

Len Kleinrock: The Theory of Packets

Charles Severance

Len Kleinrock describes his path to packet networks.

In 1958, Len Kleinrock was wrapping up his MS in electrical engineering at MIT and preparing to start at Lincoln Labs when one of his professors, impressed with his work, insisted that he continue his education and pursue a PhD. From there, the rest is history. I recently spoke with Kleinrock about his life and legacy; you can view our conversation at www.computer.org/computingconversations.

A New Direction

Kleinrock decided that if he was going to invest the time, he would only work on an important problem, one whose solution would make a difference:

I decided I would work for the best professor I knew at MIT and that was Claude Shannon—the brilliant, wonderful, magnificent Claude Shannon. Working for the man was a delight. He was a great engineer, a great mathematician, and smart as heck.

I looked around and observed that most of my classmates were working on problems involved with information theory, the field that Shannon had created. It seemed to me that these problems were “left over” by Shannon, and so were probably

hard and not of great significance. That wasn’t what I signed up for—I wanted to work on a problem that would be fun, exciting, and challenging, with a real impact.

Researchers at MIT and Lincoln Labs were building computers that would ultimately need to talk to each other. Specifically, these computers would need to have interactions with many different computers that were short and bursty, followed by relatively long periods of no communication at all:

I knew that computers, when they talk, go “blast” and then they’re quiet for a while—a little while later, they suddenly come up and blast again. You can’t afford to dedicate a communications connection for something that almost never talks, won’t warn you when it wants to talk, but that when it does talk, it wants immediate access. The circuit-switched telephone network, which was designed for continuous talking, was totally inadequate.

Kleinrock took his inspiration from the techniques commonly used in the multiuser timesharing operating systems of the time. If multiple jobs needed a large amount

of processor time, you gave each job a short time slice, and when that time was up, you gave the processor to another job for a time slice. In this way, all jobs made some progress toward completion and evenly shared the delay due to oversubscription:

I thought that time-slicing was a great idea for sharing communications. We give everybody a little bit of communications time—the little ones will filter through, and the long ones will take a little longer, and they won’t mind being interrupted by the little guys but not conversely. The important thing in this technology is to protect the very short messages from waiting behind very long ones. This automatic round robin for data communications is now called “packet switching”. You chop messages into fixed lengths and give them a small fixed amount of time on the wire; if that isn’t enough, you give them another little bit, as each small piece goes flying through the network on its own.

Packets Are the Answer

Kleinrock realized that when messages are broken into packets and those packets are sent through the network using a round-robin

scheduling approach, you could use queuing theory to build a solid model of the overall throughput and average message delay. It seemed simple enough, but as Kleinrock got into the details, it turned out to be challenging:

I set up this mathematical model and found it was analytically intractable. I had two choices: give up and find another problem to work on, or make an assumption that would allow me to move forward. So I introduced a mathematical assumption that cracked the problem wide open.

This “independence assumption” says that when a message travels through the network, it changes its length independently every time it hits a new node as it hops through the network. Mathematically, this creates a statistical independence that allows you to proceed with the analysis, but it is clearly not a true assumption. From that point, I could just sail through the solution, derive the performance behavior, optimize the design, and uncover the underlying principles.

The real question, of course, was whether the simplifying assumption skewed the theoretical results so they wouldn’t be useful as an approximation of real-world network behavior. Kleinrock addressed this by building simulation software that would test the independence assumption:

I had to write a program to simulate these networks with and without the assumption. I simulated many networks on the TX-2 computer at Lincoln Laboratories. I spent four months writing the simulation program. It was a 2,500-line assembly language program, and I wrote it all before debugging a single line of it. I knew if I didn’t get that simulation right, I wouldn’t get my dissertation.

I ran it, tested it with and without the assumption, and the results

were amazingly close. So I had my solution, I could prove the accuracy of my mathematical theory, show that the packets wouldn’t fall on the floor, and I could tell when things would work well and when they would not.

With the mathematical model and the independence assumption validated, Kleinrock finished

I could prove the accuracy of my mathematical model, [and] show that packets wouldn’t fall on the floor.

his PhD thesis in 1962; McGraw-Hill published it in 1964. The book was a clear road map as to how to build the scalable shared wired and wireless networks we take for granted today. Its model was solid and its conclusions clear; all that was needed was to get to work and build a network:

But nobody cared. I went to AT&T, the biggest network of the time, and explained, “You guys ought to give us good data communications.” The answer was, “What are you talking about? The US is a copper mine; it’s full of telephone wires...just use that.” I said, “No, you don’t understand. It takes you 35 seconds to set up a call, you charge me a minimum of three minutes, and I want to send 100 milliseconds of data.” And the answer was, “Little boy, go away.”

So little boy went away and with others developed this technology that ate AT&T’s lunch. They said it wouldn’t work and even if it did, they wanted nothing to do with it. That was the environment we faced. It wasn’t until years later when the government decided that it needed a network that suddenly I saw a way in

which I could implement the technology I had developed.

Finished with his PhD, Kleinrock prepared to begin his career as a Lincoln Labs researcher; it had supported him financially as he earned his MS and PhD:

When I prepared to go to work there, the first thing they said was, “Look, Len, why don’t you look outside before you commit to work here so that you make sure there’s nothing out there that you really would like better.” This was truly a magnificent step on their part. So I took a trip to the West Coast and interviewed with some of the aerospace companies; I wasn’t interested in a university position at all. But it turns out when I was going up to San Francisco to look at some of the high-tech companies up there, a friend suggested that I interview at Berkeley. So I did, but they changed chairmen and lost my paperwork; I never heard from them.

Kleinrock was finishing up at MIT when one of the professors at Berkeley who had met with him during the interview process came to MIT on sabbatical and they ran into each other:

He saw me in the hall and said, “Kleinrock! How are you?” He thought I was looking for an academic position, so he contacted one of his friends at UCLA, who then invited me out, offered me a job, and presented me with a dilemma: Do I want to teach? Do I want to cross the US and be 3,000 miles from the East Coast, where the world is, for a job paying half of what I could earn at MIT/Lincoln Labs to try something that I had never tried before?


So I went to Lincoln Labs and said, “Look, I have this offer. It seems attractive. It’s a new challenge. What should I do?” And they gave a wonderful and magnanimous answer: “Len,



try it, and if you don't like it, come back." So I started at UCLA 50 years ago in August 1963, and I'm still here.

In his 50 years at UCLA, Kleinrock has accomplished enough to fill several books. As the Arpanet project got off the ground, he founded the Network Measurement Center and along with UCLA graduate students Vint Cerf, Steve Crocker, Jon Postel, and many others, was an essential part of the early Arpanet development. The first two packets ever sent

across the Arpanet were sent from Kleinrock's lab at UCLA to a system at Stanford Research Institute. The network crashed after the second packet was sent.

Beneath millions of network links, billions of computers, and trillions of packets flying around the world at any given moment, a solid mathematical model developed back in the early 1960s proves that all those moving parts actually can work reliably. 

Charles Severance, Computing Conversations column editor and Computer's multimedia editor, is a clinical associate professor and teaches in the School of Information at the University of Michigan. Follow him on Twitter @drchuck or contact him at csev@umich.edu.



Selected CS articles and columns are available for free at <http://ComputingNow.computer.org>.