CS/ECE 438: Computer Networks

- Prof. Robin Kravets
- Prof. Francis Yan

Was: Communication Networks

Copyright ©: CS 438 Staff, University of Illinois

Computer Networks Are Critical

- Networks form the INTERNET

Web Browsing

Email

File Transfer

Computer Networks Are Critical

- The way we communicate & Interact

Social & Professional Networking

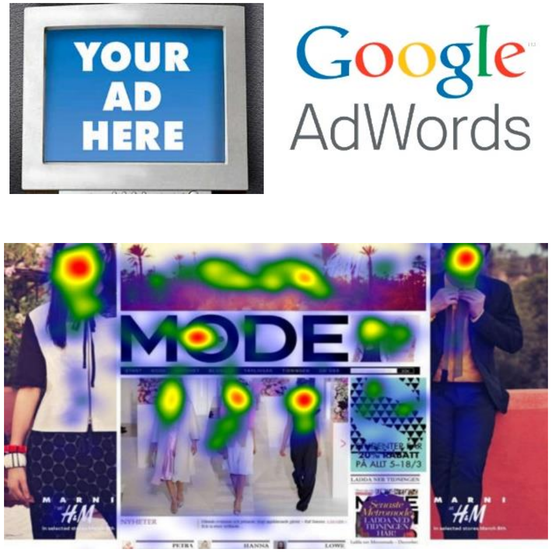
Communication

Copyright ©: CS 438 Staff, University of Illinois

Computer Networks Are Critical

- The way we do business

E-Commerce



Marketing

Cloud Computing

Computer Networks Are Critical

- The way we learn

Online Learning

Online Content

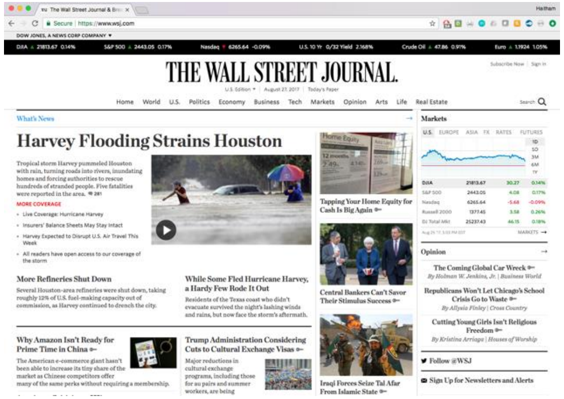
Search Engines

Computer Networks Are Critical

- The way we get news & entertainment

Video Streaming

Online Gaming

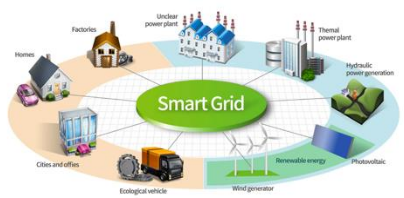


News

Copyright ©: CS 438 Staff, University of Illinois

Computer Networks Are Critical

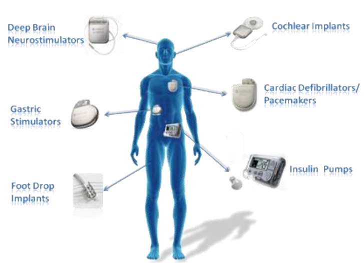
- Many more emerging applications



Networked Infrastructure

Smart Homes

Finance/Trading Networks



Networked Medical Implants

Networked Self Driving Cars

Augmented and Virtual Reality

Copyright ©: CS 438 Staff, University of Illinois

But building Networks Is Challenging

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

Copyright ©: CS 438 Staff, University of Illinois

Course Contents

- Introduction to 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

Copyright ©: CS 438 Staff, University of Illinois

Course Information

- Instructors

Prof. Francis Yan
fyy@illinois.edu

Prof. Robin Kravets
rhk@illinois.edu

Copyright ©: CS 438 Staff, University of Illinois

Course Information

- TAs
  - Martin Chong
  - Xiaojuan Ma
  - Nishant Sheikh

- Office Hours
  - Times and locations will be posted on the website

Copyright ©: CS 438 Staff, University of Illinois

Course Communication

- Use Campuswire for ALL class related communication
  - All class questions - there are no bad questions
  - All students can benefit from the discussion
  - You can benefit from answering questions
    - 1% bonus point (and a ⭐) for selected students based on semester-long contributions on Campuswire
  - Be fair and respectful
  - Do not post complete or partial solutions to any assignments on Campuswire

  - For personal or sensitive matters, please email both instructors

Course Information

- Textbook (recommended, not required)
  - Computer Networks: A Systems Approach, by Peterson and Davie
- Supplemental Text books
  - UNIX Network Programming, by Stevens

Copyright ©: CS 438 Staff, University of Illinois

Prerequisites

- Operating Systems Concepts
  - CS 341 or ECE 391 or equivalent
    - Threads, memory management, sockets
- C or C++ Programming
  - Preferably Linux
- Probability and Statistics

Copyright ©: CS 438 Staff, University of Illinois

Grading Policy

- Homework 10%
  - 4 homework assignments
- Programming Projects 40%
  - MP0 2%, MP1 10%, MP2 14%, MP3 14%
- Midterm Exam 20%
  - March 10, 7pm – 9pm
- Final Exam 30%
  - May 13 7pm – 10pm

Copyright ©: CS 438 Staff, University of Illinois

Homework and Projects

- Homework
  - 4 HW (2.5%)
  - Due Fridays 11:59pm
  - Solutions posted after 72 hours
- Late policy
  - 10% off per day late up to 72 hours
  - 7 free late days over all HW and MPs
  - No HW or MP questions in office hours or on Campuswire after deadlines
- Projects
  - Due Sundays 11:59pm
  - MP0 and MP1
    - Solo, no AI assistance
  - MP2 and MP3
    - 2 person teams, AI assistance allowed
- Validation
  - Selected groups will be asked to explain their solutions

Copyright ©: CS 438 Staff, University of Illinois

Regrades

- Within one week of posting of grades for a homework, MP or exam
- Regrades must be submitted through Campuswire
  - Please do not write on your exam

Academic Honesty

- Your work in this class must be your own.
- If students are found to have cheated or circumvented the guidelines of the course (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

Copyright ©: CS 438 Staff, University of Illinois

Is using AI tools cheating in a programming class?

- We are all trying to find a balance for this
  - If we say no AI, then using AI is cheating
  - If we say you can use AI, follow our guidelines

Copyright ©: CS 438 Staff, University of Illinois

Graduate Students

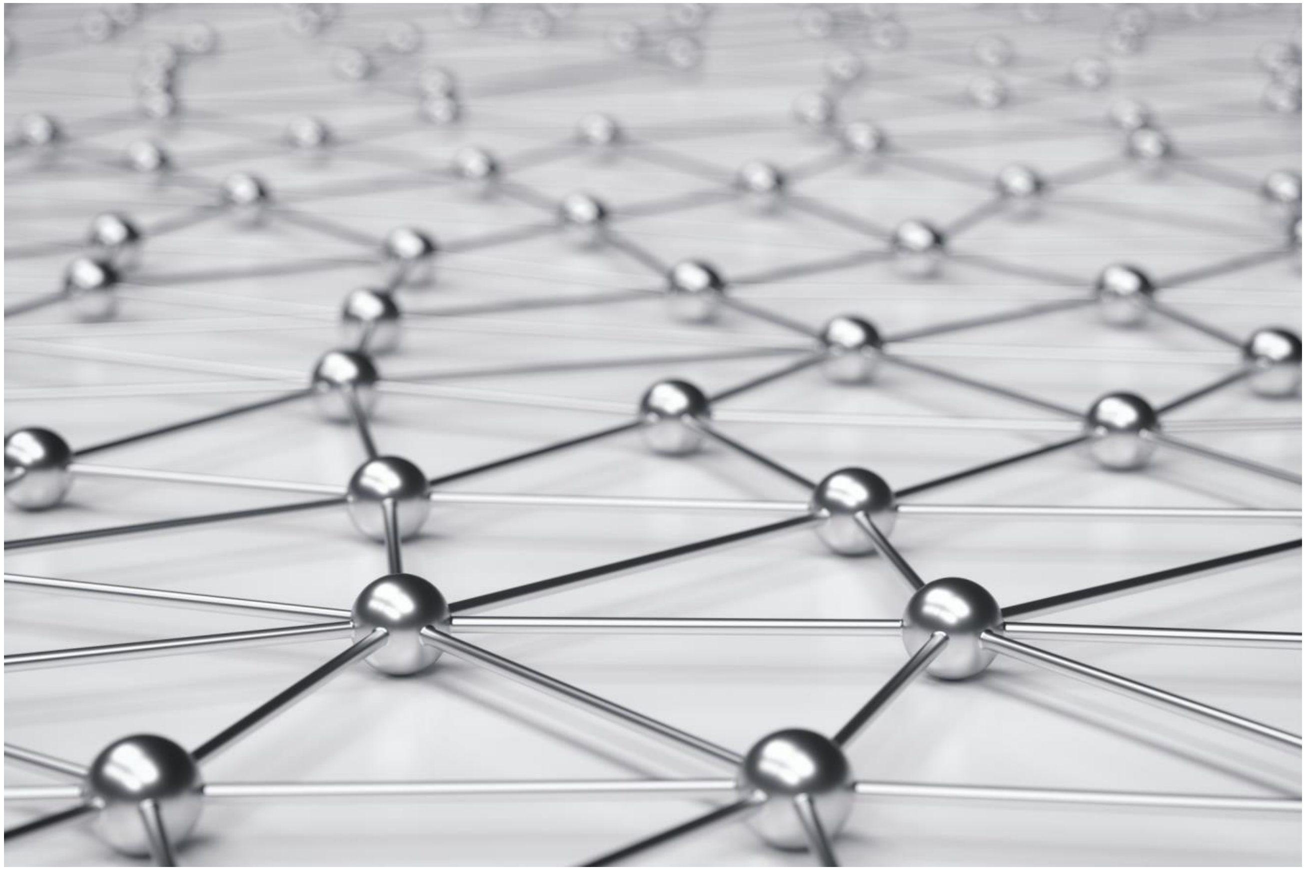
- Graduate students MAY take this class for 4 credits
  - Graduate students
    - 1-3 group members
    - Complete a mini project in a networking area of your choice
    - Project proposal due week 5
    - Present your project at a poster session at the end of the semester
    - Submit a write-up describing your project (4-pages per group member) due last day of class
  - Undergraduates
    - If you are interested in networking research, please contact us

Complete Schedule

- See class webpage
  - https://uiuc-cs438.github.io/sp2026/
- 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

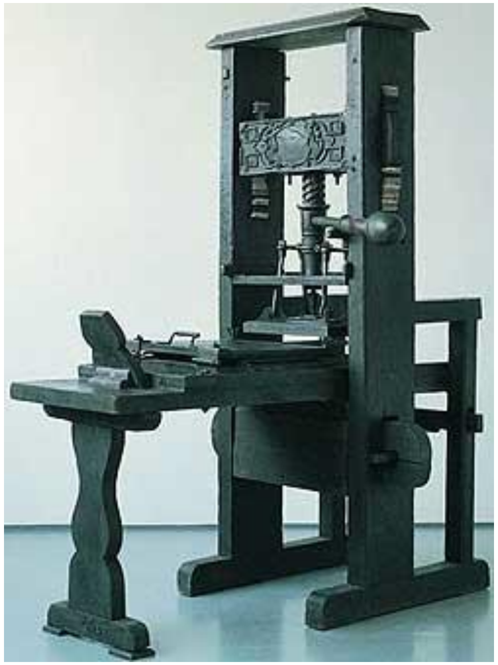
Copyright ©: CS 438 Staff, University of Illinois

Let’s gets started!

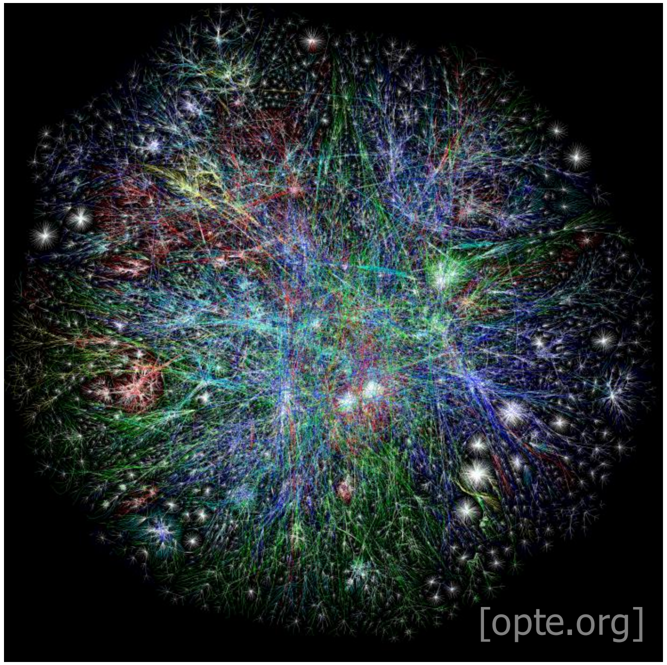


Copyright ©: CS 438 Staff, University of Illinois

What do these two things have in common?



First printing press



The Internet

Both lowered the cost of distributing information and changed human society

Copyright ©: CS 438 Staff, University of Illinois

A Brief History of the Internet

Copyright ©: CS 438 Staff, University of Illinois

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

Copyright ©: CS 438 Staff, University of Illinois

Circuit switching

1920s

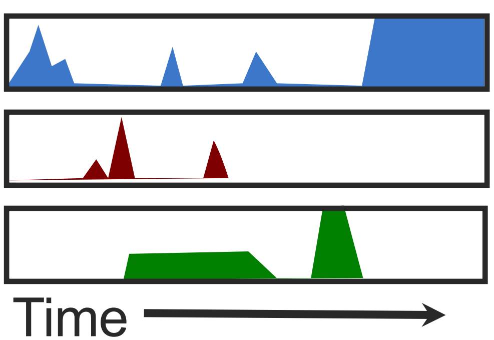
1967

Copyright ©: CS 438 Staff, University of Illinois

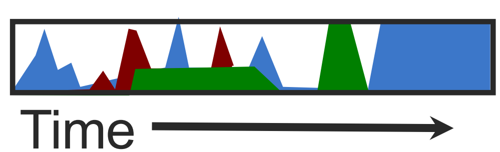
1961-64: Packet switching

- 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)

Circuit switching



Packet switching



Kleinrock

Baran

Copyright ©: CS 438 Staff, University of Illinois

1961-64: Packet switching

Circuit Switching	Datagram packet switching

Copyright ©: CS 438 Staff, University of Illinois

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	

Copyright ©: CS 438 Staff, University of Illinois

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

Copyright ©: CS 438 Staff, University of Illinois

1965: First computer network

- Lawrence Roberts and Thomas Merrill
  - Connect a TX-2 at MIT to a Q-32 in Santa Monica, CA
- Connected with telephone line
  - it works, but
  - It's inefficient and expensive
  - Confirming motivation for packet switching

Roberts

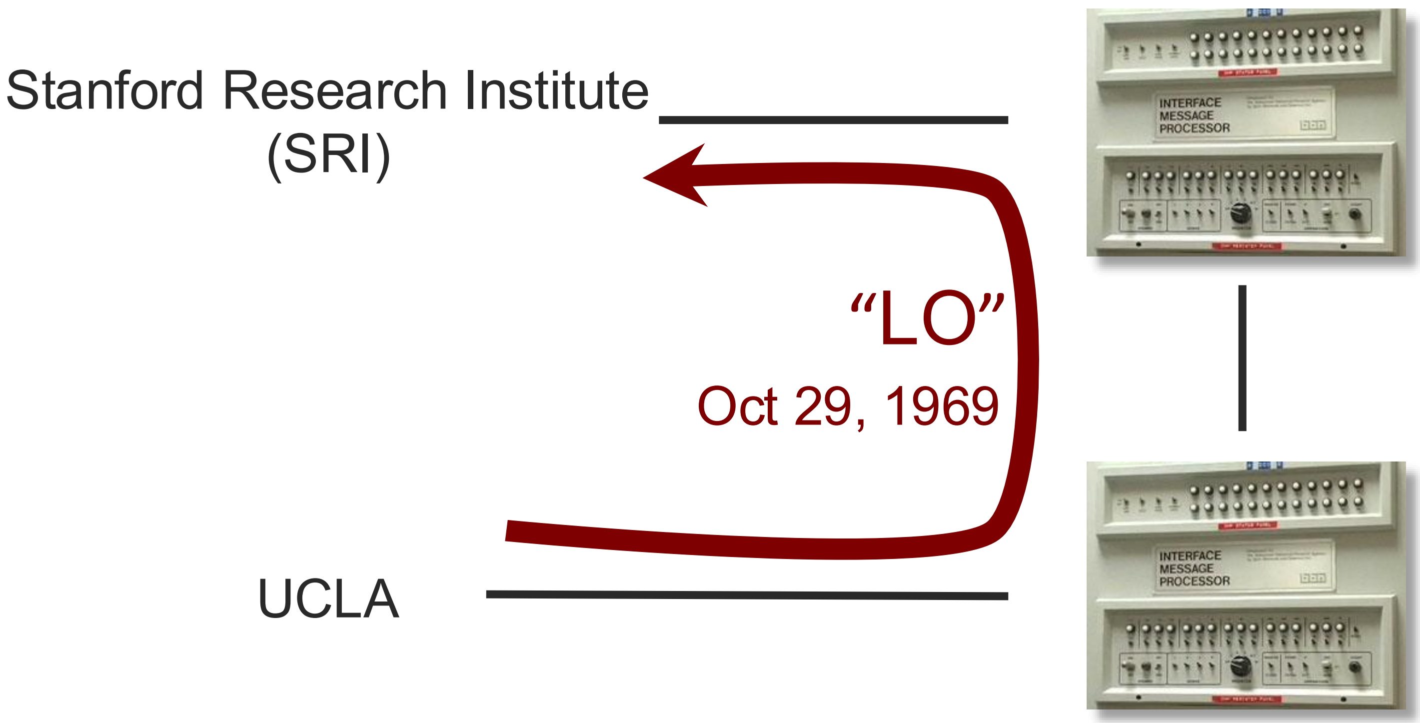
The ARPANET begins

- 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

Copyright ©: CS 438 Staff, University of Illinois

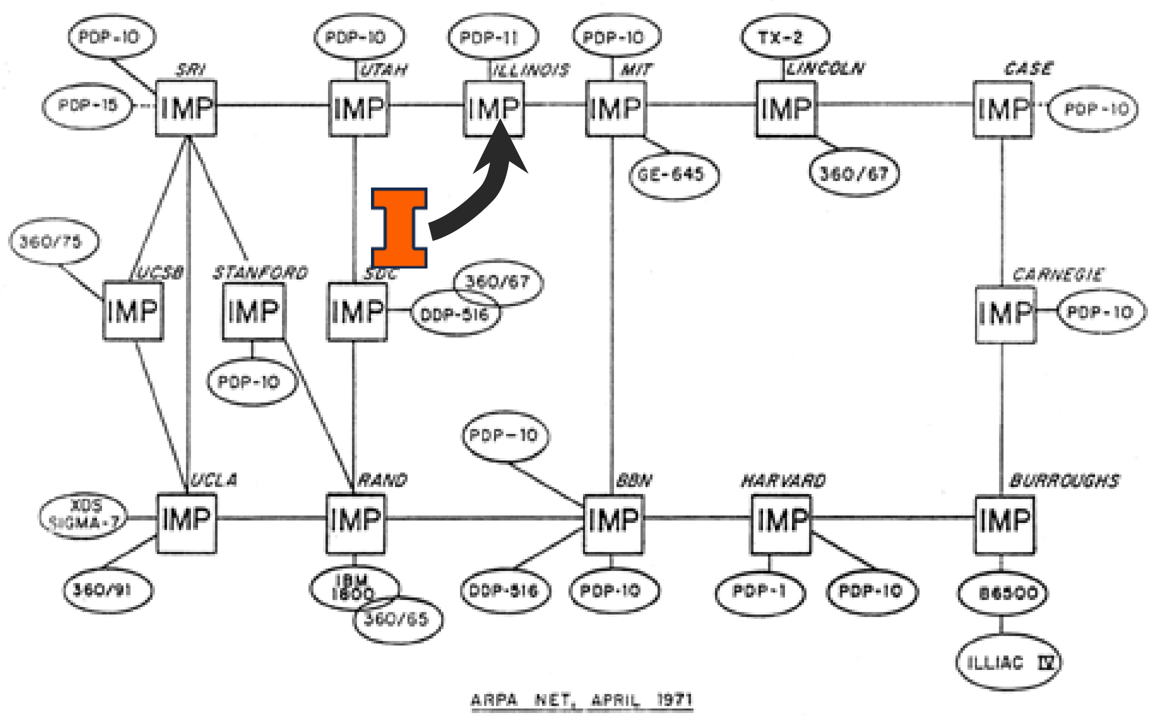
ARPANET comes alive



Copyright ©: CS 438 Staff, University of Illinois

ARPANET grows

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

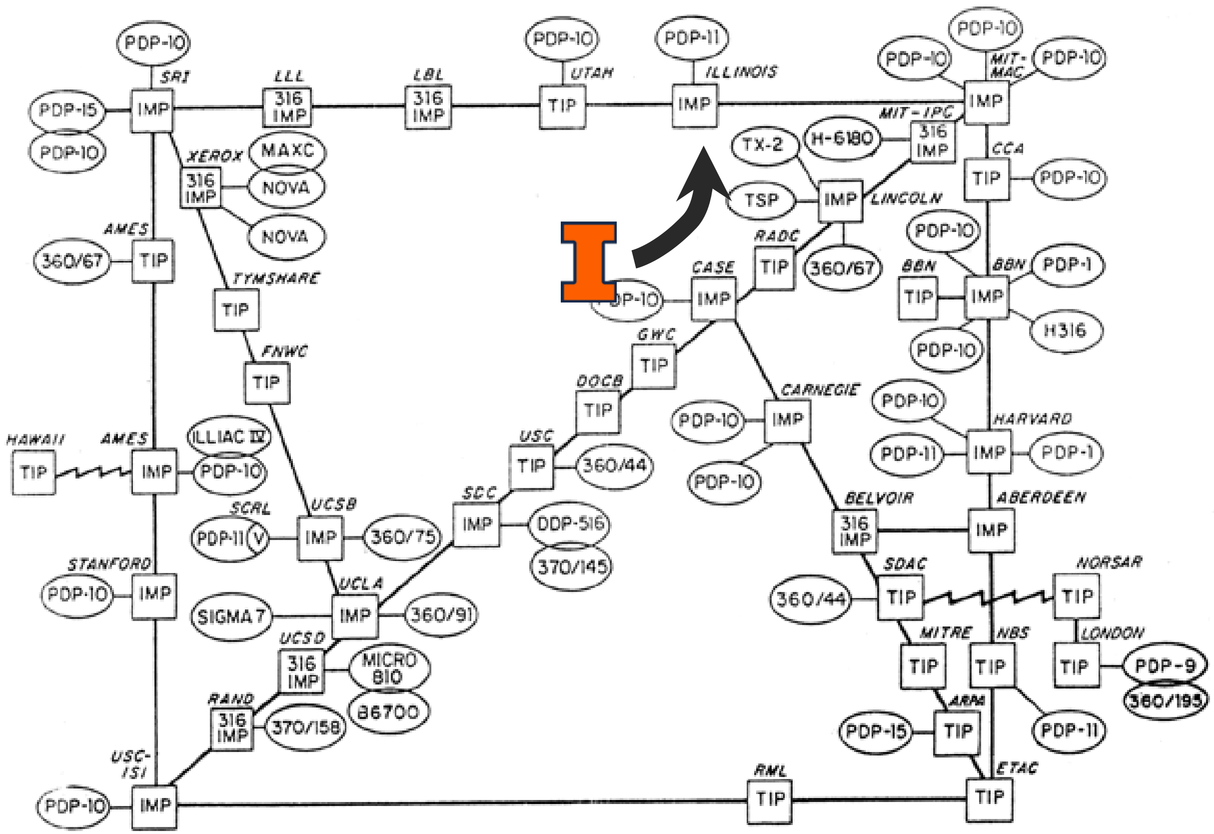


ARPANET, April 1971

Copyright ©: CS 438 Staff, University of Illinois

And grows …

ARPA NETWORK, LOGICAL MAP, SEPTEMBER 1973



77 nodes

How many do we have today?

Over 75 B

ARPANET to Internet

- 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

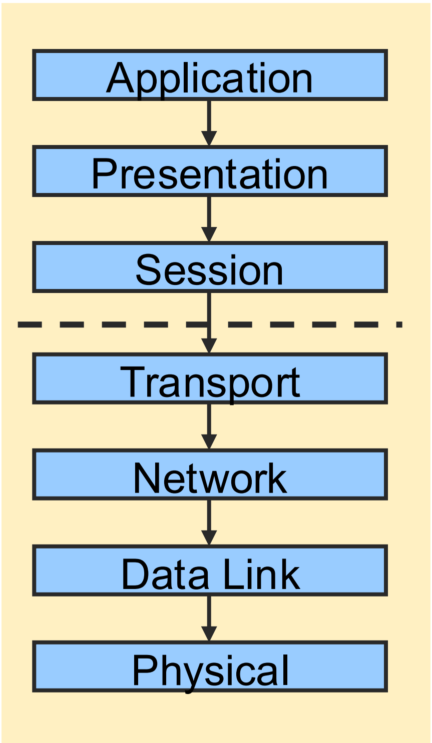
2004 Turing Award!

Cerf

Kahn

Copyright ©: CS 438 Staff, University of Illinois

TCP/IP deployment

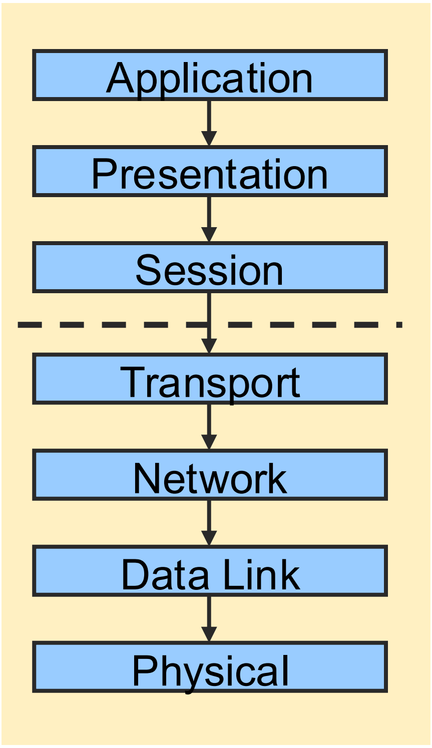


OSI Reference Model's layers

- 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

Copyright ©: CS 438 Staff, University of Illinois

OSI Protocol Stack

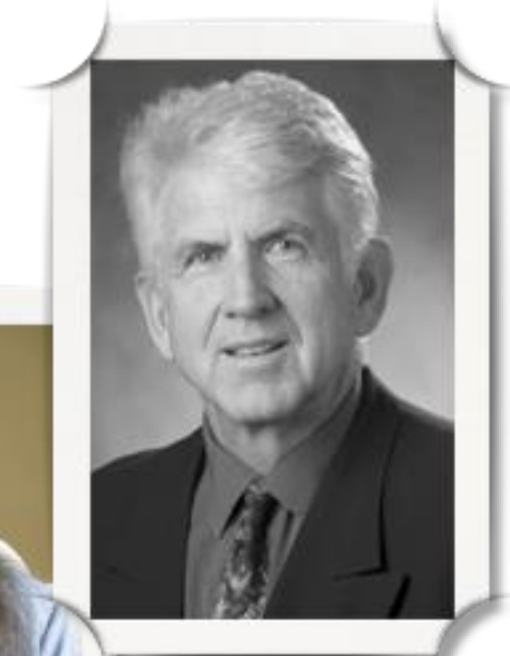


- Application: Application specific protocols
- Presentation: Format of exchanged data
- Session: Name space for connection mgmt

- 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

Copyright ©: CS 438 Staff, University of Illinois

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

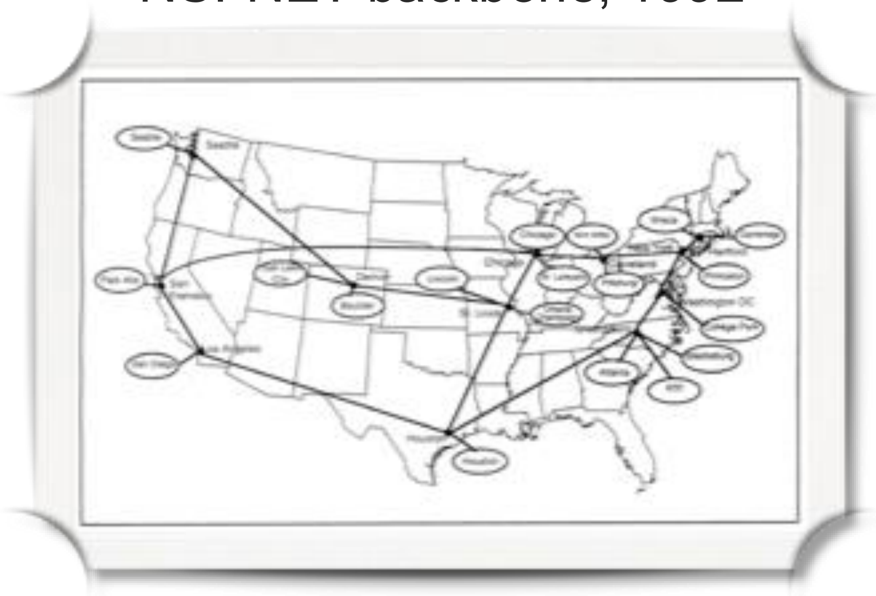
Mockapetris

Partridge

NSFNET

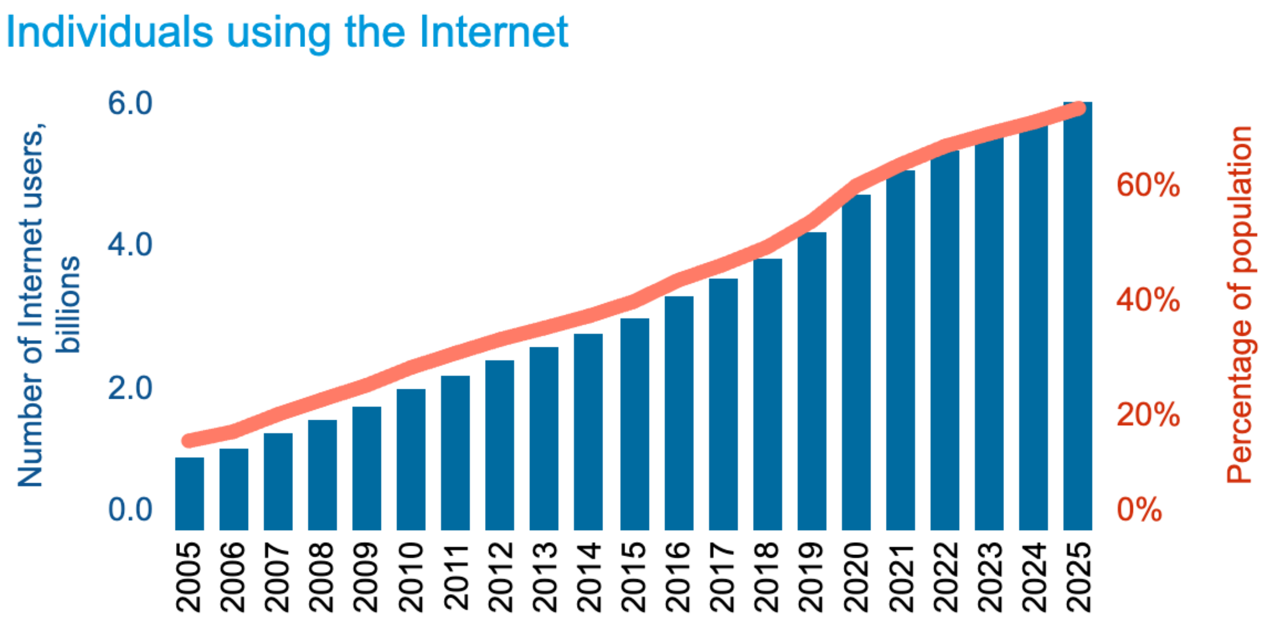
- 1984: NSFNET for US higher education
  - Serve many users, not just one field
  - Encourage development of private infrastructure (e.g., initially, backbone required to be used for Research and Education)
  - Stimulated investment in commercial long-haul networks
- 1990: ARPANET ends
- 1995: NSFNET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

NSFNET backbone, 1992



Explosive growth!

In users

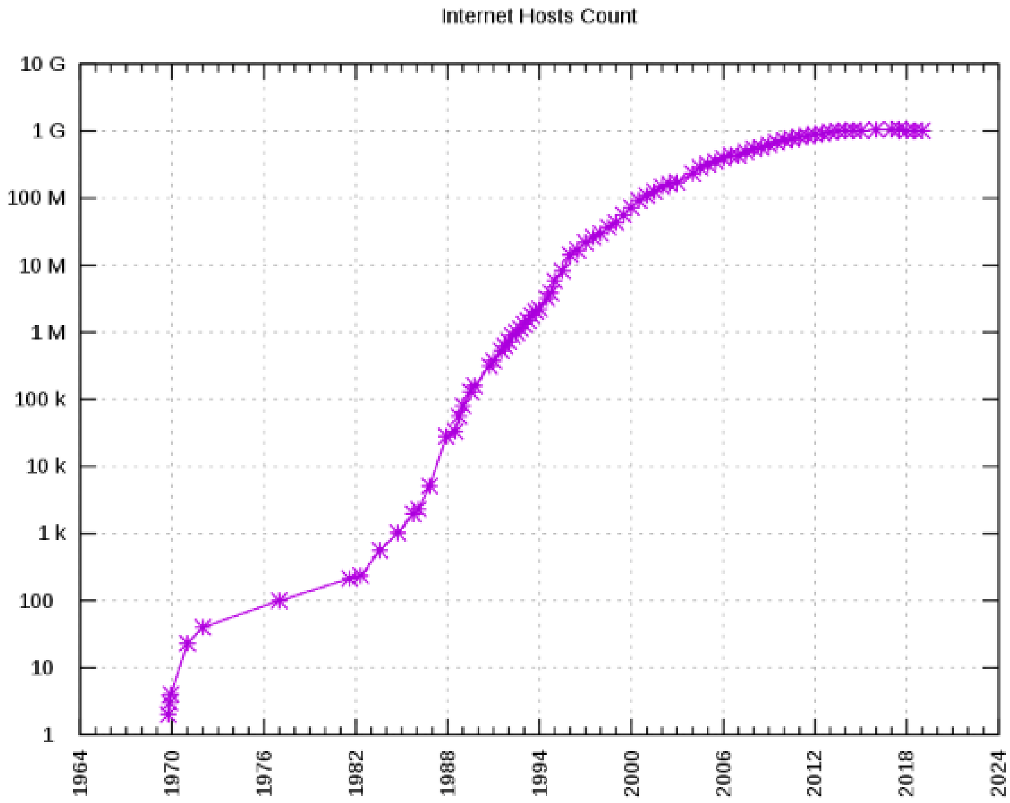


Source: ITU

Copyright ©: CS 438 Staff, University of Illinois

Explosive growth!

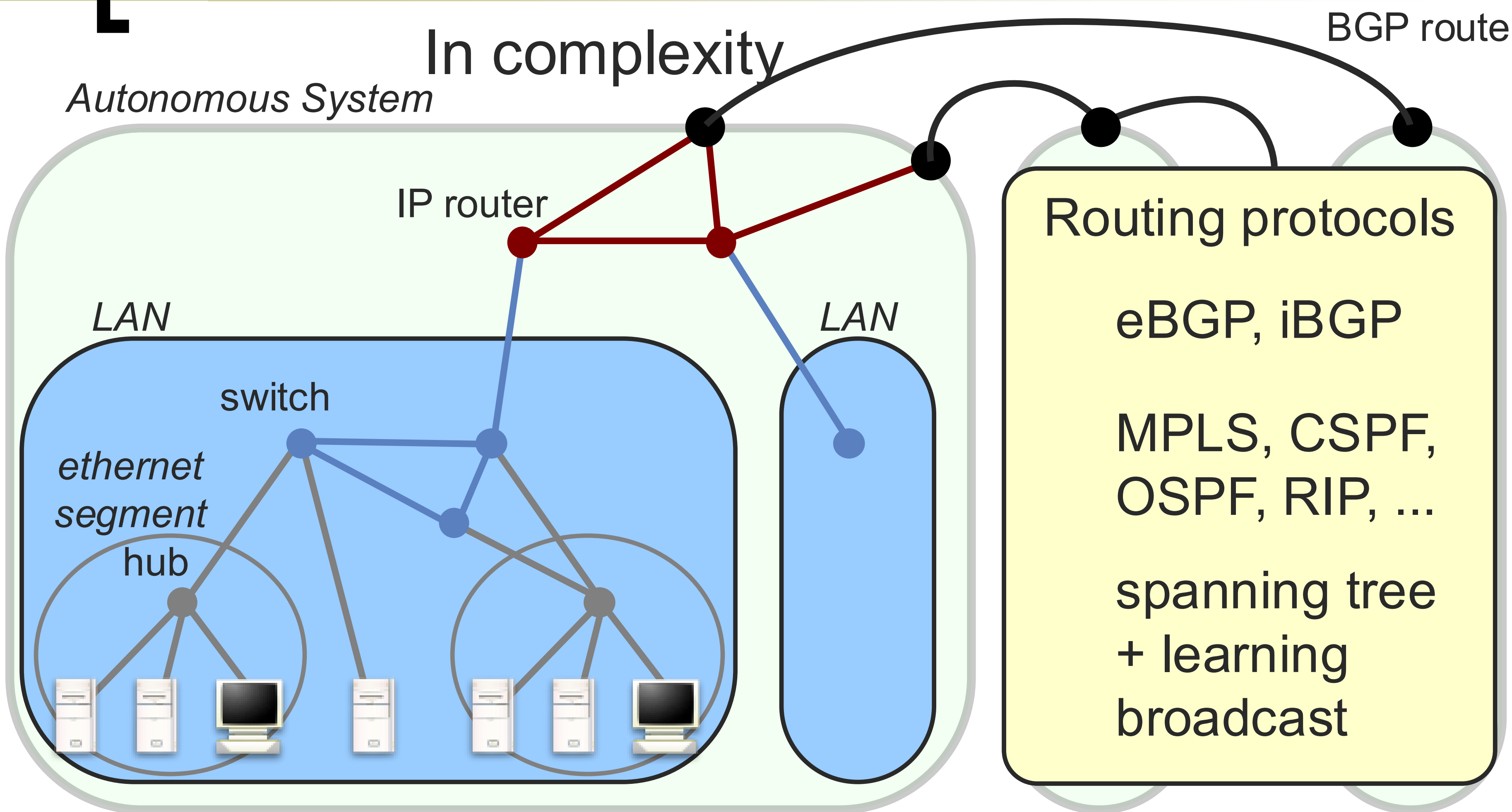
In hosts



Copyright ©: CS 438 Staff, University of Illinois

Explosive growth!

In complexity

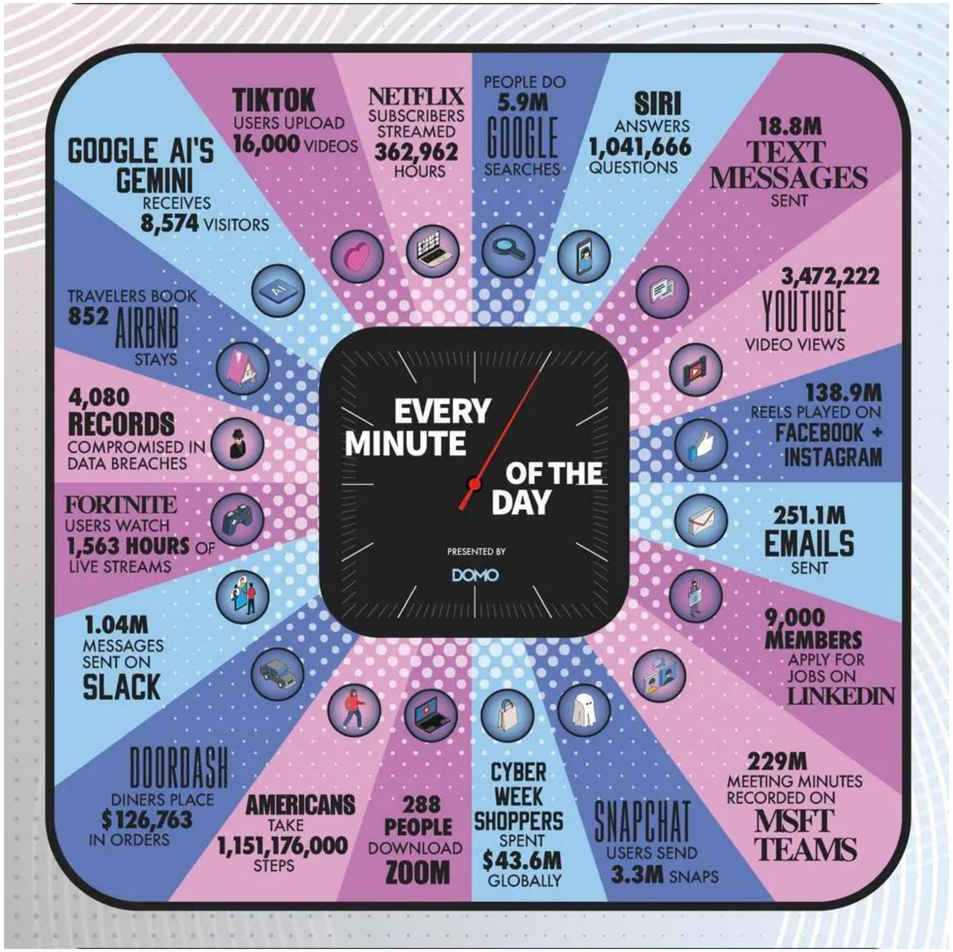


Copyright ©: CS 438 Staff, University of Illinois

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!

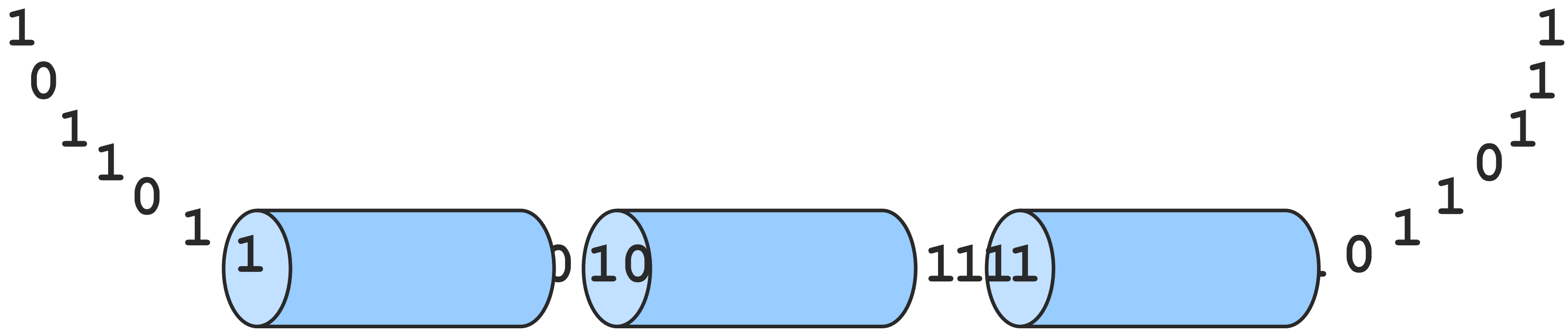


- In just one minute in 2025:
  - **694M** Spotify songs streamed
  - **231M** emails sent
  - **6.3M** Google searches conducted
  - **3.47M** YouTube videos watched
  - **625M** TikTok videos watched
  - **$43.6M** spent online during peak shopping times
  - **174K** apps downloaded
  - **66K** Instagram photos shared
  - **2.1M** active Facebook users

Copyright ©: CS 438 Staff, University of Illinois

Building Networks Is Challenging

Why is Networking Challenging



That’s it! ...right?

Building Networks is Challenging

- Networks are large and complex
  - Tremendous scale distributed across the globe
  - Rapid growth
  - Run by parties with competing interests
- Networks are hard to change & fix
  - Complex intertwining dependencies across protocols/systems, networks
  - Cannot reboot the Internet
- Networks are under continuous attack
  - Network crime is a trillion-dollar industry

Copyright ©: CS 438 Staff, University of Illinois

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

Copyright ©: CS 438 Staff, University of Illinois

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!)

Copyright ©: CS 438 Staff, University of Illinois

Fundamental Challenge: Speed of Light

Copyright ©: CS 438 Staff, University of Illinois

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)

Copyright ©: CS 438 Staff, University of Illinois

Performance

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)

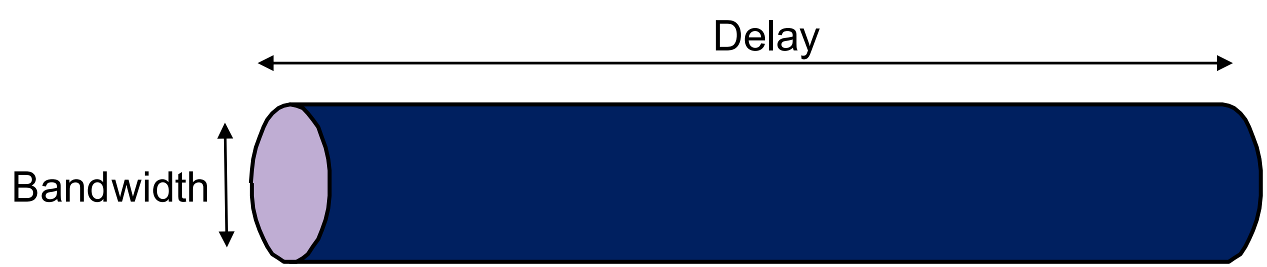
Why?

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

Copyright ©: CS 438 Staff, University of Illinois

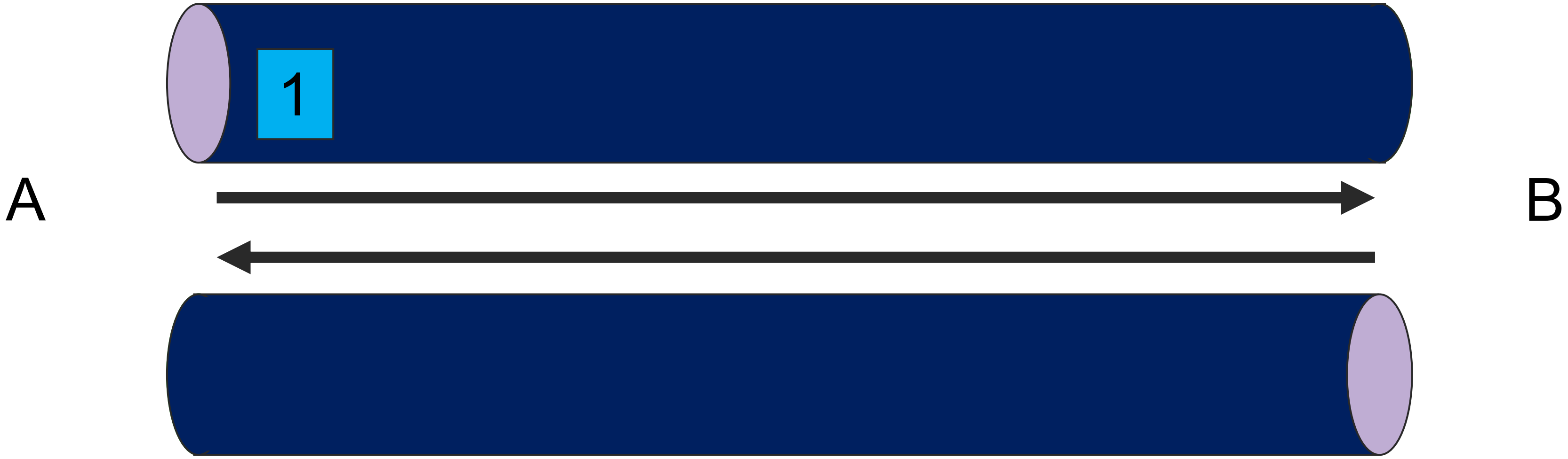
Delay x Bandwidth Product

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



Delay x Bandwidth Product

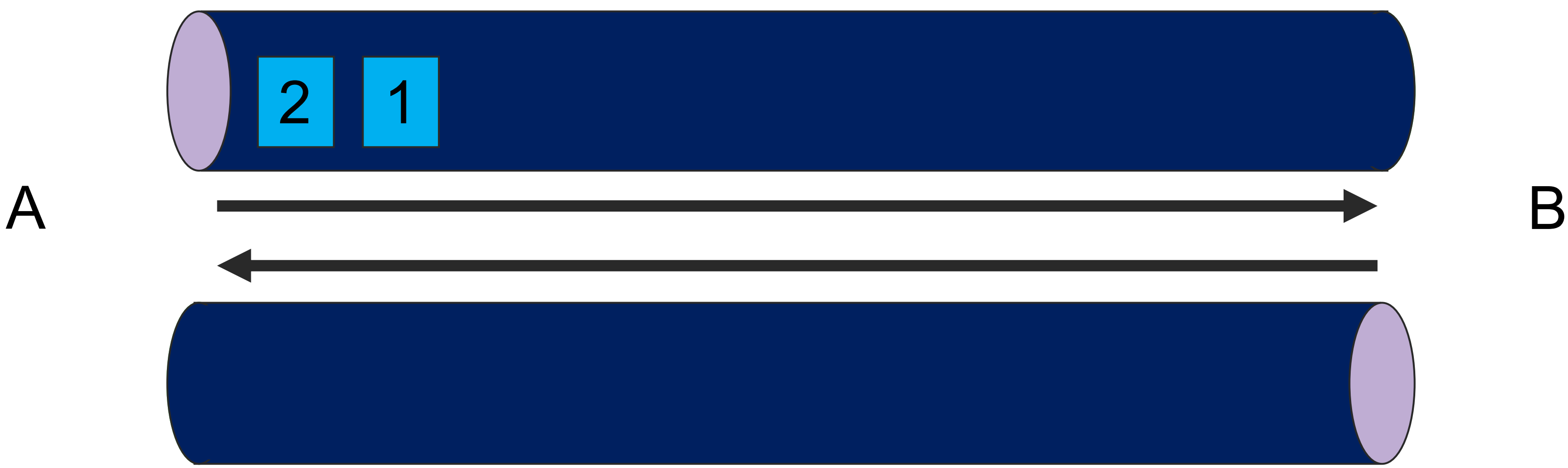
- Bandwidth x delay product
  - How many bits the sender must transmit before the first bit arrives at the receiver if the sender keeps the pipe full
  - Takes another one-way latency to receive a response from the receiver



Copyright ©: CS 438 Staff, University of Illinois

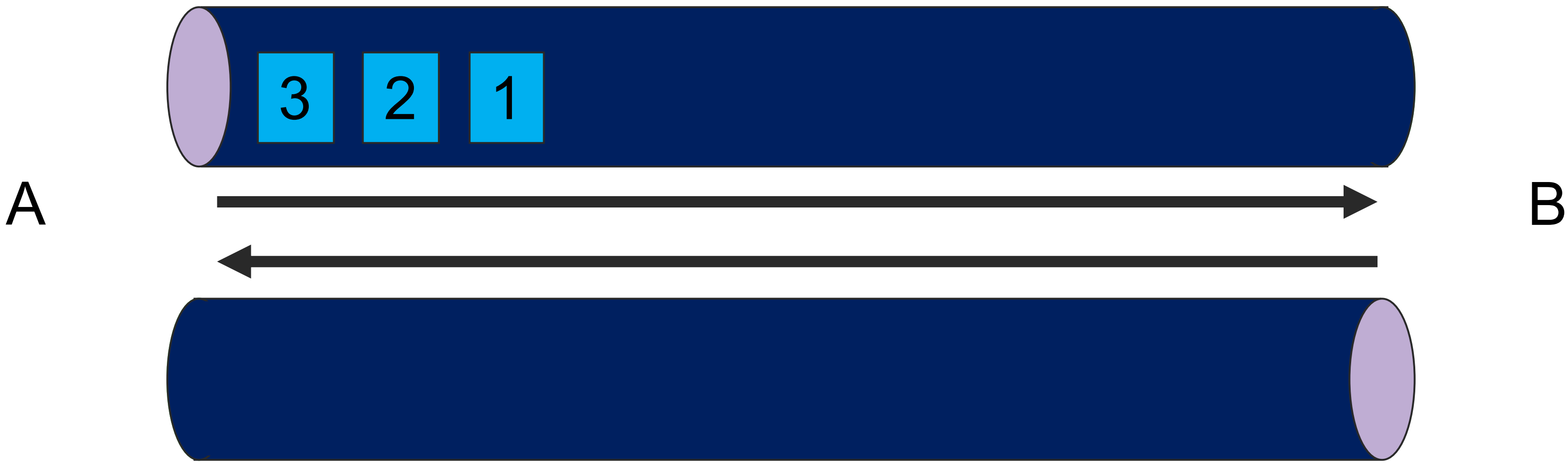
Delay x Bandwidth Product

- 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

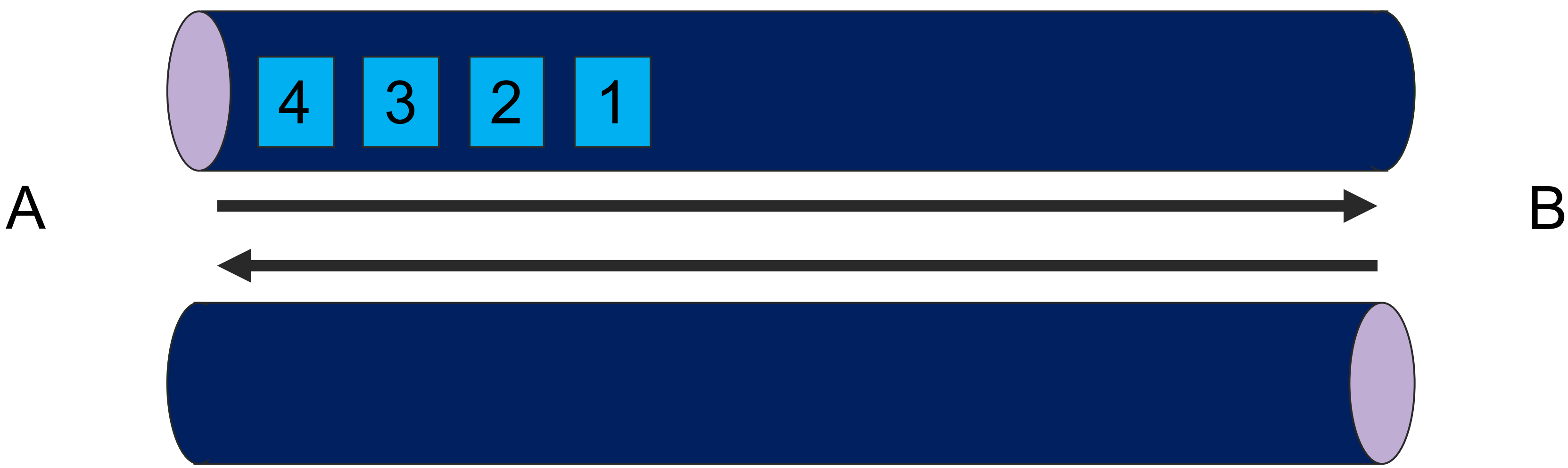
- Bandwidth x delay product
  - How many bits the sender must transmit before the first bit arrives at the receiver if the sender keeps the pipe full
  - Takes another one-way latency to receive a response from the receiver



Copyright ©: CS 438 Staff, University of Illinois

Delay x Bandwidth Product

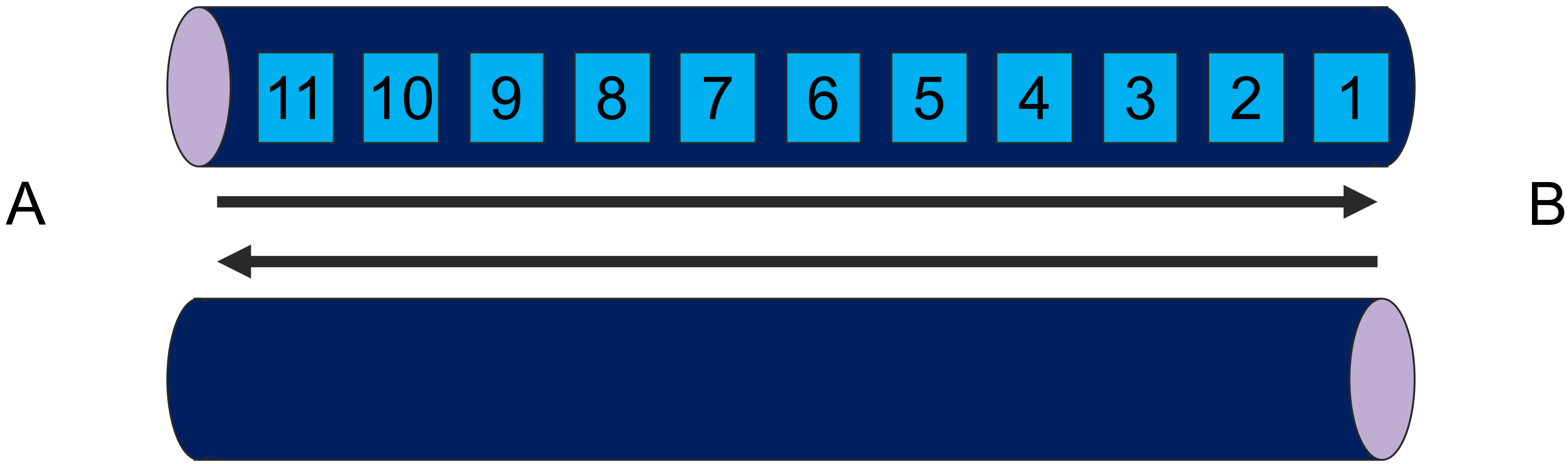
- Bandwidth x delay product
  - How many bits the sender must transmit before the first bit arrives at the receiver if the sender keeps the pipe full
  - Takes another one-way latency to receive a response from the receiver



Copyright ©: CS 438 Staff, University of Illinois

Delay x Bandwidth Product

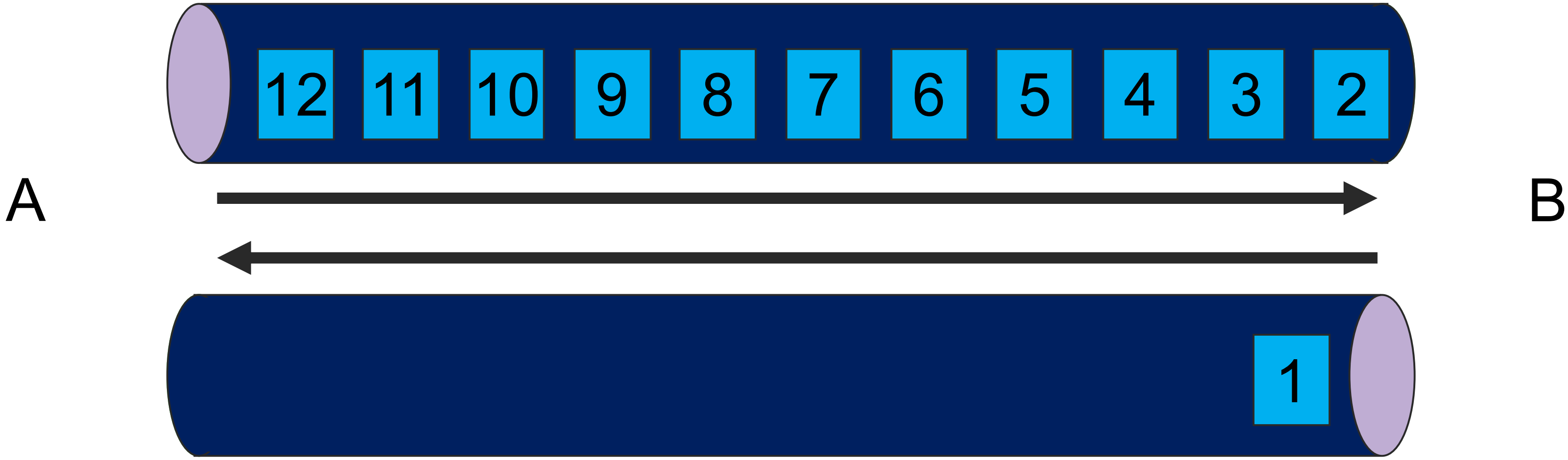
- Bandwidth x delay product
  - How many bits the sender must transmit before the first bit arrives at the receiver if the sender keeps the pipe full
  - Takes another one-way latency to receive a response from the receiver



Copyright ©: CS 438 Staff, University of Illinois

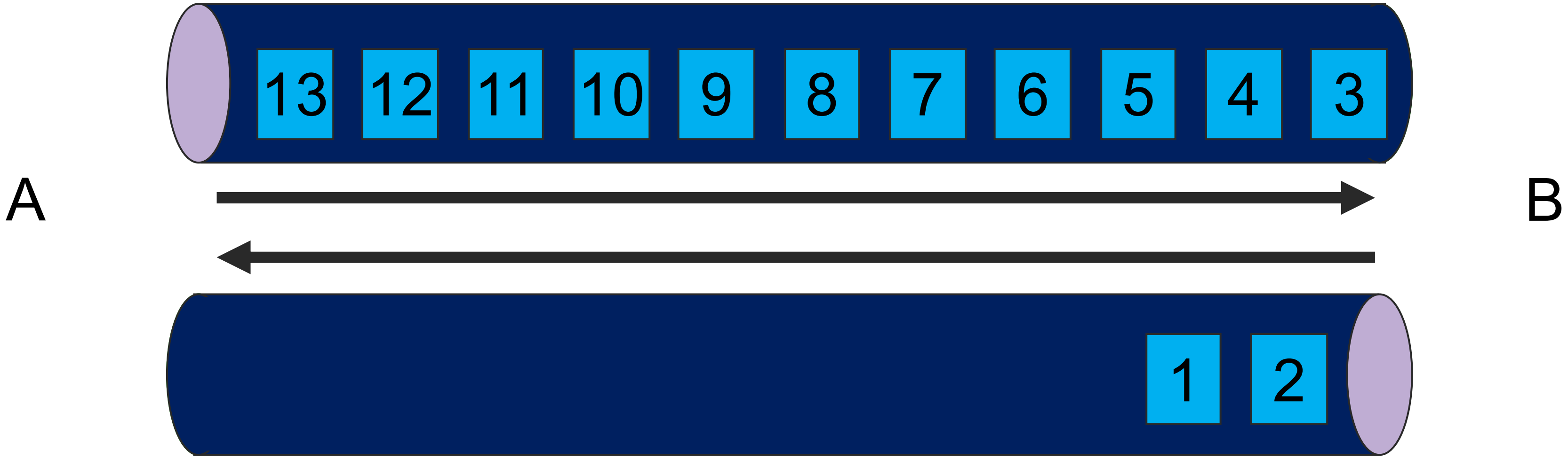
Delay x Bandwidth Product

- 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

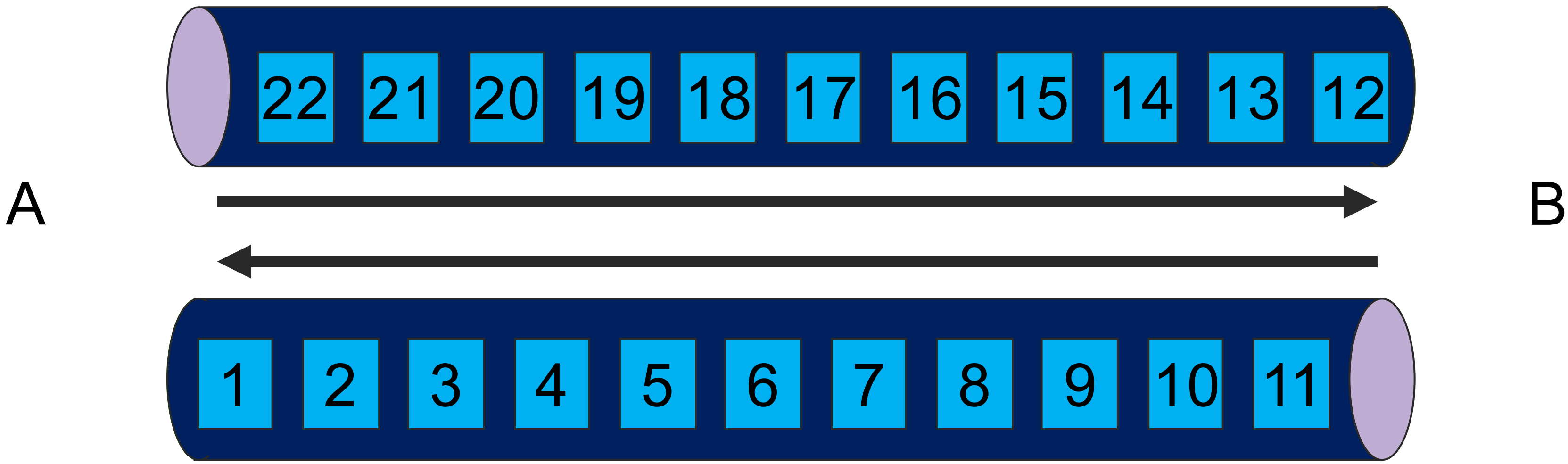
- Bandwidth x delay product
  - How many bits the sender must transmit before the first bit arrives at the receiver if the sender keeps the pipe full
  - Takes another one-way latency to receive a response from the receiver



Copyright ©: CS 438 Staff, University of Illinois

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

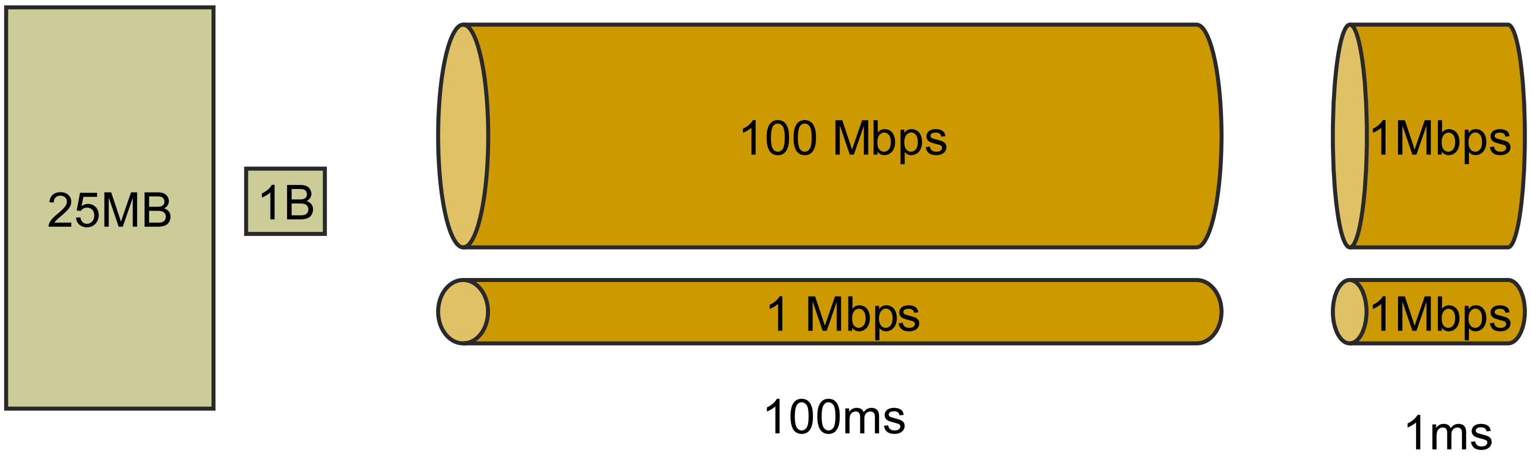
- Example: Transcontinental Channel
  - BW = 45 Mbps
  - delay = 50ms
  - bandwidth x delay product
  - = $(50 \times 10^{-3}$ sec$) \times (45 \times 10^{6}$ bits/sec$)$
  - = 2.25×10^{6} bits

Mbps

ms

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



Copyright ©: CS 438 Staff, University of Illinois

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

Copyright ©: CS 438 Staff, University of Illinois

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

Copyright ©: CS 438 Staff, University of Illinois

Fundamental Challenge: Speed of Light

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

```
 % ping www.cs.illinois.edu
PING dcs-www.cs.illinois.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…

Copyright ©: CS 438 Staff, University of Illinois

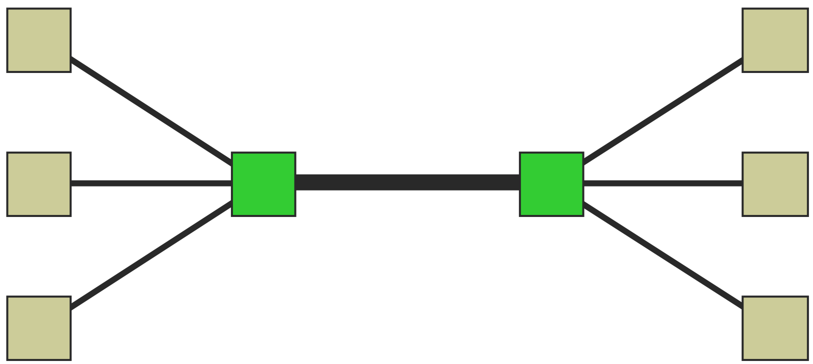
Fundamental Challenge: Shared infrastructure

- 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

Copyright ©: CS 438 Staff, University of Illinois

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)

Copyright ©: CS 438 Staff, University of Illinois

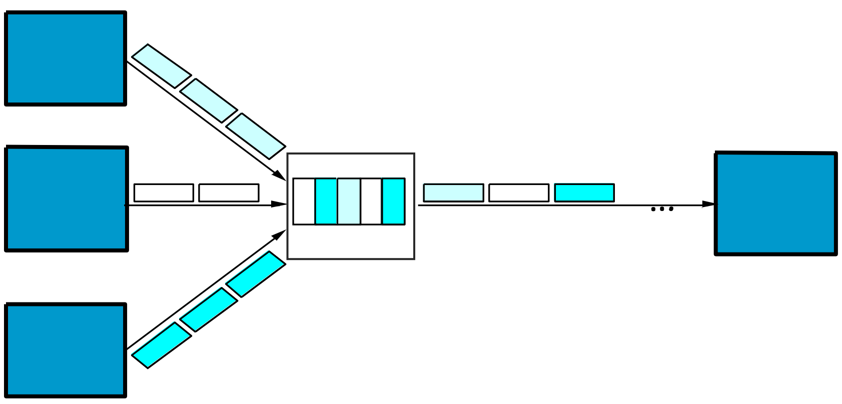
Fundamental Challenge: Shared infrastructure

- 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

Copyright ©: CS 438 Staff, University of Illinois

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?



Copyright ©: CS 438 Staff, University of Illinois

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!

Copyright ©: CS 438 Staff, University of Illinois

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

Copyright ©: CS 438 Staff, University of Illinois

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

Copyright ©: CS 438 Staff, University of Illinois

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

Copyright ©: CS 438 Staff, University of Illinois

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

Copyright ©: CS 438 Staff, University of Illinois

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

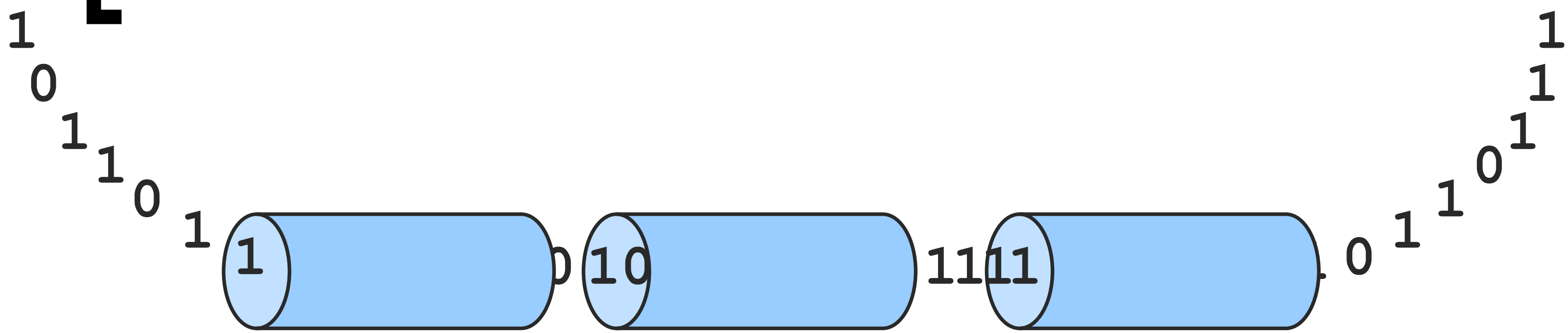
Copyright ©: CS 438 Staff, University of Illinois

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!

Copyright ©: CS 438 Staff, University of Illinois

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

Copyright ©: CS 438 Staff, University of Illinois

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

Copyright ©: CS 438 Staff, University of Illinois