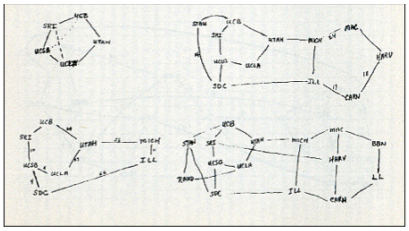
Eric Roberts
CS 54N

Handout #25
November 14, 2016

Networking

Great Ideas in Networking



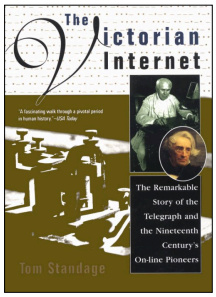
Eric Roberts
CS 54N
November 14, 2016

Central Themes

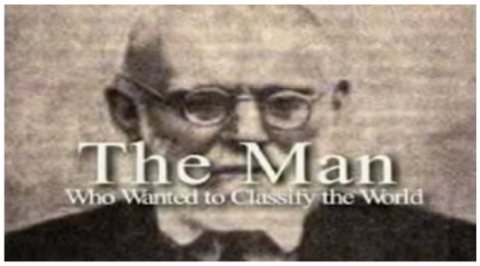
- The Internet has a long history and did not spring to life fully-formed with the advent of the web in the mid-1990s.
- Despite much mythology to the contrary, the Internet and its predecessors were funded by public research funds for most of the history of the field.
- The Internet has evolved in ways quite different from those envisioned by its creators.
- The Internet was originally designed for experimental flexibility, not as reliable infrastructure.
- The Internet is a hugely complex system and is therefore subject to both *emergent behavior* and *system failures*. Fortunately, the designers understood these ideas better than most systems developers.

The Victorian Internet

Many of the ideas that seem so new in the Internet have deep historical roots. In 1998, Tom Standage wrote a fascinating book about the history of telegraphy. In his book, Standage describes how the telegraph gave rise to many of the social structures of the Internet, including chat rooms, online romances, and its own breeds of entrepreneurs, cryptographers, and hackers.



Paul Otlet



Vannevar Bush and Hypertext

One of the earliest proponents of developing a global information network of the form we have today in the web was Vannevar Bush, President Roosevelt's Director of the Office of Scientific Research and Development. In a 1945 article in *Atlantic Monthly* entitled "As We May Think," Bush anticipated many of the ideas that are central to the modern Internet, including the idea of hyperlinked documents.

Vannevar Bush (1890-1974)

Vannevar Bush and the Memex

> When data of any sort are placed in storage, they are filed alphabetically or numerically, and information is found (when it is) by tracing it down from subclass to subclass. It can be in only one place, unless duplicates are used; one has to have rules as to which path will locate it, and the rules are cumbersome. Having found one item, moreover, one has to emerge from the system and re-enter on a new path.
>
> The human mind does not work that way. It operates by association. With one item in its grasp, it snaps instantly to the next that is suggested by the association of thoughts, in accordance with some intricate web of trails carried by the cells of the brain. . . .
>
> 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.

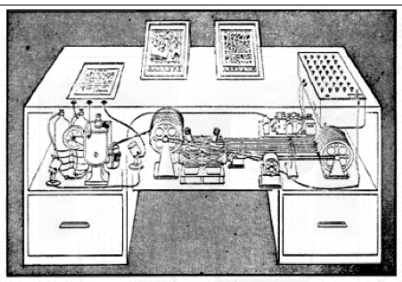
—Vannevar Bush, "As We May Think," 1945

Vannevar Bush and the Memex

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. . . .

All this is conventional, except for the projection forward of present-day mechanisms and gadgetry. It affords an immediate step, however, to associative indexing, the basic idea of which is a provision whereby any item may be caused at will to select immediately and automatically another. This is the essential feature of the memex. The process of tying two items together is the important thing.

Vannevar Bush and the Memex



Memex in the form of a desk would instantly bring files and material on any subject to the operator's fingertips. Slanting translucent viewing screens magnify supermicrofilm filed by code numbers. At left is a mechanism which automatically photographs longhand notes, pictures and letters, then files them in the desk for future reference (*LIFE* 19(11), p. 123).

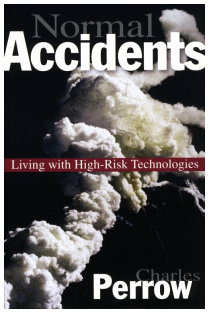
Ted Nelson and Hypertext

Many of the notions of the modern web were anticipated by Ted Nelson, who introduced the word *hypertext* in the early 1960s.

In 1974, Nelson published *Computer Lib*, a two-sided book that is more reminiscent of a graphic novel than one's traditional conception of a book. In it, he describes how computers will empower people as technology develops.

Ted Nelson

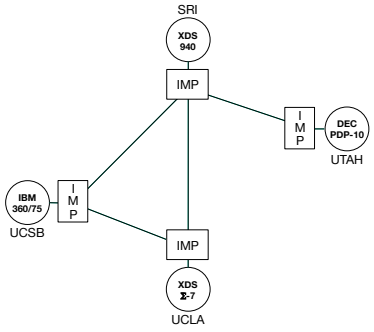
Normal Accidents



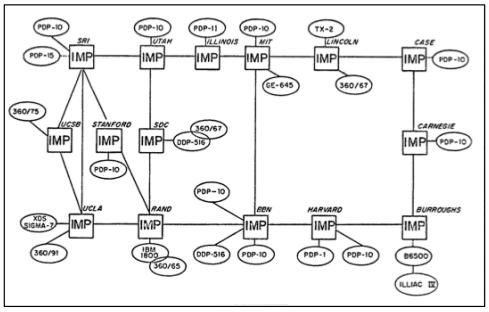
Charles Perrow, Normal Accidents: Living with High-Risk Technologies, New York: Basic Books, 1984.

Although this book does not focus specifically on programming—and indeed does not include software or programming in its index—the issues that it raises are critical to an understanding of why complex technological systems fail.

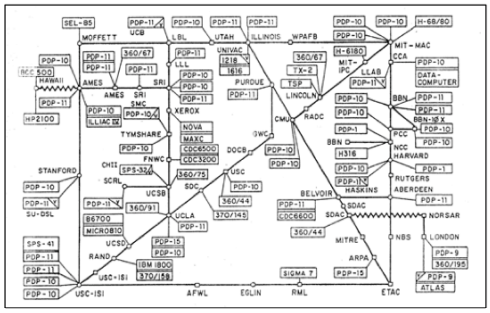
Initial ARPANET Configuration (1969)



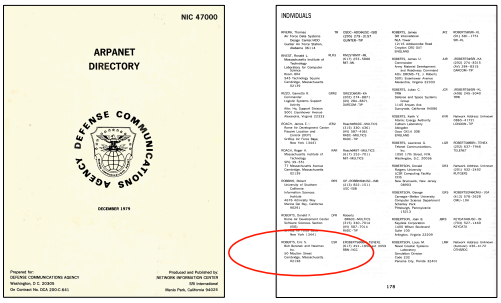
ARPANET Map (April 1971)



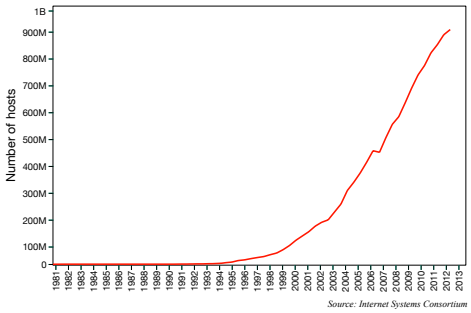
ARPANET Map (January 1975)



The ARPANET Directory



Internet Growth

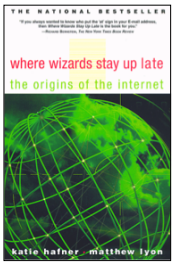


The BBN Interface Message Processor

The BBN ARPANET Team

Life among the Wizards

The history of the Internet has been told in several books. One tells the following interesting story:

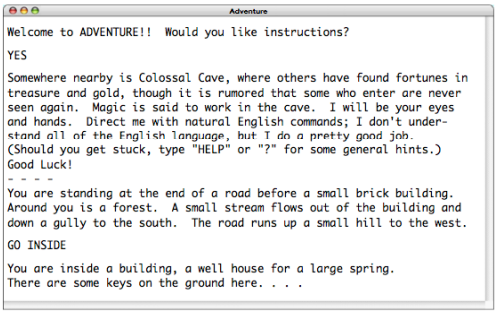


A small circle of friends at BBN had gotten hooked on Dungeons and Dragons, an elaborate fantasy role-playing game in which one player invents a setting and populates it with monsters and puzzles, and the other players then make their way through that setting. The game exists only in the minds of the players.

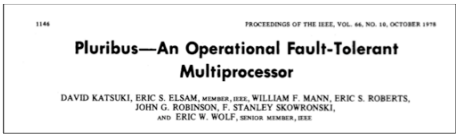
Dave Walden got his introduction to the game one night when Eric Roberts, a student from a class he was teaching at Harvard, took him to a D&D session. Walden immediately rounded up a group of friends from the ARPANET team for continued sessions. Roberts created the Mirkwood Tales. . . .

One of the regulars was Will Crowther . . .

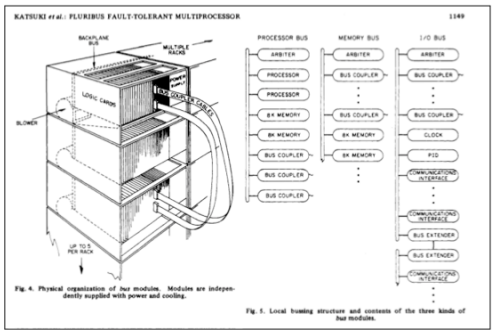
Willie Crowther's Adventure Game



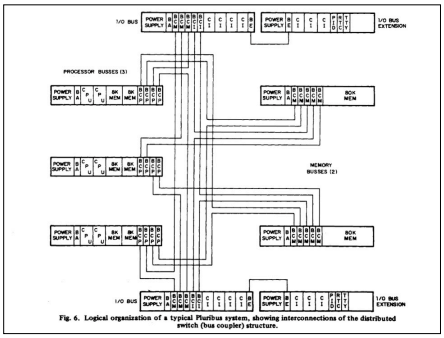
My Work at BBN



Hardware Structure of the Pluribus



Logical Structure of the Pluribus



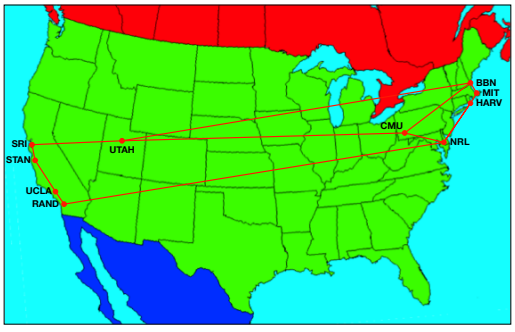
Achieving "Relaxed Reliability"

- The goal of the Pluribus was to achieve "relaxed reliability" in the sense that the machine was allowed to go down, but was expected to revive itself quickly (in particular, before anyone could know it was down).
- To do so, the Pluribus explored its own architecture as it ran, building a shared view of what resources were available by executing a sequence of ***stages***.
- Consistency was achieved by having the processors vote on their picture of the world.

TABLE I
PLURIBUS OPERATING SYSTEM STAGES

stage	function
0	Checksum local memory code (for stages 0,1,2). Initialize local interrupt vectors, and enable interrupts. Discover Processor bus I/O. Find some real-time clock for system timing.
1	Discover all usable common memory pages. Establish page for communication between processors.
2	Find and checksum common memory code (for stages 3,4,5). Checksum whole page ("reliability page").
3	Discover all common busses, PIDs, and real-time clocks.
4	Discover all processor bus couplers and processors.
5	Verify checksum (from stage 2) of reliability page code (for rest of stages plus perhaps some application routines). External reloading of missing code pages is possible once this stage is running.
6	Checksum all of local code.
7	Checksum common memory code. Maintain page allocation map.
8	Discover common I/O interfaces.
9	Poll application-dependent reliability and initialization routines. Periodically trigger restarts of halted processors.
10	Application system.

ARPANET Geographic Map (1970)

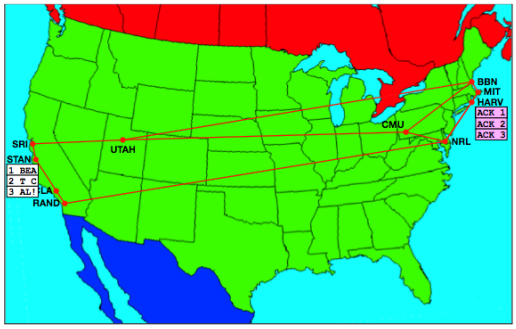


A Challenge

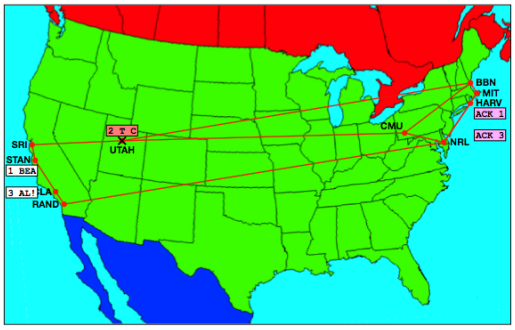
Given a map such as the one shown on the preceding slide, how would you design a reliable communication strategy for sending messages between nodes in the network, given the following facts:

- Nodes are unreliable.
- Nodes are heterogeneous, in the sense that they come from many different manufacturers, with incompatible instruction sets and word sizes.
- Connections are unreliable.
- Connections are slow.
- Network traffic is bursty, in the sense that there are usually delays between messages that are themselves relatively long.

Packet Switching

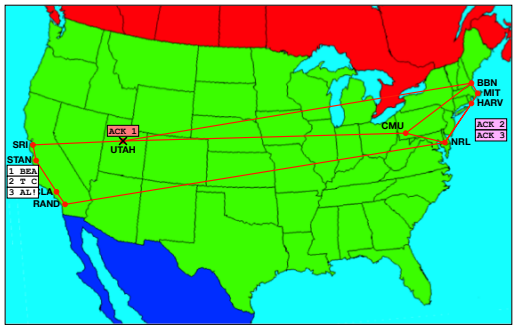


Losing a Packet



After a delay, **HARV** retransmits packet 2, which **STAN** acknowledges.

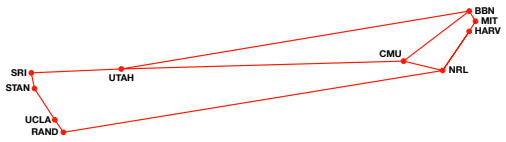
Losing an Acknowledgment



After a delay, **HARV** retransmits packet 1, which **STAN** discards and acknowledges.

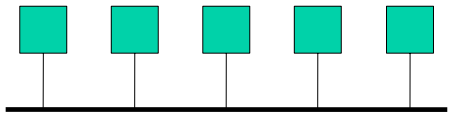
Exercise: Routing

How does each node know where to send each packet?



The Ethernet/AlohaNet Idea

Suppose you have a single communications channel (typically called a *bus*) that is shared among a number of hosts, like this:



How might you structure things to avoid interference on the bus if more than one host tries to use it at the same time?