History of the Internet Architecture

Manya Ghobadi

6.829 Fall 2020
Lecture #2

Slides credit: Hari Balakrishnan
Several pictures taken from Wikipedia

Announcements

- PSET 1:
  - Released on Monday Sept 14. Due on Sept 28.
- Office Hours:
  - TAs will have four office hours.
  - Please contact us if the timing of the default office hours are challenging for you. We are happy to make accommodations.
- Breakout rooms:
  - We'll be using breakout rooms for group discussions.
  - For smooth transitions, if you're an MIT student please sign in to zoom via your MIT account.

Announcements

- Quiz:
  - Nov. 12, online.
  - We're going to ask you to turn on your webcam. Please email us ASAP if you don't have a webcam or turning it on is problematic.
- Final project presentations:
  - Due "*Friday*" Dec 4th, online presentations
  - Please let us know ASAP if you have a conflict with this day.

Internet Architecture via the Lens of History

The whole of the populous parts of the US will, within 2-3 years, be covered with net-work like a spider's web.

When was this sentence written?

1848
About the electric *telegraph!*

58 | THE VICTORIAN INTERNET

that the whole of the populous parts of the United States will, within two or three years, be covered with net-work like a spider's web."

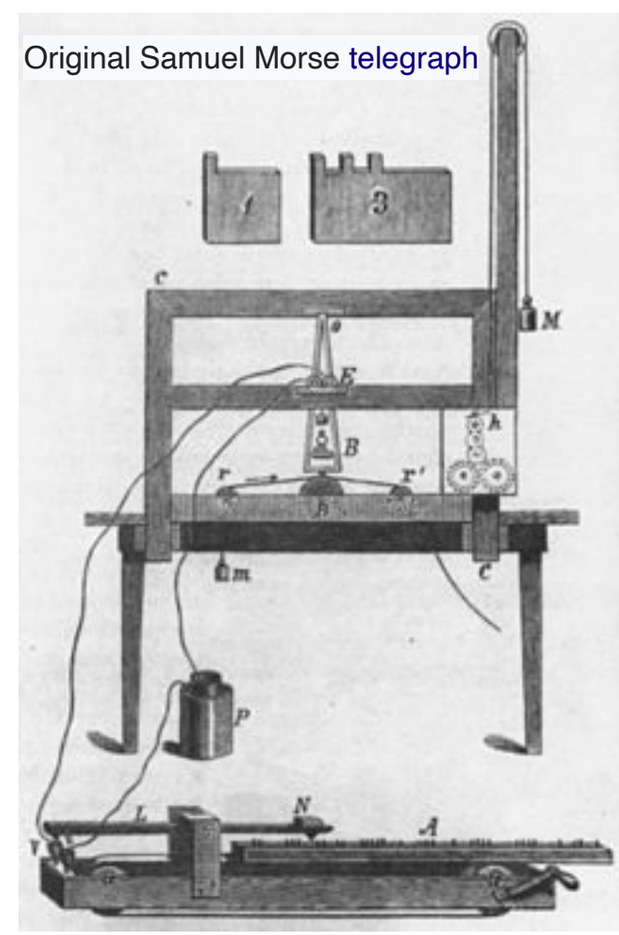
Enthusiasm had swiftly displaced skepticism. The technology that in 1845 "had been a scarecrow and chimera, began to be treated as a confidential servant," noted

The Victorian Internet, Tom Standage

- **Samuel Morse** (1791 –1872) was an American inventor and painter (born in Massachusetts!)

- After having established his reputation as a portrait painter, in his middle age Morse contributed to the invention of a single-wire telegraph system.

Morse artwork



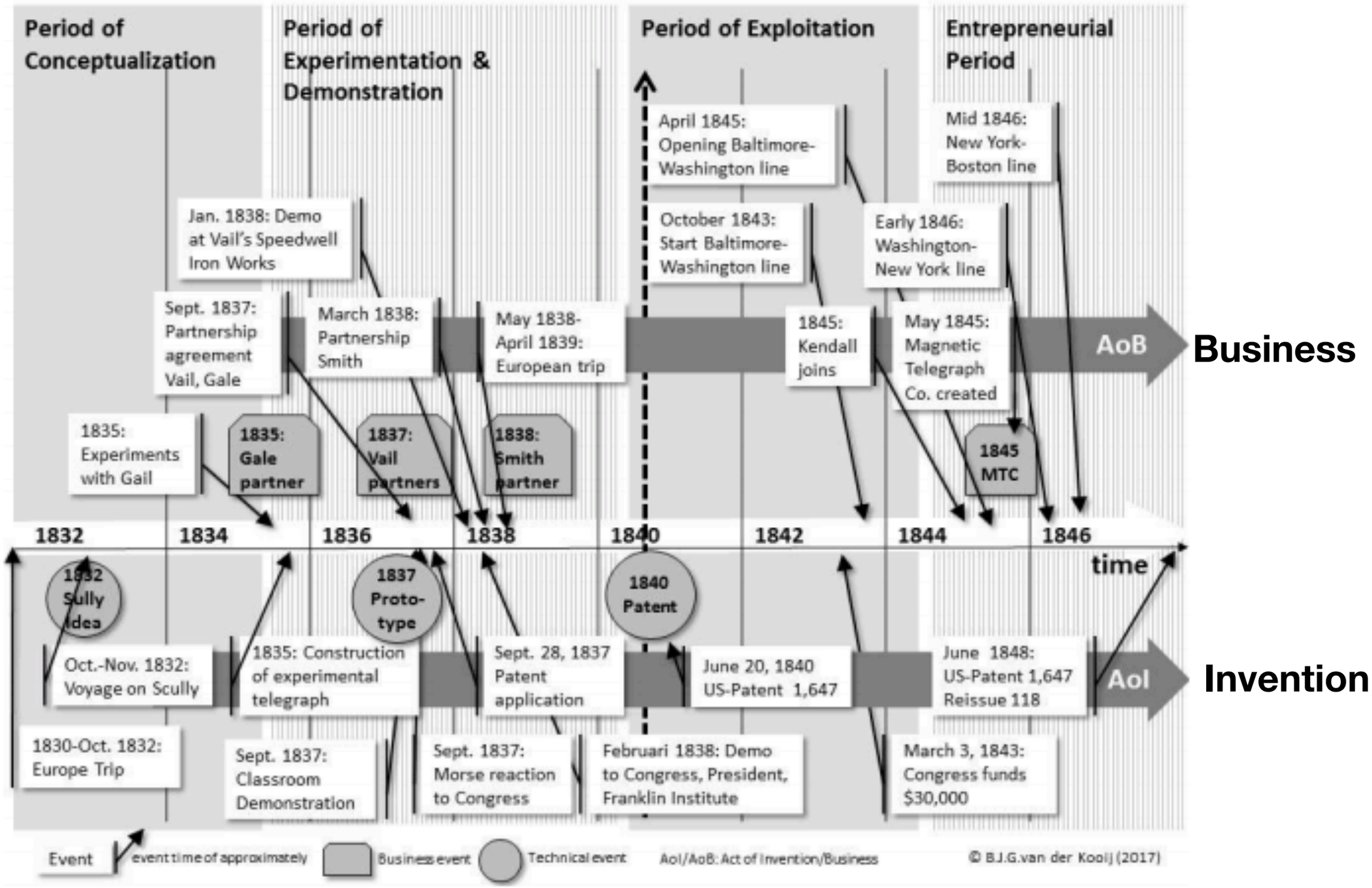


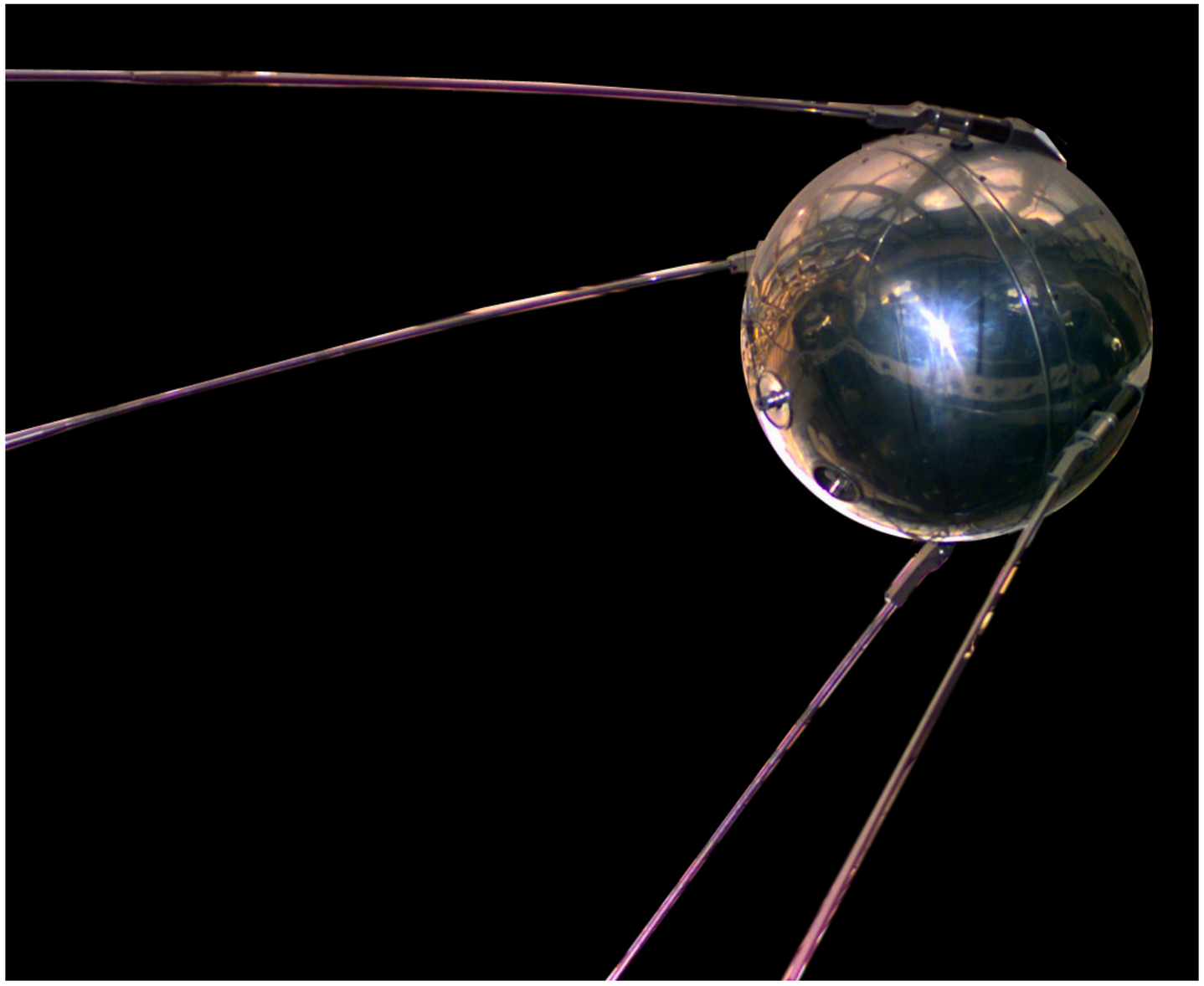
Figure 5: Timeline of the Samuel Morse's activities (1832-1846)

Source: Acts of Invention and Acts of Business in the GPT-Electricity: What did Morse, Bell and Marconi have in common? B.J.G.van der Kooij.

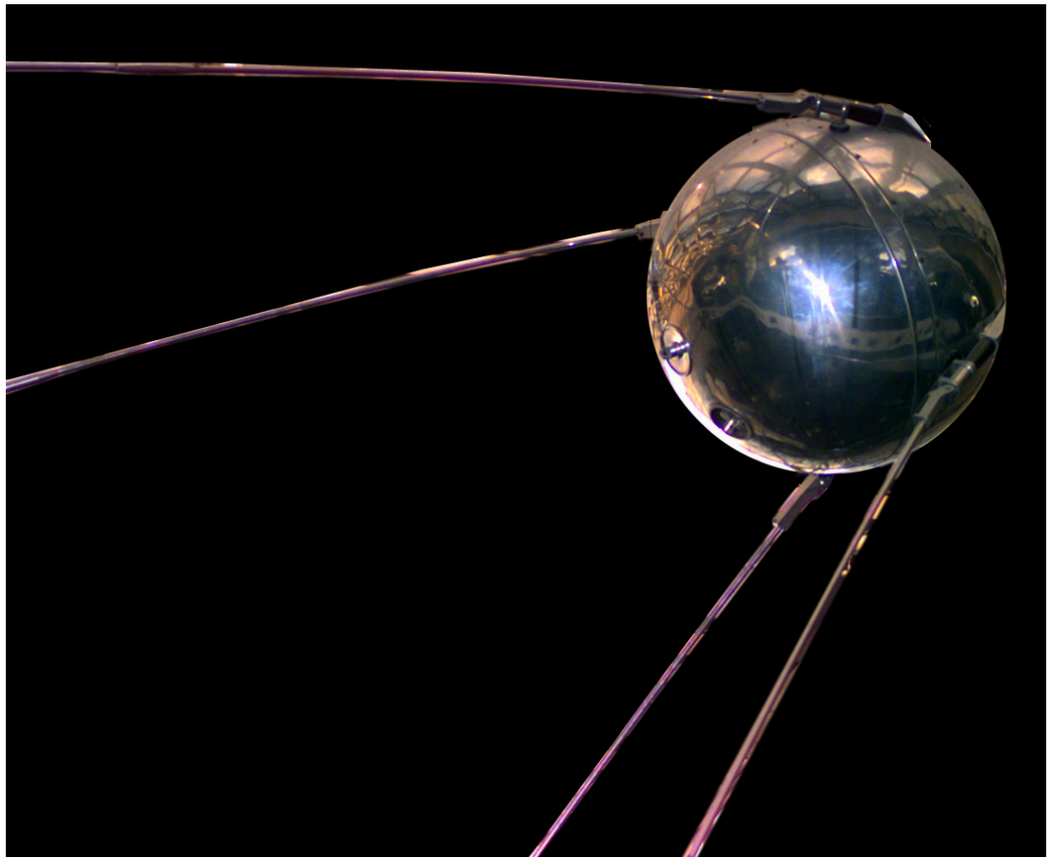
Timeline of Samuel Morse's activities: [1832-1846]

Fast forward 100 years

What is this device?



The first Earth satellite



The Soviet Union launches Sputnik-1 in October 1957

It orbited for three weeks before its batteries died and then silently for two months before it fell back into the atmosphere.

1958

Sputnik led to the creation of ARPA in Feb 1958

Now called DARPA, the Defense Advanced Research Projects Agency.

February 7, 1958
NUMBER 5105.15

Department of Defense Directive

SUBJECT Department of Defense Advanced Research Projects Agency

I. PURPOSE

The purpose of this directive is to provide within the Department of Defense an agency for the direction and performance of certain advanced research and development projects.

II. RESPONSIBILITY AND AUTHORITY

A. Establishment

In accordance with the provisions of the National Security Act of 1947, as amended, and Reorganization Plan No. 6 of 1953, there is established in the Office of the Secretary of Defense the Department of Defense Advanced Research Projects Agency. The Agency will be under the direction of the Director of Advanced Research Projects.

B. Responsibility

The Agency shall be responsible for the direction or performance of such advanced projects in the field of research and development as the Secretary of Defense shall, from time to time, designate by individual project or by category.

C. Authority

Subject to the direction and control of the Director:

1. The Agency is authorized to direct such research and development projects being performed within the Department of Defense as the Secretary of Defense may designate.

2. The Agency is authorized to arrange for the performance of research and development work by other agencies of Government, including the military departments, as may be necessary to accomplish its mission in relation to projects assigned.

3. The Agency is authorized to enter into contracts and agreements with individuals, private business entities, educational, research or scientific institutions including federal or state institutions.

4. The Agency is authorized to acquire or construct such research, development and test facilities and equipment

The Dawn of Packet Switching: 1959-1965

The Dawn of Packet Switching: 1959-1965

Paul Baran

Paul Baran (RAND Corp)

— Pioneer in the development of computer networks.
One of the two independent inventors of packet switching

— Designed "survivable" communications system that could maintain communication between end points in the face of damage from nuclear weapons during the Cold War.

– Early 1960s: New approaches for survivable comms systems; “hot potato routing” and decentralized architecture, paper on packet switching over *digital* links

The Dawn of Packet Switching: 1959-1965

Paul Baran

Donald Davies

Donald Davies (UK)

— Early 1960s – Coins the term “packet” and conceives the idea of packet switching.

— Davies proposed a commercial national network in the United Kingdom and designed and built the local-area network to demonstrate the technology.

What is ARPANET?

Advanced Research Projects Agency Network — ARPANET

- The first wide-area packet-switching network with distributed control and one of the first networks to implement the TCP/IP protocol suite.

Initial Baby Steps

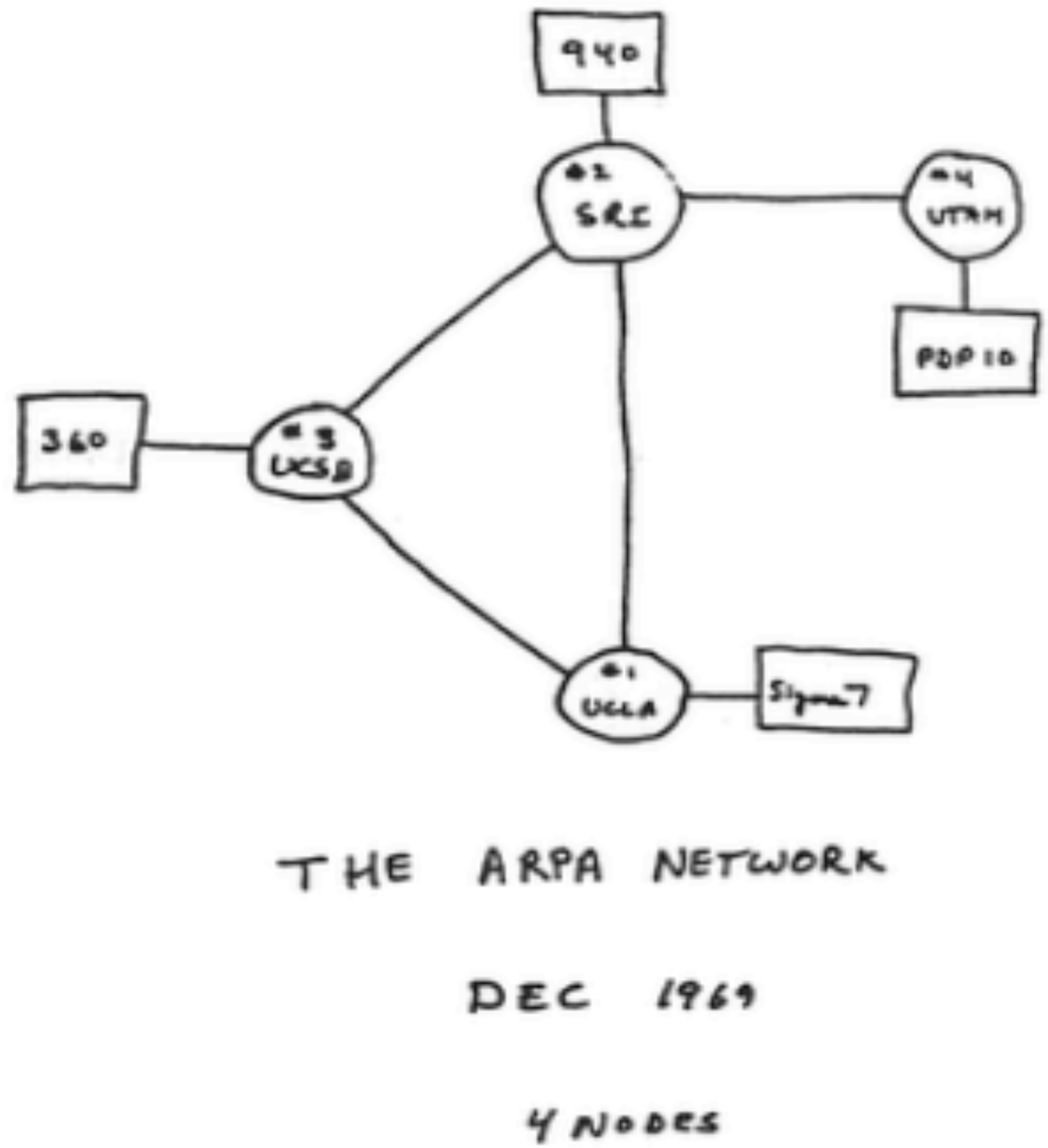


FIGURE 6.2 Drawing of 4 Node Network (Courtesy of Alex McKenzie)

In the Beginning…

Len Kleinrock

- MIT thesis: “Information flow in large communication nets”, 1961
- Prof. at UCLA
- Kleinrock carried out the theoretical work to measure the performance of packet switching in the ARPANET, the forerunner of the Internet.
- Kleinrock's mathematical work underpinned the development of packet-switching networks in the 1970s.

In the Beginning…

- Kleinrock's group at UCLA tried to log on to SRI computer: His recollection of the event…
- "We set up a telephone connection between us and the guys at SRI…

In the Beginning…

- Kleinrock's group at UCLA tried to log on to SRI computer: His recollection of the event…
- "We set up a telephone connection between us and the guys at SRI…

- We typed the L and we asked on the phone…
  - "Do you see the L?"
  - "Yes, we see the L," came the response

In the Beginning…

- Kleinrock's group at UCLA tried to log on to SRI computer: His recollection of the event…
- "We set up a telephone connection between us and the guys at SRI…

- We typed the L and we asked on the phone…
  - "Do you see the L?"
  - "Yes, we see the L," came the response

- We typed the O, and we asked…
  - "Do you see the O?"
  - "Yes, we see the O."

In the Beginning…

- Kleinrock's group at UCLA tried to log on to SRI computer: His recollection of the event…
- "We set up a telephone connection between us and the guys at SRI…

- We typed the L and we asked on the phone…
  - "Do you see the L?"
  - "Yes, we see the L," came the response

- We typed the O, and we asked…
  - "Do you see the O?"
  - "Yes, we see the O."

- Then we typed the G…
  - …and the system crashed!

September 2, 1969

1970s: Packet networks → Internetworking

- 1972: successful ARPANET demo at conference (except it failed when demo'd to skeptics from AT&T!)

1970s: Packet networks → Internetworking

- 1972: successful ARPANET demo at conference (except it failed when demo'd to skeptics from AT&T!)

- 1973: Ethernet (MAC protocol inspired by Aloha – CSMA)
- 1973-74: Xerox PUP (internetworking concepts)

1970s: Packet networks → Internetworking

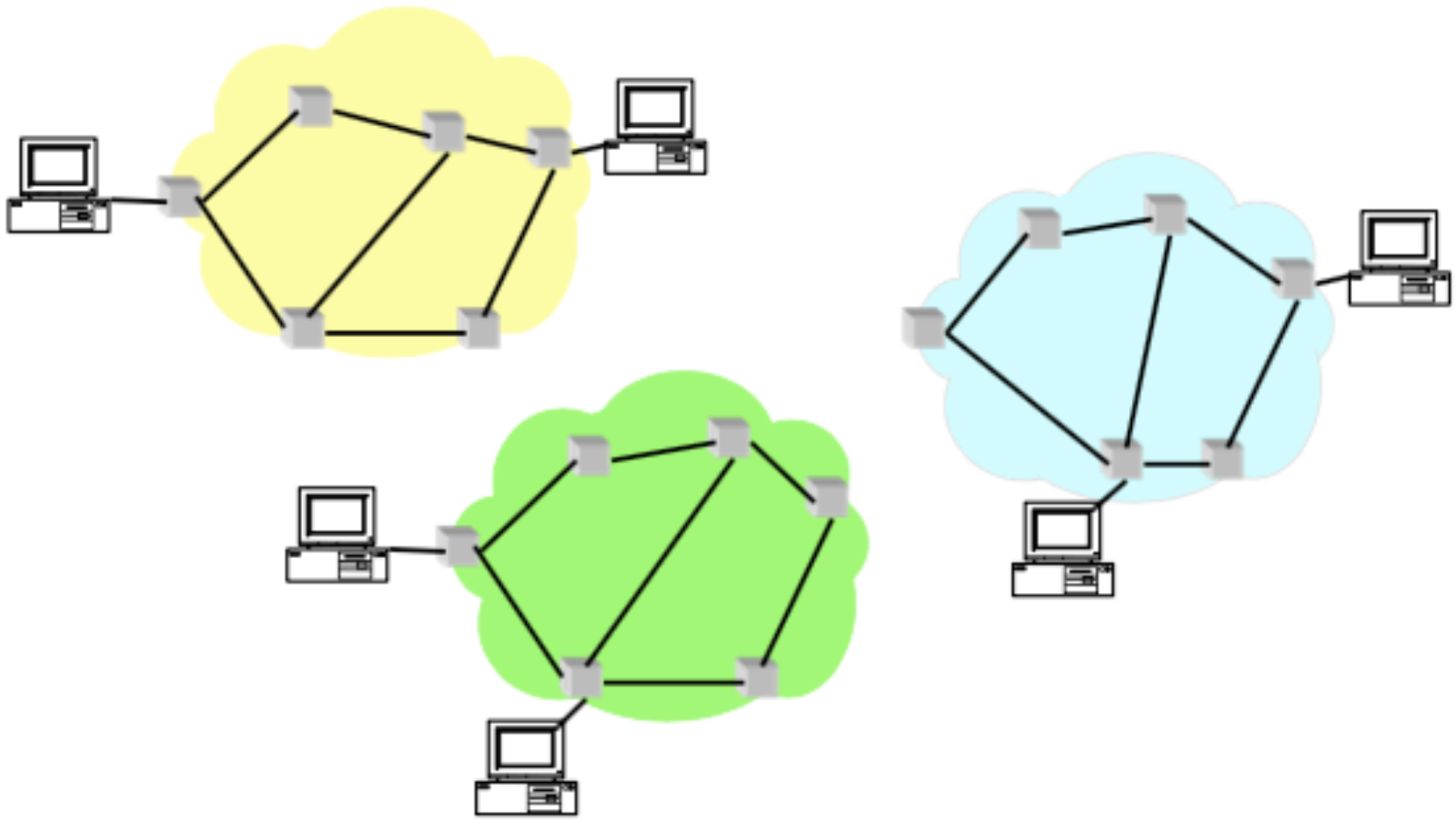
- 1972: successful ARPANET demo at conference (except it failed when demo'd to skeptics from AT&T!)

- 1973: Ethernet (MAC protocol inspired by Aloha – CSMA)
- 1973-74: Xerox PUP (internetworking concepts)

- 1973: ARPANET becomes international
- 1973-75: Internetworking effort (Cerf, Kahn, et al.)
  - Developed TCP and IP (originally intertwined) – TCP uses *sliding window*

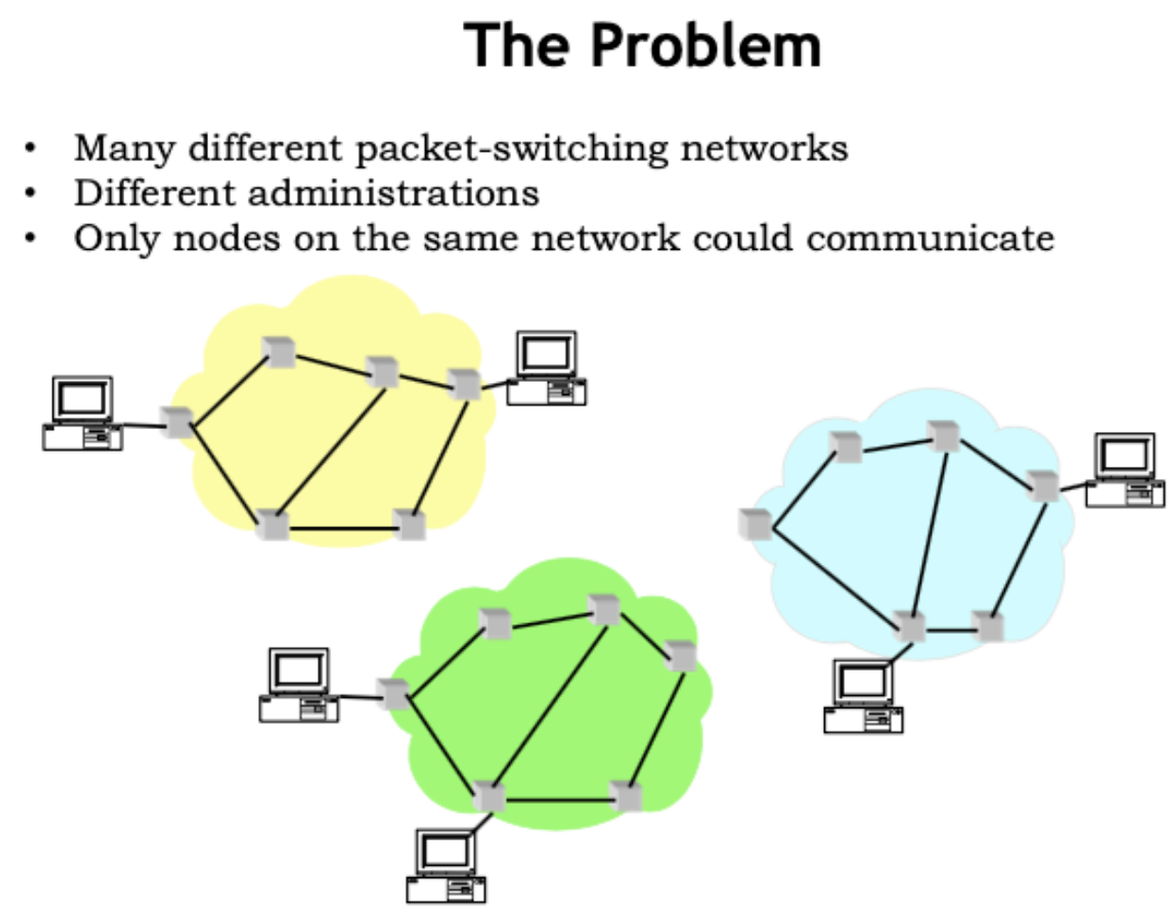
The Problem

- Many different packet-switching networks
- Different administrations
- Only nodes on the same network could communicate

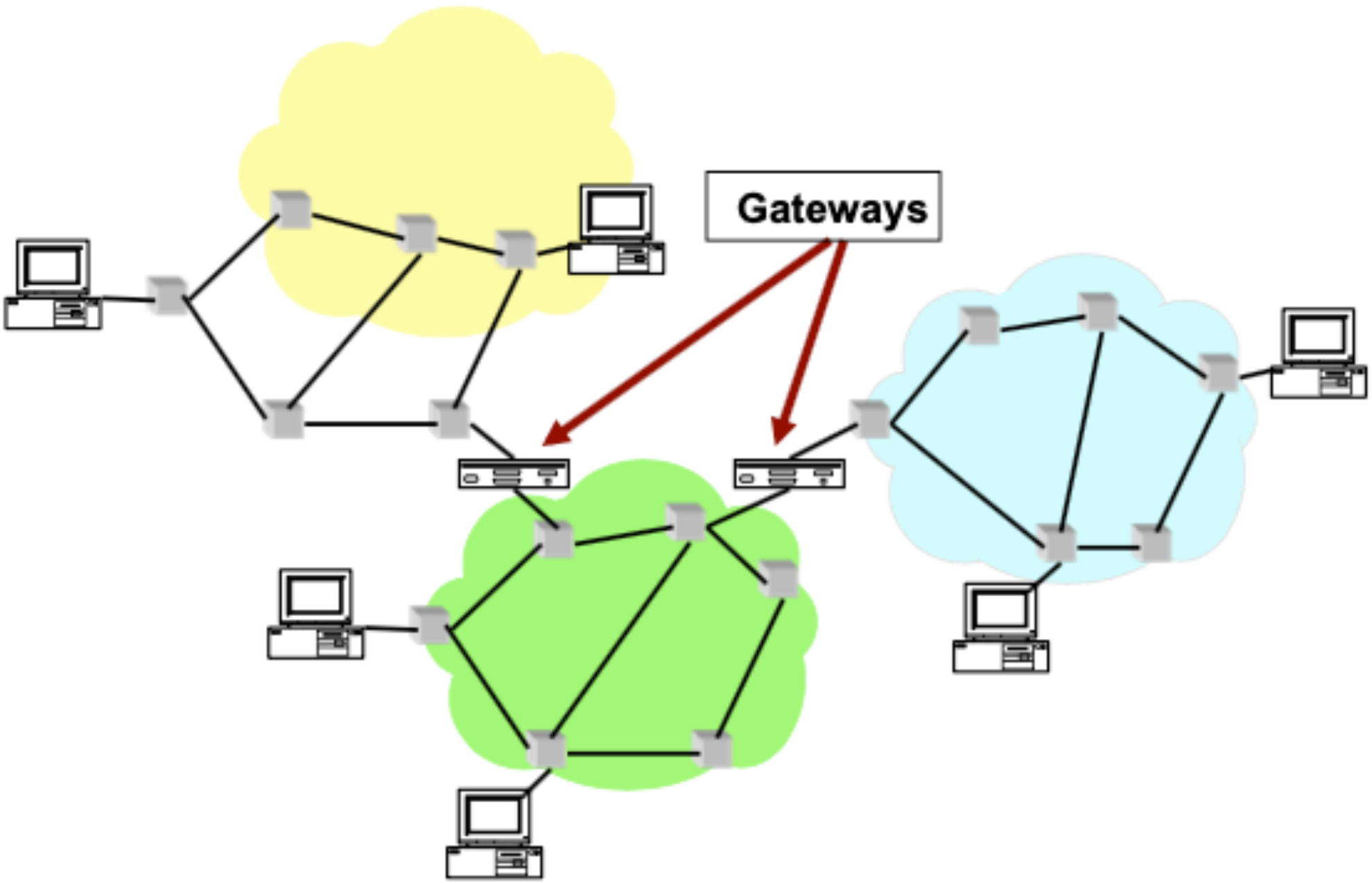


Group Discussion

- How would you solve this problem while ensuring that:
  - Each network is independent and must not be required to change
  - Maintain best-effort communication in networks
  - No global control



Solution



Kahn's Rules for Interconnection

- Each network is independent and must not be required to change
- Best-effort communication
- Boxes (then called *gateways*) connect networks
- No global control at operations level (why?)

Original TCP paper


A Protocol for Packet Network Intercommunication

VINTON G. CERF AND ROBERT E. KAHN

Cerf RFC 968

```
Twas the night before start-up and all through the net,
  not a packet was moving; no bit nor octet.
The engineers rattled their cards in despair,
  hoping a bad chip would blow with a flare.
The salesmen were nestled all snug in their beds,
  while visions of data nets danced in their heads.
And I with my datascope tracings and dumps
  prepared for some pretty bad bruises and lumps.
When out in the hall there arose such a clatter,
  I sprang from my desk to see what was the matter.

There stood at the threshold with PC in tow,
  An ARPANET hacker, all ready to go.
I could see from the creases that covered his brow,
  he'd conquer the crisis confronting him now.
More rapid than eagles, he checked each alarm
  and scrutinized each for its potential harm.
```

The Internetworking Vision

- Bob Kahn & Vint Cerf imagined there would be only a few networks and thus only a few gateways
  - "*The choice for network identification (8 bits) allows up to 256 distinct networks. This size seems sufficient for the foreseeable future.*"
  - They were a little wrong!

The Internetworking Vision

- Bob Kahn & Vint Cerf imagined there would be only a few networks and thus only a few gateways
  - "*The choice for network identification (8 bits) allows up to 256 distinct networks. This size seems sufficient for the foreseeable future.*"
  - They were a little wrong!

- Gateways would "translate" between networks
  - Evolved in the 1974 Cerf/Kahn paper as a universal network layer, later called the *Internet Protocol*, or IP
  - We now think of it as all routers supporting IP

Handling Heterogeneity

- Make it easy to be a node or link on the network (best-effort)
- Universal *network layer*: standardize addressing and forwarding
- "IP-over-everything"
- Switches: *no per-connection state* on behalf of end points
- Original addressing model (then called a "TCP address" because IP and TCP hadn't been split into different layers):

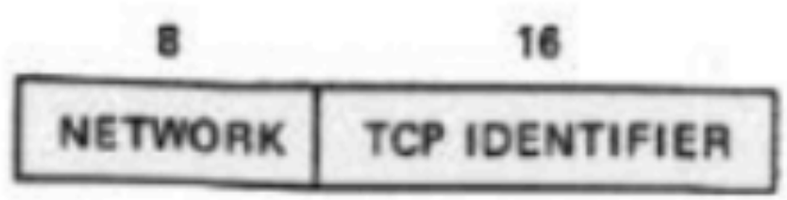


Fig. 4. TCP address.

1970s: Internetworking

- 1978: **Layering**! TCP and IP split; TCP at end points, IP in the network
- IP network layer: simple best-effort delivery

Classic Internet layering "hourglass" model

End-To-End Arguments in System Design

J. H. SALTZER, D. P. REED, and D. D. CLARK
Massachusetts Institute of Technology Laboratory for Computer Science

This paper presents a design principle that helps guide placement of functions among the modules of a distributed computer system. The principle, called the end-to-end argument, suggests that functions placed at low levels of a system may be redundant or of little value when compared with the cost of providing them at that low level. Examples discussed in the paper include bit-error recovery, security using encryption, duplicate message suppression, recovery from system crashes, and delivery acknowledgment. Low level mechanisms to support these functions are justified only as performance enhancements.

CR Categories and Subject Descriptors: C.0 [**General**]: Computer System Organization—*system architectures*; C.2.2 [**Computer-Communication Networks**]: Network Protocols—*protocol architecture*; C.2.4 [**Computer-Communication Networks**]: Distributed Systems; D.4.7 [**Operating Systems**]: Organization and Design—*distributed systems*

General Terms: Design

Additional Key Words and Phrases: Data communication, protocol design, design principles

1. INTRODUCTION

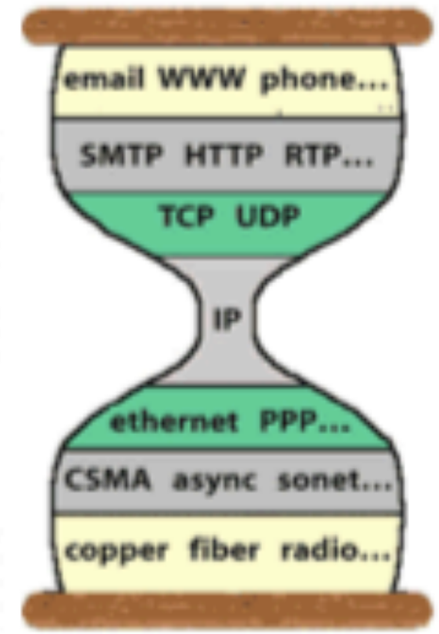
Choosing the proper boundaries between functions is perhaps the primary activity of the computer system designer. Design principles that provide guidance in this choice of function placement are among the most important tools of a system designer. This paper discusses one class of function placement argument that has been used for many years with neither explicit recognition nor much conviction. However, the emergence of the data communication network as a computer system component has sharpened this line of function placement argument by making more apparent the situations in which and the reasons why it applies. This paper articulates the argument explicitly, so as to examine its nature and to see how general it really is. The argument appeals to application requirements and provides a rationale for moving a function upward in a layered system closer to the application that uses the function. We begin by considering the communication network version of the argument.

This is a revised version of a paper adapted from End-to-End Arguments in System Design by J. H. Saltzer, D.P. Reed, and D.D. Clark from the 2nd International Conference on Distributed Systems (Paris, France, April 8-10) 1981, pp. 509-512. © IEEE 1981

This research was supported in part by the Advanced Research Projects Agency of the U.S. Department of Defense and monitored by the Office of Naval Research under contract N00014-75-C-0661.

Authors' address: J. H. Saltzer and D. D. Clark, M.I.T. Laboratory for Computer Science, 545 Technology Square, Cambridge, MA 02139. D. P. Reed, Software Arts, Inc., 27 Mica Lane, Wellesley, MA 02181.

ACM Transactions on Computer Systems, Vol. 2, No. 4, November 1984, Pages 277-288.



- In retrospect, why do you think packet switching took over the world so quickly?

1970s: Internetworking

- 1978: **Layering**! TCP and IP split; TCP at end points, IP in the network
- IP network layer: simple best-effort delivery
- In retrospect: Packet switching (& TCP/IP) won because it is good enough for almost every application (though optimal for nothing!)
- Competitor to TCP/IP: ISO, standardizing 7-layer *OSI stack*

Classic Internet layering "hourglass" model

End-To-End Arguments in System Design

J. H. SALTZER, D. P. REED, and D. D. CLARK
Massachusetts Institute of Technology Laboratory for Computer Science

This paper presents a design principle that helps guide placement of functions among the modules of a distributed computer system. The principle, called the end-to-end argument, suggests that functions placed at low levels of a system may be redundant or of little value when compared with the cost of providing them at that low level. Examples discussed in the paper include bit-error recovery, security using encryption, duplicate message suppression, recovery from system crashes, and delivery acknowledgment. Low level mechanisms to support these functions are justified only as performance enhancements.

CR Categories and Subject Descriptors: C.0 [General]: Computer System Organization—*system architectures*; C.2.2 [Computer-Communication Networks]: Network Protocols—*protocol architecture*; C.2.4 [Computer-Communication Networks]: Distributed Systems; D.4.7 [Operating Systems]: Organization and Design—*distributed systems*

General Terms: Design

Additional Key Words and Phrases: Data communication, protocol design, design principles

1. INTRODUCTION

Choosing the proper boundaries between functions is perhaps the primary activity of the computer system designer. Design principles that provide guidance in this choice of function placement are among the most important tools of a system designer. This paper discusses one class of function placement argument that has been used for many years with neither explicit recognition nor much conviction. However, the emergence of the data communication network as a computer system component has sharpened this line of function placement argument by making more apparent the situations in which and the reasons why it applies. This paper articulates the argument explicitly, so as to examine its nature and to see how general it really is. The argument appeals to application requirements and provides a rationale for moving a function upward in a layered system closer to the application that uses the function. We begin by considering the communication network version of the argument.

This is a revised version of a paper adapted from End-to-End Arguments in System Design by J. H. Saltzer, D.P. Reed, and D.D. Clark from the 2nd International Conference on Distributed Systems (Paris, France, April 8-10) 1981, pp. 509-512. © IEEE 1981

This research was supported in part by the Advanced Research Projects Agency of the U.S. Department of Defense and monitored by the Office of Naval Research under contract N00014-75-C-0661.

Authors' address: J. H. Saltzer and D. D. Clark, M.I.T. Laboratory for Computer Science, 545 Technology Square, Cambridge, MA 02139. D. P. Reed, Software Arts, Inc., 27 Mica Lane, Wellesley, MA 02181.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

© 1984 ACM 0734-2071/84/1100-0277 $00.75

ACM Transactions on Computer Systems, Vol. 2, No. 4, November 1984, Pages 277-288.

Most Useful Lesson

One should architect systems for flexibility – you'll almost never know what apps make it succeed.

(Even if it means sacrificing some performance!)

Most Useful Lesson

One should architect systems for flexibility – you'll almost never know what apps make it succeed.

(Even if it means sacrificing some performance!)

Il semble que la perfection soit atteinte non quand il n'y a plus rien à ajouter, mais quand il n'y a plus rien à retrancher.

Perfection is achieved, not when there is nothing more to add, but when there is nothing left to take away

-- Antoine de Saint-Exupery

Most Useful Lesson

One should architect systems for flexibility – you'll almost never know what apps make it succeed.

(Even if it means sacrificing some performance!)

Il semble que la perfection soit atteinte non quand il n'y a plus rien à ajouter, mais quand il n'y a plus rien à retrancher.

Perfection is achieved, not when there is nothing more to add, but when there is nothing left to take away

-- Antoine de Saint-Exupery

Or,

When in doubt, leave it out

1980s: Handling Growth with Topological Addressing

- 1978-79: ARPANET moves to *link-state routing*
- Per-node routing entries don't scale well
- Solution: Organize network hierarchically
  - Into "areas" or "domains"
  - Similar to how the postal system works
  - Hide detailed information about remote areas
- For this approach to work, node addresses must be *topological*
  - Address should tell network *where* in the network the node is
  - I.e., address is a *location* in the network

1980s: Rapid Growth

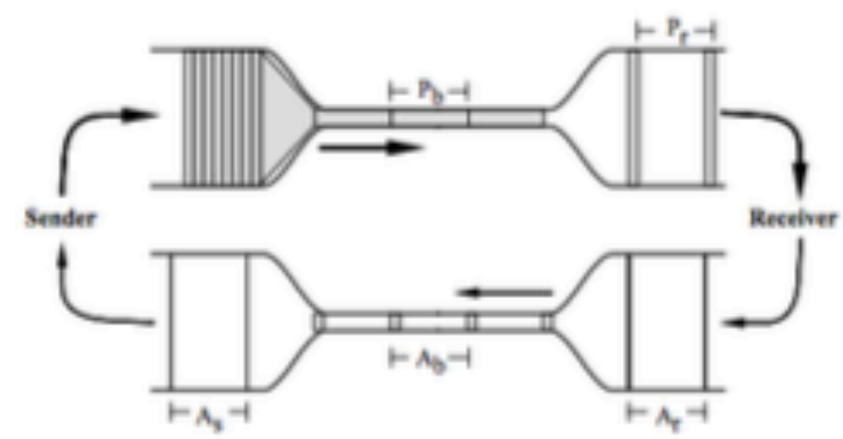
- 1981-89: Dave Clark of MIT is Internet's "Chief Architect"
  - Co-author of the end-to-end arguments (w/ Saltzer/Reed)
  - Ensures consistency of design and vision
  - "*We reject kings, presidents, and voting. We believe in rough consensus and running code.*"
- Saltzer, Reed, Clark: End-to-end arguments in system design [1984]
- 1982: US DoD standardizes on TCP/IP
  - Berkeley's computer systems research group produces BSD & sockets
- 1983: MIT Project Athena – large-scale campus-area networking
- 1984: Domain Name System (DNS) introduced
- 1985: NSFNet picks TCP/IP as standard

Growth Problems: Congestion

- 1986: Congestion collapse episodes on the Internet
  - Problems with bad timeout settings
  - Window size not appropriate for network state
  - Athena network file system congestion problems (bad timeout settings)

- *Congestion avoidance and control*
  - *RTT estimation using EWMA + new timeout method*
  - TCP congestion control by Van Jacobson (concurrent work on DECBit scheme by Ramakrishnan & Jain)
  - Adapt the window size to congestion: If congested, decrease window; else increase. Use exponential back-offs on timeouts
  - By the end of the 1980s, essentially all running TCPs had congestion control

Figure 1: Window Flow Control 'Self-clocking'

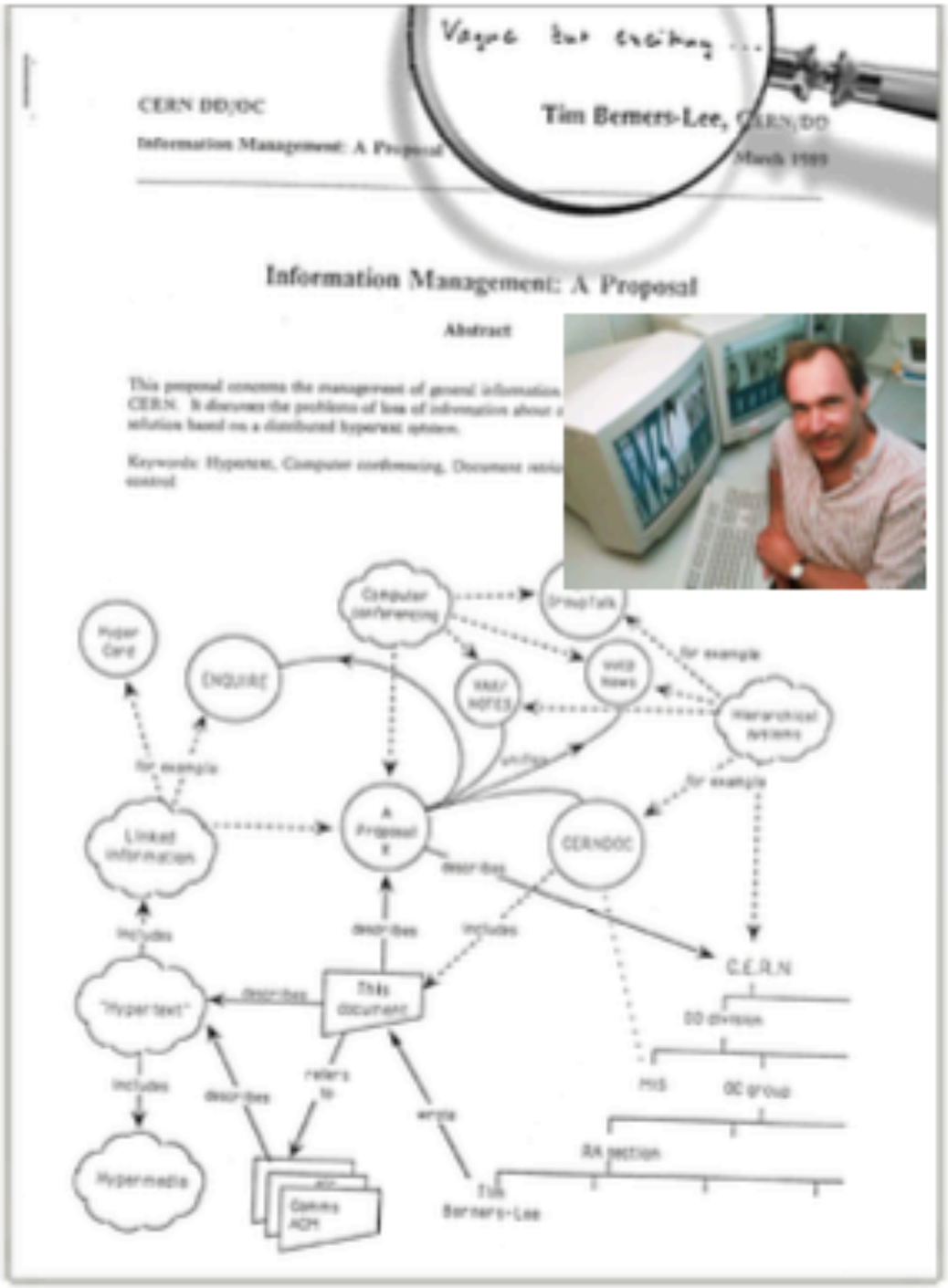


Jacobson

Jain

1990s

- 1990: no more ARPANET
- 1991: Tim Berners-Lee releases "WorldWideWeb"
- Mid-1990s: NSFNet backbone ends
  - Commercial ISPs take off
- "Classless" addressing for scale
  - And the rise of NATs
- BGP4: *Path vector protocol* between competing ISPs, who must yet cooperate



1990s: Handling Growth with CIDR IPv4 Addresses & Address Prefixes

- 18.31.0.82 is actually the 32 bit string 00010010000111110000000000010010

1990s: Handling Growth with CIDR IPv4 Addresses & Address Prefixes

- 18.31.0.82 is actually the 32 bit string 00010010001111100000000010010010
- Routers have forwarding table entries corresponding to an address *prefix* (a range of addrs w/ common prefix bitstring)

1990s: Handling Growth with CIDR IPv4 Addresses & Address Prefixes

- 18.31.0.82 is actually the 32 bit string 00010010000111110000000001010010
- Routers have forwarding table entries corresponding to an address *prefix* (a range of addrs w/ common prefix bitstring)
- 18.0.0.0/8 stands for all IP addresses in the range 00010010 00…0 to 00010010 11…1 (i.e., 2^{24} addresses of the form 00010010*)

1990s: Handling Growth with CIDR IPv4 Addresses & Address Prefixes

- 18.31.0.82 is actually the 32 bit string 00010010000111110000000010010010
- Routers have forwarding table entries corresponding to an address *prefix* (a range of addrs w/ common prefix bitstring)
- 18.0.0.0/8 stands for all IP addresses in the range 00010010 00…0 to 00010010 11…1 (i.e., 2^{24} addresses of the form 00010010*)
- 18.31.0.0/17 stands for a range of 2^{15} consecutive IP addresses of the form 00010010001111100* (1st 17 bits are the same for each address in that range)

1990s: Handling Growth with CIDR IPv4 Addresses & Address Prefixes

- 18.31.0.82 is actually the 32 bit string 00010010000111110000000001010010
- Routers have forwarding table entries corresponding to an address *prefix* (a range of addrs w/ common prefix bitstring)
- 18.0.0.0/8 stands for all IP addresses in the range 00010010 00…0 to 00010010 11…1 (i.e., 2^{24} addresses of the form 00010010*)
- 18.31.0.0/17 stands for a range of 2^{15} consecutive IP addresses of the form 00010010001111100* (1st 17 bits are the same for each address in that range)
- Hence, *subnetworks* may be of size 1, 2, 4, 8, … (maxing out at 2^{24} usually), and may be recursively divided further

1990s: Handling Growth with CIDR IPv4 Addresses & Address Prefixes

- 18.31.0.82 is actually the 32 bit string 00010010000111110000000010010010
- Routers have forwarding table entries corresponding to an address *prefix* (a range of addrs w/ common prefix bitstring)
- 18.0.0.0/8 stands for all IP addresses in the range 00010010 00…0 to 00010010 11…1 (i.e., 2^{24} addresses of the form 00010010*)
- 18.31.0.0/17 stands for a range of 2^{15} consecutive IP addresses of the form 00010010001111100* (1st 17 bits are the same for each address in that range)
- Hence, *subnetworks* may be of size 1, 2, 4, 8, … (maxing out at 2^{24} usually), and may be recursively divided further
- Forwarding uses *longest prefix match*
  - At each router, routes are of the form "For this range of addresses, use this route"
  - Pick the route that has the longest matching prefix w/ dest addr

1990s (cont.)

- 1993: search engines (Excite) 

- Mid-1990s: E-commerce starts
- Late 1990s: news on the web
  - MIT Tech: first web paper (1994)

1990s (cont.)

- 1993: search engines (Excite)
- Mid-1990s: E-commerce starts
- Late 1990s: news on the web
  - MIT Tech: first web paper (1994)
- Mid-1990s: some misplaced fears of the Internet melting down
- 1998: Google reinvents search
- 1998: Content distribution networks (Akamai)
- 1996-2001: .com bubble starts & bursts

2000-2010: The Internet Matures (Top 5 List)

- 2000-2001: .com bust
  - And 9/11 happens
  - Power of content distribution networks to handle load

1. The rise of peer-to-peer networks
   - Napster, Gnutella, Freenet, distributed hash tables (e.g., Chord), BitTorrent)
2. Security threats and defenses
   - 2000: Large-scale distributed denial-of-service (DDoS) attacks start
   - 2003: SQL slammer worm
   - Spam → phishing and pharming → complex ecosystem
   - Route hijacking by errors or malice
3. User-generated content & social networks
   - Blogs, Youtube (2004), Facebook (2003), Twitter
4. The rise of wireless and mobile data
5. Cloud computing and large-scale datacenters (Amazon, Google, Microsoft, Facebook, etc.)

- *Almost* everything moves to the Internet: telephony, video, entertainment

Group Discussion

- Discuss three items that were not considered in the original design of the Internet, but which became important concerns later.

Architectural Weaknesses & Warts

1. Designed with trusted endpoints in mind
   Unwanted traffic, lack of accountability, poor level of security
2. Mobility not a first-class goal
3. Middleboxes an afterthought
4. Management, measurement, and fault diagnosis are hard
5. Higher availability for those that need it: in architecture or as an overlay?
6. Design for economic viability: resource accounting is hard
7. Multi-homing an afterthought
8. Should data and content be first-class objects?
9. Better-than-best-effort services and support for application objectives
10. The architecture isn't evolvable
11. Cross-layer feedback is hard

This Decade and Next

- Greater growth (video, entertainment, collaboration), internationalization
- Better traffic control for application performance (human engineering, machine learning)
- Combating complexity: new methods to make things simpler (control plane: “software-defined networks”)
- Programmable data planes: doing more with network hardware
- Mobility and Wireless: anytime/anywhere access, the spectrum crunch, high variability
- Physical embedding & embodiment: sensors & actuators over the network, mobile robots and autonomous agents, vehicles
- Network security & privacy, anti-censorship, surveillance