

## Some Say the Internet Should Never Have Happened

PAUL N. EDWARDS

Nobody who reads this book will need to be told that the Internet is among the modern wonders of the world, a technological marvel that now underpins much of global commerce, communication, and knowledge. Like all wonders, and precisely because of its pervasive presence in every corner of modern life, the Internet's history lies shrouded in myth. The truth-value of such *founding myths* rarely matters as much as their dramatic and moral qualities, and the Internet is no exception.

Good myths need lonely heroes, visionaries embarked on long and arduous journeys in search of sacred grails, which they win through struggle and ordeal. The Internet's origin myth does not disappoint. Once upon a time, it goes, a tiny cadre of computer scientists at elite universities and think tanks realized that connecting computers into a giant network would bring untold benefits, all the way from sharing machines, files, data, and programs to serving as a new Alexandrian library. They knew it would take decades, but they persevered against all odds. Enduring great resistance, they built the first computer networks and then linked them together into the Internet. Funded by the U.S. Defense Department's Advanced Research Projects Agency (ARPA), and blessed by enlightened project managers who trusted their brilliance, their project survived and progressed only because of their dedication and persistence. Their community thrived on an open, meritocratic philosophy, which they embedded in the Internet's technology. In the 1980s, their hard-won victories faced a final peril, when international standards negotiations threatened to recentralize networking and sub-

ject it to government control. In the 1990s, our heroes' generation-long struggle was finally rewarded. The Internet's crown jewel, the World Wide Web, shone forth, bringing a new age of enlightenment in which billions could access the sum of human knowledge. The exponential growth of the Internet and the World Wide Web was unprecedented, unique in human history, world-transforming.

In this story, the Internet should never have happened. The forces arrayed against these lonely heroes seemed invincible. No commercial firm would have invested in internetworking, since it would undermine companies' interests in securing a customer base. Yet in the end, through stealth, persistence, and brilliant ideas, the heroes prevailed at last. According to this myth, without ARPA's long-term vision and massive financial support, computer networking might have been locked into a much more restrictive, centralized form, controlled by byzantine bureaucracies and giant corporations, preventing the flowering of innovation.

Like all good myths, this one has some basis in reality. Brilliant scientists really did spend decades working out Internet protocols and software. Enlightened managers at ARPA really did invest enormous sums and permit researchers to work almost without constraint. Early ARPANET, Internet, and web communities really did enshrine open and meritocratic (if not democratic) principles. Today's descendants of those principles guide the most vital, energetic forms of information technology, especially open-source software (Linux, Firefox), open-access knowledge (Wikipedia), and social networking (MySpace, Facebook). The Internet really has grown explosively, now reaching over a billion people—given a very generous definition of “reach”—and changing the way we live and think.

But all good myths also sacrifice details of reality in the service of exciting plots. There is always more to the story, and the Internet story is no exception.

So I will tell a different version. This tale places the ARPANET and Internet in the wider context of their times: the computing environment and the prevailing beliefs about the economics, the purpose, and the technology of computing and networking. This is not a story of technological determinism, in which one technology begets another as a matter of pure technical logic. Instead, my tale is driven by the conflicting motives of computer users and manufacturers; the incentives (financial, social, etc.) that led to sociotechnical change; visions and principles that differed from those of the ARPANET's designers; and the striking parallels between the Internet and earlier network technologies. Adding miss-

ing pieces reveals a surprisingly different picture—perhaps less exciting, but fuller, more complex, and more useful as a model for future socio-technical systems. In my story, the Internet (or something much like it) *had* to happen, sooner or later.

### The Internet Origin Myth

Before we go further, let me briefly recap the details of the Internet origin myth for readers who spent the last decade or so asleep or on Mars. Usually it goes something like this. Starting around 1960, J. C. R. Licklider, an MIT psychoacoustician turned computer geek, published seminal papers on “Man-Computer Symbiosis” (1960) and “The Computer as a Communication Device” (1968). Licklider’s visionary ideas about a “library of the future” and a “galactic network” of computers (1965) led to his appointment as first head of the Information Processing Techniques Office (IPTO) of the Defense Department’s Advanced Research Projects Agency (ARPA, later known as DARPA). At ARPA, Licklider found himself in a unique position to promote these visions, which he passed on to IPTO successors Ivan Sutherland, Robert Taylor, and others. Unlike virtually any other funding agency, ARPA’s mandate was to promote research whose payoffs might not arrive for ten to twenty years or more. Operating outside the scientific peer review system and generally under the radar of congressional oversight, the agency created “centers of excellence” at a handful of institutions such as MIT, Stanford, and SRI International, bringing the country’s best minds together in a few lavishly funded laboratories (Reed, Van Atta, and Deitchman 1990; Norberg and O’Neill 1996).

Meanwhile, also around 1960, Paul Baran at the RAND Corporation envisioned packet switching in decentralized communication networks to avoid network “decapitation” during a nuclear war (Baran 1964; Reed, Van Atta, and Deitchman 1990). In the mid-1960s, Donald Davies independently developed similar ideas for a civilian network sponsored by the British National Physical Laboratory (Davies et al. 1967). When ARPA initiated a major network project, Larry Roberts, the project’s manager, rediscovered Baran’s ideas by way of Davies. Bolt, Beranek and Newman (BBN)—Licklider’s former employer—won the contract to build a packet-switched ARPANET linking all the ARPA laboratories across the country.

The ARPANET’s purpose, in the minds of its designers, was to permit ARPA researchers to share data and programs residing on the computers

of other ARPA research centers, with a view to both intellectual productivity and efficient resource use. Today this sounds easy, but at the time, ARPA centers had dozens of different, incompatible computers. No one knew how to link them. To solve this problem, the BBN team developed the Interface Message Processor (IMP)—what we would now call a router. IMPs were small computers whose only job was to encode messages from the host computer into packets for distribution through the network, and to decode incoming packets for the host computer. Leased long-distance telephone lines linked the IMPs. By 1969, the ARPANET linked four ARPA labs. Two years later the network spanned the continent, and by 1973 the forty-node network included satellite links to Hawaii and Norway.

Much of the ARPANET protocol development was carried out by the Network Working Group (NWG), initially made up mainly of graduate students. Fearing they might offend their senior colleagues, they published early protocol proposals under the heading "Request for Comments" (RFC). This approach led to a tradition in which all protocols were published initially as proposals, up for discussion by anyone with the technical knowledge to make an intelligent comment:

The content of a NWG note may be any thought, suggestion, etc. related to the HOST software or other aspect of the network. Notes are encouraged to be timely rather than polished. Philosophical positions without examples or other specifics, specific suggestions or implementation techniques without introductory or background explication, and explicit questions without any attempted answers are all acceptable. . . . These standards (or lack of them) are stated explicitly for two reasons. First, there is a tendency to view a written statement as ipso facto authoritative, and we hope to promote the exchange and discussion of considerably less than authoritative ideas. Second, there is a natural hesitancy to publish something unpolished, and we hope to ease this inhibition. (Crocker 1969)

As the tradition evolved into the Internet era, RFCs had to be implemented at least twice on different machines before being formally adopted as Internet standards. This procedure made for rapid, robust, and relatively consensual development.

With the ARPANET working, attention quickly turned to how other computer networks might be linked to the ARPANET in an *internetwork*, or network of networks. By the early 1970s, ARPA program manager

Robert Kahn had initiated an internetworking project with a handful of others, among them Vint Cerf. Kahn and Cerf would come to be known as the fathers of the Internet. With many others, they developed and extended the ARPANET's Network Control Program (NCP) into the more powerful Transmission Control Protocol and Internet Protocol (TCP/IP), standards for linking computers across multiple networks. The ARPA internet evolved from 1974 onward, becoming known simply as "the Internet" in the early 1980s. In the mid-1980s, the U.S. National Science Foundation took on a major role when it linked NSF-funded supercomputer centers via high-speed "backbones" using TCP/IP and encouraged academic institutions to join the network.

But the Internet faced fierce challenges from competing approaches. By the mid-1970s other networking concepts were working their way through the tangled, bureaucratic structures of the International Standards Organization (ISO). The ISO's X.25 network standard (1976) and its successor, the Open Systems Interconnection initiative (OSI), were dominated by large, powerful interests such as corporations and government PTTs (post-telephone-telegraph agencies). The X.25 network model assumed that only a handful of operators would provide networking services, and early versions of OSI took X.25 as a basis. Had these other standards been adopted, the Internet might have been stillborn and all networking dominated by a few large operators. But the negotiations moved slowly. By the time the OSI standard was released in 1983, thousands of systems had already adopted TCP/IP, and a few manufacturers (such as Sun Microsystems) began building TCP/IP into their equipment. TCP/IP had become so entrenched among grassroots users that instead of destroying TCP/IP, the ISO standard was eventually forced to conform to it (Abbate 1999).

In the Internet origin myth, this was a great victory. Small and simple had won out over big and complex. Open, meritocratic, consensus-based development had defeated closed, proprietary, bureaucratic systems supported by huge, powerful entities. Ingenuity had beaten money. Had the ISO standards been finished sooner, goes this tale, TCP/IP might have been crushed, and the Internet would never have happened.

Early uses of the Internet, such as e-mail and (later) Usenet newsgroups, came to embody a libertarian, meritocratic, free-speech philosophy, widely celebrated by the counterculture as an "electronic frontier" where laws and government were unnecessary (Rheingold 1993; Turner 2006). In 1992 David Clark summed up the Internet engineering community's philosophy in a battle cry: "We reject: kings, presidents, and vot-

ing. We believe in: rough consensus and running code” (Russell 2006). Many link the Internet’s libertarian culture to its decentralized technical structure, which makes it very difficult to censor or block communication.

This culture-technology link is the principal source of the origin myth’s power. It’s a strong *technological determinism*, which sees Internet culture as directly and causally linked to such technical characteristics as packet switching and TCP/IP. Had computer networks evolved as more hierarchically organized systems, the wild cyber frontier might have looked more like a cyber subdivision, with cookie-cutter houses and lawn-care rules set up for the profit of a few huge corporations.

### Homogeneous Networks and the “Computer Utility”

It’s a powerful myth—and as I have said, much of it is actually true. But consider: if ARPA and its network were really the visionary hero of our tale, what about the other networks that needed to be joined to form the ARPA Internet? Where did they come from, and what were they for? How did contemporaries see the future of computing and networks in the early days of ARPANET development? Would the Internet have happened anyway, with or without ARPA?

To understand this crucial piece of context, we need to look back to the 1960s, when the computer industry first came to maturity. In that era, two “laws” of computing competed for the status of common sense. Moore’s law, first articulated in 1965, predicted that the number of transistors on a silicon chip would double approximately every twenty-four months. Moore’s law still holds today.<sup>1</sup> But in the 1960s—before microprocessors, personal computers, and the Internet—another prediction known as Grosch’s law held sway.

Working with some of the earliest computers, at Columbia University in the 1940s, Herbert Grosch devised a dictum initially phrased as “economy [increases] only as the square root of the increase in speed—that is, to do a calculation ten times as cheaply you must do it one hundred times as fast” (Grosch 1953, 310). This somewhat puzzling assertion later morphed into the more useful claim that “the performance of a computer varies as the square of its price,” or alternatively that “the average cost of computing decreases as the square root of the power of a system” (Ein-Dor 1985). Whatever its formulation, Grosch’s law essentially held that computers displayed dramatic economies of scale. Double your spending on a computer system and you could buy four times as much

“power.” Triple it, and you could get nine times as much power, and so on.

Economists had considerable trouble finding a real-world metric for “computer power,” and some writers occasionally wondered whether Grosch’s “law” originated in IBM’s marketing department (Hobbs 1971). But in its day the principle was very widely accepted. Larger contexts mattered; similar arguments about economies of scale dominated much corporate thinking in the 1950s and 1960s. Analogies to electric power plants, automobile companies, and other examples produced a “bigger is better” mentality that seemed to apply equally to computing. As a result, corporate data-processing departments typically sought both organizational and technological centralization, buying ever-larger, ever-faster computers. Reviewing the literature in 1983, King noted that “until the end of the 1970s . . . articles on computing centralization were nearly unanimous: . . . centralization saves money” (1983, 322).

When time-sharing technology arrived, in the early 1960s, it seemed to vindicate Grosch’s law. Time-sharing allowed several people to use the same computer simultaneously. Previously, computers could run only one program at a time (“batch processing”). With time-sharing, the computer could run many programs at once, cycling from one user to the next so quickly that users never knew that someone else was using the same machine—at least in principle. Teletype and, later, CRT terminals, either directly connected or using modems, untethered users from direct proximity to the machine. Time-sharing’s proponents sought to end the tyranny of batch processing and its necessary evil, the computer operator who stood between users and the machine. In an age when most computers cost \$50,000 or more, time-sharing promised an experience much like today’s personal computing: the feeling, at least, that the machine was yours and yours alone. Furthermore, time-sharing maximized the CPU’s value; before time-sharing, the CPU spent most of its time idle, waiting for slower input-output devices to do their work.

Time-sharing made a particular path to personal computing seem obvious and inevitable. Because of Grosch’s law, a relatively small incremental investment would bring time-shared computing to a much larger number of users. Maybe one day a single giant computer could be shared by everyone. Between 1965 and 1970, every major U.S. manufacturer announced plans to market time-sharing computers.

These arguments convinced many people that computers should be treated like power plants. It made no sense to put an electric generator in every household, when a single giant power plant could do the job

for a whole city more reliably and far more cheaply. By analogy, it made no sense for individuals to have their own computers if a single giant computer could do the job for everyone—better, faster, more reliably, and more cheaply. In 1964, in a widely read *Atlantic Monthly* article, MIT professor Martin Greenberger wrote that “an on-line interactive computer service, provided commercially by an *information utility*, may be as commonplace by 2000 A.D. as the telephone is today.”

Superficially, this sounds a lot like the World Wide Web. Yet beneath the surface of the computer utility model lay a completely different imagining of networked computing:

Computation, like electricity and unlike oil, is not stored. Since its production is concurrent with its consumption, production capacity must provide for peak loads, and the cost of equipment per dollar of revenue can soar. The high cost of capital equipment is a major reason why producers of electricity are public utilities instead of unregulated companies. A second reason is the extensive distribution network they require to make their product generally available. This network, once established, is geographically fixed and immovable. Wasteful duplication and proliferation of lines could easily result if there were no public regulation (Greenberger 1964, 65).

In Greenberger's vision, users would interact with the central computer through “consoles” or other dumb terminal equipment, sharing programs, resource libraries, and other information via the central computer. Following the power company analogy, “the distribution network” would be like power lines, a cable system for delivering the centralized resource to the end user. Thus the earliest popular notion of a computer “network” referred to a computer center accessed by remote terminals.

These early systems can all be classified as *homogeneous networks*. Organized in hub-and-spoke fashion around one or a few central nodes (computers), this network model assumed a single type of computer and a single system for communicating among them. And indeed many such systems quickly emerged. The earliest were transaction-processing systems such as the SABRE airline reservation system, which allowed remote clients to access centrally maintained databases and carry out transactions such as making a travel reservation or entering a check into a banking system. In the mid-1960s, computer service bureaus sprang up around the world, renting computer time directly and offering value-added data-processing services. As time-sharing technology matured, some systems



allowed users to access the computers directly via remote terminals, adding interactivity to the list of features.

By 1967 some twenty firms, including IBM and General Electric, had established computing service bureaus in dozens of cities. In a preview of the 1990s dot-com bubble, these firms' stocks swelled on the can't-lose promise of the computer utility:

Dallas-based University Computing Company (UCC) . . . established major computer utilities in New York and Washington. By 1968 it had several computer centers with terminals located in thirty states and a dozen countries. UCC was one of the glamour stocks of the 1960s; during 1967-68 its stock rose from \$1.50 to \$155. (Campbell-Kelly and Aspray 1996, 218)

By 1969 Tymshare's TYMNET, Control Data's CYBERNET, CompuServe, and numerous other firms had entered this exploding market. TYMNET—in a move that originated with an ARPA-sponsored project—began centralizing its network along exactly the lines envisioned by Greenberger, with the goal of “consolidating its computers into one location and transfer[ring] the computer power from that location to where it was needed” (Beere and Sullivan 1972, 511). In the language of the day, these networks were known as RANs, for remote-access networks.

The computer utilities sought economies not only in scale, but also in homogeneity. Each computer center would operate only one or a few computer models, all made by a single manufacturer, using a standard package of software and a single operating system. This aspect of the computing environment of the 1960s can hardly be overemphasized: in those days, making computers from different manufacturers work together required major investments of time, money, and training, so much that in practice most computer users did not even attempt it.

Unfortunately for most of the computer service bureaus, the technology of time-sharing never worked in quite the way that Grosch's law implied. IBM's initial foray, the System 360/67 with TSS (Time-Sharing System), sold sixty-six exemplars by 1971, but endured innumerable software problems, leading to losses of \$49 million on the 360/67 line (O'Neill 1995, 53). Similar difficulties marred all other early time-sharing projects. Though MIT's famous MULTICS project of the mid-1960s envisioned 1,000 remote terminals and up to 300 concurrent users, writing software for managing so much simultaneous activity proved far more difficult than anticipated. Constant crashes and reboots led to

wails of agony from frustrated users. In the end, no matter how big and powerful the computer, time-sharing simply did not scale beyond a limit of around fifty users. Ultimately, as Campbell-Kelly and Aspray note, Grosch's law failed because Moore's law applied. The price of computer power dropped like a stone, rapidly wiping out the economic rationale behind the time-sharing service model. UCC's stock crashed in 1971 (Campbell-Kelly and Aspray 1996, 218).

The fantasy of a single giant computer serving the entire country never materialized. Yet the computer utility model failed to die. Many of the 1960s network services, including TYMNET and CYBERNET, survived for decades, albeit in a remodeled form. The information utility model eventually morphed into a consumer product, under corporate giants such as CompuServe, Prodigy, and America Online. Grosch's law, too, experienced a long afterlife. In the 1970s and beyond, the inherent logic of the homogeneous, centralized network remained obvious, beyond dispute for many.

Users with nothing more than a terminal and access to the telephone may select from a large number of potential suppliers [of basic computing services]. Since geography is no longer of major concern . . . a sufficiently large complex, wherever located, may offer basic services to users anywhere in the country at prices lower than they can obtain locally. (Cotton 1975)

Computer manufacturers proceeded to develop proprietary homogeneous networking systems, among them IBM's SNA (System Network Architecture) and Digital Equipment Corporation's DECNET. These homogeneous networks could be decentralized, but they remained linked to a single manufacturer's proprietary standards and could not be connected to those of others. Well into the 1980s, the vast majority of computer networks remained homogeneous and proprietary. As we will see, not only did this not represent a failing on their part, it was a principal reason for the Internet's success in the 1990s.

#### The ARPANET, Packet Switching, and Heterogeneous Computing

The ARPANET project faced two basic problems: how to get the widely dispersed labs to communicate efficiently over long distances, and how to get the labs' computers to work together. These may sound like two

versions of the same difficulty, but in fact they are completely different issues.

In the standard origin story, communication between the ARPA labs gets most of the attention. Both cost and reliability mattered. Connecting every lab directly to every other one using expensive leased phone lines would drive costs upward exponentially. If direct connections were required, this would also make the network less reliable, with each new connection a potential point of failure.

Famously, the ARPANET solved both cost and reliability problems through a technique known as packet switching. In this system, each lab was directly connected to a few others, but not to all. Outgoing messages were divided into "packets," small units of data. The IMPs (Interface Message Processors) assigned each packet a message header containing its destination and its place in the bitstream of the overall message. IMPs then "routed" each packet individually, to another IMP closer to the destination. Using the address in message header, the receiving IMP would forward the packet onward until it reached its final destination. There, the receiving IMP would reassemble the bitstream into its original order to make up the entire message. The IMPs could sense when a connecting circuit was overloaded and use another one instead; every packet might conceivably take a different route. In principle, at least, no longer did each computer have to be directly connected to every other one. Nor would a down IMP delay an entire message; instead, the routers would simply choose another of the many possible paths through the network. Less busy circuits could be used more efficiently. In practice the IMP system proved far less reliable than anticipated (Abbate 1999). In the long run, though, packet switching made completely decentralized communication networks nearly as efficient as, and far more reliable than, alternative hub-and-spoke or hierarchical network designs.

Packet switching also advanced a key military ambition: building a command-and-control network that could survive a nuclear war. Since destroying even a large number of nodes would not prevent packets from reaching their destinations, a packet-switched network could endure a lot of damage before it would fail, as Baran's early RAND studies had shown. Depending on which version of the Internet story you consult, you will read either that designing survivable command and control was the ARPANET's main goal, or that this motive had absolutely nothing to do with it. In my view, *both* stories are true. ARPANET packet switching ideas came in part from Baran's work, communicated to Larry Roberts

via Donald Davies (though Roberts himself credited Leonard Kleinrock with the earliest and most influential concepts [Roberts 2004; Kleinrock 1961, 1964; Abbate 1999]). In interviews, Licklider repeatedly noted that solving military command-control problems was IPTO's original purpose. He himself wrote about these problems in a number of places, including his seminal "Man-Computer Symbiosis" (Lee and Rosin 1992; Licklider 1960, 1964, 1988).

Still, once the project really got rolling, four years after Licklider left ARPA in 1964, the data- and program-sharing, community-building, and efficiency-improving goals drove ARPANET's designers, who mostly cared little about military applications. Nor did the agency ask those designers to think about military potential, at least at first. Yet as I have argued elsewhere, in the Cold War climate many military goals were accomplished not by directly ordering researchers to work on something, but simply by offering massive funding for research in uncharted areas of potential military interest (Edwards 1996).

By the early 1970s ARPA directors *were* starting to demand military justifications for ARPA research. Unclassified military computers were connected to the ARPANET early on, and by 1975 ARPA had handed over ARPANET management to the Defense Communication Agency (DCA). In the early 1980s, nervous about potential unauthorized access and hacking, the DCA physically split off the ARPANET's military segment to form a separate network, MILNET. In the event, nuclear command and control networks never physically linked to either ARPANET or MILNET.

This half of the ARPANET problem—how to build a large, geographically dispersed, fast, reliable network—represented what historians of technology call a *reverse salient*: a major problem, widely recognized, that held back an advancing technological system and drew enormous attention from firms, scientists, and engineers as a result (Hughes 1983, 1987). By 1968 numerous corporate research laboratories and academic research groups were working on it (many with ARPA support), and solutions very similar to the ARPANET were emerging. Packet switching itself had already been tested in Davies' small UK pilot project.

Furthermore, the IMP solution exactly mirrored the dominant network concept of its day. Since the IMPs were identical minicomputers, the "network" part of the ARPANET (in which only the IMPs were directly connected) was a homogeneous one very much like its numerous corporate cousins. TYMNET, for example, also used small minicomputers to link remote sites with its mainframes. The same can be said for the

ARPANET's goals: sharing data and programs was also the principal idea behind the commercial networks. (Alex McKenzie, an early manager of the ARPANET's Network Control Center, saw the project precisely as a "computer utility" [Abbate 1999, 65].) Though often described as revolutionary, in these respects the ARPANET, while important, did not constitute a major innovation.

Instead, the ARPANET's most significant technical achievements occurred around the other half of the problem: making the labs' incompatible computers speak to each other. By the mid-1960s the agency—having left purchasing decisions to the labs—found ARPA research centers using dozens of incompatible computers. This was a deeply hard problem; as anyone who has wrestled with connecting Macs, Windows, Linux, and Unix machines knows, it remains hard even today. Some directly involved in the ARPANET project failed to see the point of even trying:

We could certainly build it, but I couldn't imagine why anyone would want such a thing. . . . "Resource sharing" . . . seemed difficult to believe in, given the diversity of machines, interests, and capabilities at the various [ARPA] sites. Although it seemed clear that with suitable effort we could interconnect the machines so that information could flow between them, the amount of work that would then be required to turn that basic capacity into features useful to individuals at remote sites seemed overwhelming—as, indeed, it proved to be. (Ornstein 2002, 165–66)

But ARPA had the money to pay for it, and (crucially) the ability to force its labs to try the experiment.

The ARPANET's "host-to-host" protocols required separate, unique implementation on each host computer, a long, painful effort. Initial host-to-host protocols concentrated simply on "direct use," such as allowing remote users to log in and run programs on the host computers as if using a terminal (the TELNET protocol). Direct-use protocols required that the remote user know the language and conventions of the host computer. Since few users had mastered more than one or two different machines and operating systems, by itself this remote login capability did little to improve the situation.

By 1971 the first protocol for "indirect use" had emerged: file transfer protocol (FTP), which permitted a remote user to download a file from the remote machine, or to upload one (Bhushan 1971). FTP was

a simple, modest initial step toward cross-platform communication (as we would call different computer/operating systems today). Depending on what it contained, the user still had to work out how to decode the file and potentially convert it into a format usable on his or her own machine. A few standards already existed, such as those defining text files (the ASCII character set), so plain text could be sent from computer to computer with relative ease. As a result, FTP became the basis of ARPANET e-mail (which simply transmits a text file from one computer to another). As many have pointed out, this very simple innovation proved a key to the ARPANET's growth. Within a year, e-mail made up half the traffic on the ARPANET. Further, much of the e-mail had nothing to do with sharing data and programs; it was social and informal. No one had planned for this use of the system, which directly challenged the ARPA managers' resource-sharing model.

Popular accounts make much of e-mail's surprise emergence, as another supposedly revolutionary feature of the network. Yet e-mail long predated the ARPANET, existing under other names in time-sharing systems from at least 1965 (at MIT). And while it is true that the explosion of e-mail traffic made the ARPANET into the phenomenon it became, similar developments have occurred in all other communication technologies. Tom Standage's *The Victorian Internet* showed how telegraph operators—who could communicate with each other over the international telegraph network at will—developed a remarkably similar culture of informal communication (1998). And the same thing happened again with early radio, and yet again with telephone chat (Fischer 1988, 1992). In all of these cases, *users rapidly co-opted media initially developed for formal, official, or institutional purposes, turning them instead toward informal social communication.*

A similar phenomenon occurred again in 1979, when users adapted another file transfer program, uucp (Unix to Unix copy), to create Usenet newsgroups. Usenet depended on the Unix operating system, then becoming popular in academia but not yet a mainstay of business or government computing (Hauben and Hauben 1997). Early newsgroups primarily treated technical topics, but they soon expanded to included discussions of virtually anything, from *Star Trek* to sex. Like e-mail, Usenet rapidly became a dominant feature of the evolving global computer network.

As a result, Usenet has been absorbed into the Internet myth, often as Exhibit A in the argument for a special Internet culture. Yet for a number of years Usenet had virtually nothing to do with either the ARPANET

or the Internet. As late as 1982, only a handful of Usenet hosts were even linked to the ARPANET:

Can anybody tell me how to send mail from this uucp node (cwruecmp) to a user at an ARPANET node (specifically, HI-Multics) and back? I have tried every path I know, and they all get returned from Network:C70. Help! (Bibbero 1982; see McGeedy 1981)

Indeed, the Usenet community experienced the ARPANET not as an open, meritocratic space, but as an elite system available only to a chosen few (as in fact it was [Hauben 1996, chap. 9]).

By the early 1980s, computer networking had already exploded on many fronts, but very few of them connected to the ARPANET or used the TCP/IP protocols (adopted as the ARPANET standard in 1982). With personal computing emerging as a major force, e-mail, listservs (mailing lists), and bulletin board systems had sprung up on many scales, from local operators providing homegrown dial-up connections in small towns to gigantic national firms. France deployed its nationwide Minitel network in 1981; within a couple of years, the *messageries roses* (semipornographic chat and dating services, much like bulletin boards) became a colossal business (De Lacy 1989). CompuServe had built its own, nationwide packet-switched network in the 1970s, providing e-mail and bulletin board services to both consumers and corporate clients (including Visa International, for which its network handled millions of credit card transactions each day). BITNET, an academic network founded in 1981, initially used IBM networking software: "By the end of the 1980s [BITNET] connected about 450 universities and research institutions and 3000 computers throughout North America and Europe. By the early 1990s, BITNET was the most widely used research communications network in the world for e-mail, mailing lists, file transfer, and real-time messaging," including listservs similar to the Usenet newsgroups (Stewart 2000). All of these and numerous other computer networks deployed a wide variety of networking schemes, including the much-maligned X.25 and proprietary schemes. Furthermore, some, such as BITNET, created their own protocols for linking to other networks—ignoring TCP/IP, but forming internetworks of their own. Although censorship did occur to varying degrees, in virtually all of these systems a culture of freewheeling, unconstrained online discussion emerged that had absolutely nothing to do with the Internet. The most celebrated of these, the Whole Earth 'Lectronic Link

(WELL) bulletin board system, was founded in 1985. If in these early years BITNET had continued to use IBM proprietary software rather than switch over to TCP/IP after just two years, would the now *path dependent* QWERTY inevitability of TCP/IP and its celebrated culture have prevailed?

In 1984, after more than ten years of development, the Internet counted only about 1,000 hosts. That was the year William Gibson coined the term *cyberspace* in his novel *Neuromancer*. If the Internet inspired him, it shouldn't have, because only after that did really serious Internet growth begin, reaching 10,000 hosts by 1987; 100,000 by 1989; and 1,000,000 hosts by 1991. The reason for this explosion, then, was simple: when other networks "joined" the Internet through TCP/IP links, each one added hundreds or thousands of hosts. By the mid-1980s, Unix had become a de facto standard in academic computing and high-end workstations (e.g., Sun Microsystems); since it provided built-in IP support, Unix networks could easily join the Internet. TCP/IP made it easy, but the thing that made the Internet explode was—other networks.

#### Conclusion: From Systems to Networks to Webs

Far from being unique, the Internet story has striking analogues throughout the history of technology. The problem it represents has occurred over and over again in the history of network technologies. In the nineteenth century, differing track gauges, car linkages, and other mechanical features of early railroads prevented one company from using another's track or rolling stock. Only after standards were agreed on (or forced) did railroad transport become a true national network in which multiple railroad companies could use each other's tracks and rolling stock. Similarly, early electric power networks used different voltages—dozens of them—as well as both alternating current (AC) and direct current (DC). Yet eventually these differences were settled and the many individual networks linked into gigantic national and international power grids. Rail, highway, and shipping networks formed separately according to their own dynamics. Yet the ISO standard container—along with railroad cars, trucks, ships, and ports all adapted to fit it—now links these networks, so that a container can travel from one to the next in a virtually seamless process much like packet switching (Egyedi 2001). Examples from telephone, telegraph, television, and many other technical systems can be multiplied ad infinitum.



All of these stories follow a similar pattern. They begin with proprietary, competing, incompatible systems. Then someone creates a *gateway* that allows these systems to interoperate, as networks. Finally, incompatible networks form internetworks or webs. This process is driven not by technology itself, but rather by the demands of users, who frequently lead the process through direct innovation (technical, social, political) of their own.

Gateways—converters or frameworks that connect incompatible systems—are the key to the transition from systems to networks to webs. For example, in the late nineteenth century the AC/DC power converter formed a gateway between AC- and DC-based electric power networks, enabling competing utilities to work together or merge (David and Bunn 1988). Devices like the AC/DC converter, however, are neither the only nor necessarily the most important form of gateway: standards (such as track gauge or the ISO container), rules/laws (such as the NSF prohibition on commercial use of the Internet, lifted in 1991), institutions (money, the stock market), languages (pidgins, lingua francas), and virtually anything else can serve the gateway function, depending on the circumstances (Edwards et al. 2007).

The history of computing follows this pattern precisely. Initially, computers were little more than stand-alone calculators. Within a few years, people started trying to connect them; the result was networks. But just as no system can provide every function a user might need, no network could provide all of the connectivity a user might desire. Linking the networks into internetworks was not a stroke of genius, but a straightforward, entirely logical step. This story *seems* unique mainly because people forget too easily.

Yet it is also true that this case of the system-to-network-to-web transition represented an extremely difficult case. Computers gain their awesome power from awesome complexity; linking heterogeneous computers, and then linking heterogeneous networks, were exceptional feats of engineering, rightly celebrated.

The end of my story, then, is this. It's all true: the Internet really is a revolution, the product of a generation-long effort by thousands of brilliant and dedicated people. It should never have happened. Yet despite its difficulty, it was completely and utterly inevitable. Both the facts of the case and the example of the past belie the notion that government PTTs or big commercial networks might have dominated networking forever.

I see two reasons why this had to be true. First, more than in almost any other network technology, computing's designers are also its heavi-

est users. This fact is arguably responsible for many computer usability problems (Landauer 1995), but it has also produced a virtuous cycle in which designers keep finding new ways to improve how computers work (Castells 2001). The second reason is that Moore's law (with a lot of help from ingenious engineers) has created a world chock full of ever-smaller, ever-cheaper computers. Computers are language machines. So are people. Our most human feature—our need to communicate, about anything and everything, all the time—would eventually have produced the “galactic network” on whose doorstep we now stand.

#### NOTE

1. Moore actually revised his prediction a number of times after the original 1965 article (Moore 1965). The historical course of chip development has varied, sometimes widely, from the smooth curve usually depicted in discussions of the “law.” Nonetheless, taken as a heuristic rather than a precise constant, it has proven remarkably accurate for more than fifty years (Mollick 2006).

#### REFERENCES

- Abbate, Janet. 1999. *Inventing the Internet*. Cambridge: MIT Press.
- Baran, Paul. 1964. *On Distributed Communications*. RAND Memorandum Series. Santa Monica, CA: RAND Corporation.
- Beere, Max P., and Neil C. Sullivan. 1972. “TYMNET—a Serendipitous Evolution.” *IEEE Transactions on Communications* 20 (3, pt. 2): 511–15.
- Bhushan, Abhay. 1971. “A File Transfer Protocol.” Network Working Group, RFC-114.
- Bibbero, I. 1982. arpa-uucp mail. Republished in Usenet Oldnews Archive: Compilation. Retrieved April 5, 2009, from [http://quux.org:70/Archives/usenet-a-news/NET.arpa-uucp/82.04.08\\_cwruecmp.65\\_net.arpa-uucp.txt](http://quux.org:70/Archives/usenet-a-news/NET.arpa-uucp/82.04.08_cwruecmp.65_net.arpa-uucp.txt).
- Campbell-Kelly, Martin, and William Aspray. 1996. *Computer: A History of the Information Machine*. New York: Basic Books.
- Castells, Manuel. 2001. *The Internet Galaxy: Reflections on the Internet, Business, and Society*. New York: Oxford University Press.
- Cotton, I. W. 1975. “Microeconomics and the Market for Computer Services.” *ACM Computing Surveys (CSUR)* 7 (2): 95–111.
- Crocker, Steve. 1969. Documentation Conventions. Network Working Group, RFC-3.
- David, Paul A., and Julie Ann Bunn. 1988. “The Economics of Gateway Technologies and Network Evolution: Lessons from Electricity Supply History.” *Information Economics and Policy* 3:165–202.
- Davies, D. W., K. A. Bartlett, R. A. Scantlebury, and P. T. Wilkinson. 1967. “A Digital Communication Network for Computers Giving Rapid Response at Remote Terminals.” *Proceedings of the First ACM Symposium on Operating System Principles* 2.1–2.17.

- De Lacy, Justine. 1989. "The Sexy Computer." In *Computers in the Human Context*, ed. Tom Forester, 228-36. Cambridge: MIT Press.
- Edwards, Paul N. 1996. *The Closed World: Computers and the Politics of Discourse in Cold War America*. Cambridge: MIT Press.
- Edwards, Paul N., Steven J. Jackson, Geoffrey C. Bowker, and Cory P. Knobel. 2007. *Understanding Infrastructure: Dynamics, Tensions, and Design*. Ann Arbor: Deep Blue.
- Egyedi, Tineke. 2001. "Infrastructure Flexibility Created by Standardized Gateways: The Cases of XML and the ISO Container." *Knowledge, Technology and Policy* 14 (3): 41-54.
- Ein-Dor, P. 1985. "Grosch's Law Re-visited: CPU Power and the Cost of Computation." *Communications of the ACM* 28 (2): 142-51.
- Fischer, Claude S. 1988. "Touch Someone': The Telephone Industry Discovers Sociability." *Technology and Culture* 29 (1): 32-61.
- Fischer, Claude S. 1992. *America Calling: A Social History of the Telephone to 1940*. Berkeley and Los Angeles: University of California Press.
- Greenberger, Martin. 1964. "The Computers of Tomorrow." *Atlantic Monthly* 213 (5): 63-67.
- Grosch, H. A. 1953. "High Speed Arithmetic: The Digital Computer as a Research Tool." *Journal of the Optical Society of America* 43 (4): 306-10.
- Hauben, Michael, and Ronda Hauben. 1997. *Netizens: On the History and Impact of Usenet and the Internet*. Los Alamitos, CA: IEEE Computer Society Press.
- Hauben, Ronda. 1996. *The Netizens' Netbook*. <http://www.columbia.edu/~hauben/netbook/>.
- Hobbs, L. C. 1971. "The Rationale for Smart Terminals." *Computer*, November-December, 33-35.
- Hughes, Thomas P. 1983. *Networks of Power: Electrification in Western Society, 1880-1930*. Baltimore: Johns Hopkins University Press.
- Hughes, Thomas P. 1987. "The Evolution of Large Technological Systems." In *The Social Construction of Technological Systems*, ed. Wiebe Bijker, Thomas P. Hughes, and Trevor Pinch, 51-82. Cambridge: MIT Press.
- King, J. L. 1983. "Centralized versus Decentralized Computing: Organizational Considerations and Management Options." *ACM Computing Surveys (CSUR)* 15 (4): 319-49.
- Kleinrock, L. 1961. "Information Flow in Large Communication Nets." *RLE Quarterly Progress Report*, July.
- Kleinrock, Leonard. 1964. *Communication Nets: Stochastic Message Flow and Delay*. New York: McGraw-Hill.
- Landauer, Thomas K. 1995. *The Trouble with Computers: Usefulness, Usability, and Productivity*. Cambridge: MIT Press.
- Lee, John A. N., and Robert Rosin. 1992. "The Project MAC Interviews." *IEEE Annals of the History of Computing* 14 (2): 14-35.
- Licklider, J. C. R. 1960. "Man-Computer Symbiosis." *IRE Transactions on Human Factors in Electronics HFE-1* (1): 4-10.
- Licklider, J. C. R. 1964. "Artificial Intelligence, Military Intelligence, and Command and Control." In *Military Information Systems: The Design of Computer-Aided Systems for Command*, ed. Edward Bennett, James Degan, and Joseph Spiegel, 119-33. New York: Frederick Praeger.
- Licklider, J. C. R. 1965. *Libraries of the Future*. Cambridge: MIT Press.

- Licklider, J. C. R. 1988. "The Early Years: Founding IPTO." In *Expert Systems and Artificial Intelligence*, ed. Thomas C. Bartee, 219-27. Indianapolis: Howard W. Sams.
- Licklider, J. C. R., and R. W. Taylor. 1968. "The Computer as a Communication Device." *Science and Technology* 76 (2): 1-3.
- McGeady, S. 1981. Usenet Network Map. Republished in Usenet Oldnews Archive: Compilation. Available from [quux.org:70/Archives/usenet-a-news/NET.general/82.01.06\\_wivax.1043\\_net.general.txt](http://quux.org:70/Archives/usenet-a-news/NET.general/82.01.06_wivax.1043_net.general.txt).
- Mollick, E. 2006. "Establishing Moore's Law." *IEEE Annals of the History of Computing* 28 (3): 62-75.
- Moore, Gordon E. 1965. "Cramming More Components on Integrated Circuits." *Electronics* 38 (8): 114-17.
- Norberg, Arthur L., and Judy E. O'Neill. 1996. *Transforming Computer Technology: Information Processing for the Pentagon, 1962-1986*. Baltimore: Johns Hopkins University Press.
- O'Neill, Judy E. 1995. "'Prestige Luster' and 'Snow-Balling Effects': IBM's Development of Computer Time-Sharing." *IEEE Annals of the History of Computing* 17 (2): 51.
- Ornstein, Severo M. 2002. *Computing in the Middle Ages: A View from the Trenches, 1955-1983*. 1st Books Library.
- Reed, Sidney G., Richard H. Van Atta, and Seymour J. Deitchman. 1990. *DARPA Technical Accomplishments: An Historical Review of Selected DARPA Projects*. Alexandria, VA: Institute for Defense Analyses.
- Rheingold, Howard. 1993. *The Virtual Community: Homesteading on the Electronic Frontier*. Reading, MA: Addison-Wesley.
- Roberts, Lawrence. 2004. Internet Chronology, 1960-2001. Available from [www.packet.cc/internet.html](http://www.packet.cc/internet.html).
- Russell, A. L. 2006. "'Rough Consensus and Running Code' and the Internet-OSI Standards War." *IEEE Annals of the History of Computing* 28 (3): 48-61.
- Standage, Tom. 1998. *The Victorian Internet: The Remarkable Story of the Telegraph and the Nineteenth Century's On-Line Pioneers*. New York: Walker and Company.
- Stewart, Bill. 2000. Bitnet History. Retrieved from [www.livinginternet.com/u/ui\\_bitnet.htm](http://www.livinginternet.com/u/ui_bitnet.htm).
- Turner, Fred. 2006. *From Counterculture to Cyberculture: Stewart Brand, the Whole Earth Network, and the Rise of Digital Utopianism*. Chicago: University of Chicago Press.