

CSE 123B: Communications Software “Networked Systems”

Amin Vahdat

April 4, 2006

Outline

- Course Mechanics
- Course Topics/Outline
 - Networking/Distributed Systems problem statement
- Introduction to Computer Networks

Audience

- Who should take this course
 - Those interested in learning about large-scale distributed systems

Learn how things work “soup to nuts”
 - Those interested in graduate school
 - Those interested in top industrial positions

Google, Microsoft, Cisco, Yahoo, Akamai, IBM, Apple, etc.

Specifics

- Instructor: Amin Vahdat, vahdat@cs.ucsd.edu
 - Office Hours: By appt, 3104 CSE
- Teaching Assistants
 - Calvin Hubble, chubble@cs.ucsd.edu
 - Office Hours: Wed 1-2, 3262 CSE
- Discussion section
 - Mon 12-1, CENTER 105
- Course Web Page
 - <http://www.cs.ucsd.edu/classes/sp06/cse123b>

General Goals

- Gain background in networking and distributed systems
 - Homework, textbook
- Understanding of some emerging (research) issues
 - Study of relevant research papers
- Building large-scale distributed systems
 - Lectures, programming projects
- Overall: emphasize “*why*” and “*how*” over “*what*”

Non-Goals

- Teach the basics of systems programming
 - Assume familiarity with operating systems C, C++, Java
- 75 minute soliloquies
 - Lectures should be interactive
- Insulate the professor from the students

Grading

- 5% Class Participation
 - Success of class depends on student participation
- 10% Homework
 - Written assignments, paper evaluations
- 50% Exams
 - 20% midterm, 30% final exam
- 35% Programming projects (3)

Paper Evaluations

- Will read 4-5 research papers on constructing high-performance distributed systems
- 1-2 page evaluation of reading for each paper
- Evaluations submitted *in advance* of class
- Describe:
 - Biggest contribution of the paper
 - Most glaring problem with the work
 - What the work implies for building robust, scalable distributed systems and networks

Course Projects

- 3 projects spanning the term
 - Hands on construction of interesting distributed services
- Assignment 1:
 - Build an HTTP Server in C/C++
 - Work in teams of 2
 - Support HTTP/1.0 and subset of HTTP/1.1 functionality
 - Details on web page
 - Due date: April 21

Outline

- Course Mechanics
- Course Topics/Outline
 - Networking/Distributed Systems problem statement
- Introduction to Computer Networks

Internet History

Vannevar Bush



- **Summary:** Vannevar Bush established the U.S. military / university research partnership that later developed the *ARPANET*.

Quote: “Consider a future device for individual use, which is a sort of mechanized private file and library. It needs a name, and to coin one at random, “memex” will do. A memex is a device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility. It is an enlarged intimate supplement to his memory.

It consists of a desk, and while it can presumably be operated from a distance, it is primarily the piece of furniture at which he works. On the top are slanting translucent screens, on which material can be projected for convenient reading. There is a keyboard, and sets of buttons and levers. Otherwise it looks like an ordinary desk.

- Vannevar Bush; *As We May Think*; Atlantic Monthly; July 1945

J. C. R. Licklider

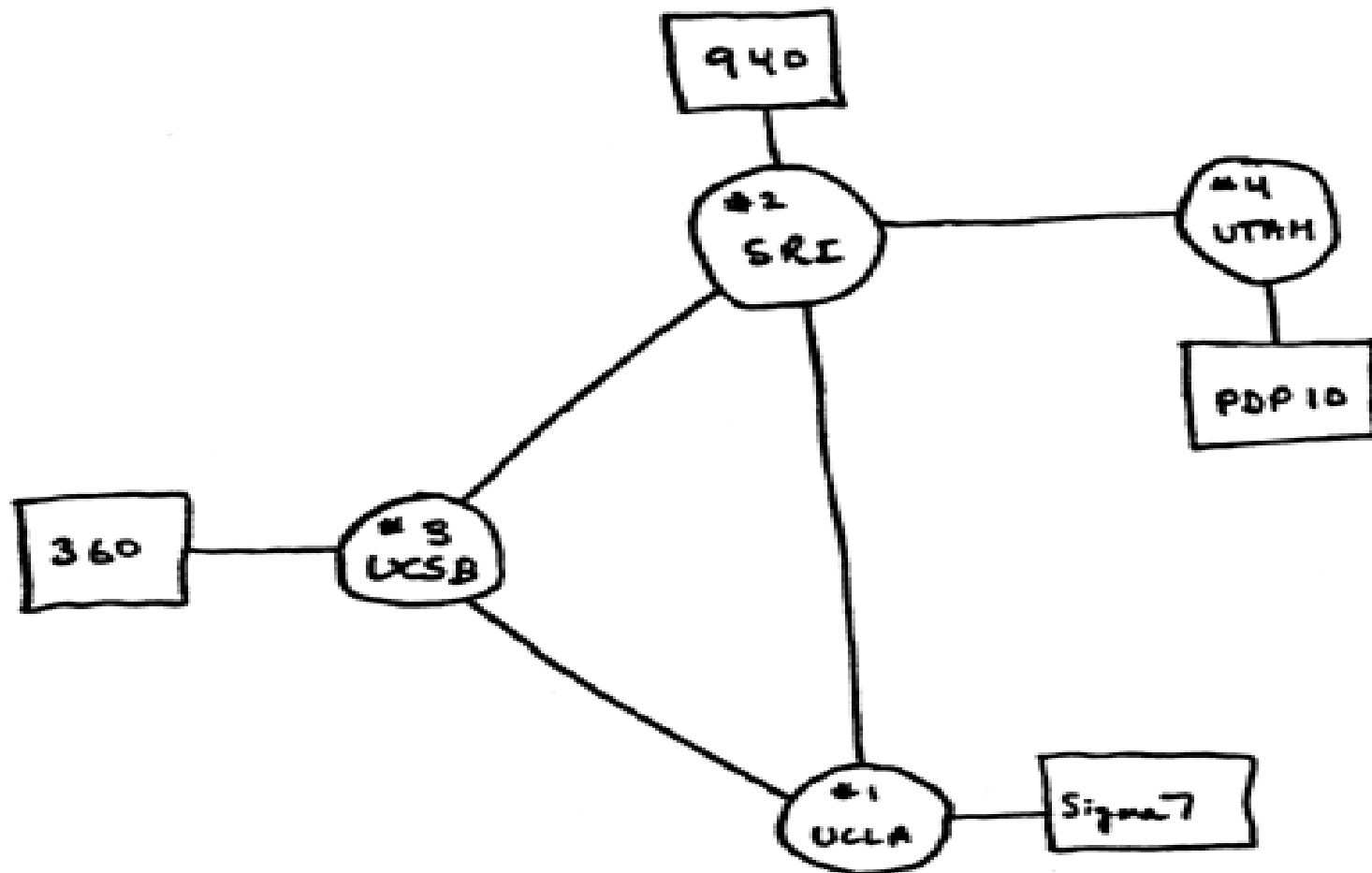


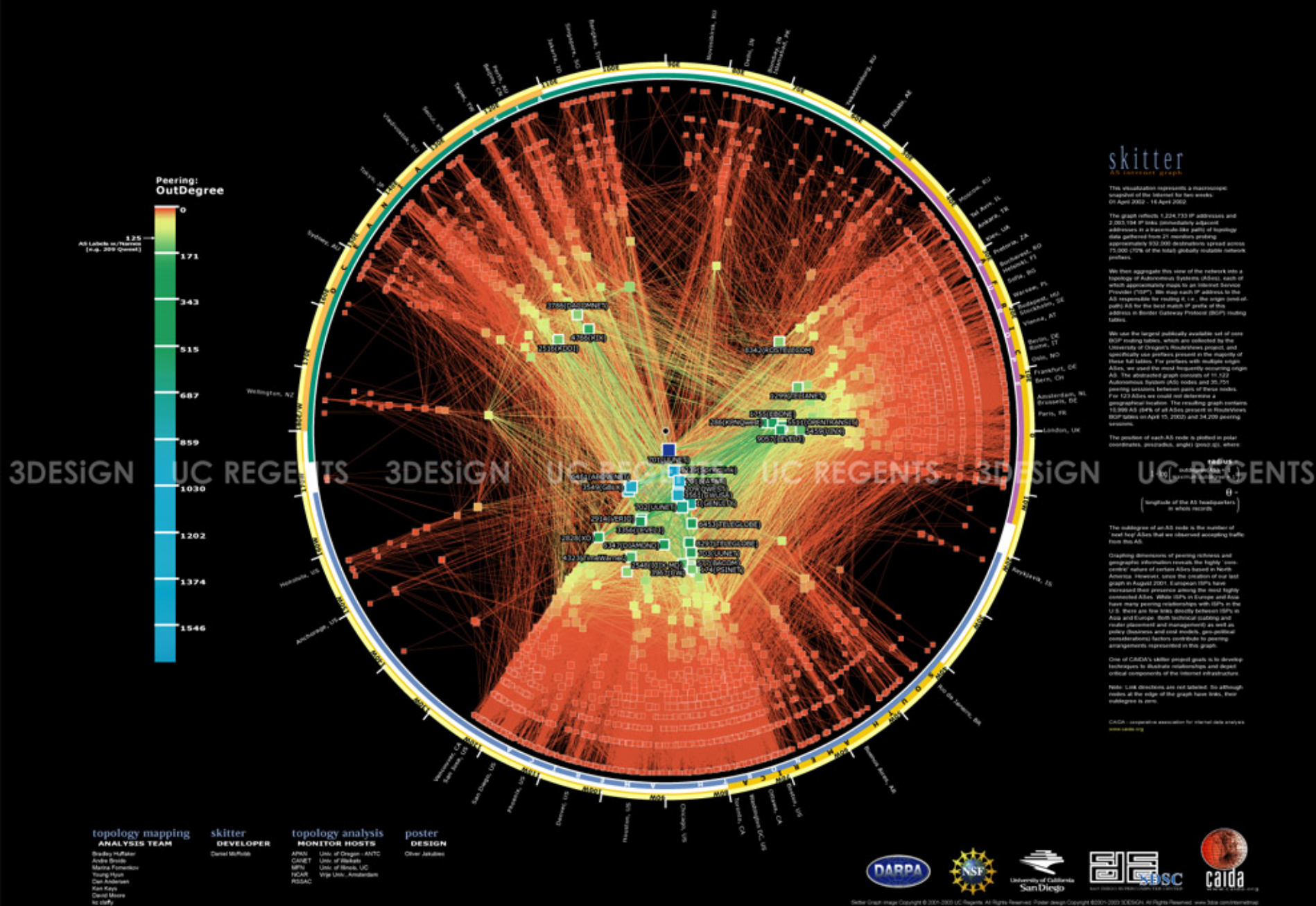
- **Summary:** Joseph Carl Robnett "Lick" Licklider developed the idea of a universal network, spread his vision throughout the *IPTO*, and inspired his successors to realize his dream by creation of the *ARPANET*.
- Quote: "It seems reasonable to envision, for a time 10 or 15 years hence, a 'thinking center' that will incorporate the functions of present-day libraries together with anticipated advances in information storage and retrieval."
- "The picture readily enlarges itself into a network of such centers, connected to one another by wide-band communication lines and to individual users by leased-wire services. In such a system, the speed of the computers would be balanced, and the cost of the gigantic memories and the sophisticated programs would be divided by the number of users."
 - J.C.R. Licklider, *Man-Computer Symbiosis*, 1960.

Background

- 1957: USSR launches Sputnik, first artificial earth satellite
 - U.S. responds by forming ARPA
- 1962: Licklider's *Galactic Network*
- 1966: Roberts (MIT) *Towards a Cooperative Network of Time-Shared Computers*
- 1967: ACM SOSP *Multiple Computer Networks and Intercomputer Communication*

1969 Internet Map





Internet Timeline

- 1971: Tomlinson develops email program, big hit
- 1972: Telnet
- 1973: FTP
- 1974: TCP
- 1978: TCP split into TCP and IP
- 1979: USENET established using UUCP between Duke and UNC by Tom Truscott, Jim Ellis, and Steve Bellovin
- 1984: 1000 hosts connected to Internet, DNS introduced
- 1988: Internet worm brings down 10% of Internet
- 1991: WAIS, Gopher, WWW released

Internet Growth Trends

- 1977: 111 hosts on Internet
- 1981: 213 hosts
- 1983: 562 hosts
- 1984: 1,000 hosts
- 1986: 5,000 hosts
- 1987: 10,000 hosts
- 1989: 100,000 hosts
- 1992: 1,000,000 hosts
- 2001: 150 – 175 million hosts
- 2002: over 200 million hosts
- By 2010, about 80% of the planet will be on the Internet

Major Course Topics

- Internetworking (mostly CSE 123A)
 - Not all computers are directly connected
 - The Internet Protocol (IP)
- End-to-End Protocols
 - E.g., provide the abstraction of a reliable byte-stream over error-prone, packet-switched network
 - Transmission Control Protocol (TCP), Congestion Control
- Modern network services
 - How is Google actually built

Major Course Topics (cont.)

- Naming
 - Given a name for an Internet resource how do you find it?
- Remote Procedure Call
 - Client/Server systems
- Distributed operating systems/file systems
 - How do you break down a centralized service to run across multiple machines?
 - Client/server, peer-to-peer
 - Clustering

Major Course Topics (cont.)

- Replication/Fault Tolerance
 - Copy service contents for increased availability, performance
 - Data consistency?
- Security
 - How to ensure authenticity and integrity of data transmitted across machines

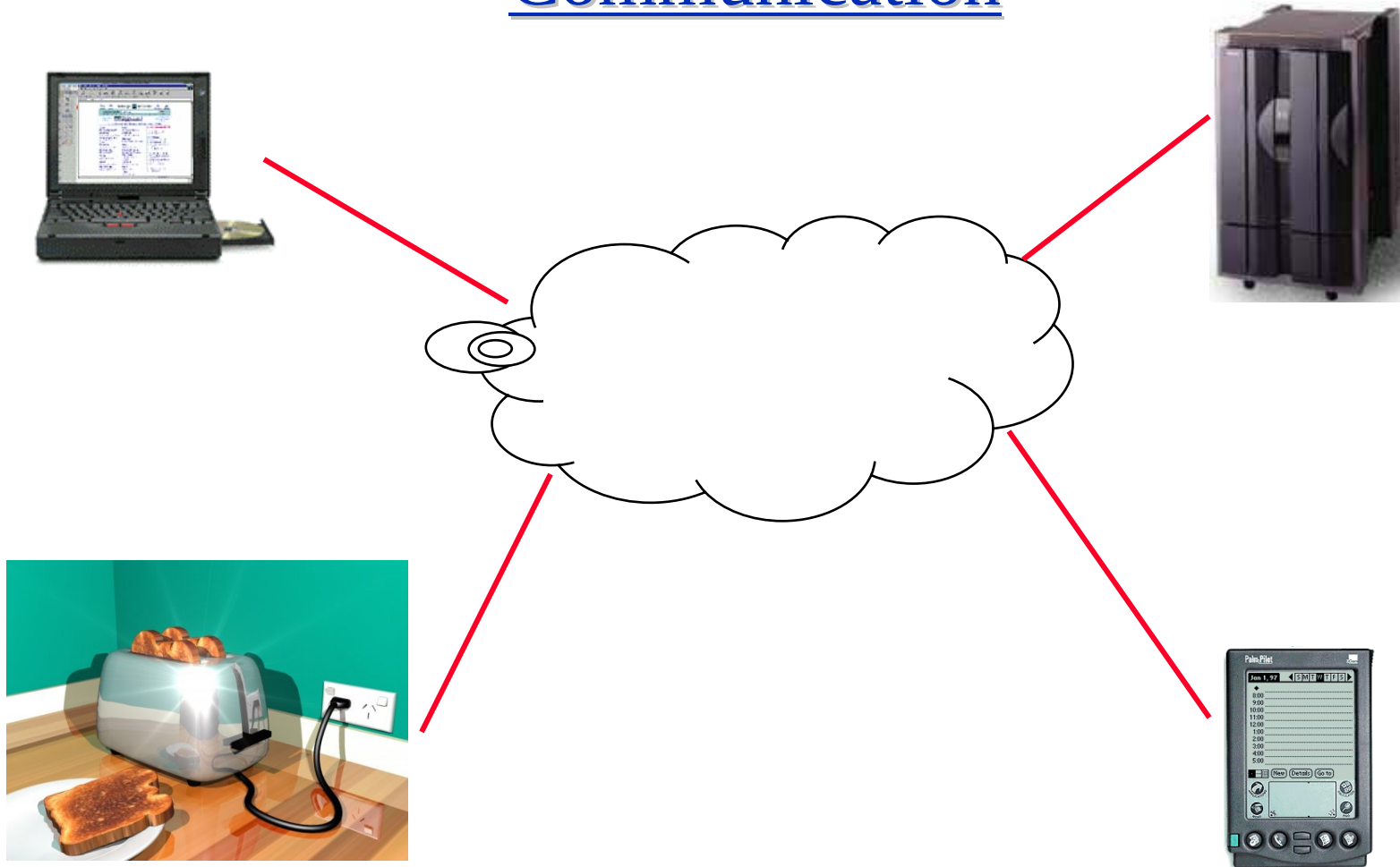
Networking Goal: Scalable, Arbitrary Communication



Networking Goal: Scalable, Arbitrary Communication



Networking Goal: Scalable, Arbitrary Communication

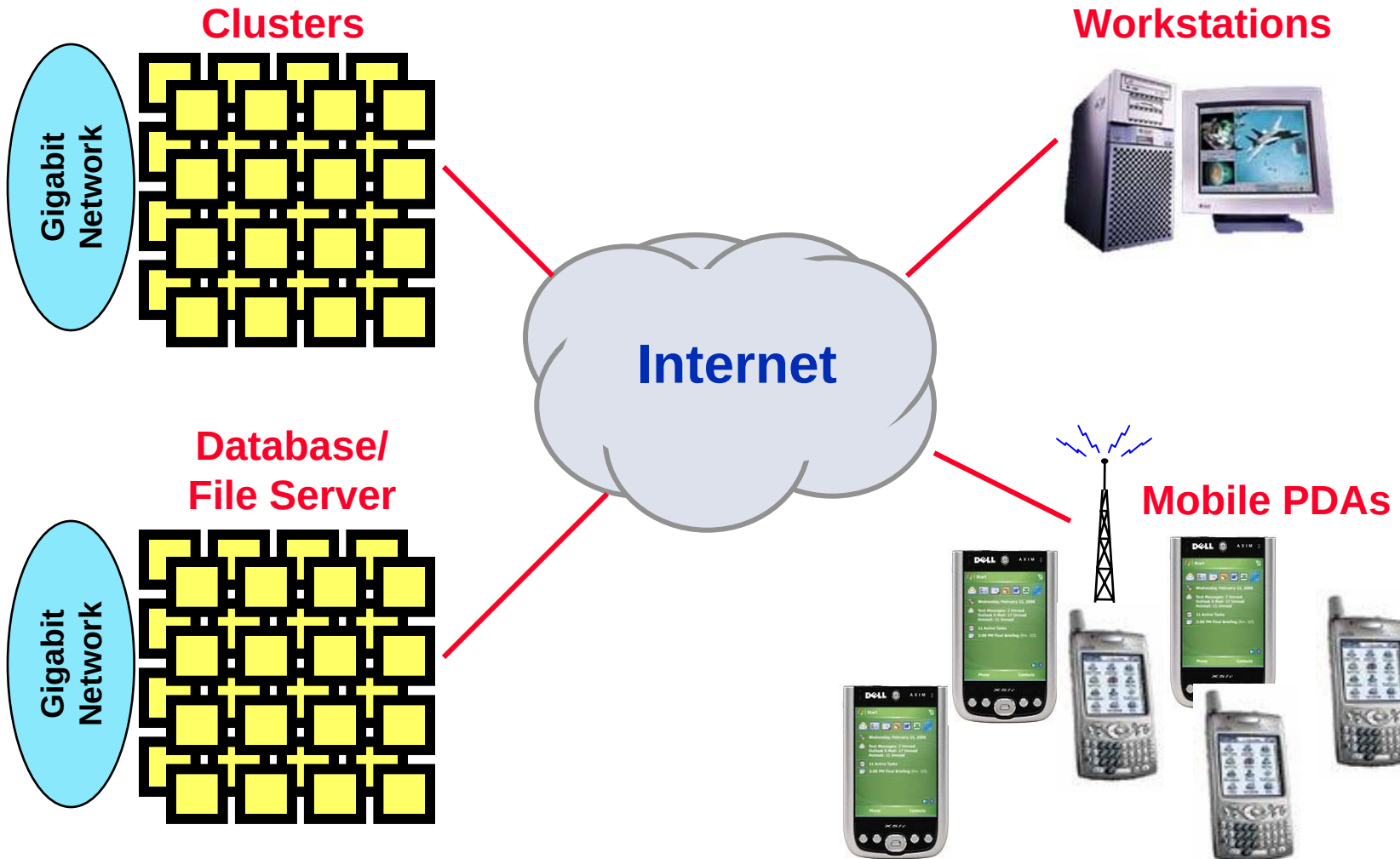


Challenges to Achieving Universal Communication

- How to connect computers
 - Cannot have all-to-all connections
 - How to *name* and *locate* computers
 - Billions of computers: translate name into physical location
 - Routing
 - Transmitting messages from one computer to another
 - Software/Protocols
 - Not just send messages, must agree on format and interpretation
 - Reliability
 - Networks drop, corrupt, and reorder messages
 - Hosts fail and are compromised
- ⇒ Common challenge is *scalability*

Distributed Systems Goals

Provide the illusion of a single large virtual machine



Distributed Systems Applications

- Why would you want to build a distributed application?
- What are some compelling distributed applications?

Distributed Systems Applications

- Why would you want to build a distributed application?
 - Access remote resources not available locally
 - Transmit information to users of remote machines
 - Bring to bear additional processing power on a single task
- What are some compelling distributed applications?
 - Telnet/FTP
 - Email
 - Telecollaboration/virtual reality
 - WWW
 - Games

Distributed Systems Challenges

- Fault tolerance
 - “Distributed computing is where a computer I have never heard of keeps me from getting my work done.”
-Leslie Lamport
 - Strive to maintain the illusion that remote resources are available locally, but *very* hard to mask failures
- Performance
 - Speed of light constraints, network congestion, unpredictable server load
- Data consistency
 - Data available at multiple sites, how to keep data consistent
 - E.g., Web caching

Outline

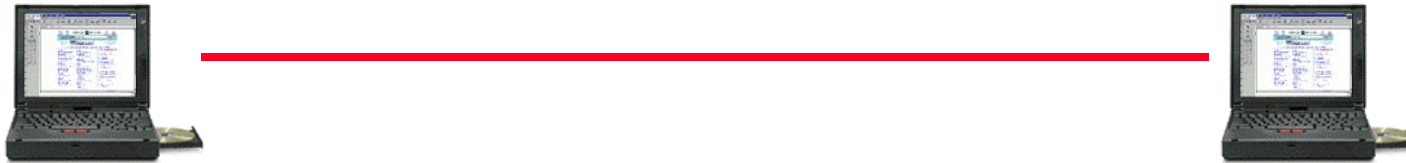
- Course Mechanics
- Course Topics/Outline
 - Networking/Distributed Systems problem statement
- Introduction to Computer Networks

Some Definitions

- Host – computer, PDA, ...
- Packet – unit of transmission across a network
- Link – transmit bits
 - Wire or wireless, broadcast or switched (or both)
- Switch – move bits between links
 - Packet switching: stateless, store & forward
 - Circuit switching: stateful, cut through

Types of Connections

- Point-to-point

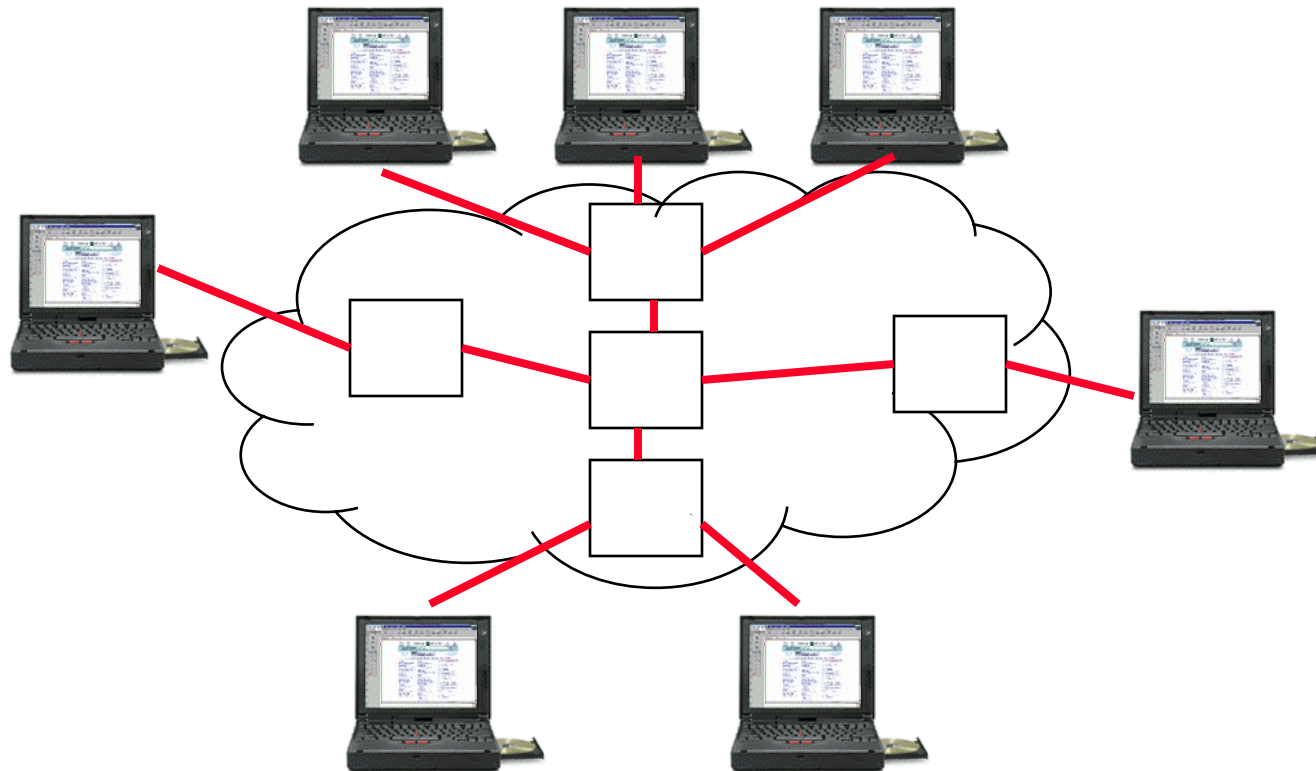


- Multiple access



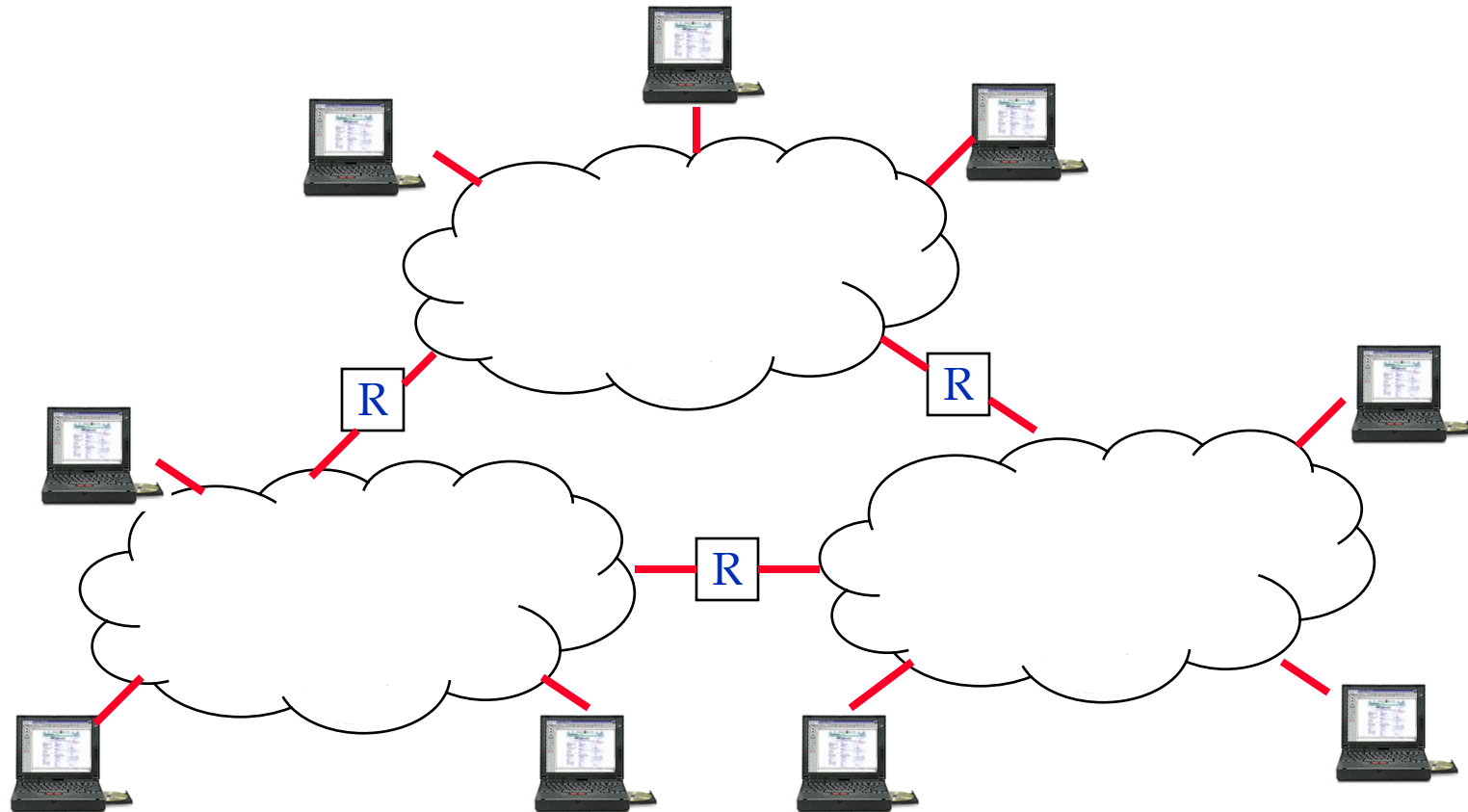
Types of Connections

- Switched



Types of Connections

- Interconnection of networks



Internet – Network of Networks

- Network delivers packets (& locates nodes)
- Router (gateway) moves packets between networks
- Software layer: IP interoperability on top of any potential network or link layer
 - Modem, Ethernet, token ring, cell phone, ADSL, cable modem, smoke signals, ...
- Minimum possible requirements on underlying networks

How is the Network Characterized?

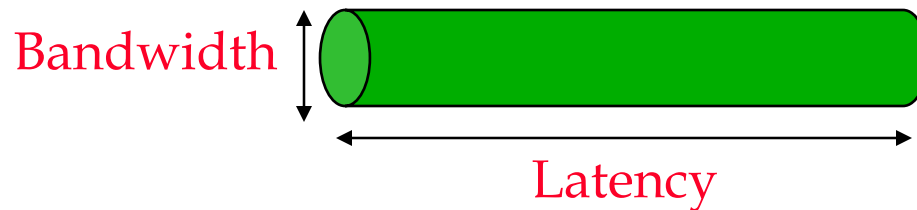
Simple Network Model

- Network is a pipe connecting two computers



- Basic Metrics
 - Bandwidth, latency, overhead, error rate, and message size

Performance Metrics



- Bandwidth: number of bits transmitted per unit of time
- Latency = Propagation + Transmit + Queue
 - Propagation = Distance/SpeedOfLight
 - Transmit = Size/Bandwidth
- Overhead
 - # secs for CPU to put message on wire
- Error rate
 - Probability P that message will not arrive intact

Bandwidth vs. Latency

1 Byte Object

	Latency: 1 ms	Latency: 100 ms
Bandwidth: 1 Mbps	1,008 μ s	100,008 μ s
Bandwidth: 100 Mbps	1,000 μ s	100,000 μ s

10 MB Object

	Latency: 1 ms	Latency: 100 ms
Bandwidth: 1 Mbps	80.001 s	80.1 s
Bandwidth: 100 Mbps	.801 s	.9 s

⇒ Desirable network qualities based on application semantics