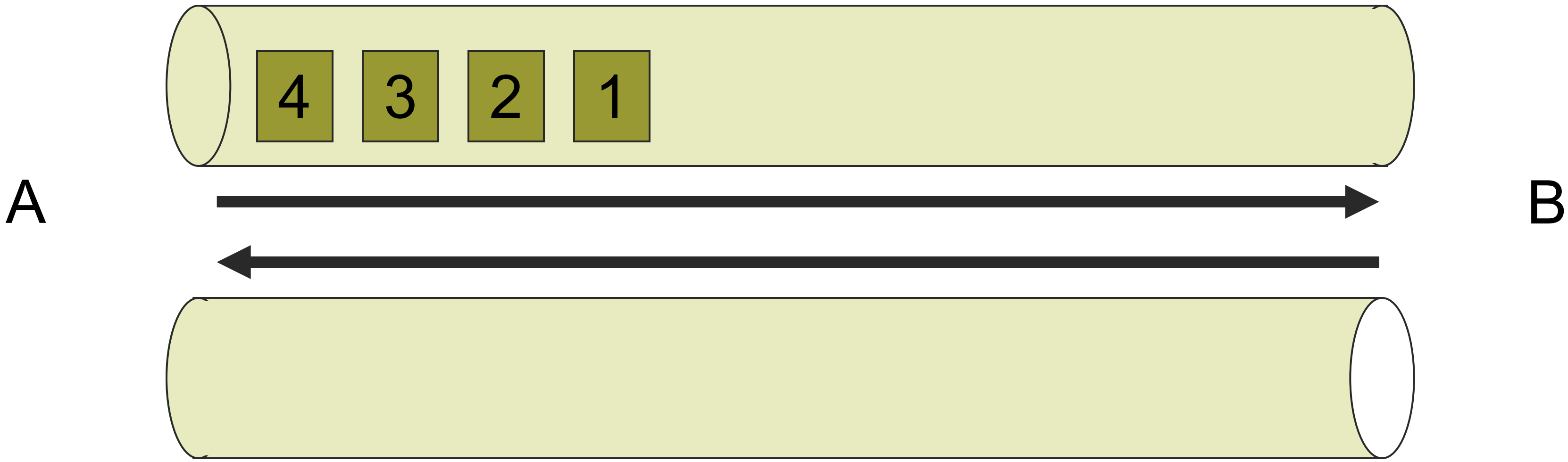
Delay x Bandwidth Product

- 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



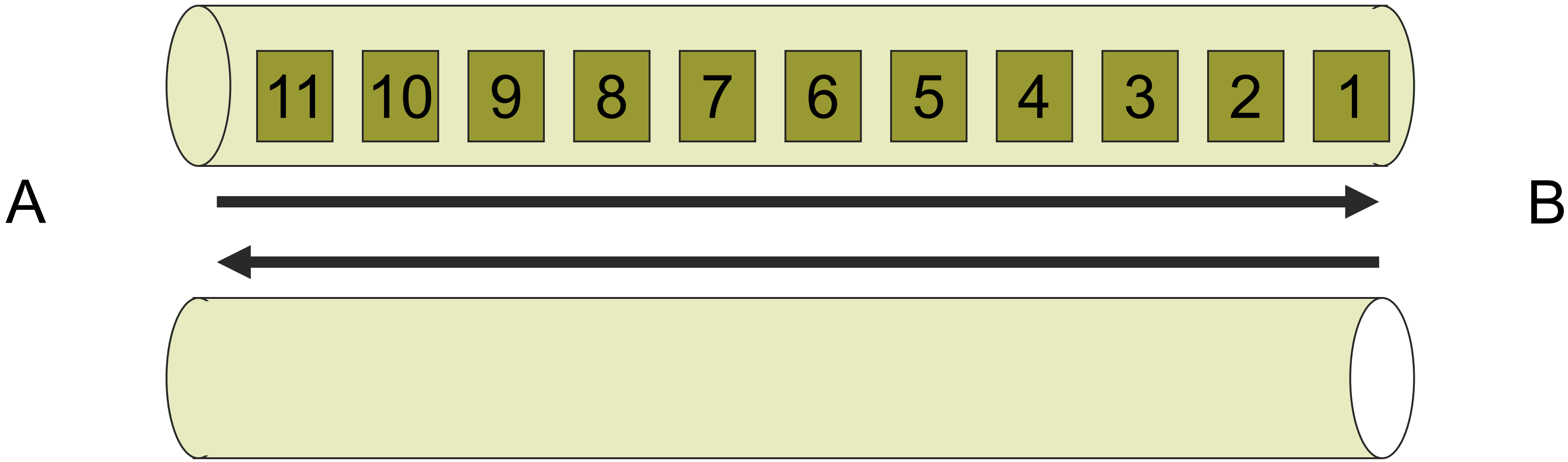
Delay x Bandwidth Product

- 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



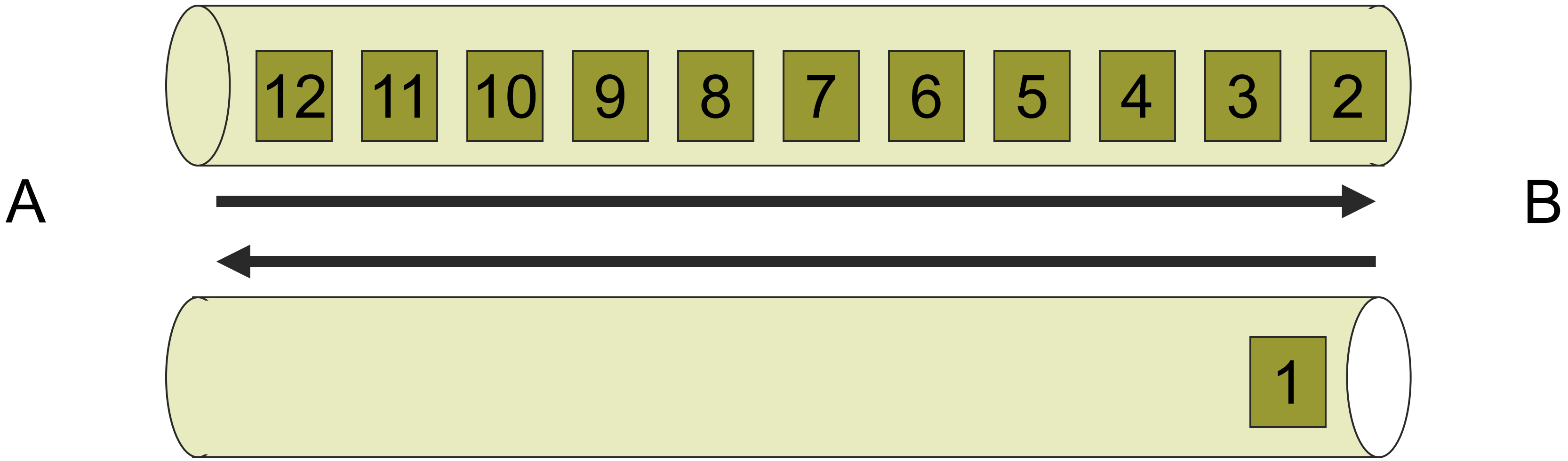
Delay x Bandwidth Product

- 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



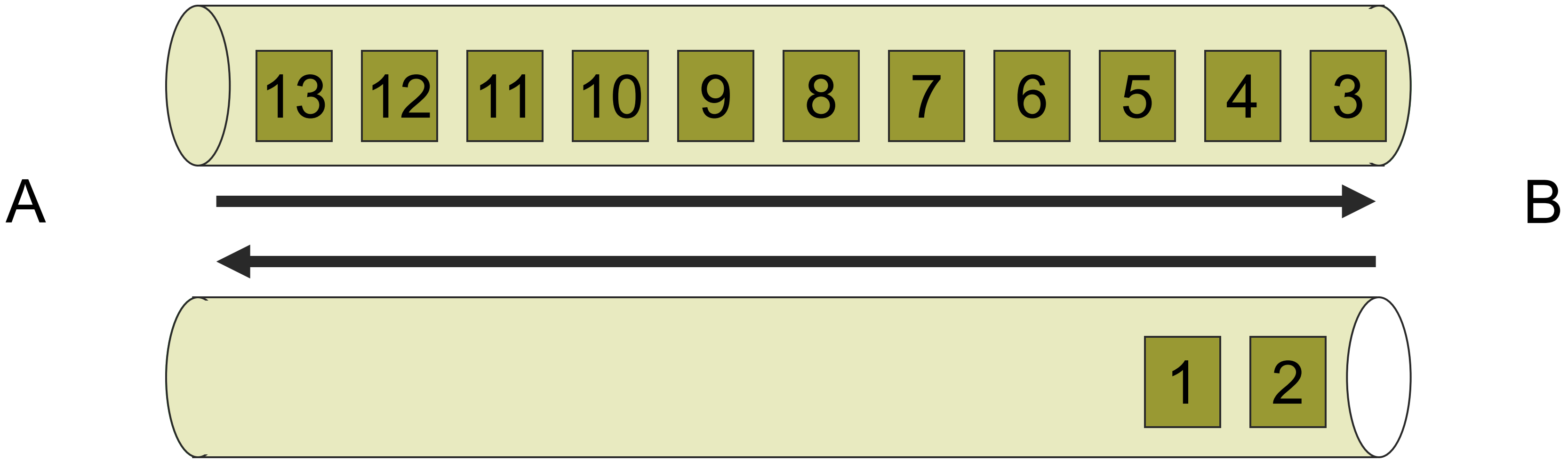
Delay x Bandwidth Product

- 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



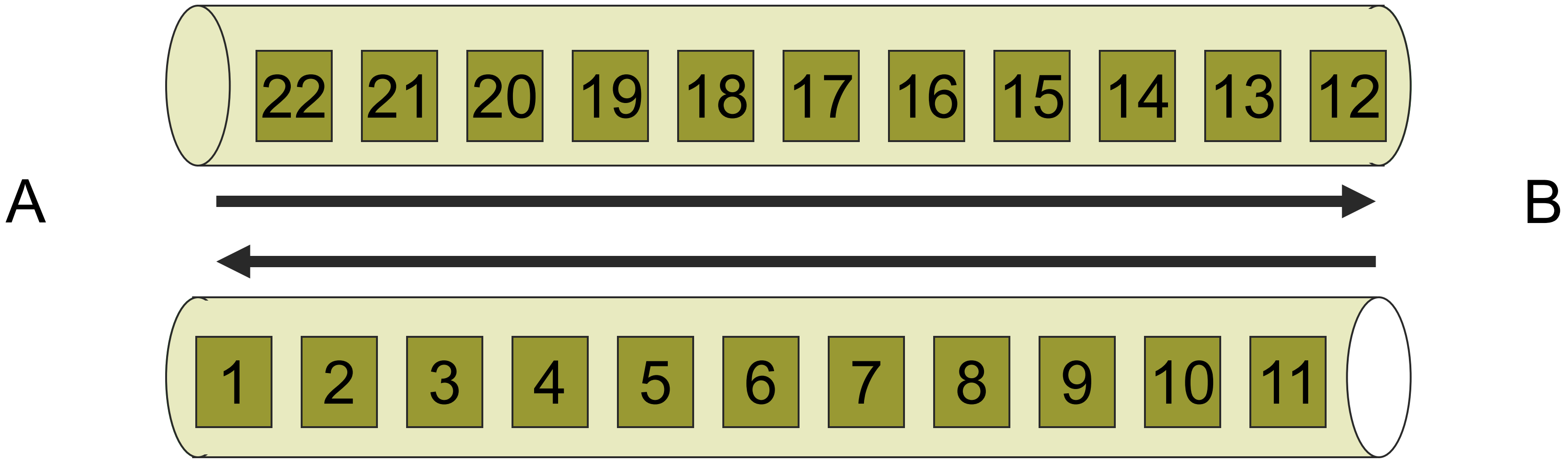
Delay x Bandwidth Product

- 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

    $= (50 \times 10^{-3} \text{ sec}) \times (45 \times 10^{6} \text{ bits/sec})$

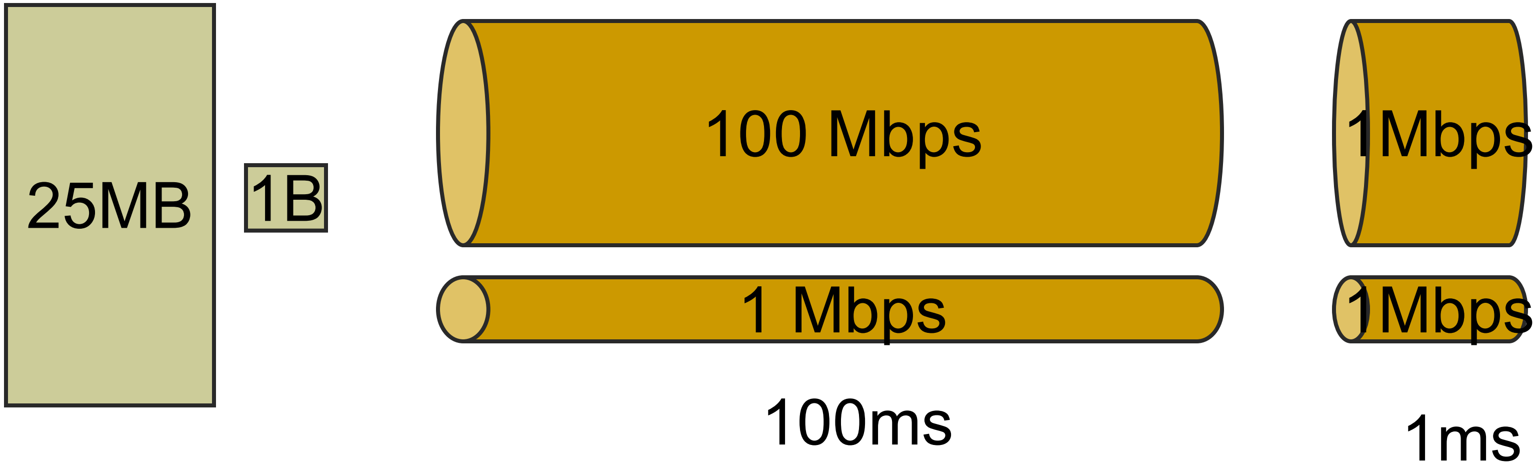
    $= 2.25 \times 10^{6}$ bits

ms

Mbps

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
- Its 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 >= 80ms
  - PC runs at (say) 3 GHz
  - 3,000,000,000 cycles/sec * 0.08 sec = 240,000,000 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
...
```

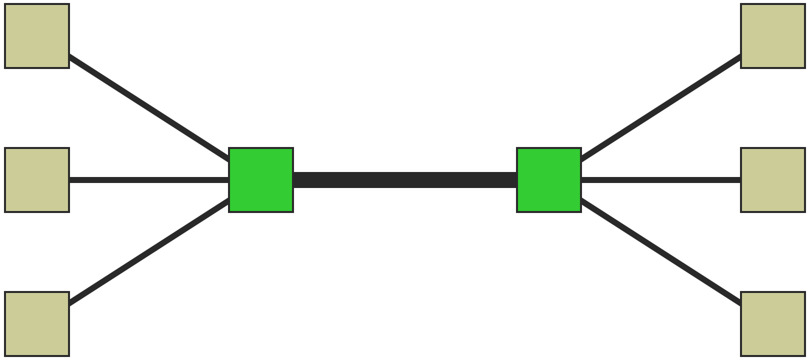
- 500us = 1,500,000 cycles
  - Still a loooooong time…

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



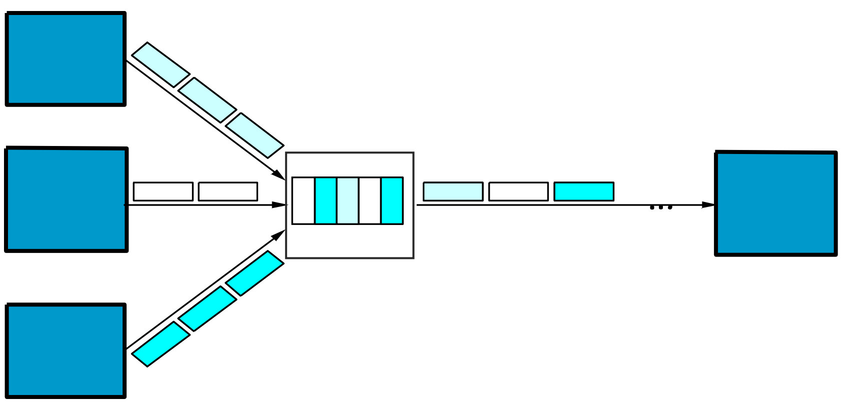
- Common multiplexing strategies
  - (Synchronous) time-division multiplexing (TDM)
  - Frequency-division multiplexing (FDM)

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!

Fundamental Challenge: Things break

- 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 loooong 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

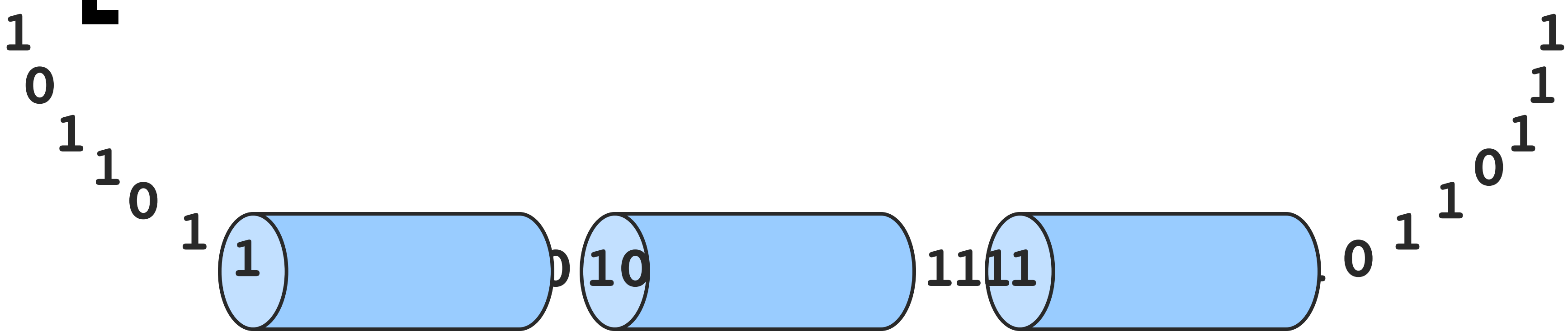
Fundamental Challenge: Security

- 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 Thursday
  - Sockets refresher
- HW 1
  - Available Thursday
- Next topic
  - UNIX network programming
- Next week
  - Technical overview of Internet architecture
  - Data link technologies