

Interview

An Interview with Leonard Kleinrock

The UCLA professor and networking pioneer reflects on his career in industry and academia.

Leonard Kleinrock, developer of the mathematical theory behind packet switching, has the unique distinction of having supervised the transmission of the first message between two computers. As a doctoral student at MIT in the early 1960s, Kleinrock extended the mathematical discipline of queuing theory to networks, providing a mathematical description of packet switching, in which a data stream is packetized by breaking it into a sequence of fixed-length segments (packets). ACM Fellow Kleinrock has received many awards for his work, including the National Medal of Science, the highest honor for achievement in science bestowed by a U.S. president.

UCLA Professor and ACM Fellow George Varghese conducted a wide-ranging interview of Kleinrock, an edited version of which appears here.

GEORGE VARGHESE: Do you remember any epiphany as a boy that led you toward communication?

LEONARD KLEINROCK: I remember early in elementary school reading a *Superman* comic whose centerfold showed how to build a crystal radio out of household items that one could find on the street: a razor blade, some pencil lead, a toilet paper roll, and an earphone (which I stole from the telephone booth in the candy store down the street). I also needed a variable capacitor and had no clue what that was,



Leonard Kleinrock.

but my mother, bless her heart, took me to a store in the electronics section of New York City, namely, Canal Street. The clerk helped me select the right part. Oh, the magic of listening to music from my newly built radio; and it required no battery or power at all. After that, I kept cannibalizing old radios and used the parts to design new radios that I put together. My mother never got in my way and allowed me a place behind our sofa to make a mess and to do my tinkering.

Your unusual college story should inspire some *Communications* readers.

My father fell ill and could not continue to run his grocery store, so I realized I could not attend college in day session and had no choice but to go to night school and bring home a salary by working full time during the day. That was a big blow. My father took me to an electronics firm where I could get a job serving as an electronics technician and eventually as an assistant electrical engineer doing

industrial electronics. So instead of attending CCNY (City College of New York) as a daytime student, I attended at night. My day work was, however, wonderfully interesting: we were involved in designing and using photoelectric devices in many applications.

The people in night school were an interesting bunch—after all, who attends night school: crazies, dropouts, motivated students who had to work during the day, and GIs coming back from World War II (this was 1951) who were disciplined and very determined. The professors at night school worked in industry during the day so they had insight into practical matters. I remember a professor bringing a germanium transistor he worked on during the day to class saying “this is a better thermometer than an amplifier,” and began to discuss ways to eliminate the temperature-dependent variations. This combination of combining practical issues with mathematical approaches has always supported my seeking to find intuition and insight behind theory. Claude Shannon, who was then—and still is—my role model, similarly had great insight and physical intuition into why things happened alongside his mathematical approach to problems.

You probably were thinking of getting a job after CCNY. How did you go to MIT instead?

I learned one day that an MIT professor was coming to CCNY at 4 P.M. to describe a terrific fellowship that would provide considerable financial support to pursue a master’s at MIT as an MIT Lincoln Labs [a well-known R&D laboratory associated with MIT—*Ed.*] staff associate. I managed to get off work early that day, but when I asked the MIT professor for an application to the program, he told me they were available from a CCNY professor sitting at the back. The CCNY professor did not recognize me and when I told him I went to night school he said “get out of here.” So I had to contact MIT directly to get a form. That I did and I was fortunate to be awarded the fellowship!

What was it like doing a master’s at MIT as a Lincoln Labs associate?

My first supervisor at Lincoln was

Ken Olsen, who later went on to found Digital Equipment and build the line of PDP computers. I worked in a group at Lincoln run by Wes Clark who built arguably one of the first PCs (the Linc computer). So there were a lot of brilliant people at Lincoln; and of course MIT professors would often visit.

What did you do your MS thesis on?

When I first got to MIT, I was interested in servomechanism systems and automatic control. Yet, my master’s thesis at MIT was on optical readout of thin magnetic films for storage and processing. I made use of the Kerr magneto-optic effect whereby polarized light rotates differently when it reflects off a magnetized surface depending on the direction of magnetization. As a result, one could use polarized light to non-destructively “read” the bits on thin magnetic films (this was before disks). My job was to improve the reading process by amplification and coding. The thesis involved experiments and models and I even constructed a special digital logic using light bouncing off a sequence of thin films. My thesis must have impressed my MS supervisor—Frank Reintjes—at MIT because he insisted that I apply for a Ph.D.

But the idea was that after a Lincoln Labs fellowship you should work at Lincoln Labs as an engineer, right? And you had a first child coming by then?

That’s right. Our first child was

Being surrounded by computers at MIT and at Lincoln Lab, it seemed inevitable to me they would eventually need to communicate with each other.

planned to come in the final summer just before I finished my MS, at which point I would be working full time at MIT Lincoln Lab and we needed the money since my wife would have to care for our newborn. So, I was not at all interested in pursuing a Ph.D. But Frank Reintjes was insistent and, amazingly, Lincoln Labs decided to offer me a follow-up Ph.D. fellowship to MIT just as they had done for my MS fellowship; this was a first for Lincoln. So I succumbed to the pressure and accepted the Ph.D. fellowship. Two others were also offered the Ph.D. fellowship: Larry Roberts [one of the founders of the Internet, see later—*Ed.*] and Ivan Sutherland [one of the founders of graphics and an ACM A.M. Turing Award recipient—*Ed.*] who both became lifelong friends.

What were the first years of your MIT Ph.D. experience like?

Our Ph.D. qualifier was legendary for its difficulty with 50% of the applicants failing out like flies. My MS at MIT made it easier since the qualifying exam was largely based on the MIT MS curriculum, but full of trick questions. Interestingly Ivan (Sutherland) came in directly from Caltech (that is, without the benefit of exposure to the MIT MS material directly) and came out on top with one month to study; he is one heck of a smart guy. When I agreed to continue on with a Ph.D. program, I decided I wanted to work with the best professor I knew at MIT, and so called up Claude Shannon (founder of information theory). He surprised me (and shocked my friends) by inviting me to his house in Winchester, MA, USA. I remember the scene looking out on Mystic Lake as an automatic lawn mower (rigged up by Shannon) mowed the grass and his son’s swinging hammock narrowly missed my head. Shannon wanted me to work on a strategy for the middle game in chess as part of a project that he and [AI Founder and Turing Award winner—*Ed.*] John McCarthy were working on.

How did you gravitate to what is considered your seminal thesis on packet communication?

I was looking for a fresh field to work on. It seemed to me that even Shannon

had stopped working on information theory and coding theory, but most of my EE graduate student classmates wanted to work on it (with other professors). It seemed to me that the problems that remained seemed harder and less impactful. And chess was not my forte. At the same time, being surrounded by computers at MIT and at Lincoln Lab, it seemed inevitable to me they would eventually need to communicate with each other and the existing telephone network was woefully inadequate for such a challenge. Shannon agreed to be on my committee and my advisor was Ed Arthurs (by the way one other student of Arthurs was Irwin Jacobs of Qualcomm fame). Arthurs mentioned a classified project he had encountered for a network between computers. Here was an unmined area, an important area, one whose solution would have impact and one for which I had an approach—this was for me! I recognized that data communications generated highly bursty traffic and that the existing telephone network, which used the static assignment technology of (slow) circuit switching, was not up to the job.

I saw that what was needed was to assign (communication) resources in a highly dynamic, demand-based fashion, that is, dynamic resource sharing, wherein a resource is only allocated to a demand request when that demand needs it, and to then to release that resource when the demand no longer needs it. This concept is manifest today in so many systems (for example, Uber, AirBnB, seats on an airplane, and so forth) in what we often refer to as a shared economy.

My thesis proposal was entitled “Information Flow in Large Communication Nets.” I was motivated by Shannon’s teachings that large systems were especially interesting since, as systems scale up, emergent properties manifest themselves.

MIT was (and still is) a place of great intellectual ferment in those days. Tell us any memories you have of those days.

I remember the amazing collection of classmates that shared office space with me. For example, Jacob Ziv (information theory pioneer and inventor of Lempel-Ziv coding) and Tom Kailath

I had loved the few courses I taught while at MIT.

(control theory pioneer) were part of the same suite with me. This lab housed Shannon’s students along with others. I remember the stimulating conversations in which we engaged and taught each other our very different fields (information theory, control theory, networking, and so forth) and spurred each other on.

Queueing theory is widely used today but you may have been the first to apply and develop this tool for computer networks. How did that happen?

Queueing theory had been invented by the telephone engineers (starting with A.K. Erlang) in the early 1900s, then taken up by the mathematicians, but after the war the Operations Research folks began to apply it to industrial problems (for example, Jackson applied it to job-shop scheduling); but it never was a mainstream tool. Yet queueing systems models had all the ingredients of a mathematical approach to performance evaluation of networks that I needed since it dealt with ways of analyzing throughput, response time, buffer size, efficiency, and so forth. Further, and importantly, it was a perfect mechanism for implementing dynamic resource sharing.

The queueing books that were available in those days were very theoretical. I tried to remedy that later by writing a two-volume textbook called *Queueing Systems*, which was queueing theory for engineers and contained the first description of the ARPANET technology and its mathematical theory that was published in a book.

Yes, your book led me to research on networking as an undergraduate in IIT Bombay. I am sure it’s influenced many others because of its clarity and strong intuition. Let’s

move on to your thesis on the mathematical theory of “stochastic networks.” What does that mean?

My thesis dealt with computers exchanging messages—whose size and inter-interval times were governed by a probability distribution—across a network of what we would call routers today. Networks of steady deterministic traffic flows had been studied (for example, Max-flow Min-cut theorems had just appeared) and one node systems with stochastic arrivals had been well studied (queueing theory). However, very little had been done on the combination of those two, and this led to stochastic networks. This was a very hard problem to solve analytically, and to this day, the exact analysis is still intractable. However, I was able to crack the problem by making an assumption that the stream of traffic entering each router queue was a stream of independent traffic (the “independence assumption”); I was able to show via simulation that network behavior was accurately predicted with this assumption.

You were inducted into the Internet Hall of Fame in the first year when it opened, along with (Vinton) Cerf, (Robert) Kahn, and others. Your nomination says Leonard Kleinrock pioneered “the mathematical theory of packet networks, the technology underpinning the Internet.” So how did stochastic networks morph into the Internet?

Once I recognized that the key issue was how to support bursty traffic in a data network, it became clear that the mathematics needed to represent the network had to be based on stochastic networks; this meant that I needed to extend queueing theory to the environment of networks, hence stochastic networks. There were at least two other independent threads that were nearly concurrent. Paul Baran at RAND corporation was tackling the problem of how to design a network for the military that was resilient to attack and so he hit on the notion of breaking messages into packets and dynamically routing them around failures and had simulations to show the effectiveness of this routing. He also proposed distributed network topologies that provided protection against

partial network damage. His work was classified and so I did not see his papers until my thesis was completed. His work was right on! Another important thread was from Donald Davies who was working independently at the National Physical Laboratory in England and who realized that the packet switching was good because, as we had articulated, data was bursty. He coined the word “packet” and pointed out that long packets were more likely to contain an error than were small packets; hence he suggested packets of approximately 128 bytes, which was later used in the ARPANET design. He promoted packetization as having the desirable advantage of allowing small messages to swoop past large messages; interestingly, I had shown the exact form of this trade-off mathematically in my dissertation years earlier. Moreover, he recognized that once messages were packetized, then retransmission of packets rather than whole messages would reduce delays in overall transmission. Further, he noted that the ability to pipeline packets reduced latency through the network.

So in some ways, Paul, Donald, and you explored different facets of the benefits of packet switching. Paul focused on routing resiliency, Donald on packet-level error resiliency, and you on mathematical performance evaluation and optimization of packet-switching networks using stochastic models. Is that accurate?

That is a fair characterization. I would add that Paul and Donald were looking mainly at critical architectural issues whereas I was more focused on extracting the underlying principles and developing a mathematical theory of packet networks. Among the principles were: dynamic resource sharing is key in an environment of bursty demands; large shared resources supporting lots of traffic are far more efficient than small resources supporting less traffic; and distributed adaptive control is efficient, stable, robust, fault-tolerant, and it works.

Your thesis also anticipates and analyzes other benefits of packet switching we rely on today that are complementary to those pointed out by

While we think of the Internet today to send email and support social networks, the motivation then was to share the expensive computers ARPA was funding.

Davies. These include techniques such as priority queueing (for example VoIP is queued before data packets in today's routers) and splitting packets for the same destination across multiple paths (called ECMP). Is there anything else you want to mention before we move on from your thesis?

Some other aspects include the effects of scaling. I showed for the first time that in terms of performance, a single link of capacity C is better than N links each of capacity C/N (this was an example of the second principle I mentioned previously). I investigated how to optimally design network topologies, which contributed to the field of network flows that Howie Frank and Ivan Frisch and others made major contributions to. I also investigated distributed adaptive routing control but I modeled that by having each router precompute an ordered sequence of favorable routes for each destination and use the first route that was not congested locally.

Interesting! That's different from dynamic routing in today's Internet where routers use Dijkstra's algorithm to compute shortest paths. However, that technique takes longer to respond to failure. Some networks today (for example, MPLS protection) use your idea for faster recomputation of (possibly less optimal) routes after failure.

It is true that I did not suggest a Dijkstra-type updating procedure dynamically based on networkwide shortest paths. I introduced a simpler,

albeit less optimal, protocol where the dynamics of the network were reflected in which links of interest out of a node were idle, indicating that the route to the destination using a queued link was not desirable and that the uncongested links leading out of a node to that destination were currently better choices.

Your thesis defense was a remarkable event ...

Larry Roberts, Ivan Sutherland, and I were very close friends and did our thesis defenses at the same time because we had all heavily used the MIT Lincoln Lab TX2 computer as part of our Ph.D. research. The union of our three Ph.D. committees came out to Lincoln Lab to view our work including Claude Shannon, who was on each of our committees, Marvin Minsky, and Peter Elias. It was a big heyday and a bit stressful given the credentials of those committee members. The projects were very different; we were just all using the TX2. Ivan did this great work on Sketchpad, Larry did his on machine perception of three-dimensional objects, and I did mine on communication networks. The TX2 allowed me to run an enormous simulation to verify the accuracy of my mathematical approximations.

You submitted your thesis and clearly it was well liked at MIT since they suggested you publish it as a book. How did you end up at UCLA and not at Lincoln Labs?

Morally, after their fellowship I felt I should work for Lincoln. But they were remarkably generous and offered to have me look around the academic and industrial circuit to see what opportunities were there. I received some great offers of research positions: Bell Labs, Lincoln Labs, Hughes, and many more. And then there were academic offers, including the one from UCLA for a tenure-track position (at half the salary I would get at Lincoln). But I had loved the few courses I taught while at MIT and realized I could augment my salary by consulting. So with the West Coast weather, the Wild West appeal, and a university position, I drove my family all the way across

the country. Lincoln Labs was extremely gracious and even said that I could come back if I did not like it at UCLA—but it has been 56 years and I am still here!

Fast-forward to the birth of the Internet in a UCLA office. On October 29, 1969, you and Charley Kline, one of the Ph.D. students on your software team, transmitted the first message between computers hundreds of miles apart. What was the backstory?

In my software team, besides Charley there was Steve Crocker who headed the software group, Vint Cerf, and Jon Postel, all UCLA graduate students at that time and subsequently Internet luminaries. The backstory starts with Ivan Sutherland who became head of IPTO for ARPA in 1964. Ivan visited UCLA in 1965 and suggested we network the three nearly identical IBM 7090s on campus. But the three administrators didn't want to share their computers, so that network was never implemented.

How did it finally happen?

Bob Taylor (who later led Xerox PARC) took over IPTO after Ivan. Bob was convinced that IPTO needed a computer network to link the sites he was supporting so that they could share each other's computers and applications. Bob convinced Larry Roberts to come to Washington in 1966 and head up this idea of deploying a computer network. While we think of the Internet today to send email and support social networks, the motivation then was to share the expensive computers that ARPA was funding at sites like Utah (for graphics), Stanford Research Institute (databases, Doug Engelbart was there). Larry was familiar with my networking research and publicly credits my thesis for giving him confidence to spend millions of dollars of ARPA money on this crazy idea. Larry was also well aware of Baran's work and that of Davies (who had even built a single-node packet switch) and incorporated their ideas in the ARPANET design.

How did Larry get everyone together to create the ARPANET, the precursor to our Internet?

In 1967, Larry brought a bunch of us together to help him specify what this network would look like and what performance characteristics it would have. We specified the network and created the spec and then Larry put it out for bid. In December 1968, BBN was granted the contract.

In September 1969, BBN delivered the first IMP to you at UCLA. Why UCLA and not SRI or Utah?

My role in this ARPANET project was performance evaluation, design, experimentation, and measurement. At UCLA we had specified the measurement software BBN later implemented in each switch. It was natural that we would be the first node so that we could begin to conduct experimentation and make measurements of what was going on.

The first message on the Internet was "Lo" which seems to have Biblical connotations that go along with the Creation Story. Was this deliberate?

Not at all. We were trying to send the text "Login" to login to the SRI host but there was a bug and the software crashed after sending "Lo." Of course, the bug was in SRI's software, not ours nor in the network itself!

How did we get from the first ARPANET to the Internet we know today?

The first host-to-host protocol was called NCP but soon it became clear that the ARPANET would shortly be

One other aspect of today's Internet we did not foresee was the emergence of the dark side (in all its manifestations) that plagues us today.

just one of many networks in an evolving "internetwork of networks" where every network would have a network number. The need for a more advanced internetworking protocol became clear and this was Cerf and Kahn's great achievement of TCP/IP for which they justly were given the ACM A.M. Turing Award.

The rest of the story, the commissioning of the NSF backbone, the decision to transition to multiple commercial backbones who had to cooperate, and so forth, are all well known. We had no clear idea of how the Internet would be used, but we caught our first glimpse when Ray Tomlinson introduced email in 1972 and it very quickly took over a major portion of the traffic; that was when it became clear that a major use would be to support people-to-people communication. Put another way, we completely missed social networking as a major use of the Internet. Indeed, it has been the case over and over again that the Internet community has been surprised each time major new applications have exploded in use (for example, the World Wide Web, peer-to-peer file sharing, blogs, user-generated content, search engines, shopping engines, social networks). What we are good at predicting is the underlying infrastructure of the Internet (networking technology, IoT, wireless access, mobility, and so forth). One other aspect of today's Internet we did not foresee was the emergence of the dark side (in all its manifestations) that plagues us today.

While the Internet was gaining steam, you trained several generations of remarkable students whose Ph.D. theses and papers with you greatly influenced the Internet and analyses of time-shared systems. Tell us more ...

There is so much to tell, so let me provide a small sample only. My first student was Ed Coffman, who worked on some extensions to priority queueing and time sharing. Most of my students who followed concentrated on various performance analyses of aspects of the Internet as it emerged. For example, the early ARPANET did synchronous (periodic) routing updates but Gary Fultz's thesis analyzed the benefits of

asynchronous updates, something we take for granted today. Mario Gerla provided optimal routing design and provided an effective protocol. Parviz Kermani's thesis introduced the idea of cut-through routing, that is, starting to forward a packet as soon as the router read the destination address, thereby reducing latency, which is pervasive today in Local Area Networks. Farouk Kamoun's thesis introduced and showed the enormous benefits of hierarchical routing, which we see in OSPF areas today. Simon Lam and Fouad Tobagi initiated the analysis and design of wireless networking and provided the early analysis of Slotted Aloha (Lam) and CSMA (Tobagi). And so on.

And the triumvirate: Gerla, Tobagi, Lam—all full professors at UCLA, Stanford, and Texas (Austin) respectively—and winners of major networking lifetime awards.

Mario and I worked on network design techniques that went beyond my thesis while Howie Frank and Ivan Frisch were concurrently working on different techniques at Network Analysis corporation. Mario went to work for Howie Frank but we later hired him back at UCLA. Simon's thesis came out of the satellite packet switching meetings we had where he was my right-hand man. Out of that came his dissertation on the analysis of the instability of Aloha that Abramson had created in Hawaii. Then ARPA started moving to packet radio on a metropolitan area networking basis. The application was to foot soldiers or possibly tanks moving across a battlefield; that led to the SURAN survival radio network, and the whole packet-radio project. Tobagi and I started studying CSMA—carrier sense multiple access—which eventually contributed to Ethernet.

Their theses led to a cottage industry in both network design (popular in the 1970s) and to media access schemes (which continue to be popular today because of 802.11 and WiFi).

That early work caught the attention of researchers and industry as we continued studying the behavior of networking at large. We were fortunate to be working on these problems at an early stage.

I fear the power of the Internet is being lost.

Let's wind up by asking you about the future. In the old days, they would say "Young man, go West." You did that literally (Los Angeles) and metaphorically (the Internet was the new frontier, the Wild West of communication). Is networking now merely boring infrastructure like plumbing. What advice do you have for young researchers?

I think there is an enormous amount of exciting work to be done in networking and distributed systems in general. For example, areas that are in need of innovation, research, and development include IoT, distributed ledgers, the introduction of biologically inspired principles to networking (and engineering in general), distributed intelligent systems, advanced network architectures, network security, and much more. The space is awash with great problems to dive into. In the case of distributed ledgers, the technology that underlies bitcoin, I am excited on the one hand, but concerned on the other hand. What concerns me is that billions of dollars were poured into blockchain technologies soon after its birth, thus distorting its path to proper maturity; this is because profit-seeking companies and speculators jumped on the bandwagon right away, which may lead to brittle designs. By contrast, we had 20 years without commercial interruptions in designing the Internet and those years of careful curation helped, I believe, to make the Internet more robust.

What about future applications?

As I said before, it is hard to predict. We missed the social networking revolution completely. So, in some sense, we have created an Internet that is destined to continually surprise us with new, exciting, and explosive applications; that is a good thing

because it represents creative opportunities for future generations of researchers. I think it would be interesting to study retroactively why certain applications (for example, Twitter, Facebook) grew so popular while others failed so that we can try to be better at prediction. Perhaps a lens based on behavioral economics of the sort pioneered by (Nobel Prize winning economist) Daniel Kahneman and his colleague, Amos Tversky may help.

What concerns you about the Internet?

I am concerned about the Balkanization of the Internet as nation-states cut off and censor Internet traffic and corporations create closed enclaves. I fear the power of the Internet is being lost. I realize this is partly because of security concerns but am confident about advanced technology developments that can ameliorate these concerns.

You have had a remarkable career of academic excellence (network performance evaluation clearly begun with your thesis and your later work with your UCLA students), real-world impact (you were heavily involved in the evolution of the Internet including helping write the famous Gore proposal), and entrepreneurship (you have started several companies including Linkabit (with Jacobs and Viterbi, which later led to Qualcomm) and Nomadix (an early mobile wireless company). What advice do you have for ACM members?

First, think deeply about the results of your work. It is not enough to evaluate your ideas. You need to keep thinking about them to distill principles before moving on to the next big thing. Second, try to bounce ideas among brilliant people. I have had the fortune of doing so with folks like Shannon, Sutherland, and colleagues like Gerald Estrin at UCLA. Third, aspire like Shannon to combine physical intuition with mathematical analysis. While I have done a fair amount of mathematical work, I am an engineer at heart. My early building of a crystal radio remains a watershed event in my life. ■

George Varghese (varghese@cs.ucla.edu) is Chancellor's Professor, Computer Science, at UCLA, Los Angeles, CA, USA.

Copyright held by author.