Journal Title: Computing in the middle ages ; a view from the trenches 1955-1983 /

Vol. , Is.

Month/Year: 2002

Pages: 156-187

Article Title: Severo Ornstein; Chapter 14

Article Author: Ornstein, Severo M.

For: Edwards, Paul N
At: 106 Hatcher Graduate Library ILL - University of Michigan (EYM)

Supplied to you by
Baylor University
Access Services

CONNECT BU LIBRARIES

OCLC = IYU

- Not On Shelf
- Not As Cited
- Lack vol/issue
- Not Yet Received
- Holdings end before vol/issue
- Holdings start after vol/issue
- At Bindery
- Poor Condition

BU Library Use Only

Group: IFM Libraries
MaxCost: $26.00IFM
Aging Date: 20100214
ZYU,ZYU,*IYU,IHY

BL

Borrower email:

ILL Number: 62666546

Odyssey: 141.211.175.136

Ariel: 141.211.175.21

Ship Via:
Ariel
Document ID (ILL#): 62666546

Patron: Edwards, Paul N

To (Ariel Address): 141.211.175.21

Ship Via: Ariel -- To: EYM 2/14/2010 4:06 PM BU OsoFast TN# 371221

Chapter 14

I join BBN; some comments on ARPA; Encounters with TENEX; some coffee-tables, and a teletype through the wall. The ARPANET is born

Bolt Beranek and Newman (BBN) was a somewhat unusual firm: it had many ties with MIT and liked to think of itself as half-way between a company and a research institute of some sort. It had been founded in the 1940s by three MIT professors in architectural acoustics who had given the company its name, but by this time it had grown considerably and had strayed into computers as well as other fields only remotely related to acoustics.[30] J.C.R. Licklider, who became a grand old man of computer science, had worked there on an early experimental Time Sharing system on a DEC PDP-1 computer. (The PDP-1 was the first full-blown computer that DEC made and BBN's was the very first of the series. Ben Gurley, a former TX-2 engineer, had designed the PDP-1 with some oversight from Wes Clark.) By the time I arrived on the scene, BBN was running a commercial Time Sharing system

---

[30] One wit quipped that the only reason BBN didn't run a brothel in Cambridge was that they didn't have the appropriate talent. This was unduly harsh, but after years of academic association, BBN did seem comparatively "commercial" to me. *Time-sheets,* for heavens sake!

(TELENET) and had two computer divisions. The one I was joining was headed by Frank Heart, and its mission was to design and build a wide variety of computer systems for clients. The other, headed by Jerry Elkind, was more explicitly research-oriented and, among other things, was developing experimental Time Sharing systems supported by government money.

In computer research, government money had increasingly come to mean ARPA money, because ARPA had become the agency that provided by far the largest share of government funding for computer research.[31] Initially hired by Jack Ruina, then director of ARPA, J.C.R. Licklider had set up an Information Processing Techniques Office (IPTO) within ARPA to oversee and fund research in the burgeoning computer field. The National Science Foundation and the National Institutes of Health continued some support, but far below the level of the funds available to ARPA through the Defense Department's always ample budget.

Of all the money that the Defense Department spent, that which was funneled through the ARPA office (as IPTO was known within the computer community) was unquestionably some of the very most productive. It fostered much of the computer research and development that took place in this country during the middle-ages. Much of that work took place at universities, but a few places with close connections to Universities such as BBN

[31] As mentioned earlier, some of the support for our Macromodule research had come from ARPA.

and the Stanford Research Institute (later SRI[32]) which were centers of excellence outside of a university, also gained ARPA support for computer research.

I was personally always unhappy about this situation. It was part of a more general concern that so much of government-funded research was focused on military matters and funded through the Department of Defense (DOD). I felt that this distorted national research priorities and left far too little for other, non-military concerns. However, although DOD of course hoped that military applications would eventually be found for the products of the research ARPA funded, the money from IPTO was largely dedicated to very general, quite unspecific computer research. When developing tools as broadly applicable as a "general purpose computer," it is impossible to set limits on what one might be used for. In any case, during the mid-1960s it was nearly impossible to do significant computer research anywhere in the U.S. (except perhaps at IBM) without working indirectly for ARPA one way or another.

As first head of IPTO, Licklider had established a standard of excellence that was to persist for many years beyond his own tenure. The people who followed him—Ivan Sutherland, Bob Taylor, Larry Roberts, Bob Kahn—were outstanding individuals who dedicated a few years of their lives to overseeing federally-sponsored research in the field. Of course it was an important and powerful position,

---

[32] At that time, SRI was a part of Stanford University. It was changed into a separate nonprofit corporation sometime in the 1970s, which was when its official name was shortened to "SRI." That happened during the Vietnam War as a result of student activism that banished all classified research from the University.

but I also came to realize that the job entailed elements of personal and professional sacrifice in that these individuals could have had (and eventually did have) jobs with many more emoluments outside of government, perhaps even as ARPA-funded researchers. I remember riding to the airport one day with Larry Roberts when he was head of the office, and discovering that as a government employee he was constrained to use the cheapest available car rental, which meant that we had to be ferried from some distant rental site to the airport.

It may seem strange that I ended up in what was nominally the less research-oriented division of BBN. This had primarily to do with my long-standing friendship with Frank, but it also had to do with the fact that the other division was focused heavily on research in Time Sharing systems. This Big Deal concept was anathema to the Little Deal "Gospel according to St. Louis," that I arrived bearing. But beyond these factors, it's also incorrect to characterize the construction of dedicated computer systems as non-research. Designing and building the kinds of systems Frank's division was working on involved plenty of research. It's just that in addition to the research, there were commitments to finished systems, hard deadlines, etc. It seems that often one person's research is another person's routine engineering. In any case, the term "research," (like terms such as "executive" and "mansion"), has become so over-used that it has lost much of its original pith.

Although in Frank's division a significant effort was under way in medical computing, it was of a very different sort from the direct, real-time laboratory use in which I'd been involved. I also wanted to explore a totally new application area and so I began working with Wally Feuerzeig who, together with Seymour Papert of MIT, was researching applications in education. Wally and Seymore had designed the special computer language LOGO, which they felt would be easy and instructive for children to learn. I began looking for ways to design a machine that directly implemented that language.

I had also agreed to teach the computer design and programming course that Harvard had wanted, so it was arranged that I would be hired through BBN to teach a graduate seminar at Harvard. This would enable me to pull the material together for an undergraduate version of the course in ensuing years and would provide me with teaching assistants. Not surprisingly, since this was the first time I'd taught such a course, preparation took a good deal of my time and perhaps for that reason I failed to make progress with the LOGO project and eventually became discouraged about my ability to make a contribution to it. I also came to realize that I didn't share my colleagues' deep conviction that computers could revolutionize education. So my enthusiasm waned and I began to think that perhaps after all I really belonged in the other BBN division which was developing hardware, something I by now understood. So I moved over into Jerry Elkind's division and began to learn more about Time Sharing.

There was plenty to learn and plenty of bright new people to get to know. Danny Bobrow, like me, was teaching a course at Harvard and at that time I met Ray Tomlinson, who would later become famous as the originator of the @ sign for email addresses (totally swamping the far more significant fact that he had extended the intra-site message exchange system into an inter-site system, thereby creating the first primitive email system—not to mention his more fundamental contributions). Ed Fiala, who would turn up again later at Xerox PARC, was, as always, deeply buried in the most esoteric complexities of the system software. Perhaps the most memorable figure of all was Dan Murphy who, because of his last name, was blamed for virtually everything that went wrong. Our boss, Jerry Elkind, would later hire me to work in the Computer Science Lab at Xerox PARC in California.

The group was preparing to shift from an SDS-940 system to a DEC PDP-10. That was an upgrade to a more powerful machine and the BBN engineers had designed special hardware to make the PDP-10 more amenable to Time Sharing. The new system, consisting of hardware and software felicitously intertwined, was referred to as TENEX and would become a de facto standard at many ARPA-supported sites over ensuing years.[33] One of the most important features of TENEX was a virtual memory system that allowed programmers to believe they had more high-speed memory than actually existed (by surreptitiously swapping information between high-speed memory and disks). And of course the system allowed multiple programs to share the machine and run "at the same time." These features (virtual memory and multiprocessing) are a standard part of virtually all computers today. I received my first introduction to these concepts from Jerry Burchfiel, a senior member of the TENEX group.

A substantial number of research projects at BBN depended on this Time Sharing system and some of them involved real-time processing. As I've indicated, real-time processing doesn't mix well with Time Sharing since such processing demands the attention of the computer in reaction to events beyond the computer's control. The

[33] TENEX was eventually licensed to DEC as TOPS-20

computer must be ready to jump up and salute whenever it is needed. But under Time Sharing, the computer's attention is directed to the needs of the multiple users sitting at their terminals. There is an inherent conflict in trying to serve these two demands simultaneously. Nonetheless, plans were afoot to design a special hybrid processor that would be attached to TENEX, enabling it to provide some level of real-time processing.

I tried to put my attention on this undertaking but ultimately failed, because—as in Don O'Brien's earlier nightmare—I felt that indeed we were trying to attach wings to a steamroller. When I questioned a colleague about the wisdom of the whole approach, to my dismay he simply quoted "Ours not to question why; Ours but to do or die." This was a long way from the sort of response I was accustomed to, and I wondered what sort of outfit I'd fallen in with. But then a consultant, Chuck Seitz, appeared on the scene and we quickly found that we had a good deal of common ground and shared many of the same attitudes. Among other things, he was teaching a course at MIT similar to the one I was teaching at Harvard and we showed one another what we were up to.

A large disk system was to be added to the PDP-10. Today the capacities of disks are measured in megabytes, or, increasingly, in gigabytes. Back then, disks were measured in feet. We're talking about a set of disks roughly four feet in diameter. As the disks were being installed, they filled the halls near the machine room and then gradually disappeared, one by one into its maw. In order to be able to get at information quickly, they turned at alarming rates, and in doing so, they swept a film of air along on their surfaces. In order to allow the bits of information to be closely spaced so that lots of them could be crammed in, the heads that read and wrote the disks needed to be very close to the disk surface. As the heads moved in close, the film of air offered substantial resistance. (Recall that your automobile rides on the air in your tires and that metal and rocks can burn up upon reentering the atmosphere; air becomes a material to reckon with when sufficiently compressed.) The heads thus had to be forced toward the disks to overcome the air resistance.

Perhaps you're wondering, "But what happened when the disks slowed down and the resistance of the air film decreased?" Excellent question. The designers had considered this matter and in such a situation—when the power failed, for example—a safety feature instantly retracted the heads from the disks. The day arrived when the disks were working and everything had been tested—everything, that is, except this safety retraction mechanism. Alas when the power switch was shut off for testing, there ensued a frightful screeching sound as the heads failed to retract and instead ground their way into the surfaces of the disks. Shortly thereafter the halls were once again filled with disks as replacements began to arrive and the damaged ones were converted to coffee tables or discarded. The safety feature was redesigned and after a few weeks, things were once again ready for action.

The question then arose whether or not to test the redesigned mechanism. After considerable debate it was decided NOT to test it. Power failures were infrequent and should one occur, we would know soon enough if the mechanism worked. In such contests between man and the perversity of nature, man is invariably the loser. Within days a power failure occurred and once again the halls were filled with disks. But engineers are a persistent lot, and in the fullness of time the disks were tamed and became part of the working TENEX system.

One day, for a reason I can no longer recall, I needed to talk to John Barnaby, one of the TENEX programmers. When I entered his office, I couldn't help noticing a huge, ragged hole in the wall by his desk. I could see right through it into the neighboring office, but somehow I knew that it would be unwise to ask how it had come about. As we talked, that hole stared at us like a pink elephant whose presence everyone has tacitly agreed to ignore. Later I learned that, despite a mild manner, this was a fellow who could become quite distressed if he encountered trouble with his program or the system on which it was running. I can't say for sure which of these problems had arisen, but I did discover that on a night not long before my visit, something had infuriated him to the point that he (a largish fellow) had picked up his roughly fifty-pound model-33 Teletype (the terminal of choice at the time) and hurled it bodily through the wall. Programmers of his caliber were hard to come by, so this minor infraction was overlooked. The hole remained for some time and I thought of it as vivid testimony to the frustrations induced by over-stressed Time Sharing.[34]

Although people raised an eyebrow at such behavior, many understood it full well and secretly sympathized with

[34] Some years later my wife Laura became a victim of a Time-Sharing system known as TENEX, the computing environment with which she was then interacting via a model 33 teletype. She is a skilled knitter and in between responses from TENEX she had ample time to beaver away at an elaborate sweater containing thousands of very fine stitches. When it was finished she dubbed it her "TENEX Memorial Sweater" since most of it had been knitted while waiting between responses from TENEX.

and even admired such a forthright response. I've often thought that if more people responded as Barnaby did (or the guy who shot the 1401 full of holes), expressing their frustration with similarly overt action, perhaps the computer industry would be forced to shape up and give us less defective products. As it is, we may never know how many personal computers continue to be secretly abused by their frustrated users.

I realized I didn't belong in this Time Sharing division—I'd settled among disciples of the wrong religion. I was becoming quite discouraged by this point and felt that perhaps after all I'd made a serious mistake in turning down Ivan's offer at Harvard. I'd found no place at BBN to apply my skills constructively, but when I talked to some of my colleagues at Harvard their lives sounded differently bad as they described their struggles to avoid committee work and obtain funding. I was having a serious talk with Frank about the situation, when he pushed across the desk a document that had just arrived from Washington. He said it was a request for proposals (RFP in the lingo) for building some sort of network of computers and suggested that I take it home and look it over. With very little enthusiasm I put it in my briefcase to read that evening. I didn't understand all of the details, but I got the general drift and decided that what it described was a relatively straightforward, if not simple, engineering job. The next morning I went into Frank's office and, putting the document on his desk, told him I felt that we could certainly build it, but that I couldn't imagine why anyone would want such a thing. The network the document described was to become known as the ARPANET, forerunner of today's Internet.

There is no question that, in retrospect, my initial sentiment ("Who would want such a thing?") seems ludicrous. However, hindsight is far easier than foresight, and the request for proposal that came from ARPA made no mention of e-mail or the World Wide Web.[35] These things, which actually caused networking to "take off," were to come later, more or less as afterthoughts. Instead, at the outset, there was talk of eliminating duplication and fostering "Resource Sharing" —the sharing of programs, results, and access to computers among workers at the various sites, mostly universities, that ARPA was supporting. But these things seemed difficult to believe in, given the diversity of machines, interests, and capabilities at the various 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. A long road and many years of difficult, ground-breaking work lay between the interconnection of the first few ARPANET sites and the world we now inhabit in which a computer that can't access and utilize the Internet would be deemed virtually useless.

As originally conceived, the network was to consist of the large host computers at each site "talking" directly to one another. It was Wes Clark who, one day riding in a car with Larry Roberts (and others), urged the innovation

---

[35] Amazingly, today there exist almost no copies of the original RFP from IPTO. I take this as some indication that I was not alone in failing to grasp the historic significance of what was happening.

wherein the network traffic handling was off-loaded from the main ("host") computers at each site to a small auxiliary computer. This small computer was dubbed an Interface Message Processor or IMP. It is probably not overstating the case to say that this suggestion was critical to the success of the entire network project. In addition to relieving the host of the work of handling network traffic for other sites, it had the advantage that each host had only to deal with its IMP in a standard fashion, rather than having to interact with the different types of computers at all the sites to which it was connected. Eventually, of course, since the IMPs really only passed the bits along blindly, the host programs *did* have to deal with and understand one another. Standard host-to-host protocols, by which the hosts communicated with one another, would be defined and refined over the coming years. Think, for instance, of the simple protocol whereby we all say "Hello" and "Good-bye" when we use the telephone—a device which, like an IMP, simply passes the data (our voices) along without understanding anything we say. If someone violates these conventions, it causes at least momentary confusion.

Anyway, here at last was something I could get my teeth into—a challenging engineering problem. Although I didn't know where it would lead, it didn't seem like a bad idea, so back I came to Frank's division as we began to put together a proposal in answer to ARPA's request. Before we could write a sensible proposal, we felt we needed to understand in detail how what we were proposing would work, and that meant that we needed to do a pretty thorough system design. It soon became evident that some very large corporations and defense contractors were also interested in bidding on this job and we felt that tiny BBN would need to have a really superior proposal if it were to stand any chance against these giants. We had another problem as well. Several of us knew Larry Roberts from Lincoln days and we knew that he would be concerned about any appearance of favoritism and would therefore be cautious about giving the job to BBN. We therefore had to write a proposal that not only Larry but *anyone* could see was the best of the lot. Fearlessly we plunged in and set about not only figuring out how the system should work but actually proceeding to design it to a level of detail unusual for a proposal.

Our old colleague Will Crowther was still working at Lincoln, but we thought that for a job like this he might be induced to come to work with us. We discussed it with him and even before he had completed the formal transfer, he began acting as an informal consultant in the design. At that time Bob Kahn, who would later become head of the ARPA IPTO office, was working at BBN. He was interested in the potential of the network, had contributed to the error detection scheme contained in the proposal, and wanted to learn some fundamentals of hardware design. So I commenced working with him on the design of the interfaces and other special hardware that would be required on the IMPs. Teaching is an excellent vehicle for coming to grips with and understanding a problem, and we enjoyed working together as the interface designs quickly took shape.

The hardware design was relatively straightforward since it was obvious what was required and the choices were somewhat limited—it was just a matter of being careful and thorough. The more complex design decisions lay in the software, where most of the character and behavior of the IMP, and thus of the network, would be determined. The software team was made up of Dave

Walden, Will Crowther, and Bernie Cosell, with Frank also deeply involved and the rest of us making occasional suggestions and comments. It was just the right level of difficulty and we all enjoyed ourselves enormously, feeling that the design of systems such as this was ideal grist for our mill. We had high regard for one another's abilities and our mutual understanding was such that a great deal of abbreviated language was used in communicating with one another. I was getting to know Dave Walden for the first time and I remember being impressed that such a bright young guy had such good judgment as well. Bernie was already well known to me (and, in fact, to the entire BBN community) as a programming wizard. He was notorious for having inadvertently caused Bob Newman (the "N" of BBN) to imagine briefly that he was typing to a person (Danny Bobrow) at another terminal, when in fact he was actually typing into Bernie's cobbled-together version of Eliza, a simple simulated-psychoanalyst program originally concocted by Joe Weizenbaum at MIT.

As more and more time was spent in preparing the proposal, the "overhead" costs began to pile up. The company's management, feeling that the probability of BBN winning the contract was vanishingly small, was appalled at the amount of money that was being spent in preparing the bid—more money (I believe) than BBN had ever before spent on such a thing. But by the time we finished, we had great confidence in our design. As I recall it did not fully comply with ARPA's RFP; we felt we had found better ways to do a few things. We had already designed all of the special hardware and had actually written the time-critical inner loop of the program, as well as designing the rest of it in some detail.

We, and a small number of other bidders, were individually called to Washington for discussions with Larry. We defended our design decisions with great vigor and Larry quizzed us unmercifully about every detail. As the weeks passed and we remained in the running, our hopes began to rise. Then, to the astonishment of many (probably especially to some of the larger bidders, not to mention the BBN management), we were finally awarded the contract. That evoked a hilarious telegram from Ted Kennedy, congratulating us for winning the contract to build the "Interfaith Message Processor." But aside from the hilarity, we were elated as well as appalled at what lay before us.

Recognizing that we would need to beef up our forces, we shortly hired Will Crowther and began hiring the best students from the course I'd been teaching at Harvard. The first of these was a very bright young fellow by the name of Ben Barker. Shortly after he came on board, Ben joined us for a meeting with AT&T people at their New York headquarters in order to discuss details of their lines and the 50 kilobit modems to which we would be connecting. AT&T was somewhat reluctant about the whole endeavor; they didn't really want novices like us mucking about with their terminal equipment, and moreover felt that the entire enterprise was somewhat silly.[36] But Larry held a heavy governmental sword over their heads so they laid on a rather formal meeting, replete, it turned out, with cigars, candy, and nameplates at each seat. As we entered the

[36] Later, when we complained about interruptions in the lines on the order of a few hundredths of a second, they simply could not comprehend why anyone would possibly care about such a tiny matter.

room, I thought to myself that this must be rather heady stuff for a young student such as Ben, but later, when our plane home was canceled by a snowstorm and we were reduced to riding the train back to Boston, suddenly Ben snapped his fingers and said "I should have thought of it earlier; we could have taken Dad's helicopter." Clearly Ben could take AT&T and their nameplates in stride.

We had selected the Honeywell 516 computer as the basis for the IMP[37] and had presented extensive justification for the choice in our proposal—including the fact that we planned to use a hardened, military version built like a tank. Now it was time to convey to Honeywell all of the additions and modifications to the basic machine that would be necessary to turn it into an IMP. At that point we were not familiar with the particular logic packages that Honeywell used and so the designs that we gave them, though detailed, were in terms of general logic rather than explicit packages. It seemed to us a straightforward task to render the design into Honeywell's own logic modules, but to be sure that there were no mistakes, we spent a number of sessions explaining everything carefully to the designer they had assigned to the project. This was to turn out to be an extremely painful part of the project. The software was under our control; we were doing it ourselves. But for the hardware we were dependent on Honeywell's special

---

[37] This was based in part on work that Dave Walden and Alex MacKenzie had done earlier evaluating the Honeywell 516 for another project. Honeywell, furthermore, was willing to help BBN bid on the proposal (they were hard-pressed to respond quickly to the several bidders that wanted to use the 516), and quickly agreed to build the special hardware for us. Many computer companies hate doing special hardware, wishing to reserve any available talent for new machines that can be sold in quantity.

systems division and that turned out to be a hotbed of incompetence.

Week after week we struggled to get the Honeywell engineers to understand what was wanted. In the process we became familiar with their logic elements and were thus able to see that one mistake after another was being made. As time passed it became evident that the design was converging too slowly for the schedule and ultimately we were forced to accept a machine that we knew we would have to rework substantially. Ben was a great help in all of this and together we finally managed to get it going. We then gave a set of revisions to Honeywell so that ensuing copies would, we naively presumed, be correct. Alas, we were to learn, it would take many months before all the corrections were finally incorporated. In the meantime we had to make repairs to every machine that arrived. We instituted a plan under which we gave Honeywell test programs and insisted that they run them in their plant in our presence so that we could certify each machine before it was shipped. One day, despite our previous day's refusal to accept a machine, it showed up on a truck at the BBN loading dock. Frank, watching the proceedings through a window, registered disbelief as he saw me refuse delivery, turn the truck around, and send it back. This caused some chaos at the Honeywell plant and, along with some of my more forceful language, apparently garnered sufficient attention that things then began to change.

In the meantime the software crew had been busy and a test version of the program began to operate in the first fully-functioning machine. Since we had only that one machine to work with, for testing purposes the communication interfaces (both to the host and to the modem lines) were shunted so that outgoing channels were connected directly to incoming channels. We were elated that, despite the troubles with Honeywell, things were going well, but then a strange thing began to happen After many hours or days of continuously successful operation, the machine would suddenly stop in a strange state, never the same way twice. We had what amounts to a system designer's worst nightmare—an extraordinarily rare, seemingly random, intermittent failure. Despite numerous attempts, we were unable to catch it in the act in order to capture some symptoms. Then suddenly Ben or I (or perhaps the two of us together[38]) remembered the synchronizer glitch problem that I'd encountered several years before in a totally different environment. Could this be a manifestation of the glitch problem, and what could we do to verify this suggestion?

There were lots of 516s out in the world serving faithfully in many other settings—that was one of the reasons we had selected it—and no such problems had been reported before. But we also knew that our several high-speed interfaces were driving the machine's input-output section much harder than most other applications. And most other applications didn't require the machine to operate continuously, 24 hours a day, for months on end without a single hiccough. Ben pored over the logic diagrams of the machine and finally, in the part where requests for service by external devices are handled, he found a possible culprit. He designed a clever piece of hardware that aggravated this part of the machine even more intensively than our interfaces did, and lo, it stopped

[38] I had described the problem to my Harvard class as part of the introduction to synchronization.

almost at once. We then excitedly hooked up an oscilloscope so that we could observe what was happening and, dimming the room lights, so that we could see a very faint occasional failure trace amidst all the bright correct ones, we peered at the scope's face. And sure enough, there it was—an occasional failure that Ben's device had made just frequent enough to become visible.

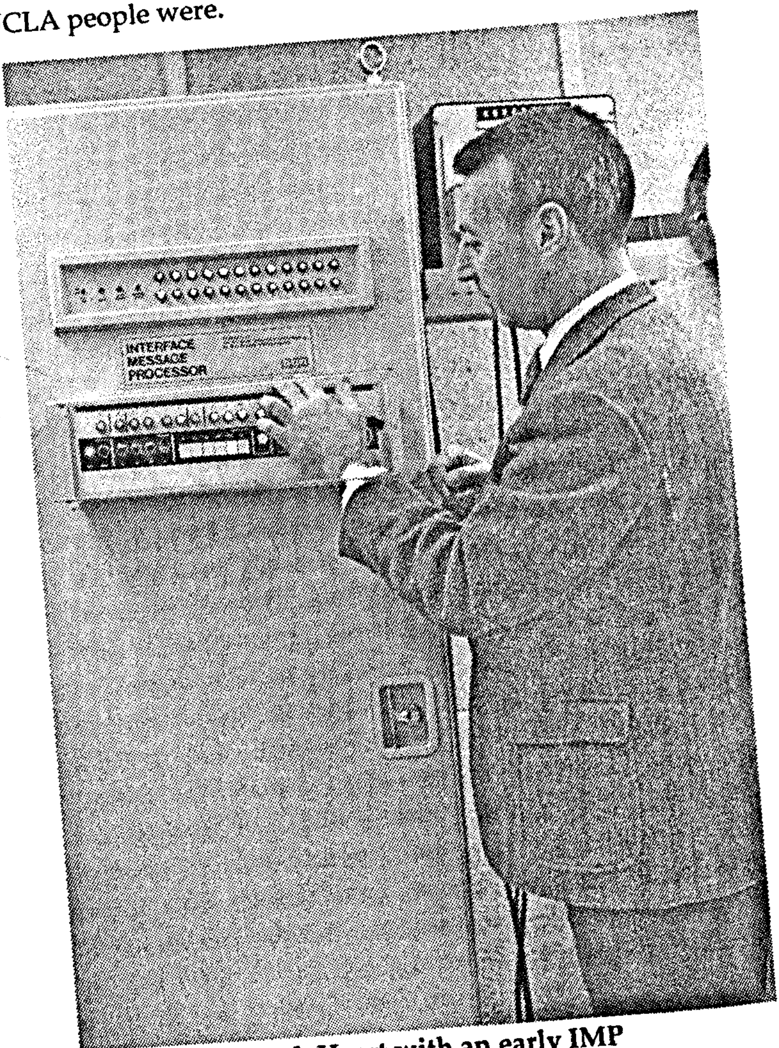
This was one of the best pieces of hardware detective work any of us had ever experienced. We were thrilled and, now having hard evidence in hand, we immediately called Honeywell. Their preliminary reaction was defensive: the troublemakers at BBN were not only complaining about their interfaces and the schedule, but were now questioning the very design of their basic 516! After considerable persuasion we got them to produce the machine's original designer from the back room, the first really competent Honeywell engineer we'd met. He expressed considerable skepticism at first but agreed to come have a look, and after listening carefully to our explanation and peering with us at the evidence on the scope face, he was finally forced to concur. In the meantime, Ben had devised a simple fix to the machine that would cure the problem.[39] We instituted the change and the failure never occurred again. We recommended that Honeywell adopt the change, install it in all future machines, and retrofit existing machines. I believe this happened; we certainly checked all future machines that came to BBN to be sure the fix was installed.

---

[39] The problem can't be truly *eliminated,* but Ben's fix reduced its probability to the point where its occurrence was measured in hundreds of years rather than hours.

The day approached when we were to deliver the first IMP to UCLA and in short order packers arrived to encase our precious cargo in a sturdy wooden box. Given that the version of the machine Frank had chosen was "hardened" —it had a rugged steel case, and appeared capable of withstanding a direct nuclear hit—this extra packaging seemed a bit redundant, but it provided a place to affix a destination address label. Finally I added numerous arrows saying "This Way Up," and a note that said simply "Do It To It Truett." Frank, in an excess of compulsiveness, had decided that someone from the team should accompany the machine on its journey—I mean *ride with it in the cargo plane!* This proved impossible—the air transport company simply refused—but it was watched carefully onto and off the plane. Truett Thach, a member of the BBN team, was waiting in Los Angeles and shortly more of us joined the festivities when it finally arrived at UCLA.

Given the careful testing the machine had undergone at BBN, including not only the self-test, but also testing with another IMP in the same room through a pair of modems, we were not surprised that the machine worked as soon as it was plugged in and turned on. In fact, since it was shipped with the program installed, I believe it came alive of its own accord when it was plugged in, thanks to a "watchdog timer" we'd provided to reset and restart it automatically should it ever falter. We had met the schedule, something quite unprecedented in the computer field. The UCLA crew had expected that we would spend several days getting the machine working, and in fact had relied on this to allow them time to finish their own preparations of the program and interface for their host machine. Thus they had to scurry some, but shortly the connection was made and the much-touted initial message was sent through the IMP from one part of the Host to another. Having done a great deal of such testing on our home turf, we were not as impressed with the event as the UCLA people were.



Frank Heart with an early IMP

The initial contract had called for four IMPs, and right on schedule the other IMPs were delivered to their prescribed sites. There was a good deal of reluctance by some of these initial site managers because they viewed the network as an intrusive nuisance that interfered with their routine operations. But once again Larry held a trump card (funding) and they were forced to knuckle under and provide suitable hardware and software interfaces. It would be several years, however, before the fruits of this trouble were to begin paying significant dividends. During those years, the software communities at the ARPA sites would gather their forces and begin to develop standards by which diverse computers could communicate with one another.[40] Given the diversity of the hardware and software systems that would be joined into a single network, this was a complex and daunting undertaking. Steve Crocker, then a graduate-student at UCLA, would become visible on my radar screen as one of the principals in this enterprise and it was some years later that Bob Kahn and Vint Cerf, together with others who have received considerably less credit, would define the discipline that would allow networks to be joined together to form the Internet as we know it today.

By this time my second marriage was beginning to falter and would soon break-up. Throughout this difficult period I continued teaching at Harvard and in the second year we turned my course into an undergraduate one. On opening day nearly 100 students showed up and we were forced to turn quite a few away as we had insufficient lab space and equipment. Fortunately by then I had lined up

[40] The fact that TENEX and the IMPs all came from one place (BBN) facilitated much of the early ARPANET protocols and application work.

some superb teaching fellows (including Ben Barker) and together we went to work. I had a good set of lecture notes from the preceding year and felt much more secure about what I was doing. I began to relax and enjoy the teaching. It was exciting to see understanding dawn and my informal manner shortly led to friendly badinage with the class. We were working together, bringing them on board for an exciting journey. I asked Dave Walden to give a lecture on Time Sharing systems and he did a superb job. At one point he lost the thread of where he was going and simply turned to the class and said, "Come on guys—help me out here." They were a bright bunch and quickly put him back on track. When the time came for the final exam we put together some really intriguing problems and after it was over, as one student was leaving, she turned to me and said "That's the best final exam I ever flunked!"

Other things bound us together as well. It was a time of anti-war protests, and the students had clearly noted the resist button on my jacket and knew that I was on "their side." When I came across some of them gathered in a crowd outside a meeting in which the faculty was considering taking a stand against the war, they hurried me inside. At BBN I felt considerable ambivalence about my work. On the one hand I was strongly against the war but on the other, here I was working hand in glove with a branch of the Department of Defense. I was able to see that the military connection of our work was pretty tenuous, that the implications of what we were doing were extremely general, but nonetheless, under the circumstances I had some level of discomfort drawing my pay indirectly from the Department of Defense. Frank knew this and one day as we departed for a meeting at the Pentagon, I mentioned that I was considering moving my resist pin onto "the general's" jacket when we all hung up our coats at the commencement of the meeting. Frank wasn't sure whether or not I might actually do such a thing. Neither was I.

Meantime, Ray Tomlinson, a bright colleague from my brief TENEX excursion, noticed that it would be relatively simple to extend the already-extant intra-TENEX inter-person communication feature to permit communication between individuals at TENEX systems at remote sites. Ray inadvertently achieved immortality through his adoption of the "@" sign to indicate a remote network address. More importantly, his "hack" marked the beginning of inter-site email which promptly blossomed into the most widespread use of the network. As email started to become ubiquitous, other mail programs began to show up with improved user interfaces. However, truly "user-friendly" systems would appear only after the switch from teletype-style, terminal-based systems to the sorts of display screens virtually everyone uses today.

As the ARPANET grew and prospered, it shifted imperceptibly from an experimental vehicle into a utility as people began to depend on it more and more. We had, of course, realized from the outset that reliability would be critical and had taken numerous steps to assure that when an individual IMP died, the network as a whole would continue to function as traffic was rerouted around the sick IMP. Right from the outset, features were designed into the IMPS that allowed them to be monitored, reloaded, etc., from what soon became a Network Control Center (NCC) at BBN. Alex McKenzie, a member of the team, was an early and ardent advocate for viewing the network as a utility. Alex, a meticulous fellow with a booming voice as forceful as Frank's but in a lower register, took over management of the Network Control Center and was to find himself increasingly acting as a buffer between the users of the network and the providers at BBN.

Alex showing the NCC to some Chinese visitors

Because of its special rôle in overseeing the network, there were particular sensitivities at the BBN node. One day it became necessary to install a jumper—a single short piece of wire—between two points on the back-panel of BBN's own IMP. The IMP was busy running in the network at the time and, because of its critical role at the network control center, we decided to ignore the normal discipline, whereby one would turn off a machine before touching the wiring. The place where the jumper needed to be installed was awkwardly close to the base of the machine and as I got down on my hands and knees, with the crew surrounding me shouting suggestions, I peered at the pins to which the jumper needed to be attached. Finally I reached forward gingerly—and almost immediately touched something I shouldn't have, abruptly shutting off the machine. Chaos ensued as the control center crew erupted into action, kicked us out of the way, and quickly restarted the machine—so quickly, in fact, that it was up and running before we could take advantage of the time to install the jumper while the power was off!

So there we were again, right back where we'd started. One thing was clear—I'd had my chance, and it was now up to someone else. I couldn't believe it when Ben Barker stepped forward and said, "Here, let me." Ben was a very tense fellow and I'd noticed that intense concentration on his part was often accompanied by a shaking hand. Here, I thought, was a catastrophe not only waiting but impatient to happen. Ben crouched down and as I knew it would, his hand commenced shaking. Then, as I watched in disbelief, his hand suddenly stopped shaking as it shot out, shoved the jumper into place, withdrew, and immediately resumed its shaking. To this day I don't know how he managed it.

All of the early sites were Time Sharing systems and users of those systems accessed remote sites on the network through terminals attached to their local "Host" system. The IMPs acted somewhat like the phone company, merely passing information along. The Host computers dealt with the terminals and connected them logically with the desired remote Host. But there were users, not associated with any nearby Host, who wished to access remote sites by connecting directly from their terminals to an IMP and thus to the network. This unanticipated development required an upgrade to the IMP that added a sort of mini-host inside of the IMP through which terminals could speak with remote Hosts. In addition to the mini-Host software, we needed hardware through which the terminals could be connected to the machine—these terminals were of many different types and speeds. Ben came up with a clever idea for this hardware which accommodated a wide variety of different terminals. While the software team made the necessary additions to the program, we set about designing and building the hardware.

This was an era in which a technique known as "wire-wrap" was the method of choice for building prototype devices. By this time we were using integrated circuits (ICs) of the sort that would now be described as small scale integration. These little bug-like gadgets plugged into sockets on one side of a board. On the other side, the prongs of the sockets extended about half an inch, forming something that looked like a miniature, dense bed of nails. The wires that provided the required interconnections between the ICs were wrapped around these prongs. Special machines (owned and operated by sub-contractors) did the wire-wrapping, but the mechanisms for providing the machines with the information from our drawings regarding precisely *which* sets of points to connect together, was, to say the least, primitive. Today such mechanisms are fully automated, but in those days most wiring lists were made up by hand[41]. Thus one had to contend not only with design errors, but also with clerical errors introduced during the transcription process. This made debugging something of a nightmare and caused harsh thoughts, and sometimes even words, to be exchanged between us and the company that did the wire-wrapping for us.

---

[41] Some research institutions were then busy developing automated systems but they hadn't yet percolated to places such as BBN.

After the hardware was built, Tony Michel, another member of the growing IMP team, and I worked together in debugging it. As always, there was considerable time pressure and as we worked late night after night, Tony started to fall asleep even as he was running oscilloscopes and moving probes around. He later confessed that I'd pushed him so hard during that period that he had seriously considered quitting. But he didn't, and shortly we had working hardware. The software required major upgrading and the whole thing took a number of months to complete, but eventually we got it all working.

As people began using the Terminal IMPs (called "TIPs"), they were plagued by problems with the telephone connections between their terminals and the TIP. Strictly speaking, these weren't BBN's problems (just as the connection from your computer to your Internet Service Provider isn't part of the Internet), but the users didn't care about that; they viewed their terminals as connected to the remote site and weren't interested in the distinction between the individual line connecting them to the TIP and the further network connections to the remote sites. If any part of it failed to work, BBN got the blame. So we developed a system wherein periodically a computer at BBN dialed each of the terminal ports on every TIP in turn, connecting to and testing each one to be sure that the terminal lines were working properly. These, of course, were just plain old regular phone lines, and sure enough one day one of them seemed to be failing. Unable to figure out what was wrong, the technicians decided to listen to the line as it was dialed and tested. They heard it ring and try to whistle at the TIP (think FAX), but instead of the expected answering signal, to their astonishment, they heard an angry voice shouting "Oh, it's YOU again is it!"

and the receiver was slammed down. Someone had entered a wrong number in a table of TIP phone numbers, and an innocent victim had been receiving the repeated test calls. At that time faxes hadn't yet come into widespread use and the recipient of the calls, never having heard such a thing, assumed it was some prankster repeatedly calling and then whistling into the receiver.

There was considerable interest abroad in networking, particularly in England where much parallel thinking about so-called "packet-switched" networks had taken place. Shortly Donald Davies, who had been a pioneer in thinking about such networks, came by for a visit. Davies might easily have built a network in England before the ARPANET, had he had access to the kind of funding ARPA was able to provide. Interactions with our group began to develop in other parts of Europe as well, and within a year or so we were making trips both to report on our work in various meetings and also to explore with potential customers their need for computer networks. Once again Frank's instinct for seeking and finding new applications was an enormous asset.

I myself think the verdict is by no means in regarding the ultimate benefits and costs to society of computer networking. Obviously it's already had an enormous impact, but things are moving so fast, and the air is presently so full of hype, that it's impossible to say how it will all ultimately settle out. Like so much technology, the final effect will depend on the way society chooses to utilize it. Today the glowing predictions are largely based on optimistic assumptions about how this will work itself out and about potential economic benefits. But there are other impacts as well, and I can envision possibilities that are far more ominous than those presently being touted. I suspect it would be wise to reserve judgment until the present wave of euphoria settles down and matters clarify themselves as developments unfold over coming years.

Most of us have forgotten the sorts of hype that accompanied the advent of television in its early years. It was hailed then as a great educational vehicle. But against the limited educational programming that actually exists today, one must balance the far more prevalent entertainment programs, which, by and large, have had a dumbing-down effect on society—not to mention the enormous perversion of the political process that has resulted. Perhaps computer networking, because it permits multi-way, rather than just one-to-many communication, will provide a countervailing force, as some are hoping and predicting, but much remains to be seen.

Before I leave the ARPANET (and its offspring, the Internet), I feel obliged to comment on what has happened, in recent years, to its paternity. As its importance has become obvious to everyone (even to me), so-called "fathers" have been cropping up all over the place. It's the same old story of the press identifying and celebrating certain individuals as "the Father of the Internet," whereas in truth the thing came about as a result of the convergence of numerous technical developments and the ideas and energies of a large number of individuals. Although some people were obviously more central and influential than others, trying to point to any one person or a few individuals as responsible for either the end result or the vision is absurd. Nonetheless, regrettably, a number of former colleagues have allowed themselves to be singled out and celebrated as particularly important figures, whereas others, probably even more central, who are by nature more reticent, have received far less attention.

I have come to believe that it was J.C.R. Licklider and his disciple Bob Taylor, who provided the principal impetus for networking in the U.S. Not that they themselves "invented" any part of it. But they foresaw and believed deeply in the tremendously cohesive impact it would have and used their influence to push research and development in the direction of networking. The depth of their conviction and Taylor's extraordinary persistence in pursuing their vision of the computer as primarily a *communication* device, are, to me, absolutely stunning—especially given that neither one was anything like a technical whiz. It may be precisely because they themselves were not directly in the trenches, where the many formidable obstacles were often all that one could see, that they were able to hang on to their vision. But hold to it they did, and my hat is off to them. It's also off to Larry Roberts, the person who not only ran the ARPA office during that period but also had ample technical ability to climb repeatedly into and out of the design trenches during the crucial early years of the network. And likewise to Frank Heart, whose firm hand on the tiller and general paranoia, produced a design that set the kind of high standards for later developments in networking that would enable it to become a new kind of utility for the entire world.

A number of years ago, BBN held a self-congratulatory party to commemorate the twenty-fifth anniversary of the beginnings of the Internet. Forgotten was the horror with which, at the time, the earlier management had viewed the cost of writing the proposal. Instead, now there was much raucous breast-beating. Frank, however, who had spearheaded BBN's initial involvement in networking and had overseen its blossoming, was a model of reserve, saying only, with considerable eloquence, that it was a rare privilege to have had the opportunity to ride the crest of such an important technological wave. Would that all who have been involved could emulate the reserve such a statement reflects.