



/97/\$25.00 © 1997 IEEE
Published by the IEEE Computer Society

Len Kleinrock on the Origins of the Internet:

Web Extra

Today he moves through his busy schedule carrying an arsenal of modern communications devices—a Motorola two-way e-mail pager, a Psion PDA, a cell phone, and often, an IBM ThinkPad. To many people, these are tools worn as badges to indicate membership in the tech-savvy army of business professionals. To Leonard Kleinrock, <http://millennium.cs.ucla.edu/> they're a different kind of badge—more like campaign ribbons for the work he did to make the Internet a reality.

Article Contents

[Web Extra](#)

Download Citation

- [ASCII Text](#)
- [BibTex](#)
- [RefWorks Procite/RefMan/Endnote](#)

Download Content

 [DOWNLOAD PDF](#)
PDFs Require [Adobe Acrobat](#)

Known as the "father of modern data networking," Kleinrock developed the underlying principles of packet switching, the communications technology of the Internet. He also laid out some of the key functional specifications for the Arpanet, predecessor to the Internet. Destined for a career in science, Kleinrock's early interest in radio cast him through the Bronx High School of Science, City College of New York, and eventually a full graduate fellowship at the Massachusetts Institute of Technology in the Electrical Engineering Department.

Kleinrock joined the faculty at UCLA in 1963. Through his ongoing participation in establishing the data networking technology for Arpanet, his UCLA lab became first node on it. This first switch (known as an interface message processor, or IMP) arrived at UCLA on Labor Day weekend, 1969. Kleinrock's team of 40 people connected the first host computer to the IMP and started sending packets of data. Along the way, Kleinrock was the first person to use the Arpanet for personal e-mail/chat—not a function in the original specification, but he needed to retrieve an electric razor left in England.

Since the pioneering days that led to the birth of the Internet, Kleinrock has continued his participation in its growth and development. Most recently, new technology for nomadic computing and communications has been on his mind. He's developing technology that will support the nomadic user in his computing and communication needs as he travels from place to place.

Internet Computing's EIC, Charles Petrie, got a chance to talk with Kleinrock at the Supercomputing Conference in Pittsburgh, where Kleinrock received the Computer Society's 1996 Harry Goode Memorial Award for "fundamental contributions to packet switching and queuing theory, two of the principal technologies that led to the Internet, empowering the global community to participate in worldwide economic, political, and cultural processes."


What follows is the behind-the-scene story of how the Internet came to be, as told by one of its founders.

Petrie: As I understand it, your doctorate thesis outlined the ideas behind packet technology 10 years before the Arpanet was launched, is that right?



SAVE 20%
on 100's of Wiley-IEEE Computer Society Press titles!

Use code 2009C at checkout



WILEY-BLACKWELL
IEEE
IEEE COMPUTER SOCIETY

Kleinrock: That's correct. My dissertation laid out the basic principles for packet switching and data networking. I developed not only the basic analytical tools but the basic network design principles as well. And I talked about what the tradeoffs were and why they were so attractive.

Petrie: So you actually did some of the design as well? Design in terms of the tradeoffs?

Kleinrock: Yes, that was a major part of my research. I developed a way to calculate the optimal channel capacity assignment. I invented effective dynamic routing procedures and also established the analytic model by which you could calculate delay . . . and to simulate it I had to make some fundamental assumptions—I simulated the hell out of it to show that the assumptions worked. It's the kind of problem where if you try to solve the exact problem, you'll never get it solved; it's totally intractable. And it's intractable for two reasons. The first reason is that the queuing theory just kills you.

Petrie: Is it inadequate, or is it just too big?

Kleinrock: Inadequate. We don't know how to solve these multidimensional queueing problems. The second reason you can't solve the exact problem is due to the topological complexity. Network traffic flow is a multicommodity flow problem, and you can't solve multicommodity flow problems when more than three commodities flow in the network. As an example of a one-commodity flow problem, consider that we want to ship grapefruit around the country on the railroad system. Each link of the railway network has a certain capacity—so many tons a day can be shipped on that link. The problem is to figure out the maximum tonnage of grapefruit that can be shipped through the railway system per day, assuming that the entire system is devoted to that task.

That problem is solvable. It's called the max-flow problem. Suppose instead that I want to send apples and oranges from different locations. That's two commodities. That's a little harder. Three commodities you can do in some cases. Beyond that, there's no efficient computation algorithm.

Now in networking, every pair of users is a different commodity. So from a theoretical point of view, you just can't solve these problems. So I asked, what can I do? And I made an assumption. I called it the independence assumption. And that cracked the problem wide open. The analysis became totally doable.

Petrie: And the assumption is?

Kleinrock: The problem involves the sending of a sequence of messages. If you look at what's happening on any link, you will see a sequence of messages arriving, and the time between the arrival of the first of two adjacent messages is related to the service time of the second message. In queueing theory, when service times and arrival times are correlated, you just can't handle the analysis. In the case of sequential messages, we see a very high degree of correlation, because the minimum time between two arriving messages is the service time of the second one.

My assumption was that they're not correlated. This was the independence assumption. I assumed that every time a message hit a node, its length was randomly reselected, guaranteeing that its length was independent of its arrival time. Sounds like a bold assumption but it turns out to be fine. The reason is that it's not only my stuff moving down the line, but other people's traffic is also joining the fray. In addition, my messages go out to different places—one goes here, one goes there—so it busts up that dependence very effectively from an engineering point of view.

That was the assumption needed to allow the analysis to go forward. But you must recognize that I was the first to develop an exact (and very simple) formulation of the problem. I can tell you what the equation is, it's trivial. It's a one-liner. It's exact. But it contains a term you can't solve for, and that's why you have to use the independence assumption. It totally broke the back of the problem. Once you have that, you can do a design based on that assumption. It has now come to be known as Kleinrock's Independence Assumption.

Petrie: But your design is based on your simulation?

Kleinrock: The simulation served to show the accuracy of the independence assumption. The design procedure is based on an analytic

fit to tariff data. It turns out that the design problem was really very simple; in fact, once you have the analytic fit, you can do an exact analysis for the channel capacity assignment.

Suppose you have so much total capacity. Where do you put it in a network to minimize the average delay? How many bits per second do you put in each lane? To do this, you should know the traffic matrix. Once you have that, my simplest formulation led to what I called the optimum square root channel capacity assignment, and I generalized that to far more complex formulations.

Petrie: How did you generate a workload in your simulation?

Kleinrock: In the simulation I chose a variety of workloads. They were randomly generated, uniformly generated, generated with hot spots, and all the rest. I tried a variety of streaming stuff. I tested over a whole range. And so my thesis, which was filed in 1962, became a book that was published by McGraw-Hill in 1964. Recall that my first articles came out in 1961. I actually started work on packet switching in 1959. I mention the date because that's before other work that has been given credit for the invention of packet switching. I looked at the pure analytical case and had the exact formulation, and an analytic solution proved correct by simulation.

I also studied the effect of slicing things up into small pieces. This time-slicing algorithm showed the effectiveness, i.e., the gains in response time, that you get from "packetization." Of course, the word packet was not used in the data network field until the late 1960's when Donald Davies coined it for this application.

Petrie: But it was published; when was the article published?

Kleinrock: The first article was in *RLE Report, Research Laboratory for Electronics Report* in July of 1961. That predated everybody else's stuff. My thesis was handed in in December of 1962.

Petrie: How much of your analysis eventually made it into the design of Arpanet?

Kleinrock: Well that's the other part of the question, because we started this conversation with you saying I had developed my ideas 10 years before Arpanet. The genealogy is this: I had some very well-known PhD classmates at MIT. One of them was Larry Roberts. Others included Ivan Sutherland, Tom Kailath, and Jacob Ziv.

Petrie: In the same class?

Kleinrock: Well the same group, going through together. I don't know who else you would know, but anyway, a powerful group of guys. You know, Claude Shannon, the father of information theory, was on my committee. It was a wonderful time. Larry Roberts, a very close friend of mine to this day, is a brilliant guy, and he did his work on picture processing. When I graduated I went directly to UCLA. My entire graduate education was fully funded by the MIT Lincoln Laboratory through their staff associate program. I worked for them in the summers. I also worked for them for six months after handing in my dissertation. They encouraged everyone to examine the market for research positions, just to see what's out there. They were very generous. As it turns out, I was offered this terrific job at UCLA. Lincoln Lab was sincere in their encouragement for new graduates to find the best positions possible, so I accepted the UCLA faculty position. Larry stayed on at Lincoln Lab. He was also a staff associate.

Now the name Licklider comes into the picture. Let me tell you about ARPA. Do you know the Sputnik story?

Petrie: No.

Kleinrock: In 1957 the Russians launched Sputnik, the first space satellite, and caught us with our pants down. The United States government said, "We can't ever let that happen again," so in 1958 they formed the Advanced Research Projects Agency. And they started doing work in a variety of areas, including some very early computer work. In 1963 they created an official office for computer research, the Information Systems Technology Office, ISTO. It's called ITO (Information Techniques Office) now.

J.C.R. Licklider was the first head there. He was the head before it was official. He was a visionary with a background in psychology. Now that time, the 60's, was the era of timesharing. Every time ARPA came to a new investigator and said we want you to do research for us, the investigator would say, "fine, you want me to do computer research? Buy me a computer". So they bought him a computer. After a while, it got a little old, because every guy that got a computer made something special out of it. And every time a new one was bought, they wanted the capability of all those others. So Licklider conceived the idea of what he called "the galactic network." He actually used that word. He wanted to create a network whereby if you had some unique application, others could share it through a network. That was his motivation. He was the only one who had that long-term vision. Now of course I don't know if he foresaw what we have today. But he was thinking much further ahead than anybody else. None of the rest of us down in the trenches could see it.

Petrie: It seemed utterly impossible.

Kleinrock: You have to remember that PCs weren't invented yet. So Licklider wanted to have a network developed. He brought in a fellow named Bob Taylor to take over the office, and he stepped aside. But he influenced Bob to make a network happen. Bob Taylor hired Larry Roberts to come to the office and oversee his operation. This was in 1966. Larry said, "I'm going to get Len Kleinrock, since he knows just how to do this." So he brought me in, not to the office, but to the project, while I was at UCLA.

Petrie: This was two years after you published your dissertation?

Kleinrock: That's when the book came out. The dissertation was officially published when I handed it in, in 1962. Larry saw my early drafts, the whole thing. He was totally influenced in his thinking by my work on networking. That's why he accepted the job—he knew that the technology was doable.

So Larry called me in, and in 1967 he gathered half a dozen people together to try to write the specification for what this network would be. Fred Baskins was there. He was a time-sharing buff, and he banged his fist on the table and said, "If this network can't give me a half a second response time, I can't do time sharing." So we said okay, the spec is a half a second roundtrip.

Then I banged my fist on the table and said, "Look, this is an experiment. We have to know what's going on, we have to put measurement hooks in the network switches." These measurement hooks in the software should contain artificial traffic generators, measure traffic flows, collect and display histograms and other measurement tools. There was also the issue of reliability. How do you specify reliability? Well you could do it the way the military does and ask that the network be operational 0.9999721 percent of the time . . . which is nonsense. We chose a far more pragmatic measure. We said that if any single link breaks, the network should still function. That meant that there would have to be two independent layers. So we wrote the spec. In January 1968, we released the spec. It was a request for a proposal. In January 1969, BBN (Bolt, Beranek, and Newman) won the bid. Their job was to select a minicomputer, modify the hardware and software to meet the spec that we had created for the switching node, what we called an IMP (interface message processor). They also had to manage the Arpanet—you know, worry about the lines staying up, talk to AT&T, talk to all the modem manufacturers, lease 50 kilobit-per-second lines, and more.

Because I had done the pioneering work on networking, it was decided that my UCLA lab would become the first node on the network, and we would become the network measurement center as well. It was our job to stress the network, run experiments, try to test it and crash it if we could.

Petrie: So UCLA was the first node. Did you have the IMP yet?

Kleinrock: No, not yet. We had to get ready for this IMP that BBN was putting together. And their switch was coming in September 1969, and we needed to get the IMP to host the specification that BBN was supposed to deliver.

Petrie: What was the host machine?

Kleinrock: A Scientific Data Systems, Sigma 7-SDS Sigma 7.

Petrie: And the IMP was a DEC?

Kleinrock: No, it was a Honeywell minicomputer. Honeywell announced this machine in 1968, and they showed it, I believe, at the Fall Joint Computer Conference, 1968, in one of the big cities. They had a military hardened version with four eyehooks on top. At the show, they had it suspended from the ceiling, lights running, swinging in the air, and a big strapping guy with a sledgehammer was whacking the side of the machine to show it wouldn't fail. And it didn't fail; it kept working. And by the way, I think that's the machine we got. There were no dents in ours, but we have a hardened version at UCLA.

Petrie: It's still there?

Kleinrock: Yep, it's still there. And the heck of it is, that neither the UCLA Chancellor, nor the city of Los Angeles make any publicity about it. You know, Philadelphia is the City of Brotherly Love, right? Los Angeles is the city where the Birth of the Internet took place. You'd think a politician would take that and run with it!

Petrie: This wasn't a very big machine?

Kleinrock: No—the size of a public telephone booth or a refrigerator. It had 16K of RAM and 16 levels of interrupt. It could handle eight lines including host interfaces and lines out into the network. That was the max.

(Note: See <http://millennium.cs.ucla.edu/LK/Inet/birth.html> for a photo of the first IMP.)

Petrie: So now you've got a team of guys working to write the interface?

Kleinrock: I put together a software team and a hardware team, a total of 40 people. We had to build the interface and write the protocol according to spec that BBN finally released to us. The hardware team was headed up by Mike Wingfield. The software team was headed by Steve Crocker. And underneath him were Charley Kline, Vint Cerf, Jon Postel, and others. They were a fine team of people and they were writing the code for the IMP.

Petrie: And what were they writing in?

Kleinrock: It was assembly language I think. I can't recall. I didn't do any of the programming, didn't want to.

Petrie: And Crocker was the head of the team?

Kleinrock: He was the head of the software team. They were aggressive guys, as software guys tend to be. Shortly after the IMP was installed we had the problem of creating the host/host protocol. We had developed the IMP/host protocol, and now we needed host/host. That was a major, major effort. We didn't realize how hard it was. I gave it to this distributed group of graduate students around the community, and it took them two years to write it, but it finally got written. During that period the network was not very well used because there was no easy way for hosts to talk to each other. About the only way that happened is if somebody, say from Utah, took a job at Santa Barbara and wanted to talk to his machine back in Utah, he knew just how to do it. So it was migration of people that allowed the early work to get done. And finally the protocols were put in place.

I want to backtrack on one thing about Larry (Roberts). Larry certainly had strong credentials in networking. He conducted the first experiment trying to connect two computers together. He connected the machine at Lincoln Lab to a machine at SDC (Systems Development Corporation) in Santa Monica. And it was terribly hard. This I believe was in 1966. It was a failure, basically. I mean it was guessemly hard because there was no protocol, no understanding of what to do. That's another reason why he was prepared to try to create this network.

Petrie: So he was not only very good at motivating government and industry to fund this project, but he's also knowledgeable?

Kleinrock: More than that. First off, he's one of the smartest guys I know—he's a brilliant, brilliant engineer. I mean he was not an administrator, he was an engineer. He got his hands very dirty. He was there all the time. He was designing topologies, you know, he's an engineer's engineer. Very smart guy.

Another name that's important is Robert Kahn. He was at BBN in those early days. He was on their side implementing the algorithms and adding his own inventiveness to the algorithms. He did a good job. Bob is a very innovative guy. Then he moved; he replaced Larry at ARPA when Larry went on to become the first president of Telenet. And he continued many of the programs Larry began, such as the packet radio and packet satellite stuff. And now he's heading up CNRI.

Okay, so the IMP was supposed to arrive on the Labor Day weekend, September 1969. And we were late in our preparations for its arrival. We knew they were also late at BBN, and were thrilled since we could really use the time. And it was going to take them two weeks to ship it out, thank God. Except they decided to air freight it. [laughter] And it arrived on the Labor Day weekend. Monday was Labor Day, and Tuesday, the big event. The IMP arrived, we were going to connect the first machine, the first IMP. So everybody and his brother was there. UCLA Computer Science department was there. My guys were there. ARPA was there. Honeywell was there. BBN was there. AT&T long lines was there. UCLA School of Engineering was there. The administration, GTE. Everyone was there, ready to point the finger if it didn't work. [laughter] But it worked. We had bits moving the first day. We had messages working the second day. It worked like a charm. Like a charm! It was unbelievable.

Petrie: You were sending e-mail?

Kleinrock: No, we were sending and watching bits and messages, there was no content to speak of because this was not host to host. This was host to IMP.

Petrie: What do you say to an IMP?

Kleinrock: You just send a message, you know, is it correct? Any bits in error? And if it gives you acknowledgment, then the message got there. You know, it was being packetized and all of that.

A month later SRI (Stanford Research Institute) got their IMP and connected their host computer as well. It was a DEC PDP machine I believe. Now there were two hosts. So one evening shortly after they were connected, Charley Kline and I got together and arranged with a fellow up at SRI to be prepared to receive a message. As it turns out, along with the digital line you could basically take off an audio line, a voice line, as well. So we could talk to our SRI collaborator as well as send to him.

And send we did. The idea was to login to the SRI machine from UCLA; this required that we type in LOGIN. Now the PDP machine was one of these systems that had a character-oriented command language. With such systems, you would type in a few characters. And once the string is unique, it would fill out the rest of the command. So if you were typing in the word "login," and once you typed the L-O-G, it filled in the I-N.

So, Charley typed the L, and he said "Did you get the L?" "Yes, I got the L" came back the reply from SRI. "Did you get the O?" "Yes, I got the O." "Did you get the G?" Crash!

A few hours later we got it running. This is not "What hath God wrought?" This is login. And we didn't even have a camera to record the event. Following that, the network grew. We paused after the first four nodes to test our little network. We were testing things like alternate routing, for example.

Petrie: But that first crash, that was because the DEC machine wasn't complete?

Kleinrock: We don't know what happened. Somewhere in the protocol some funny thing happened, a line error? The other interesting

story is that one machine was talking ASCII, the other was talking EBCDIC, and Charley had to map the character codes in his head. He happened to know both. He had to go from one and convert it to the other, God bless him. [laughter] You know, these are the things that made it human. So then we paused, we tested, we started doing our measurements, and then by summer of 1970, we had 10 nodes and we spanned the country. And it just grew.

And at my Network Measurement Center, we could crash the network at will.

Petrie: How?

Kleinrock: We would stress it. The point is that the software, the flow control software, was written in ad hoc fashion spread all over the place. And then you add something here and you don't know what the effect there is going to be. I'll give you one example, a very simple one. This is a trivial thing. So much memory was put aside for reassembling packets back into messages. And what's the maximum number of messages you might be reassembling at the same time? Well, the smallest thing you can reassemble is a two-packet message. With a one-packet message, there's nothing to reassemble. Take the number of packets, divide by two; that's the maximum number you can be reassembling simultaneously. Good. Assign that many pointers, whatever the number is.

But one day they found out 16K was too small, so they added 16K more. Now you have much more memory for reassembly, but they forgot to increase the number of pointers. Natural, easy mistake, right? So when memory got full, you couldn't point to it. You'd allocate it, stick it in, and you couldn't find a pointer. Crash! Simple thing. And the reason we discovered that is because our measurement messages were two packets long. And we generated a lot of traffic from all over the network which was designed to back to UCLA. BBN called up and said we crashed the network. What you mean? we said, we're taking measurements. So they restarted, and it crashed it again. And then we finally told them what it was.

Petrie: What about the host interface? There's a different host interface for every host? Was there a different type of host at every site?

Kleinrock: No. There was a preponderance of PDP-10s. In fact so much so that ARPA stopped buying PDP-10s because this almost became a homogeneous network, and they wanted to develop a heterogeneous network.

Petrie: So how many hosts were there initially?

Kleinrock: One per IMP. Typically one, occasionally two per IMP. So the number of nodes grew. It was 10 in summer of 1970.

Petrie: At that point, how many different kinds of hosts?

Kleinrock: Oh, I suspect about four or five different hosts. But we didn't have the host/host protocol then. October 1971 is when we finally got the protocol implemented in most of the host machines. And the DEC PDP-10 was the main host machine. For example, by August 1973, there were 16 PDP-10's, eight other PDP-class machines, six IBM 360-class machines, one 370-class machine, one Burroughs B6700, the TX-2, some miscellaneous machines, and, of course, our lonely SDS Sigma 7. It was just a weird mix of things. Each one required its own surgery to adapt to the system.

Petrie: By a different team? It's a wonder that it worked.

Kleinrock: Yes, and that was its strength, too. You can imagine how much BBN liked us as we kept crashing their network any time we wanted. So what would happen was the network would crash, but BBN didn't give us source code for the IMP operating system. They held it close to their chest. So when it crashed, we would say, "Here's what happened, please fix it." They would reply, "No, it's too complicated to fix; it's gonna take us six months." We got that answer continuously. So one day the government pressured BBN to release the source code. After all, the government had paid for it, and BBN had no right to keep it from the government. So now we got source code. Then we could see what happened, we'd see what the trouble was, we'd write the code to fix it and send it BBN. And it still took six months. [laughter]

Petrie: But every site was doing this now?

Kleinrock: No, we were the only ones stressing the network. We were the Network Measurement Center for the entire Arpanet. Each IMP had the ability to measure itself and send us back messages. So we turned on and turned off the measurements.

Here was a really interesting problem, what we referred to as reassembly deadlock. I published the first book about the Arpanet in 1976 (*Queueing Systems: Computer Applications*, John Wiley & Sons) and described reassembly deadlock there, along with many other deadlocks. Let's assume that a number of messages are in the process of arriving at some destination IMP, which is reassembling these messages. But there's only so much space for reassembly. When a packet for a new message comes in, it might be eight packets long (that was the max), and so eight packet buffers are put aside for reassembling that message. So you have a bunch of eight-buffer segments put aside, partially filled, none of them complete; let's refer to the missing packets as "pink" packets.

Let's further assume that all of the reassembly buffer space is currently in the process of reassembling these messages. Now if you look at all the switches in the IMPs attached to this IMP, they're also sending packets to this same destination IMP. Let's further assume that all of the storage in these IMPs is full of packets, but none of them are the missing pink packets. They're all from newer messages (say "green" packets).

Where's the missing pink stuff? One layer behind that. Now these missing pink guys can't get to the destination until these green guys get through; and the green can't get through until the pink get through, complete the reassembly of some messages, and release the reassembly buffer space! Bang! That's your re-assembly deadlock. It forced BBN to change the whole operating system for the IMP. That was deadly.

Petrie: They should have anticipated that one.

Kleinrock: Yeah, well. These were new times, you know? And it was complex. I mean in order to get a message into the network, you had to grab an available logical link, you had to get storage at the other end, and you had to get a token to move it. So those three things had to be collected. They're all in different locations with different control processes collecting them. You can imagine the craziness.

Flow control is still the Achilles' heel of our networks. That's what took down the AT&T network, takes down the Internet. Little simple things. Signaling System 7, the signaling control structure of the telephone network, took down AT&T. Of course, it's a digital system. One of the switches thought it was overloaded—it wasn't—and it told the other switches next to it, "Stop sending," and that propagated. Crash! Once you see these things occur, you can easily fix them. The trick is to anticipate the problem, and that is very very difficult.

Petrie: You're saying this brings down the Internet. Has it brought down the Internet?

Kleinrock: The Internet goes down every so often. Sometimes you don't know why. But in those early days we knew what was going on because we had hooks all over the place. They removed the Network Measurement Center in the mid-70s. BBN continued with the maintenance function, but DCA (Defense Communications Agency) took over the Measurement Center in 1975. After that we don't know what happened. So that was the growth of the network.

Petrie: When you finally did get the host to host protocol, what was the first use?

Kleinrock: Well, there were some uses before that, but e-mail was put in in 1972 as an ad hoc add-on by Ray Tomlinson. Just a trivial thing to add to a network. It caught fire and dominated network traffic immediately. That was again another signal of what was coming today. The people-to-people thing.

Petrie: What was the use before e-mail?

Kleinrock: Some military Air Force bases, Tinker and McClellan, sent some stuff. We don't know what it was. We have no idea. Mostly sending a message to someone else or trying to use a remote machine. But it was a crude kind of e-mail. Just testing machine-to-machine

communication. File transfers, sending documents around, very low-level use.

Petrie: Are we talking TCP at this time?

Kleinrock: Oh, no. The original control protocol was NCP (Network Control Protocol).

Petrie: So at this point what were you doing for the equivalent of FTP and telnet?

Kleinrock: TCP doesn't play telnet. Telnet was there ahead of time, early on. There was ICP, which is the Initial Connection Protocol. There was the host-to-host protocol. There was even a voice protocol. There was a file transfer protocol. And there was telnet. All those things were being developed, and the idea of layering these protocols was already present in the early Arpanet days, way before IBM came out with theirs or the ISO/OSI seven-layer architecture. It was all in the early Arpanet design.

Petrie: And e-mail quickly began to dominate the early use?

Kleinrock: Almost immediately. I'll tell you the first illegal use of the Internet. I did it. [laughter] In 1973 we had a meeting in the University of Sussex in Brighton, England, on computer communication networks. We were housed in dormitories on the campus. I had to leave a day early and when I got home, I noticed I'd left my electric razor in the dormitory. I wanted to get it back. The conference had set up a temporary link from London, which was on the Arpanet at the time, down to Sussex. And I said, "Gee, who would be logged on somewhere in the network. It was 3:00 in the morning Sussex time. What fool would be logged on at that time?" I thought, "Ah! Probably Larry." There was a program called Resource Sharing Executive. It knew it lived in a network. So you could say "Where Roberts," and it would log onto every host, look at the "who" list, see who was logged on. The network was still small enough in those days that you could do that.

Petrie: That was the early equivalent of Finger?

Kleinrock: Exactly. A couple of minutes later it came back, "Roberts logged on as TT [teletype such-and-so] at BBN." I connected to him with a "talk" session. I said Larry, blah-blah-blah. He said, "Don't worry." Next day, I got my razor!

Petrie: [laughter] The first personal use.

Kleinrock: [laughter] As far as I know.

Petrie: In that particular example it sounded more like chat than e-mail.

Kleinrock: It was chat. Not e-mail. We had a split screen. Actually, it wasn't a split screen. It was line after line; it used "GA" for "go ahead," that kind of thing.

Petrie: But there you are. What year was this?

Kleinrock: It was 1973.

Petrie: So in 1973 you already have e-mail, chat, telnet, everything.

Kleinrock: Yeah. It was all there. What happened then was that Larry initiated a satellite packet radio program in about 1972 or 1973.

That's where Norm Abramson introduced the Aloha technology. Aloha had been in use to connect the various islands of the Hawaiian chain to the University of Hawaii, and we adopted it for satellite use. I did the analysis for that as well. And we got a satellite link across the Atlantic doing Aloha, broadcasting from four nodes. The interesting part here is that we were the Network Measurement Center in California, measuring and controlling a satellite over the Atlantic, using the Arpanet to get to it. It was a real kick. Before Larry left he began to start a program in packet radio, ground radio, which Bob Kahn then took over.

Petrie: This is the idea of what some people have called the data sphere.

Kleinrock: Yes. No base stations, just plop down the radios. The military environment is what we were using as a model application.

Petrie: You would parachute in a whole bunch of radios?

Kleinrock: With people or whatever, and they form a network. The key idea was that there were no "base stations." Rather, multi-hop technology was used to cover distances. This was a very hard problem to solve. The first thing you have to choose is the channel access method. Once again, Abramson's group from Hawaii had been trying something called CSMA, carrier sense multiple access.

Let me back up. At one of the first meetings Abramson came to, he talked about Aloha. I was there with one of my PhD students, Simon Lam. We looked at ourselves and said we just have to analyze the behavior of Aloha. So I put him on to it and we solved Aloha. A couple of years later, I was at another meeting for packet radio with another of my PhD students, Fouad Tobagi. Abramson comes again, this time with CSMA, and writes an equation on the blackboard describing the throughput-traffic relationship. I looked at it and said, "That's wrong."

On the airplane ride home, we analyzed it and got the correct and exact answer. In fact, we analyzed a more general version of the problem. That was the first analysis of CSMA, which then led to CSMA/CD, which is the Ethernet protocol.

Petrie: Really?

Kleinrock: Bob Metcalfe took the CSMA protocol we had analyzed, which means you listen before you talk, and he added CD (collision detect)—which means you listen while you're talking—and put it on a wire. By the way, Ethernet is the wrong name. It's not over the ether; it's over a wire. But at least it had CSMA/CD.

The latest version of Ethernet no longer has CSMA/CD. Gigabit Ethernet has abandoned CSMA/CD for some of its modes. So what does it have? It's not Ethernet anymore. It has a frame format that's common, that's all. Anyway, the point I'm making is we had the Arpanet, we had the satellite network, now we had the packet radio network. Three networks. They couldn't talk to each other very well. And that's what motivated Bob Kahn to conceive the idea of TCP/IP. He later described the ideas to Vint Cerf, and suggested to Vint that he install it at the operating system level. It was Bob's idea to put it together.

Petrie: What a great genealogy. That's wonderful.

Kleinrock: You know, there's a book about this, the one by Katie Hafner and Matthew Lyon, it's called *Where Wizards Stay Up Late*, <http://www.simonsays.com/titles/0684812010/>. It just came out about six months ago. She got much of the story. She interviewed everybody, but the result has some gross distortions.

Petrie: [laughter] Now are you saying that the press doesn't always get it right? But just to recap, you got three networks and the protocol that you were using for early Arpanet was NCP?

Kleinrock: NCP.

Petrie: What did the satellite and the wireless use?

Kleinrock: In the wireless ground radio, they were more interested in the link level and the network level. We're just getting stuff across, not so much worried about host to host. So it wasn't easy to use it. And the satellite was really piggy-backing on part of the Arpanet protocol.

Petrie: Did TCP slowly migrate onto the Arpanet or was it adopted all at once?

Kleinrock: It migrated slowly, but then it was later ruled that you cannot join this network of networks unless you implement TCP. It was dictated.

Petrie: Your work with distributed and wireless networks has led to an interest in nomadic computing. What constitutes nomadic computing?

Kleinrock: Well, I recently formed a company, Nomadix, <http://www.nomadix.com> and we are developing software and services for nomadic computing. Nomadic computing is designed to provide the system-level support for users who travel. Understand that "travel" may be as simple as moving from the high-performance workstation on your desk, with its excellent connectivity to a LAN, to a conference table in the same room to use your laptop whose screen, keyboard, and capability are less than your workstation's and whose connectivity may be zero or simply wireless. The system support we're developing is meant to make this move as seamless and transparent as possible.

Indeed, one can talk about all kinds of mobility. Geographic mobility is the obvious one. The second is social mobility—which refers to movement among roles for a given individual (engineer, father, husband, et cetera). The third is appliance mobility, when I change the communications or the computing platform I'm working on. And the fourth is application mobility—that is, in a given social context, I may move between word processing to graphics processing to a database application, and so on. The key is to provide services like file updating, synchronization and reconciliation, location identifiers, et cetera. Overall, nomadic computing support seeks to provide the illusion of connectivity, even when you are disconnected. There are some severe implications here, because none of our systems right now can do that properly.

Petrie: Just like the model of Eudora for mail?

Kleinrock: Exactly. I want to transmit something; it doesn't get transmitted now, but so far as I'm concerned it was transmitted.

Petrie: The POP server.

Kleinrock: Yes, a personal portable POP server.

Petrie: Pretty hard to do that with a Web browser.

Kleinrock: Yes and no. You can pre-cache but you can't click on something you did not anticipate ahead of time. It is hard, that's right.

Petrie: Well I do that manually now. I do cache . . .

Kleinrock: You do HTML files . . .

Petrie: Yeah, and I carry them with me and I call up Netscape on my Macintosh and look at them, even though I'm off-line. It would be

real nice if that was somehow automatic and done on a large scale—on a CD-ROM, for example.

Kleinrock: We're actually looking at some of that, my colleagues and me. There is a person who's working a project called Seer, as in prophet. It gathers information regarding which files you touch when you work on a project. Like do you know that your ATM font files are coming in? Or some other links? It will bring it all in to you. But that's a really hard problem. I have another student working on predictive caching in the Web environment while you're connected. That's another issue.

Petrie: Predictive caching? Sounds hard.

Kleinrock: It is hard. For example, it accounts for the cost of communications, all that stuff. Or suppose I had done some work while disconnected and I suddenly enter a region where there is some connectivity. What should we exchange?

Petrie: What should be exchanged first?

Kleinrock: That's right. Because I may come out of a communication-rich environment soon, or I may only be connected for a limited time, and what about compression? If I have weak connectivity, please only send me a fax instead of full text. And don't send me a full-motion video clip. So all these issues are very important. And our systems have to be nomadically enabled to begin with. You can't easily build this in afterwards.

Right now, a significant change of connectivity or latency in your application or your environment is considered a failure or an exception. What communications line broke? It's should rather be considered a usual event—you must build it in. And that's a whole shift in the way we're going to be working.

Petrie: So you treat disconnects as a normal event?

Kleinrock: That's right. And more than that. I am interested in all the middleware to support that and give you the right environment and give you the transparent architecture. The first Nomadix product is rather interesting. You know those trade-show badges, the little plastic card with a magnetic stripe? You go to an exhibit and you swipe the card at the exhibitor booth—and the exhibitors are complaining like hell because they're only getting your name and company. So you get more sophisticated cards; there's all kinds of things like that.

We have a new device which replaces the swipe-card type of technology with a floppy disk and we call it the IDisk. Your name and other identification information is stored on the floppy when you register. When you go to a booth and insert your floppy disk into the exhibitor's laptop, he gets your information and gives you back information—either what he wanted to give you or what you select from a menu. It's right in his laptop, and it's a floppy disk which is cheap. You go from exhibitor to exhibitor. And what are you doing? First of all, you're going to take the trade show home on a disk—the brochures they give you and so on stored electronically on your disk. Secondly, we have been forming a Web page on your disk in the process. This stuff is in HTML format; you've got graphics on the disk and all of that. You go on the airplane and use Netscape, or any other browser, to navigate through all the information you have collected. Now none of the links are hot because you're not connected. If you try to click on an exhibitor's logo, it'll indicate that you can't connect. But if you *are* online, then you're linked. You can go right to any exhibitor's page.

Petrie: But the local internal links are alive?

Kleinrock: Absolutely, and that's one of its really attractive features. We refer to the page on your disk as the MobileWeb. So that's our product. We've used it, for example, at the ATM Year 96 conference. It was a big hit. And it's in the process of rolling out as we speak.

Petrie: Yeah. It's like replacing key punches, isn't it? Matter of fact, IBM got rich doing that.

Kleinrock: Yes. The nice thing is that we can interoperate with any of the existing registration systems that produce magnetic stripe badges or any machine-readable badges. Either at the registration site, you can swipe your badge and out comes our floppy disk, or you

could swipe it at an exhibitor's station and get your floppy disk there. And then when you go the next booth, you've got it.

Petrie: As a parting note, and an appropriate one, what I think you're saying to the readers is that one way or another, things are gonna get a lot better in the next few years.

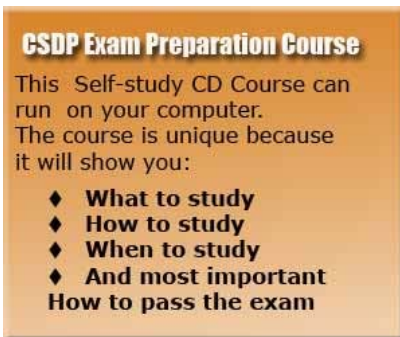
Kleinrock: And a lot more exciting.

For more information on the origin of the Internet, see

<http://www.isoc.org/internet-history/>.

IEEE Internet online Computing

Send general comments and questions about the *IEEE Internet Computing Online* website to internet-computing@computer.org



CSDP Exam Preparation Course
This Self-study CD Course can run on your computer. The course is unique because it will show you:

- ◆ **What to study**
- ◆ **How to study**
- ◆ **When to study**
- ◆ **And most important**
How to pass the exam