CASTING THE NET

The or igins of the Internet, like the messages it carries, are diffuse and fragmentary. But one fall day in 1969 all the pieces came together.

BY KATIE HAFNER AND MATTHEW LYON

WHEN ROBERT TAYLOR JOINED THE federal government's Advanced Research Projects Agency (ARPA) in 1965, his division's suite in the Pentagon included a small terminal room. There, side by side, sat three computer terminals, each of a different make, each connected to its own mainframe computer. Each mainframe, in turn, had its own operating system and programming language. ARPA was at the leading edge of computer research, but the terminals were irksome to use. Having three computers was like using three television sets, each dedicated to a different channel. "It became obvious," Taylor said years later, "that we ought to find a way to connect all these different machines."

Established in 1958 as a response to the technological challenges posed by Sputnik, ARPA was generous with its money during Taylor's time. Come up with a good idea for a research program, some program directors joked, and you'll get funding for it in about thirty minutes. In 1966, when Taylor was promoted to head of his division, formally known as the Information Processing Techniques Office, he decided to take the joke at face value. He headed straight to the office of ARPA director Charles M. Herzfeld. No memos. No scheduled meeting.

Each new ARPA investigator seemed to want his own computer, Taylor told Herzfeld. A great deal of work was being duplicated, and it was getting damned expensive. Computers weren't small and they weren't cheap. Why not try tying them all together? If machines were linked electronically, investigators doing similar work in different parts of the country could share resources and results more easily. (It was an idea inspired by J. C. R. Licklider, the renowned psychoacoustician turned computer scientist at the Massachusetts Institute of Technology, whose grand vision of an "intergalactic network" had sparked a revolution in computer engineering that was carried out by Taylor's generation.) ARPA, Taylor suggested, could fund a small test network, starting with, say, four nodes.

"Is it going to be hard to do?" Herzfeld asked.

"Oh no. We already know how to do it," Taylor responded with characteristic boldness.

"Great idea," Herzfeld said. "Get it going. You've got a million dollars more in your budget right now. Go."

When Taylor left Herzfeld's office, he glanced at his watch. "Jesus Christ," he said to himself softly. "That only took twenty minutes."

Every culture has its creation myths, and cyberculture clings dearly to its own. Taylor's request truly launched the experiment that spawned the Internet, yet few people know of it. Instead, a very different story has been passed from hacker to hacker as the Internet has sprawled across popular culture: in the beginning the Internet was a military invention; its original channels were built to keep critical information flowing in the event of a nuclear attack.

Like most myths, that one is a fiction rooted in fact. In the early 1960s, before Taylor made his pitch to Herzfeld, an engineer named Paul Baran wrote a series of papers that brilliantly foretold the structure of the Internet. A short while later, a physicist named Donald Watts Davies at the British National Physical Laboratory independently came up with many of the same ideas. All told, dozens of people helped invent the Internet, improving on the central concept, now known as packet switching. And for Baran, at least, that concept was indeed born of cold war fears.

IN 1959, WHEN BARAN WAS HIRED BY THE RAND Corporation in Santa Monica, California, the Americans and the Soviets were building arsenals of nuclear missiles set on a hair trigger. Baran knew that the nation's long-distance communications network could not withstand a nuclear attack. Yet for the president to order a nuclear strike—or to call one off—he would need to use at least some of that network. Designing a robust system was not simply an intellectual challenge; it was a necessary response, as Baran put it, to "the most dangerous situation that ever existed."

RAND was one of the leading think tanks for military analysis during the cold war. But few people there knew much about digital-computer technology, and fewer still were interested in it. "Many of the things I thought possible would tend to sound like utter nonsense, or impractical" to his RAND colleagues, Baran recalled.

Baran just dived deeper into his work. The key to more-robust networks, he believed, lay in redundancy. Communications structures had to function as cohesive entities even after many of their components were destroyed. Looking well beyond mainstream computing, to the future of digital technologies and the symbiosis between people and machines, Baran chose the human brain as his model. When brain cells are damaged, he realized, neural networks sometimes simply bypass them, taking new pathways through the brain. Theoretically it was possible to set up a network with numerous redundant connections. But there was one problem. Analogue signals deteriorate each time they are sent across more than one link—like video recordings copied across several generations. For that reason, it was pointless to connect any two points in the telephone system via more than five intermediate connection points.

The solution, Baran thought, lay in computers. Digital signals could be stored efficiently and replicated an unlimited number of times with almost perfect accuracy. If computers could be taught to speak to one another, a redundant network might be created—one that, in a modest way, resembled the astonishingly complicated linkages among neurons in the brain. Best of all, computers offered speed. Almost any digital-switching technology, it was thought, could beat the twenty or thirty seconds mechanical telephone switches needed in order to establish a long-distance connection.

THE LAYOUT FOR BARAN'S THEORETICAL NETwork was as simple as it was dramatically new. Telephone networks have always had central switching points. In the most vulnerable networks, all paths lead to a single nerve center. In decentralized networks such as the ones in use today with long-distance telephone systems, links are clustered around several nerve centers interconnected by a few long lines. Baran came up with a third kind of design, which he called a distributed network. He imagined a network of many nodes, each redundantly connected to its neighbor, in a lattice reminiscent of a fish net.

Baran's second big idea was still more revolutionary: Fracture the message too. By dividing each message into parts, one could flood the network with what he called message blocks (or "packets," as Donald Davies later called them), all racing over different paths to their destinations. Upon their arrival, a receiving computer would reassemble the packets into readable form.

Conceptually, Baran's and Davies's approach seemed to borrow more from freight movers than from communications experts. Imagine that each message is a large house. How best to move that house from, say, Boston to Los Angeles? Theoretically, one could move the whole structure in a single piece. House movers do that over shorter distances all the time—slowly and carefully. It is more efficient, however, to disassemble the house if possible, load the pieces onto trucks and drive them over the nation's interstate highway system—another kind of distributed network. Not every truck will take the same route; some drivers might go through Chicago and some through Nashville. If the driver coming out of Nashville learns that the road is bad around Oklahoma City, he may go through Kansas City instead. But as long as each driver knows where to deliver his load, all the pieces should quickly arrive at the destination. Once there, they can be reassembled in their original order.

That innovation, in a communications network, helped solve a number of problems at once. At the time, all such networks were circuit switched, which meant that a line was reserved for one call at a time. A call between two teenage girls, for instance, would tie up a telephone line for as long as they commiserated over boyfriends—even during pauses in the conversation. That made a lot of sense, given that most people keep up a fairly steady flow of talk during a phone call. But data are different from conversation. They usually pour out in short bursts followed by empty pauses that leave the channel idle much of the time. It would cut costs dramatically if packets from different messages could share a line.

Baran envisaged a network of unmanned switches or nodes, each incorporating a routing table. That table would indicate the best routes for packets to take, constantly updating them on traffic and mechanical conditions around neighboring nodes—much as human dispatchers warn truck drivers over CB radios about obstacles on the roads. If the best path were busy—or blown to bits—the message packet would automatically take the next-best path.

FOR FIVE YEARS BARAN WROTE OUT HIS IDEAS and lobbied AT&T officials to accept his logic. Condescending and smug, AT&T dismissed him. But a few technologists at Bell Laboratories listened; RAND came around; and soon the air force was on board. In 1965 the Pentagon agreed to fund Baran's network but wanted to place the newly formed Defense Communications Agency (DCA) in charge of it. Baran pictured nothing but trouble. The agency was run by a group of old-fashioned communications officers with no experience in digital technology. "It would have been a damn waste of government money," he recalled. "The DCA would screw it up and then no one else would be allowed to try, given the failed attempt on the books." It would be better, Baran thought, to wait until a competent organization came along.

He did not have long to wait. In 1967, two years after he had moved on to other projects at RAND, Baran was drawn into a circle of experts gathered by a deep-thinking young engineer named Lawrence G. Roberts. Although Roberts had no real interest in nuclear-war scenarios, he was intrigued with Baran's insights. He was convinced that everything worth doing *inside* a computer had already been done; the future lay in communications *between* computers. Now Roberts, the man Taylor had recruited to create the first network for ARPA, had a chance to test his conviction.

At twenty-nine, Roberts had done more in the field of computing than others achieve in a lifetime. A year earlier, for instance, he had completed a groundbreaking proof-of-principle experiment linking two computers across the country. Yet Taylor had chosen him not only for his scientific background. Roberts also had a knack for management, and he was a quick study. Even before his first day at ARPA, he had an outline of the network figured out. Then, and for years afterward, he drew meticulous network diagrams, sketching out the data lines and the number of hops between nodes. Interconnecting a matrix of incompatible machines, Roberts realized, would require calling on every expert he knew in every area of computing and communications.

EARLY IN 1967 ROBERTS LAID OUT HIS INITIAL plan at a meeting for ARPA's principal investigators in Ann Arbor, Michigan. The idea, he told them, was to interconnect all the big nodes directly, over dial-up telephone lines. The networking functions would be handled by host computers at each site. The hosts, in other words, would do double duty, as research computers and as communications routers.

The idea was greeted with little enthusiasm. No one wanted to relinquish valuable computing cycles to a network of dubious value. Moreover, workers from the East Coast universities saw no reason to link up with campuses in the West. They were like the Boston woman on Beacon Hill who, when told that long-distance telephone service was available to Texas, echoed Thoreau's famous line: "But what would I possibly have to say to anyone in Texas?" In any case, they wondered, how could a computer such as the TX-2 at the MIT Lincoln Laboratory in Lexington, for instance, talk to the Sigma-7 at the University of California, Los Angeles?

Just before the meeting ended, Wesley Clark, a computer scientist at Washington University in Saint Louis, passed a note up to Roberts. "You've got the network inside out," it said. On the way back to the airport, sharing a car ride with Roberts and others, Clark sketched out his idea: Leave the host computers out of it as much as possible. Instead, insert a small computer between each host computer and the network of transmission lines to handle the message routing. (Donald Davies had independently come up with much the same solution and was already fleshing out the functions of an interface computer in England.) Those helpful little computers would come to be known as IMPs, short for interface message processors.

Clark's suggestion solved several problems: It placed far fewer demands on the host computers and on the people in charge of them. The smaller routing computers that made up the inner network all could speak the same language. And that language would serve as a lingua franca for what would otherwise become a Babel of computer languages and operating systems. Each host computer would have to learn only the language of the routing subnetwork.

With Clark's suggestion, the rest of Roberts's design quickly fell into place. It would be engineered according to a few basic principles and specifications. First, the subnet's essential task would be to transfer bits reliably from one location to another. Next, the average transit time through the subnet was to be less than half a second. Third, the subnet would have to be able to operate autonomously. Computers of that era required several hours of maintenance a week. But the IMPs had to be able to continue operating whether or not the host or other individual IMPs were running.

In Cambridge, Massachusetts, the firm of Bolt Beranek and Newman spent most of 1969 designing and building the first IMPs. The IMP Guys, as the firm's hardware designers and programmers called themselves, took on a panoply of crucial problems, with amazing results. They invented the algorithms that would pull packets into one IMP, figure out the best place to send them and push them out to the next IMP down the line. They discovered ways of processing packets ten times as fast as Roberts had required, and they wrote computer code so concise and elegant it was a kind of poetry.

The one responsibility the IMP Guys did not have was figuring out how to get the host computers to understand one another. The IMPs, being go-betweens, had been designed to read only the first thirty-two bits of each packet—the part specifying its source, its destination and its location in the file into which it would eventually be reassembled. The contents of the packet were left for the host computers to translate. To enable translation, research teams at the host sites had to design communication protocols in advance. It was among the hardest jobs in creating the network, and the teams had little more than a year in which to do it.

Roberts had chosen four sites to start the ARPA network: the University of California, Los Angeles; the Stanford Research Institute (SRI) in Menlo Park, California; the University of Utah in Salt Lake City; and the University of California, Santa Barbara. In the summer of 1968, more than a year before the first IMP was scheduled for installation at UCLA, a few graduate students from each site met in Santa Barbara to talk over their task. "We found ourselves imagining all kinds of possibilities—interactive graphics, cooperating processes, automatic data base query, electronic mail," one attendee says, "but no one knew where to begin." To speed up the process, the group decided to meet regularly, eventually calling itself the Network Working Group, or NWG.

The NWG was an ad hocracy of intensely creative, sleep-deprived, idiosyncratic but well-meaning young computer geniuses. And they always half-expected, any day, to be thanked for their work and promptly replaced by the field's true professionals. There was no one to tell them that they were as professional, and as official, as it got.

A computer, circa 1968, was an extremely egocentric device. Like a monarch surrounded by its subjects, it spent its time telling other devices what to do. (In computer parlance the process is known, aptly, as master–slave communication.) If another device tapped the computer on the shoulder with a signal and said, "Hi, I'm a computer, too," the receiving machine would be stumped. The NWG's goal was to get the mainframes to talk as peers, or at least to acknowledge one another's existence.

First the programmers had to ask themselves a few basic questions. What form should the protocol take? Should there be a single, foundational protocol on which to build all application protocols? Or should it be complex, subdivided, layered, branched? Because any protocol was a potential building block, they thought, it was best to define simple protocols, each limited in scope, with the expectation that they might be joined or modified in unanticipated ways. In some sense, the protocols would be to network builders what two-by-fours are to framing carpenters. That design philosophy broke ground for what came to be widely accepted as the "layered" approach to the protocols. As the talks grew more focused, the graduate students agreed that their first two projects would be to write protocols to enable users to perform remote log-ins and file transfers between host computers.

As exciting as it was, the NWG's work was also terribly complicated. By the spring of 1969 the first protocols were far from done, and the IMPs were slated to arrive in the fall. Rather than try to rush something out in time, the NWG decided to tell every site to patch together its own makeshift protocols. At SRI, one worker wrote a clever program that fooled his computer into thinking it was communicating with a "dumb" terminal rather than with another computer. It was a stopgap solution (the two computers would hardly be communicating as equals), but it would do.

On October 1, 1969, a month after the first IMP was installed at UCLA, the second one arrived at SRI. A few days later a lucky student named Charley Kline, then an undergraduate at UCLA, picked up a telephone headset in Los Angeles and pressed a button that rang a bell on the IMP in Menlo Park. One of the group members at SRI answered it, and the two began the connection. Nearly ten years after computer networking had first been envisaged, two computers were at last on the verge of talking to each other.

Unlike most systems today, which prompt the user for a log-in name and password, the SRI system waited for a command before acknowledging a connection. "L-O-G-I-N" was one of those commands. The quality of the connection was not very good, and both men were sitting in noisy computer rooms, which did not help. So Kline fairly yelled into the mouthpiece: "I'm going to type an *L*." Kline typed an *L*.

"Did you get the *L*?" he asked. "I got one-one-four," the SRI worker replied; he was reading off the encoded information in octal, a code using numbers expressed in base 8. When Kline did the conversion, it was indeed an *L* that had been transmitted. He typed an *O*.

"Did you get the *O*?"

"I got one-one-seven." It was an *O*.

Kline typed a *G*.

"The computer just crashed," said the worker at SRI. The failure was caused by a bit of programming that was probably too clever by half. Once the SRI machine recognized the letters *L-O-G*, it completed the word. "When the SRI 940 system received the *G*," Kline recalls, "it tried to send back *G-I-N*, and the terminal program wasn't ready to handle more than one character at a time."

Later that day they tried again. This time it worked flawlessly. Sitting in Los Angeles, Kline logged in to the machine in Menlo Park and executed commands in its time-sharing system. The SRI computer responded as if the UCLA computer were a bona fide dumb terminal. There is no small irony in the fact that the first program used over the network made the distant computer masquerade as a terminal. All the work to connect computers ended up with the same master–slave relation the network was meant to eliminate.

But it was only the beginning. Within a year, four IMPs had been installed and the NWG, under pressure from Roberts, had finally devised a workable host-to-host protocol. The network that had begun as a high-risk experiment was well on the way to becoming a reality. The hardware worked; the software worked. Above all, the concept that had been dismissed as impossible, a technology on which the entire enterprise turned—packet switching—proved splendidly reliable. Computer networks, the most revolutionary two-way communications tool since the invention of the telephone, were born.

"Technological development is like building a cathedral," Baran remarked in retrospect. "Over the course of several hundred years, new people come along and each lays down a block on top of the old foundations. If you are not careful, you can con yourself into believing that you did the most important part. But the reality is that each contribution has to follow onto previous work. Everything is tied to everything else."

Baran's simile captures the collaborative spirit of the Internet's creation. But an invention so momentous can hardly fail to breed any number of creation myths. At the end of 1989, dozens of network pioneers gathered in Los Angeles to celebrate the ARPANET's twentieth anniversary. There, at a symposium called "Act One," one attendee told the following story:

"In the beginning ARPA created the ARPANET.

"And the ARPANET was without form and void.

"And darkness was upon the deep.

"And the spirit of ARPA moved upon the face of the network and ARPA said, 'Let there be a protocol,' and there was a protocol. And ARPA saw that it was good.

"And ARPA said, 'Let there be more protocols,' and it was so. And ARPA saw that it was good.

"And ARPA said, 'Let there be more networks,' and it was so." ●

KATIE HAFNER is a contributing editor specializing in technology at NEWSWEEK *magazine and is coauthor of* CYBERPUNK: OUTLAWS AND HACKERS ON THE COMPUTER FRONTIER. *MATTHEW LYON, assistant to the president of the University of Texas at Austin, is a former journalist. This article is adapted from their forthcoming book,* WHERE WIZARDS STAY UP LATE, *being published in September by Simon & Schuster.*