

# A TEI Project

## Interview of Charles Kline

### Contents

#### 1. Transcript

##### 1.1. Session One (February 16, 2014)

### 1. Transcript

#### 1.1. Session One (February 16, 2014)

FIDLER

This is an interview with Charles Steven Kline. I'm Brad Fidler, and it is February 16, 2014. Why don't we start by you telling me about your early childhood.

KLINE

Okay. Well, I was born in New York, and for the first several years grew up in New Jersey just outside of New York. My dad was an electronics engineer and worked for an electronics company. It doesn't exist anymore, but it was actually a fairly well-known electronics company of its day, Allen DuMont Laboratories. Every once in a while, you'll see something on TV and they'll talk about the old DuMont Network. They had a nationwide television network that competed with NBC back in the forties and fifties, and my dad designed oscilloscopes and stuff like that. Anyhow, in the mid-fifties, they decided to move him out to California to manage the operation in Los Angeles. Of course, since all the relatives were back east, my mom didn't really want to move, but they moved. So we moved to L.A., and I grew up in L.A., and other than first grade, everything else I went in Los Angeles, so I think of myself as a native Angeleno even if I wasn't born there.

FIDLER

When did you figure out that you wanted to be an engineer or do something with computers?

KLINE

Well, I'd always been good in math. I always liked playing with things. My dad and I would build transistor radios and things like that. There was a company that still exists, actually, Heathkit, which used to sell little kits for everything from oscilloscopes and volt meters to little kits for learning stuff, radios, stereos, TV sets later on, and so on. So we built some of those, and I liked playing with electronics, and my dad was in electronics, so it was sort of something I enjoyed doing and I got to play with, and if had any questions, he could always answer them.

FIDLER

Was this during high school when a lot of this interest was even further developing?

KLINE

This started in elementary school and junior high school. So my dad kept going back to school. He'd gotten a bachelor's degree, then he'd gotten a master's, then when he was out here, he got another master's. He went to a program they had called the Engineering Executive Program, which was to train executives in engineering management. Later he taught the Engineering Executive Program and he got a master's of engineering, and then later decided to go back and get his Ph.D. in systems engineering. Anyhow, so as part of doing that, you had to have—you still have to—you have to have a major field and a couple of minor fields, and one of his minor fields was, quote, "computers." This was just before they had departments at UCLA. There was engineering and there were sort of disciplines, but there weren't formal departments. There wasn't a computer science, but there was a field called computers. So he had learned how to program. In junior high school, I was taking algebra, and I had all this homework to do which involved, among other things, factoring equations. Now, if you've had algebra and they give you an equation,  $x^2 + 3x + 2$ , that's  $x + 2$  [unclear]  $x + 2$  and  $x + 1$ . The problem is coming up with whatever the factors are that are going to add to that middle number, and they didn't give you things like  $x^2 + 3x + 2$ . They gave you things with these weird exponents, and you're trying to figure out what are the factors of these exponents, so maybe you can try to play around and figure out which one are going to add to that middle term. So I said to my dad, "Do we have a table of prime numbers?" In those days, you just didn't go down to your local bookstore and buy a cheap table of prime numbers. By the time I was in college, you did, but when I was in junior high school, you didn't. He said, "No. They have them at the library." And he said, "But why don't we ride up, we'll go over to UCLA and I'll show you how to write a program to generate one." So we wrote a program in FORTRAN and I learned a little FORTRAN to generate a table of prime numbers. So I generated a table of prime numbers from like one to a hundred or one and couple hundred on what was almost a personal computer of its day, about the size of my piano, actually bigger than that, the IBM 1620, which UCLA had recently gotten. That wasn't in Boelter Hall; that was in the building that was called Engineering 1, which isn't there anymore on the fourth floor. Programming's kind of neat, it makes sense, it seems fairly intuitive, fairly obvious. You tell the computer to do this, and it does that. Of course, I really didn't know how things really worked down at the hardware level yet. I didn't really understand it at the machine-language-level yet. I just knew FORTRAN, although over time, I got some books on things that related. I found a book called The 1620: The Hands-On Approach, which talked about the 1620, its actual instructions, and I could begin to understand how the compiler would convert what I wrote in FORTRAN into the instructions the machine was going to execute. Anyhow, okay, so I go through high school in West L.A., and I'm trying to decide where to go to college, and I pretty much decided I was going to go into a science and I pretty much decided to go into engineering. Now, things were a little different than they are now. In the late fifties and sixties, the Master Plan for the University of California was that something like 25 or 30 percent of the people could go to a University of California campus and you would have your priority at your local campus. The idea was to encourage people to have campuses scattered around and you could go to your local campus. So to get into UCLA, which was my local campus because we lived close to UCLA, I needed a 3.25 GPA, I needed—there was a list of you had to have so many years of English, math, history, this, that, whatever, and things, and that was pretty much it. And if I met these requirements, I was guaranteed

acceptance. They didn't have the situation like they do now where they get fifty thousand applicants for five thousand slots. So I didn't even apply anywhere else, because acceptance was guaranteed. A little different today. [laughs]

FIDLER

Very different now.

KLINE

Very different now. My wife has given lectures to high school students about the process of getting into UCLA. They have College Fairs, and she's done, as a volunteer for UCLA, explaining why you might want to go to UCLA, but how they do their analysis of people and how they rate the GPAs, and they have multiple GPAs they use now. When I did it, they didn't even have the five-point A's. So I started in '65 at UCLA as a freshman, engineering, and at least for the first couple of years of engineering, you pretty much didn't have any electives. You had chemistry and physics and math and this and that, and you had very few choices, and there were certain engineering courses that were required because they didn't have departments yet. So you were an engineering student, and you would, later on, as you got to your junior and senior year, you might take courses that were more focused in a particular discipline of engineering, but those first couple of years were you had a class in nuclear engineering, you had a class in just general lab engineering, and so on.

FIDLER

This is when it was a Department of Engineering, not the school.

KLINE

It was College of Engineering.

FIDLER

College of Engineering.

KLINE

So I think it was my—I'm trying to remember whether it was my freshman year or my sophomore year. I think it may have been my sophomore year, I had to take a FORTRAN programming class, and I looked at the book that I was going to have to get. I said, "I know this stuff. I wonder if I can test out of this rather than taking it and go on to the next class." So I said to my dad, "Who should I talk to?" He says, "Well, talk to Jerry Estrin," who may have been the head of the computer group at that point or not, I don't know. So I made an appointment to see Jerry Estrin, and I went in and saw him and said, "I'd like to test out of this class or maybe be a reader for it." Pretty much T.A.'s were graduate students, so I knew I wasn't going to be a T.A., but readers sometimes read people's homework assignments and whatever. So he says to me, "Well, do you know programming?" And I said, "Yeah." He said, "Well, write a program on the board." I don't remember what he asked me to write on the blackboard, but he had me write something on the blackboard, and so I did. He said, "Yeah, you don't need to take this class. I'll sign a waiver so you don't have to take it." And he said, "Would you like to work on my research project?" I said, "Sure." So I was going to be hired as a lab helper.

FIDLER

Did this interface at all with the kinds of questions you were becoming interested in as an engineering student?

KLINE

Well, I was leaning towards I was interested in computers, but I didn't specifically have questions yet as to what I was going to do with computers. So I got introduced to a bunch of people that were working with Jerry Estrin either on his project or just students that were working with him. Some of his students had built a device—he had a hardware lab where they could actually lay out circuit boards and build things, and they had built a device that interfaced to the IBM 7094 that was in the next room up from 3420, the next room north, 30—I don't remember the number. [laughs] And that had an IBM 7094 in it, and they had built an interface to that, and Jerry was going to try to use that to do some measurements of the operating systems of the 7094 and how programs ran on the 7094. One of the guys had used that interface and built an interface to a Digital Equipment Corporation DEC 340 display scope that we had gotten, which was a thing the size of a refrigerator. You've probably seen a picture of it in—I can show you a picture of it in some of the pictures of the SIGMA 7 room. There's a big display scope.

FIDLER

And wasn't Gary Fultz working on that for his dissertation later on?

KLINE

Maybe later on, but earlier it was Mike Wingfield, I think, worked on that. I'm trying to remember the name of the guy who—I can't remember the name. But anyhow, so they had actually gotten it sort of they could write a program on the 7094 that would draw pictures on this display scope. Years later, we were able to connect that display scope to the SIGMA 7, but that's a whole—so to begin with, I was doing things like, "Well, these guys need some programs run on the 7090, so would you schedule time on the 7094 and go in and then run them on the 7094 and get them their results," that kind of thing. I was actually assigned to work for Steve Crocker. I got introduced to Steve Crocker, and Steve was effectively my sort of supervisor or manager, and we were going to, among other things, instrument FORTRAN programs. So we were going to write a program that analyzed a FORTRAN program, modified it so that it would add to the program additional instructions to keep counters of which paths in the program were executed and how many times, and then when that was done, it would take those results and would generate an output that showed here's the paths that were run. Okay. That turned out to be a little harder than I expected, because it was almost like building a FORTRAN compiler to analyze the syntax of FORTRAN. Well, one of the guys was building what later became called metacompilers. Metacompilers were compiler languages. In theory, you could use this language to write a compiler, and one of the guys there was building a metacompiler called META 5, and so I was trying to use META 5 to write my program that would analyze the FORTRAN program and put in the instrumentation stuff. Steve Crocker figured this would take me a few months. A couple years later, I was still working on it. [laughs]

FIDLER

This was a measurement team, I guess, that was being established.

KLINE

Jerry Estrin's work, a lot of his work was—it was general computer science, but he had a number of people that were doing research into measurement, and his work, if I remember correctly, was funded by the Office of Naval Research.

FIDLER

And this initial work in this lab was from around 1966, was it?

KLINE

Yeah.

FIDLER

And you were at that time doing a combined bachelor's and master's.

KLINE

Not yet. I was just an undergraduate student. But the minimum wage at that point was \$1.35 an hour. That was the U.S. or at least the California minimum wage. I think that was the U.S. minimum wage. And a lot of students had jobs on campus and various places at \$1.35 an hour, but I was getting paid \$1.80 an hour, so that was a pretty good wage for—they only would let me work half-time at most. They said, "We don't want this to interfere with your schoolwork." There was a history, I guess, of students getting so involved in their computer and other work that they never got their degrees done. So years later, when I wanted to work more than half-time and they were trying to decide whether this was actually a university rule, they discovered it wasn't. It was just sort of a policy. So we talked to the office upstairs in engineering that was sort of in charge of that stuff, but I think it was the assistant dean I was talking to, and he said, "Well, what's your GPA?" I said, "4.0." He said, "Okay. If you have a 4.0 GPA, we'll say you can work more than half-time." And they sort of created a new rule on the fly. [laughs]

FIDLER

And at that time, were you learning on your own through guidance with co-workers?

KLINE

I was learning on my own through—I wasn't taking very many computer programming classes yet. That didn't happen much for another year or so. But I was reading computer books and I had stuff to do, so I was trying to learn how to do it.

FIDLER

How close was your contact with Steve Crocker?

KLINE

It was quite close, and then he went to MIT for six months or a year, and in order to keep things going, he invited Vint Cerf, who was a close friend of his from high school days, to come back and work on his degree and sort of take over managing this project, so then I sort of got assigned to Vint Cerf. Okay. So, approximately '67, the SIGMA 7 was coming, and I loved playing with computers, so when I found there either was a group of people going down to see the SIGMA 7 on the test floor in Santa Monica, I said, "Can I come?" They said, "Sure." So I went with them and saw this SIGMA 7, which was this neat computer, as far as I was concerned. By that point, the Computer Center in the math science building had gotten a 360 Model 40, first model of the 360 series, and a lot of people were playing with the 360/40. They were later to get bigger 360s. So when I looked at the programming manual for the 360 series and the programming manual for the SIGMA 7 series, I discovered it was almost like that he had cloned the 360 series for the SIGMA 7. There were differences, but one had a 32-bit load instruction, one had a 32-bit load instruction, one had a 16-bit load, and so on. There were 8-bit machines, whereas, for example, the 7090s were 36-bit words rather than 32-bit words, and the 7090 series and a lot of other computers used 7-bit characters, where this used 8-bit characters. A lot of the computers in the world were using ASCII for the code set, but this computer, the IBM machine and the SIGMA 7, were using EBCDIC for the codes for the various characters. There were instructions on the 360. There was one called Translate,

which would take a string and take each character and look it up and use it to look into a table to replace it by another character. Well, SIGMA 7 had the same instruction. It was almost like they had almost copied the instruction set, but not exactly. There were a number of differences, and, among other things, the SIGMA 7 had an architecture that could be paged, whereas the 360 did not.

FIDLER

And when the SIGMA 7 arrived, did this coincide with a shift in your research focus, or were you still doing—

KLINE

Still doing stuff for Jerry Estrin. The research focus change didn't happen till sometime in '68-ish time frame.

FIDLER

Before we go to that, let's spend a bit more time on the measurement. Steve Crocker leaves, Vint Cerf comes in. Did you work with Vint Cerf much in the same way, and then were there other people that you were working with, maybe horizontally?

KLINE

There were a whole bunch of people horizontally. I won't say a whole bunch, but there were people working on hardware. There were people working on that metacompiler that I was trying to use, that I was constantly waiting for. But what happened by this point was we had this SIGMA 7, and since I liked to play with the computer a lot, I was sort of there a lot, so I sort of became the guy who knew the operating system, knew how to boot it. If there was a problem and we had to call SDS out to repair it, I worked with the engineer to repair it, and so I sort of became the go-to guy who knew how that machine worked.

FIDLER

And can you tell me about making the operating system for the SIGMA 7?

KLINE

So Steve Crocker wanted to build an operating system, and he started designing one and he decided that our group needed a name. At MIT they had Project MAC, and at Stanford they had the Standard AI Lab, and this and that. So somewhere he came up with the name Spade, and he called our group the Spade group, S-p-a-d-e, which later on, jokingly, he called the operating system we were building Some Poor Ass' Design. [laughter]

FIDLER

Did you see yourselves as analogous to Project MAC, for example?

KLINE

I don't think we were analogous to Project MAC, but Steve had just come back from going to MIT for a year and had seen the development of Multics there and the things they were doing there and said, "We need to build an operating system for this computer." And he was sort of influenced. That's the first time I heard the word "hack." Steve came back and he was telling me, and we would talk about this, "Oh, there's this really clever hack we can do." And I said, "What's a hack?" Well, a hack was a term from MIT which meant a clever piece of programming, and a good programmer who wrote clever hacks was called a hacker. Over the years, it's changed so that in the common vernacular it's normally thought of as a person who's trying to do bad things with computers, but to those in the know, it still has a meaning. In fact, at Facebook, they refer to themselves as hacker. Their address is 1 Hacker Way. [laughter] But MIT's

Project MAC was much more famous and much bigger and much more involved in a lot of different things.

FIDLER

Was this a time when your duties were expanding then? You started with the measurement group. Now you're becoming known as someone who knows his way around the SIGMA.

KLINE

Well, what happened was not only was I the guy who's becoming known as the guy who sort of knew how this particular computer worked, but we went up to Lawrence Livermore because Steve, I think, had heard that they had this operating system they were building, GORDO. And we went up there and visited, and they explained to us how it worked. There were some funny stories. Lawrence Livermore was the place where all the United States nuclear weapons were built, so they had all kinds of security. I remember I was in there and they had a lot of big CVC computers because those were the fastest number crunchers, and they had all these simulation things, and they had this printer that was just shooting paper incredibly fast. So I started to walk over to this printer, and these guards come running over, blocking, and saying, "No, no. You can't look at that. That's all classified. You can't look at that." I said, "I'm sorry. I didn't realize." [laughs] I mean, whenever we were in the area, they had these big signs that said "Caution: Uncleared Visitors in the Area." But they explained to us what they were doing, and we took back the stuff, and I sort of made it my—no one assigned me this task, but I sort of made it my task to see if I could get this software working on our SIGMA 7.

FIDLER

Was there a general sense that you wanted GORDO to be working and then you just took it upon yourself to—

KLINE

Right. And we had brought it back, and I think even before we had had any real meetings about, "How are we going to go about getting this to work? What are we going to do with this?" I said, "Well, we've got the source. Let me see if I can compile this and put it in a form where I can boot it up and see if I can get it to work." And they had sort of given us instructions, "If you compile this and do this and put it on this disk, it'll work." Although we had different disks than the disks they had, so I had to modify the driver for the disk drive. We actually used that operating system. There were some other people in the department who used that system for research. For example, there was a professor, Dick Muntz, who I think has retired, but he was chairman of the department for a long time. He probably hangs around there every once in a while, but he was into operating systems as part of his research focus. So we tried different—this was one of the first paged operating systems. By page, meaning that not all of the program had to be in memory at a time, as when you tried to access a part of your program that wasn't in memory, it would get a fault, the operating system would say, "Oh, the reason that faulted is because that page isn't there. I've got to go to the disk and get that, bring it in, put it in memory somewhere, change the hardware register so that that address is over there." Well, there are different algorithms as to—well, when you run out of space in memory, to bring in a new page, you've got to push something out to swap something out. Well, you could swap out the one you haven't used in the longest time, that hasn't been accessed, or you could keep the one that's been used the most, and get rid of the one that's been the least,

even if it's just been used. There are all these different things, and people had done queueing models of what might be the best page-replacement algorithm. So we could test some of that stuff on the SIGMA 7 by modifying the paging algorithms, and I think a couple of Dick's students did that.

FIDLER

So was there any formal scheduling that would happen? So, for example, there's this interest which you're describing. Then there was still, I imagine, the measurement work that was going on.

KLINE

There was some measurement work, but I was beginning to do the measurement work under the operating system. But I was spending most of time working on the operating system, keeping it running, making it so that multiple people could use it. I wrote a text editor. At the beginning, the only way you could do anything with it was still by cards, even though we had this time-shared system. If you wanted to write a program, you had to type it up on cards and then read in the cards, and then you could compile the program and try running it. I said, "First I want to write myself a program so I can edit on a terminal instead of editing on cards." Among other things, keypunches don't have a backspace button. You push a button and it punches holes in the cards. They do have a button that's, "Oh, I made a mistake," and it copies the entire card other than the column you just made the mistake in, but it wasn't very convenient. And not only that, if you've got this program and then you made some changes, "Oh, I want to make some changes," you'd just rather bring it up and do an editor, make the changes, and say, "Compile it." So I wrote myself a text editor to be able to do that, and then I could edit not only programs I was writing, but I could edit the operating system online, compile the operating system that way. Because I remember the operating system was several big boxes of cards originally. [laughs]

FIDLER

And you were punching your own cards until then?

KLINE

Oh, yeah.

FIDLER

Now, how did this research transition to the Network Measurement Center with Len Kleinrock?

KLINE

Okay. So sometime, I'm guessing it was around '68, mid-'68 or so, the ARPANET was beginning to happen. Bob Taylor decided he wanted to build the ARPANET. He had decided to hire Larry Roberts to manage the project. Larry had gotten in touch with Len. Len was really interested, said, "Look, you need to do this packet-based thing." There began to be meetings of the various principal investigators and also of some researchers, for example, Steve Crocker, who created a group called the Network Working Group and started the RFC series. So there were a bunch of people that were interested in this at UCLA. Besides Len and Steve Crocker, there was Jon Postel and Vint Cerf and myself, and we were all interested in it, but I was more worried about how do I keep the operating system going, and I heard this IMP was coming. But also Len had money and needed people, and so we all sort of got transitioned, or a lot of us got transitioned to Len's group. By this point, I don't think Jerry had that level of funding anymore.



FIDLER

When was the first time you heard about the ARPANET? You mentioned the IMP arriving, but there was—

KLINE

I heard about the ARPANET sometime in '68, and I heard that by September of '69 we were going to get this IMP. And probably late '68, early '69, by early '69, Mike Wingfield was beginning to think about how he was going to design an interface. There wasn't just off-the-shelf hardware. You couldn't just go get a cable and plug in the IMP to your computer. You had to build the hardware that connected your computer to the IMP. The BBN had built a spec, BBN Report 1822, I think it was, which has the hardware and technical specs of how the IMP was going to communicate to the computers. Some places went out and contracted some hardware company to build them an interface. At UCLA, I don't know whether Mike was assigned to it or whether he just took it on on his own, but he had already built hardware using the labs we had, the acid baths and things to etch circuit boards and so on. So he designed the circuit boards that he was going to need to plug into the SIGMA 7 to implement an interface, a very simple interface.

FIDLER

So did you work with him on the host IMP?

KLINE

Not on its design, no. But then when he built it, I wrote the software to interface it to the operating system. He wrote a standalone test program, a program that ran by itself without the operating system, just, you know, shut down our system, ran his little test program to test his interface, and then I took his test program and said, "Oh, so that's how it works." So I looked at his specification of how you programmed his interface and wrote a driver for the SIGMA 7's operating system. So, in theory, it was possible for an application to transfer data to and from it. So being the kind of guy that just liked to play with things, I was there the day the IMP arrived. My recollection is it was supposed to arrive September 1st. My recollection is that it arrived August 29th, and the BBN guys were happy that it arrived early because this was the first one—they had a hard deadline in their contract, September 1st. I'd have to look up what day of the week that was or whatever. And several of the guys from BB&N flew out to set up the hardware and get the software working. Now, you're used to computers, when you turn them on, there's a BIOS that has to boot up. That's in read-only memory that will then get executed and then read from the disk the operating system, okay, and that's because when the power's off, the memory in the computer doesn't retain any storage. But the IMP and the SIGMA 7 and most computers of that vintage used core memory, and core has the properties—little magnetic cores that they retain the state they're in. So literally when the IMP arrived and they unpacked it and checked it out and made sure everything looked okay and plugged it in and turned it on, they could push the start button, and it continued where they had left it, turned it off. It still had the software in its memory.

FIDLER

Had you heard at all about BBN's progress as they received the Honeywell machine, as they debugged?

KLINE

I didn't hear anything about that. I didn't know anything other than we were getting this thing called an IMP and that there was this report that explained how you used it and some discussions about what the IMP was and why we were doing this, but I wasn't following the progress. Len may have been, certainly the guys at ARPA were, but I wasn't following the progress prior to our getting the IMP. Then I was more interested in following their progress. Well, have you fixed this? When is that going to happen? One of their challenges was the first few IMPs had a high-speed paper tape reader, and they would send out the software on paper tapes, and we could put the tape in the tape reader, and it would read fairly fast. So that wasn't very convenient for them to be sending out paper tapes all over the country, and not every IMP had a high-speed paper tape reader. So one of their challenges when they got connected to the net was to make it so that they could reload IMPs remotely. So what they did is they came out and they changed the bootstrap ROM. They put a bootstrap ROM board in our IMP, in each of the IMPs, and that bootstrap ROM board had just enough software on it to try to go out to the network and ask a neighboring IMP, "Please send me a copy of your software." And all the IMP software was identical except for the machine number, which was also on that circuit board, so our circuit board was identical to everybody else's except ours said "machine one" on it, and somebody else's said "machine two" on it, and so on. So it would just say to its neighbor, "Please send me your software." It would load it up and keep going. Later on, they got more sophisticated about that, and they could actually force the systems to download a new version and so on, things we're all used to today for downloading software, but this was new stuff back then.

FIDLER

Because I'd heard about the time when they'd use, I think, a PDP and their node to update software, but you're describing a process where it's from IMP to IMP.

KLINE

How they got their software—when they were trying to update things, they—I'm not sure their process, but if your IMP crashed, rather than your having to reload the software in the IMP, you could just boot it, it would get its software from a neighbor. And eventually they modified that so they could tell an IMP, "Crash and reload your software from a neighbor," and they could, meanwhile—and I think they may have even put in something where, you know, you had version seventy-three and your neighbor had seventy-four, it would automatically update to the newer one or something. Somebody asked me about that a few years ago because he was involved in a patent litigation, a friend of mine who was involved in it, and because somebody was claiming that they had a patent on automatic updating of software, and we were saying, "Wait a minute. The IMPs did that forty years ago." [laughs] So he was looking for [unclear].

FIDLER

I understand that it was challenging to find the specific documentation for that investigation. That's what I've heard.

KLINE

It was challenging to find the documentation for him to prove that particular—what exactly was done. I basically told him his best bet was to get in touch with some of the guys from BBN that are still alive.

FIDLER

So it sounds like you had a lot of responsibility for the SIGMA 7. What kind of responsibilities did you take on when the IMP arrived?

KLINE

Well, when the IMP arrived, you know, I was there when they moved it in. I watched them set it up, and they showed me how to reboot it and how to load the paper tapes, and so if it crashed, I would do that. Or if I got a call from the guys at BB&N saying, "We're trying to test it. Can you stop it and tell us what's in this—push these buttons to give us the value of this register," or whatever. You realize there are vending routing protocols. So the IMPs are trying to keep track of how many hops is it from us to this other site. Well, let's see. We can go this way, and that IMP is telling me it's three to get to there, this IMP is telling me it's four to get to there, so I guess we should go this way. Anyhow, those things had bugs in them. There were even some famous cases where they got loops in them. I don't remember the details, but there are people who could tell you the stories of when packets got lost. When it came time to do TCP/IP, they added a field in the IP header called the time to live, and every time you passed a packet on to the next host, you decremented that, and if it got down to zero, you threw it away. And that would deal with any issues with there's a bug in the routing and that you don't have packets just floating around the net forever.

FIDLER

How closely were you connected to fixing or then improving the routing algorithm?

KLINE

I didn't do anything on the routing algorithms. Other people did, but I didn't. Len had people that were trying to do cubing analysis of all this stuff, and so some of his students were involved on that, but I wasn't. I'd hear about it, well, they found the following bug in the routing algorithm and here's what it did. Oh, that's kind of neat. But I wasn't involved in that. There were people, I can't remember who, but in our group that were trying to beat on it, trying to pump as much traffic as they could and see what would break and see whether routing protocols would break or whatever, but I wasn't doing that. But I did write the NCP and I did make sure it worked with our operating system, and I wrote one of the versions of Telnet. I think somebody else wrote the Telnet program we used.

FIDLER

Why don't we stick with NCP because it's a big story and we hear that, for example, there were no adults in the room, we hear that it was a somewhat decentralized process running this.

KLINE

Well, the whole ARPANET development was largely decentralized. You had working-group meetings on occasion, you had drafts of thoughts written and sent around as either just random notes or as RFCs, and people would kibitz on them and say, "No, we don't think that's a good idea for this reason," or, "Oh, that's neat, but we ought to add this field to it or whatever." And then little by little, people would say, "Okay, I think we've got a draft of what we think we can make the official protocol," so it would be protocol. And then people, "No, that won't work." And I think if you go through RFCs, you'll find draft proffered, official RFC, whatever. And I certainly read all those early RFCs and kibitzed. If they were being written at UCLA, I was kibitzing with the people who were actually writing them, or if it was somewhere else, I may have been sending off notes. In

the early days, we didn't have email and that kind of stuff, but we sort of did. Pretty early on, we had a form of email, and the way it worked is there were a few sites that had their own single-system email. The first one I used was at SRI, but also at Multics. I think we used Multics more for this then. And so if we all had accounts at Multics, we could leave each other messages at Multics, and I could log in as Charley or CSK and I could say, "Well, I want to send a message to Vint," but I would do it by logging into that system as if I was there. We just would agree to use—it wasn't until the early seventies that we actually had a form of email the way we think of it today where the message is created at one place and transmitted to another place.

FIDLER

Let me see if I've got this straight. So on the one hand, the protocol proffering you mentioned, that was a mid-summer RFC, 1970, I think. In terms of the messaging, you've got email on the one hand. On the other end of a spectrum, you have completely local messaging on a time-shared system.

KLINE

Right.

FIDLER

But you're talking about this kind of intermediary stage where you—

KLINE

I'm saying the completely local on a time-sharing system predated the seventies' stuff. That happened in—

FIDLER

Of course.

KLINE

—the late fifties, early sixties, mid-sixties, and we started using that, either by literally dialing with a modem into one of those systems or, as Telnet developed, Telnetting into the system and we'd say, "Okay, we're all using the system at MIT to send our messages to each other."

FIDLER

So local systems being accessed remotely that you'd use.

KLINE

Right.

FIDLER

That's really interesting.

KLINE

And that was one of the original goals for the ARPANET. It was resource sharing. Bob was saying everybody wanted a bigger, faster computer—Bob Taylor. He only had a finite budget. It seemed like a waste to every couple of years be buying everybody big, fast, multimillion-dollar computers. Everybody wanted a different computer or a different operating system, and they weren't compatible, and, gee, you wanted to run this graphics program that they have at University of Utah, but you don't have the right hardware for that to run that, or you want to run this chess program that they're working on at MIT, but you don't have—well, it made more sense to run it there, to share the resources. Not only that, in the middle of the night MIT time might only be eleven o'clock your time, and so it made more sense to share resources. That was the whole goal of the ARPANET, resource sharing as well as a platform for doing research on networking. And you've

probably heard this story, but most of the places that were ARPA contract or ARPA research places weren't really interested in connecting to this ARPANET. They said, "No, I'm working on an operating system, I'm working on A.I. I don't want to do that. I've got my own work to do. Why do I want to spend resources connecting to this ARPANET and why do I want to let other people use my computer?" And Bob Taylor pretty much said, "You're going to." [laughs] "You want your funding and you're getting a lot of funding from me? You're going to do this." And it turned out, of course, that it had a very positive long-term benefit, but it wasn't so obvious at the time.

FIDLER

Let's stick with this NCP topic because it's so important and you were right in the middle of it. You talked about the decentralized structure and how you'd utilize ARPANET resources to implement that. Were there sources of authority, though?

KLINE

The closest thing to the source of authority was Steve Crocker, Jon Postel, and a couple of others that were sort of considered the gurus, and they would sort of bless what we did.

FIDLER

And was there authority instituted by, for example, formal institutional titles, or was it—clearly it became these—

KLINE

It just sort of happened. Steve took an initiative to create this Network Working Group and get people together in physical meetings as well as exchanging a lot of documents, and we just sort of self-managed until we sort of came to an agreement that we'll keep working on these designs until we're all sort of satisfied with it. An expression that I remember developed as we would do things and then discover, well, we sort of did this wrong. There never seemed to be enough time to do things right, but there was always enough time to do it over. [laughs] Because we ended up doing things over several times. So we finally got the NCP because we needed a standard protocol that we could then write applications on, like Telnet and FTP. Even though the implementation would be different and the actual interface, what today is called an API, Application Programming, would vary from system to system, the functions, what you could do with the protocol, how did you open connection, how did you name sites, what kind of flow control was built into the protocol, all these kind of things. Now, NCP took advantage of some of the features that were provided by the IMP. That was a mistake. In hindsight, that was a mistake. For example, the IMP, you would send a message and it would tell you when the message had been delivered, and you couldn't send another message on a thing called a link until you got the thing back that this one had been delivered. So that provided a sort of built-in flow-control mechanism, and the NCP took advantage of that. Well, that had several—first of all, suppose you're using hardware that isn't an IMP, suppose you have a different kind of a network. Well, that protocol doesn't work. So in the seventies, Vint had gone to Stanford. By this point, there was already clones of the ARPANET. BB&N had sold a few clones. There were other networks being developed. There was CYCLADES in Europe and other networks that were being developed. There were various local area networks. That was something going on at Xerox PARC and so on. And the question was, well, how do you interconnect those. So it was time to build a new protocol that wasn't tied to the IMP and that separated some of the things that were messed up. We had sort of lumped together addressing and flow control and connection

opening and closing. Those had all sort of gotten smashed together in the NCP. So in TCP and IP, those got separated.

FIDLER

That's such a big story. I'm wondering if we can preface it with a bit more talk about the NCP before we get into it.

KLINE

Sure.

FIDLER

Because that's huge. I'm curious about with NCP if there was a back-and-forth with ARPA and then also with the sites that were producing the various implementations between the Network Working Group and then these particular sites in ARPA.

KLINE

I don't recall a back-and-forth. It was more of, you know, we need to do something so we all have compatible protocols that will talk together. So if I write a Telnet program, I can log into any site that's implemented an NCP and a server Telnet. Telnet had a user side, the part where I said, "I want to connect to you," and a server side, which is the thing that's sitting there waiting for people to try to connect to it. Well, if somebody implemented a server Telnet and an NCP, then I, user Telnet, should allow me to connect there. And, of course, we had to worry about issues like, well, some sites are using ASCII and some sites are using EBCDIC. Some sites always want us to end each line with a carriage return. Some types want us to end with a carriage return and a line feed. Some sites want the local terminal to echo. The physical terminals that they used at Multics back in the early sixties were basically IBM Selector typewriters that, literally, you push the button, it printed, whereas some of the other [unclear] systems didn't want to do that. They wanted to control the printing from the computer because sometimes they wanted to print different things than what you typed. The classic example is when I typed the L-O-G to log in, and it saw the G and said, "Oh," and it was going to type G-I-N for me. Or not echo, for example, passwords, not print a password. You'd type a user name, and then when you typed your password, it didn't print anything. These days, they'd probably print X-X-X-X-X, but if I remember correctly, in those days it didn't print anything. You just typed your user name and you typed your password and a carriage return, and then it either said you're logged in or it said password error or something. So we had discussions of things like to try to accommodate all these different systems. I don't think there was anybody at ARPA who was saying, "You have to do it this way," or, "I want to approve any of this." It was more of, "You guys make it work. Do something that's reasonable. Make it work." And because we had a bunch of reasonable people who wanted to make it work, that worked. If you have smart people who want to work together, who want to make things work, you can do a lot of good stuff. You can have smart people, but if they don't want to work together, things don't work. You can have people that want to work together, but if they're not smart enough, things don't work. We see this in Congress all the time. [laughter]

FIDLER

And speaking of that, when the RFC started getting produced, was that really a reflection of how the work was already structured, or did that introduce something new?

KLINE

Pretty much both. It introduced something new, because there was no structure. It was pretty clear we had this first IMP, there were some more IMPs happening, we all knew we needed to do something with this stuff, we needed to find a way to make it work. So Steve Crocker said, "Let's create a working group," which he called the Network Working Group. And he wrote an RFC number one, which he says, "Let's create a set of notes for this working group. Anybody can write one. Here's sort of the rules. It shouldn't be a waste of time, but you can write one on any subject." And that's how that happened. Even though the RFCs are still effectively the documents that document the Internet, there now is a bunch of rules. You have to write draft RFCs and sort of get them improved before you—they're trying to keep the RFCs rather than be an informal Request for Comments to pretty much a specification or close to a draft specification. But Steve sort of created that, and it sort of fit with the personalities of the researchers that were trying to make this thing work, and I don't think ARPA had—so people were sort of working that way, and Steve said, "Let's formalize this a little bit so that we can have a way of doing that." Some of the earlier RFCs said, "Here's who's going to get a copy of them." And it listed, "We'll send one to this site and this site." It even had the addresses, if I remember correctly, in there, and I think it was pretty much one copy was sent per site, and it was up to that site to distribute them to whoever needed them. Later on, of course, when things were online, things get a lot easier.

FIDLER

You mention personalities as intelligent people that want to work together, and then you mention personalities as maybe a source of things being organized the way they were. So much is said about the particular organizational structure of this and how influential it was. Do you think if you had a group of people with a different mindset, it would have gone differently? And furthermore, was this a particular social or cultural group that they're coming from that made that more likely?

KLINE

I don't know about the latter. I'm guessing that the answer is yes, because people came from MIT and other places where people were pretty much laid-back, informal, wanted to just do things, had a culture of just doing things. You compare that, for example, to the way things were done by the various agencies, for example, the ITU, and later on, the OSI and so on, we could have spent years working on standards and getting approval before anything happened, rather than—today this is called Agile Development Methodology. You start building things and you adapt rather than waiting until you've got it all specced down and every "i" dotted and "t" crossed. And because we didn't know what we were doing, if we wanted to get something done, we just sort of had to start moving forward and experiment. And the fact that ARPA was willing to allow us to experiment, they weren't saying, "I want to see a thirty-page spec before you guys do anything." Even BB&N, there was an RFP, Request for Proposal, that was sent out, for which a lot of companies refused to bid. If I recall correctly, AT&T said it would be a waste of time and they wanted not to use packet-switching. They wanted to use—IBM said nobody would ever want it and so on. Then BB&N said, "We can do this." But they had a lot of freedom in specifying what the protocols would look like. I've seen the original RFP, but I don't have a copy of it, but it didn't have that much detail in it as to what would the interfaces with the computers be, what would routing look like inside of the IMPs. It was more of they needed to cost about this much money and I want them

delivered by this date, and they need to handle about this many communications per second.

FIDLER

So do you think that the structure of ARPA's request mirrored their general philosophy about [unclear]?

KLINE

They had a philosophy—at least in the computer group. I can't say about the other group. ARPA had—the history of ARPA had a lot of groups. They started as a response to Sputnik, and first they were going to go into space stuff. They turned out not to do much in space stuff. That pretty much got handed off to other agencies. But they did behavioral science research and they may have done some biochemical research. They certainly did computer research. They did research on semiconductors and various things. They did a lot of stuff. And the program managers, at least in the computer departments, had a lot of freedom to both fund what they thought was interesting and to manage it however they thought, and they tended to want to manage it in do good stuff, and if you do good stuff, everything's fine. [laughs]

FIDLER

Were you aware of the difference between, for example, the IPTO at ARPA and then other offices at ARPA and maybe—

KLINE

I wasn't. I mean, I knew that there were other offices. I didn't know anything about how they were run or what they did. You know, if you talk to people who had been at ARPA, Vint, Steve, others from ARPA, you could probably find out about how ARPA was managed in other offices other than IPTO. But I only had interface with IPTO, and occasionally somebody would come out who was like—the director of ARPA would come out and visit, but that was more like, "The boss' boss is coming." But how they were managing things and whether they managed other groups, I had no idea.

FIDLER

Did you participate in any demonstrations for ARPA?

KLINE

I'm sure I did. I can't specifically—not at conferences and things like that. Jon Postel did, and some others went to some of these conferences and set up things and showed off things. But I mean, when people came by, or if ARPA brought somebody by, I showed things to people, you know, "Go grab Jon. Go grab Charley. Go grab whatever and have them show this guy how this thing works." Also, even though I was like a chief programmer type, I wasn't the head guy. The head guys were Vint and Steve and Jon Postel and Len and those. So when they were sending somebody to a conference, that was more likely to be them.

FIDLER

And briefly about your titles, you haven't mentioned your formal staff titles at all.

KLINE

My titles changed. Started off as lab helper. Then I got promoted at some point to what they used to call junior coder—Coder 1, I think it was called—and then later Coder 2, then later Programmer 1, which I think later they eventually changed to assistant programmer, and then programmer and senior programmer, these over a period of years. Each one was a significant raise because each of those had steps. It was like Assistant



Programmer Step 1, Assistant Programmer Step 2 or something. So it was probably a 20, 25 percent pay range, and those things probably overlapped slightly. Then there was senior programmer, principal programmer, and then the top programmer ladder at UC at that time was called computer systems designer. So I eventually got promoted to computer systems designer. That was sometime in the late seventies. When I was promoted to computer systems designer—these are all staff titles. I mean, these are the same titles that somebody who might have been working on the accounting systems for the university might have or whatever. Anyhow, computer systems designer was a pretty hard title to get. When I got computer systems designer, I was actually getting paid more than my professor, Jerry Popek, was getting paid as an assistant professor. [laughs] He wasn't very thrilled at that.

FIDLER

I can't imagine he was. It's curious because you've mentioned all these different roles and responsibilities, but you didn't bring up these staff titles. Besides the money, I'm wondering if they had much of a bearing or determining power on what you were actually doing, or did they just follow along.

KLINE

They just followed along, but as we started doing more interesting things, my role became partly a managerial role. I didn't really have a managerial title and people weren't specifically reporting to me in a formal org chart, but I was telling people what to do. When we were building the secure UNIX system that we were doing, and later when we were building a LOCUS system, I was effectively the senior technical guy in the group, but I don't think I had people who were officially reporting. I didn't do periodic annual reviews for people and that kind of stuff.

FIDLER

And is that part of that same local culture, if we can call it that, that we were talking about a moment ago?

KLINE

Partly that and partly—by this point I was working for Jerry Popek, and I think he didn't want to establish a big hierarchy internally in his group. He sort of wanted everybody reporting to him. That changed when we started LOCUS, I mean the company, because then I actually was managing people.

FIDLER

Did you see a difference between working with Jerry Popek and then working under Len Kleinrock at the Network Measurement Center for how that organization, those management strategies would work?

KLINE

Well, with Len, I switched from working for Len to working for Jerry about, I'm guessing, '74-ish, something like that. What happened is Jerry came in late '72, early '73, and he got Len to effectively let him have some of his money/student—basically students to do some work, and at some point Jerry got his own contract from ARPA. Jerry and I are only a year difference in age. Jerry's one of these guys who went right through and got his degree like that [snaps fingers]. I mean, I started in '65 and didn't finish my Ph.D. till 1980—three degrees—but Jerry started in, I'm guessing, '64, and was an assistant professor by '72, but he was one of these guys just, you know, was gung-ho on stuff. Jerry and I hit it off. I mean, this new assistant professor was coming, and they said, "You

might be interested in meeting him. His research is in computer security,” and I was sort of interested in computer security at that point. So we hit it off. So the difference was with Len, I was just a student who was working in his group. With Jerry, I was almost a co-equal working with him on stuff. He was officially my boss and he was also officially my thesis advisor, but he was more of a co-equal.

FIDLER

Did everyone report directly to Len the way you describe with Jerry Popek?

KLINE

I’m trying to remember. The answer is I don’t recall. I mean, the students worked for Len. Len’s students reported to Len on what they were doing. In terms of the people that were sort of doing staff work, some were reporting to Steve, some to Vint, some to Len. There wasn’t much of a hierarchy.

FIDLER

I’m wondering on this topic of how research was structured and the culture of that, is there more that you can say about how ARPA did things and how things were done locally at UCLA and the relationship between those things?

KLINE

Well, my understanding, from what I can remember and from what I’ve seen, was people at different universities had gotten reputations that they were doing good work in graphics, they were doing good work in AI, they were doing good work on operating systems, they were doing good work on man-machine interaction or whatever, and if they were doing good stuff and they could write a good proposal to ARPA, and if something ARPA was interested in funding, there was a good chance they could get some funding. One of the tricks was knowing what ARPA was interested in funding. So I have no idea how ARPA actually ran their office, although I’ve gotten the impression from many people that it was, again, very informal, very loose. The program managers had a lot of freedom to sort of fund what they wanted within their budgets. They had a budget. When Bob Taylor decided he wanted to do the ARPANET, he went in to see Steve Lukasic and explained why he wanted to do this, and he came out with five or ten million dollars more added to his budget an hour later. [laughs] But you did have to write real proposals, real grant proposals to ARPA for money, but you sort of knew whether you were going to get the—I don’t think they got all that many just sort of out of the blue.

FIDLER

So it was based on previous, maybe informal?

KLINE

Informal or being introduced to people and them hearing about you and learning what you were up to and sort of making a decision as to, “If this guy writes us a proposal and it makes some sense, we’ll fund him.” In the late sixties, ’67, ’68 time frame, ARPA was funding about half of all computer science research in the world. Everybody thought, oh, IBM must have these huge research labs and this and that, but if you added it all up, ARPA was funding a lot, certainly of the advanced research. IBM might be funding better work on the physics of disk drives, but in terms of software and new ideas, ARPA was funding a lot of it. That eventually got ARPA in some trouble because some congressman said, “Why is ARPA funding this? ARPA stuff should be specifically military-focused. If it’s not a military focus, it shouldn’t be done by ARPA. It should be done by NSF or somebody else.” And that’s part of the reason it changed names from the

Departments of Defense's Advanced Research Project Agency, ARPA, to the Defense Advanced Research Project Agency, DARPA, and it changed back and I think it's DARPA again. I think it's gone from ARPA to DARPA to ARPA to DARPA. [laughs]

FIDLER

In the nineties, I think they got the D-MAC for a few years. And I wonder are you referring to Mansfield like around in the early seventies when—

KLINE

I'm guessing mid-seventies, something like that. I don't remember the exact time frame. You can probably look it up online. But there was, you know, something, "Why is ARPA doing this? ARPA should be doing—if they're doing research on how we can build a new bomber, that's great. If you're doing computer work for what we need for the military, sure, but if they're doing general computer science, why is that not just general research?" Well, there were two reasons for that. One is they needed that general research, they needed the state of the art of the computer industry to grow and some other industries to grow for military reasons, but also they had the money, and these other agencies couldn't get the money.

FIDLER

Were these shifts perceptible at the time, or is this something you learned about after the fact?

KLINE

It's something I learned about later, yeah. At the time, none of that was obvious to me. There may have been people who it was obvious to. I mean, Len may be able to tell you stories and may know more of what was going on. To me, I learned about that later when I would talk with people like Vint later on when they were at ARPA.

FIDLER

So it sounds like you were pretty shielded from those kinds of concerns.

KLINE

I was shielded from all that. I helped Jerry write proposals to ARPA, but I was shielded from the internal politics of what was going on inside of ARPA and how it related to Congress. Apparently, Jon Postel, who you never met, Jon was a guy who liked to walk around in sandals and bare feet, had long hair. You've probably seen pictures and heard stories. Well, apparently there were some interesting stories when he showed up at the Pentagon or when he showed up to testify in Congress or whatever.

FIDLER

Anita Coley recalls him walking around in bare feet as a common occurrence.

KLINE

Yeah.

FIDLER

I wonder if we can go back to a bit more on NCP, because I notice in December 1972 you've got RFC 417, this little link-use violation. I think it was 10X, if I remember correctly.

KLINE

They were doing something wrong. I don't remember what it was.

FIDLER

And then in February 1973, you've got RFC 460, which is this NCP survey, and both of these seem like you're monitoring NCP implementations from UCLA and suggesting—

KLINE

There had been NCP specs, and people were building NCPs and, of course, we're trying to use NCPs, and I'm trying to debug NCPs, and if I got a—you know, I'd get back—I think my code would say, "Now, you've got a bug here. This message didn't come back correctly." And I'm thinking, "Why? What am I getting," or, "Oh, why are they using that link or whatever?" The survey—somebody had suggested, and I don't remember who; it may have been Jon—that we do a survey of the implementations and see who implemented what and who didn't implement what, what features were implemented, because there were some optional features, if I remember correctly, in the NCP. And since I was reading all the RFCs, I knew the protocol, and since I had built an NCP, I was pretty familiar with the protocol at the time, so if I saw somebody doing something wrong, I said, "Wait a minute. Gee, why are people—" You know. RFCs were considered an informal set of notes, so you could write an RFC that said, "Let's have a meeting next week," or, "What do people think about using this field in this way instead of that way?" So if I saw that, "People are using this link field in a way that I don't think is correct. What do people think about that?" And that wasn't necessarily a criticism; that was more of a question.

FIDLER

Oh, that's interesting. So you had specific implementations. Was UCLA the main site that would then be monitoring these and suggesting, raising questions about how it had been implemented?

KLINE

To some degree, yeah. Not completely, but to some degree as we would try to measure things or test things or see what was up and down, because we had this thing that was running around periodically seeing which sites were up and what features were up, we were interested in who was doing what right and who wasn't. You could say we were tasked with that, but you'd say we weren't. We just were doing it.

FIDLER

It sounds like there's a relationship there between what you ended up doing on NCP and the broader role of the Network Measurement Center on the network.

KLINE

Right. Yeah. The Network Measurement Center really never was the way you think of a—there are—I'm trying to think of the name. Network ISPs have, effectively, control rooms that keep track of stuff, measure stuff, try to see what's going on to deal with problems and put out fires. We never really got to a point we were at that level. It was more of, gee, Len had these cubing models of what he expected how the network ought to behave, and we're trying to do some measurements to see if it actually does behave as the models predict it will, rather than as a measurement center that's constantly measuring it. On the other hand, BB&N was constantly getting reports back from the IMPs periodically about the status of things like, "This circuit is down." So BB&N called up AT&T Long Lines one time and said, "Circuit Number"—and they gave some big long ten-digit number or whatever—"was down from 12:02 a.m. to 12:04 a.m. on the following day, and was down from such and such and such and such on the following day. We want a refund." And AT&T was going, "Huh? How can you know that? We can't even know that. We don't even have any technology to know that." [laughs] But that's because the IMPs, when they weren't sending real traffic, were sending packet writing table updates

and “are you alive” messages, and if they didn’t get the responses, they could see, “Well, I guess that circuit is down.” In fact, the IMPs had lights for each of your circuits, and when the circuits were working, the lights were off. When the circuits were not working, the lights were on. So when the IMP booted up, these lights would all be on, and then as the IMP started talking, the neighboring IMP lights would turn off. So you could actually look at the IMP and see which of the lines were up or down. Anyhow, so BB&N had sort of statistics about the failure rates of circuits that effectively AT&T had no way of doing, because they didn’t have any way of collecting statistics from the—nowadays, some people complain about it, but we have these smart meters which electronically transmit your bill, but also can show how your energy usage changes during the day. Some people don’t trust—there’s two kinds of people who don’t trust them. There’s those who think that somehow they’re transmitting stuff and you’re going to get cancer or something from it, like being near a power line. [laughs] Then there’s the people who don’t think the meters are accurate. So nowadays, the power company can get actual data on, well, how does the usage go, and which blocks get more of it and which get less? They never could get that before. All they could see is out at the transformer, more power was going out. Well, similarly, AT&T, unless they got a call saying, “The circuit’s down. We need to have somebody come out and fix it,” they couldn’t monitor how often the circuit was failing and recovering. But BB&N was getting those statistics. [laughs]

FIDLER

It’s interesting about BBN, because from what I understand, BBN would be accessing the same measurement tools in the IMP program as UCLA would, but then there’s somewhat different purposes for why they were doing it. I wonder if you can elaborate on that.

KLINE

Well, we took advantage of some of the features of the IMP where you could send it packets and it would respond to you. You could ask it for some of its routing tables or whatever, and it would send them to you for our measurement purposes or to see what was going on. BB&N was getting this all the time just to manage the network, just to keep it working just as an operation center to know that the IMP was up or the IMP was down, or a phone line broke. That reminds me. Our IMP was one of the first, and it was a Honeywell 516. And a few months later, say maybe six months later, they came out and came and changed the control panel, because the original one we had was a standard 516 control panel, and the light bulbs plugged in and so on. They changed them to these little bulbs that screw in because the ones we have weren’t RFI-protected, radio frequency interference-protected. And the original spec said everything was supposed to be milspec. It was supposed to meet certain criteria for a radio—so if you open up the IMP, you’ll see all this sort of shielding and stuff. That was, in theory, to [unclear] radio frequency interference. I don’t know if that was because they thought the bad guys were going to try to do something or whether it was just because they figured it would be more reliable if it wasn’t going to be affected by other noise in the room. You have to understand the equipment wasn’t all that reliable in those days. [laughs] But I remember, “So you’re changing out all the light bulbs in all the panels?”

FIDLER

That was within, like, the first year or something?

KLINE

That was within the first year—

FIDLER

Interesting.

KLINE

—and I guess the later IMPs came that way.

FIDLER

Those measurements experiments, I understand initially the data was accumulated and processed on the SIGMA 7. Did you have any role in establishing setting up the programs that would do that, running the experiments?

KLINE

No, no.

FIDLER

Did you notice them being run? Were there times when the signal would be reserved?

KLINE

I mean, it pretty much didn't affect anything, and there may have been times when somebody said, "I'm going to run an experiment and I'm probably going to slow down the network." I would have just said, "Sure. Thanks for telling me." [laughs] And then once they got the data, whether they wanted to process it on the SIGMA 7 or whether they wanted to do some more sophisticated statistical processing and use the 370 next door that had statistical packages they could use depending on who was doing the work.

FIDLER

And a brief specific question. Did you know of ever accessing the routing tables on the IMPs from the Network Measurement Center? Because you mentioned the routing tables briefly, and I think that might have—

KLINE

I know you could get at them. I don't remember doing it personally. I believe there was a message you could send to the IMP which would [unclear] tell it to send you the routing tables, and I think somebody did that, but I wasn't doing very much actual measurement. I was keeping the systems working, and other people were doing the measurements.

FIDLER

While we're moving on from NCP, before we get to TCP, can you tell me more about what you do? Were there other projects that you contributed to, people that you'd just discuss things with?

KLINE

Well, a bunch of us were reading all the RFCs that were being generated, whether generated at UCLA or generated at other places, and kibitzing on them and discussing do we think this is a good way to go or not. We were all taking classes, so if somebody was doing research on something, whether it was—I was taking Len's cubing theory classes, so I was learning all that stuff. I remember being in meetings where there was discussions going on about the research on the packet radio, the satellite radio work, but I wasn't specifically involved with it other than going to meetings. I may have listened and I may have kibitzed, but it wasn't a focus of mine. I was impressed when I heard about the packet radio stuff, which was effectively based on the—at the University of Hawaii, they built a system called ALOHA, which basically used radios to transmit data among the islands, and the packet radio used some similar stuff to transmit packets over radio. Now, what was interesting was the radios they got, which were made by Collins Radio, which was a big radio company, used spread-spectrum technology, which I had never heard of

at that point. In fact, at one point in the past, it was classified. Spread-spectrum technology basically spread the data over a lot of different frequencies in various different coding ways, and the claim was that there was enough redundancy in it and the power in any individual frequency was so low that it was pretty much immune to interference. So people were explaining to me the coding and how that all worked. I thought, "Wow. That's really neat." And then, of course, when Bob Metcalfe at Xerox PARC was developing the Ethernet, he was basically saying, "Well, we've got ALOHA over the radio. Why can't we do the same thing over a wire, basically the same protocol, but we'll transmit the signal over a wire." If two radios transmitting at the same time on the same frequency will interfere with each other, you'll get garbage. Well, if two people are transmitting on the wire at the same time, they'll interfere with each other, they'll get garbage. They'll both see the garbage. They'll retry. And if you put a little random delay in there, then one of them will probably get through and the other one not, and then it'll—

FIDLER

Before the TCP experiments, were you aware of these attempts at interconnection with networks? So, for example, there's ALOHAnet in Hawaii, there's UCL in the U.K.

FIDLER

I was aware of the attempts to interconnect various networks to the ARPANET in one ad hoc way or another, not by building a whole new protocol, but by coming up with some way of interconnecting. They'll use ALOHA to here and then they'll have some kind of a gateway that will transmit it onto the ARPANET. But in terms of—TCP was the first attempt that I really knew about to try to build a general framework, although the Xerox PARC, PARC Universal Packet, PUP protocol, was effectively an IP-like layer that predated TCP/IP slightly, so they claimed they invented the Internet protocol because they had a protocol that you could use to interconnect different kinds of networks and encapsulate other things inside of it. I don't think it was as general as IP, but—

FIDLER

Once we've gone through some more of this, I'd really love to hear your thoughts on the invention debates that you're—

KLINE

Oh. Well, I'm not sure I want them on tape, but I— [laughs]

FIDLER

We'll talk after. Did you ever talk to people, either telephone or through the network, that were involved with ALOHAnet, for example, or other network projects?

KLINE

I did, yeah, because either they came by UCLA or I talked to them for some other reason. I can't think of the guy's name, but the guy in England on the CYCLADES. Kirstein.

FIDLER

There's Pouzin on CYCLADES and then Peter Kirstein on UCL.

KLINE

Peter Kirstein, yeah, on UCL, and I talked to somebody in Hawaii. I don't remember who. It may have been answering questions about NCP, or it may have been they came by UCLA for some reason. I didn't have a lot of involvement, but certainly met all those people.

FIDLER

Do you think that was the extent of the connection, for example, between UCLA people and ALOHAnet or UCL, was that you'd have meetings or telephone calls about protocols? It wouldn't be someone at a terminal in Hawaii that you'd be later sending an email to or something like that?

KLINE

Maybe later, but only if I had a reason to. And also that wasn't so much my function. Jon Postel was involved with all these people, so he was sort of the knowledge repository as well as the—so he was always keeping track of who was interconnecting to the ARPANET, what they were doing. If I needed to know, well, what are these guys in England doing, I would go to Jon and he'd say, "Well, this group are doing this, and Peter Kirstein, oh, he's doing this and this, and what they're going to do is that." But I didn't have a big interaction, and most of the people didn't. Some of Len's students did, because they wanted to analyze the queueing analysis of ALOHA, and they wanted to figure out how that—when they tried to analyze packet radio, because I forget which student—one of his students did his Ph.D. on packet radio. I'm trying to remember. I don't think it was Mario. It was somebody—

FIDLER

I know that there was SATNET and PRNET.

KLINE

Right.

FIDLER

Some of it was, I think, organized at least in some capacity through the Network Measurement—well, post-Network Measurement Center Lab.

KLINE

Right.

FIDLER

Earlier, you talked about design decisions for NCP that were linked to developments in TCP, and this connection between NCP and TCP is something that's—

KLINE

Well, it was more of over time we had seen that there were errors in NCP, errors in the sense that we could have done it better. You always discover, "Oh, I could have done it better if I had only done it this way. Gee, if I had only made that screen on that phone another quarter-inch wider, I would have been able to put another icon on that screen. It would have made all the—." You know. [laughs] So as we're learning things we could have done better, okay, at the same time, there was pressure to start interconnecting the—when I say pressure, some people wanted to. I'm not sure there was any pressure from ARPA. But Vint was at Stanford and he wanted to interconnect all these networks. He had a student, Carl Sunshine, and between Vint and Carl Sunshine—and Vint, apparently, and Bob Kahn, supposedly over dinner one night, on a napkin sketched out their thoughts for TCP/IP. I remember getting, just like I got all the drafts of all these other random documents, a draft of TCP. I don't remember exactly what year it is. One day I said it was some particular year, and Vint said, "No, we didn't have a draft by then," so I had to be off by—but the first version of TCP had most of the same concepts, but it was one protocol: TCP. It wasn't TCP/IP. And then at some point, they realized it made sense to separate the addressing and routing of packets from the flow control and connection establishment, so that's why TCP and IP separated. And then you say, well, IP is doing



the transmission, and the routers and things are actually moving the packets. You could have other protocols besides TCP. For datagrams, you can have the datagram protocol, and you can have other protocols that go directly on IP. So there's some protocols that just use IP, and there's some protocols that use TCP/IP and work on—so, for example, ACTP, the web protocol, is a protocol that works on top of TCP on top of IP, but there are some protocols, like some of the protocols for packet voice, for Voice-Over-IP, work directly on IP rather than working on top of TCP, because TCP has a layer of overhead that is great when you're streaming data, but isn't necessarily great for short little bursts where you don't want to do a big setup. Anyhow, so they wanted to be able to interconnect these different kinds of networks and also, again, NCP was built when the only thing really we had was the IMPs and the ARPANET. Now we have to worry about other different kinds of communications hardware. We've got local networks, we've got Token Rings, we've got Ethernets, we've got packet radio equipment, we've got other communications protocols that aren't even anything like the ARPANET, but we'd like to sort of encapsulate them and transmit them over the ARPANET. There are people beginning to work on Voice and wanting to put Voice on the ARPANET or on the—so it made sense to have a more general set of protocols, and that led to TCP/IP. And in the process, they fixed up a bunch of things. They said, "Well, you know, the IMPs were designed for up to 64 sites with four hosts per site, 256 hosts. That's clearly not enough. Gee, how big should we make it? Oh, why don't we make it 32-bits. That'll last forever." [laughter] Turned out not to, but—

FIDLER

Were the uncontrolled packets that they experimented with on the ARPANET, was that influential at all in the development of—

KLINE

The uncontrolled—

FIDLER

Like for Voice experiments?

KLINE

Well, people who did Voice experiments learned about the kind of effects of delays and that what—Bell Labs mostly already knew that, that you couldn't have more than a certain amount of delay or you'd hear it, and they were gradually beginning to learn about roughly how many instructions you could afford to go through in processing a packet and still have it not add too much delay. Of course, computers kept getting faster and faster. That helped. And the bandwidths of the communications kept getting faster and faster. In the late nineties, a T-1 line was really expensive, and that was 1.544 megabits. Today we're at home and I get between 25 and 50 megabits down and about 10 megabits up, which is plenty for my uses, and delays are in the low numbers of milliseconds and that's the critical number for Voice. It was funny, in the late nineties I went to a conference where they were talking about—they had the outgoing head of the FCC and the incoming head of the FCC, and as part of their talk, they said, "Well, it's clear that the whole world is going to go IP at some point and that Voice is eventually going to be over IP. It's not an issue of if, it's only an issue of when." Well, only recently now, they're talking about switching off the landline phones and converting to an IP-based network, and there are some issues, some technical issues there having to do with

reliability of an IP-based network, not in terms of voice quality, but in terms of reliability, in terms of power outage.

FIDLER

You mentioned errors or things that you quickly realized you'd like to fix on NCP, and I hope you can tell me more about that. And also were these errors things that you'd noticed before TCP development started?

KLINE

No. I mean, I did notice before TCP development that NCP was too tightly coupled to the IMP, that it took advantage of features of the IMP to do things that really should have been built into the protocol so that they would work on other networks besides an IMP-based network. But aside from that, and because of that, also, the NCP was based on, again, 256 hosts. So we needed to fix those things. We needed to have a network protocol that didn't depend on IMPs that could work on—you know, with bigger address base than 8-bits could be 32-bits, which we thought would have been plenty. [laughs] But then people still weren't thinking of billions of sites. With billions of sites, you need a naming system. I used to know in the top of my heads, one is UCLA, two is SRI, three is Santa Barbara, four is Utah, five was—

FIDLER

BBN.

KLINE

—BB&N, six was MIT, seven was—I don't remember.

FIDLER

Lincoln?

KLINE

I don't think so. Anyhow, there were a lot of sites I just knew the numbers of, so if I wanted to connect Telnet to them, I could say, "Telnet to six," and that would connect me to MIT, to Multics. But, okay, so then we made tables where I could say, "Connect to Multics," and my software would look in a table and say, "Oh, Multics. That's six." That works fine when you've only got a few hundred things and you're distributing these tables literally on paper, eventually typing them, sending them, but you need a system that scales to billions of things, that works when sites are down and so on. That ended up with DNS. Okay. Well, nobody even thought about that back then, that you're going to need directory systems and naming systems. So it wasn't so much that it was an error that we recognized at the time. In hindsight, as this thing grew, it becomes an obvious error that this doesn't scale. I guess there were two errors that are obvious in hindsight that weren't so obvious at the time. One is that it didn't scale to speeds and sizes of networks, and the other is it didn't have any security built into it. And you'll hear everybody who talks about—if you hear a talk by Vint Cerf and he's talking about lessons learned from the ARPANET and from TCP/IP, and he'll talk about, "Well, of course, none of us realized at the time, we wanted to put security in, but we just didn't think it made sense to do at that time." [laughs] I'm not even sure they wanted to put it in back then. And the computers were too slow to do encryption in software, and hardware encryption chips were just beginning to happen in the eighties.

FIDLER

Of the errors that you saw sooner, there's the dependence on the IMP. Two things about that. One, I'm curious if there's more you can say about how and why you noticed that, and then, two, if this was part of this broader concern with separation of function.

KLINE

It's part of a broader separation of function that if we're trying to build a general-purpose protocol that applies to more than just the ARPANET, well, the communications hardware might be different, so the function needs to be—if you go back to PC and MS-DOS, the PC had certain spec of hardware. You could access the display hardware. You could actually write in the display memory. People did. But if you did that, you discovered, "Wait a minute. If I get some different display hardware, my software isn't going to work anymore." So you really needed to separate the functions so that there's system functions that you call to draw a line on the screen or to put up a character or whatever, so that you can replace the driver because now you've got a different graphics card, and the rest of your software still works. Well, it's the same idea. You really want separation of function, and that didn't really become obvious until probably around the time of TCP, when you're saying, "Wait a minute. We know we need to fix some things. We know we want to add some more capability. Hmm. We really need to separate out the functions here so that this isn't dependent on that, and if we change this, we can change it without having to rechange everything else."

FIDLER

Were there moments in everyday use when that would come up, or was it, conversely, you start thinking about interconnecting networks, and then that gets identified?

KLINE

It gets identified when you think about what could we do better or how would you interconnect networks. If things are working, you don't notice the fact that, "Gee, this would have been better." If the network had been designed so it could have 32-bits instead of 8-bits and we had a million nodes, some of those things, for example, the naming issues, would have come up a lot sooner. And, of course, once you had email, then suddenly the issues of, "Well, gee. I want to send it to Brad Fidler at UCLA. Now, how is the mail delivery system going to know what 'at UCLA' means? Well, gee. I need a system to do that." Well, that wasn't too bad when, again, when it was only 64 or 256 sites, and I could say, "Send it to Brad Fidler at UCLA Computer Science," or at UCLA Math Sciences. But when you get millions of sites, you've got to have an automatic mechanism that could convert and find out how to deliver stuff. So scaling issues and separating of functions so that you have the freedom to evolve, to change, to improve, you know, I don't really care what kind of engine I have in my car; I just want to know that I can put gas in it, step on the gas, and it runs. I don't really care if it's a four-cylinder or a six-cylinder or a gas-burning or an electric. I just want it to work. But in order to change it, I've got to separate some of those functions.

FIDLER

So it really was—well, for you—

KLINE

Change is what forced the noticing of mistakes more than the mistakes were so visibly obvious that, "Oh, my god. We really screwed that up." Now, in some of the generations of the drafts along the way, we'd say, "This doesn't work," or, "There's a race condition here," you know, just plain bugs, things that just aren't going to work, but in terms of

some of these other things, you didn't notice it until you discovered it just didn't expand or you wanted to make some other change. You wanted to put in priority, and you said, "Well, wait a minute. There's no field for priority. But I need these packets. These are Voice packets. They need to go with higher—" Or, "These are video packets. They need to have a higher priority." And then later on, there was a discussion of can we do a multicast; that is, "I want to send this to three sites. Do I have to send out three copies of the message, or can I send out one and have the IMP or the router or whatever deliver the three copies?" In fact, even better is if it's got to deliver three over here and three over there, if it sends one to here, and then over here it gets divided into the three copies, well, the multicast protocols that are in the routers today will do that if they're used. Nobody uses them, that's a different— [laughs] But that evolved. That requires the ability to have these functions separate so that they're not tied into—when you're writing your application and you say, "I want to send this," you don't want to have to worry about what's going on underneath to get it sent.

FIDLER

And is layering another thing that would link NCP to TCP for you?

KLINE

Yeah. Layering makes a lot of sense in that, okay, this layer deals with this set of functions, it's a separation of function, and I can just assume it's there and use it. But layering in theory is less efficient, because in theory I have to make this nice clean architecture and clean layer, so I'm going to have to do something and translate it to what that layer wants versus just cheating and going right down to there, which is why game programs on PCs will sometimes cheat and go directly to the graphics hardware. You'll see options in some applications where it'll say, "Do you want us to use the hardware if we're able to directly?" I think Windows has some options to do that if you go down on the control panel in some places. But if you're trying to design things that are clean so you can replace things, you want everything cleanly layered.

FIDLER

Were you exposed to many of the decisions about how to layer these functions in the early planning of the ARPANET?

KLINE

Well, not in the planning of the ARPANET in the sense of the first meetings that were going on back with Bob Kahn and Larry Roberts and Len Kleinrock and—oh, I can't think of the names.

FIDLER

Shapiro?

KLINE

There's a guy I was thinking—the guy who said, "You guys have it backwards. You should build these separate little computers."

FIDLER

Wesley Clark.

KLINE

Wesley Clark. Anyhow, I heard the results of those conversations, but I wasn't involved in those, in that kind of planning or those kind of discussions at that point.

FIDLER

And then when it came time to make design decisions about how these layers were going to function, interact to each other, was this also something that you were—

KLINE

The stuff that, you know, like how will the NCP work with the IMP layer, yes, I was involved in that, but how is the IMP going to work with the hardware layer or how will routing work or whatever, later on when I was, like, at Cisco and that kind of stuff and were doing a different layer of protocols and we're worried about, you know, what functions belong in the routers and what functions belong in gateways above the routers, and where should a Voice-Over-IP gateway be, and what functions, for example, should be in the cable modem at the actual modem, what functions should be in the cable head end, and what should be in the router that's beyond that. But in terms of things like TCP and IP, only vaguely, only peripherally. I'd read the drafts and say, "This makes sense. I might have put that function here, but—."

FIDLER

What else would you say, if anything, about the relationship between NCP, either in retrospect of errors in NCP or in functions that were developed there then migrated in some sense over to TCP?

KLINE

Well, to be honest, NCP was an attempt to just "Let's get something working." We've been doing everything ad hoc up to this point. People are just saying, well, we've got an IMP and we can send it things. We need to have a protocol so that everybody agrees that here's how transmission works so that we can build things like Telnet and FTP and those kinds of things on top of it. So we have to do something. So let's compromise on a protocol. So NCP was never—I won't say it's got a lot of errors as much as I'll say no one really thought about it as a long-lasting thing. It was always something that we'd use for a while while we figured out something better to do. NCP had a connection establishment, and this is closing of connections, and it had flow control, and TCP had a connection establishment and flow control. NCP also had the actual communication, whereas TCP didn't. TCP left that to IP. IP said, "Here, IP. Get this from here to there." So IP had things like communicating with whatever layers at the hardware with whatever—IP would say, "Gee, I may need to fragment this packet into smaller packets, because I happen to know my hardware doesn't allow things to be that big." NCP didn't think about those kind of issues. NCP assumed we have a network that handles an IMP-size packet, 8,000-bits, and so people divided it into IMP-size packets and that's all, whereas TCP said, "I'll leave it IP to transmit." And IP said, "I might have some hardware that can only transmit 100 bytes at a time, and others that can transfer 1,000 bytes at a time, and others 10,000 bytes at a time. So whatever I get, I'll get this hitted and a count, and if I have to divide it, I'll subdivide it, and I'm allowed to fragment it and subdivide it."

FIDLER

Is there anything else you want to say about NCP or the transition to TCP?

KLINE

Not that I can think of.

FIDLER

Okay. During that whole time, are we talking about a similar management style, lack of org chart?

KLINE

Yeah. And we weren't quite as involved in the TCP development. We were reading the documents. By "we" I mean the guys at UCLA. We read the documents, we kibitzed, but we were doing other things. I was working on security by this point. And so TCP got implemented basically by other people at other places. They got an ARPA contract at Berkeley to put it on Berkeley UNIX, which became sort of the reference implementation that everybody started from after that. And it took a while before there was real pressure for everybody to move off of NCP and move on to TCP. There's a famous date—I forget when it was—where the cutover from NCP to TCP and that NCP was going to go away.

FIDLER

I believe there were badges that were given out for that day. It was January 1st, '83.

KLINE

Something like that.

FIDLER

"I survived the—."

KLINE

Yeah. But it was largely visible to me. By '83, I was away from UCLA working on stuff, and we were already running Berkeley UNIX, so it already had TCP in it. And since we didn't have a direct connection to the Internet/ARPANET in our company, we were using UUCP to interconnect through UCLA to get to the ARPANET. So I could email people everywhere invisibly, but it was later on that we actually got an Internet connection. But I have friends who made a ton of money building TCP/IP for various companies.

FIDLER

In accounts of the really early inspirations for the ARPANET, there's talk about seeing all these computers in a room and they can't communicate with each other. When the ARPANET was being developed, did you notice it solving preexisting challenges that you'd had or frustrations you'd had using computers, or was it opening up new areas?

KLINE

Both. For me, when the computing facility at UCLA, the big facility which had the 360s, got connected to the IMP and we were connected to the IMP, then I could submit jobs that I suddenly didn't submit on the 360s by typing them up on my SIGMA 7, pushing some buttons, and sending them in as a job, getting the printout back to the SIGMA 7, and either looking at it on my terminal or printing it out on the printer on the SIGMA 7. So that solved just—I didn't have to use cards anymore. A lot of the campus was still using cards. They had—I wish I had a picture of it because a friend wanted one. I never got one. They actually had vending machines like candy machines, you know the candy machines where there'd be this kind of bar and that kind of bar. When you pulled the lever, that thing would move and the bar would fall out, okay? Well, they had card machines where they had like a rubber band with fifty cards, compute punch cards on it, and you could put your money in and pull the thing, and you could get your punch card so you could go to the keypunch and punch your cards. That's how students got cards to use for their classes. [laughs]

FIDLER

Those sound like repurposed cigarette machines as much as anything.

KLINE

They were repurposed candy machines.

FIDLER

Candy. Okay.

KLINE

So one of my friends—I forget who it was; might have been Mike Bablusi [phonetic], I don't remember—he was saying, “You never got me a picture of that.” Because he had come to UCLA and he saw this and just was in hysterics. But for me, it solved that problem, but it opened challenges. And, of course, the whole ability to use email to send things, and those really opened up new things for me.

FIDLER

Were other people using—because what you're describing with the IBM 360s, you're using the IMP to make a local area network or you're connecting to—

KLINE

I was actually using it as if it was a local area network. IBM had remote job-entry stations where for a certain amount of money, you could get a printer and a card reader, and they could be connected by a phone line, and you could submit jobs on those and get your printouts there. And they may have had a couple of those at UCLA. But for me, this meant I could easily do stuff from—and I took a Teletype home, and so from the late sixties—I don't remember exactly what year, but about '68, I had a Teletype at home, and so I could submit jobs, look at stuff, look at my email from home. [laughs]

FIDLER

That's interesting. How prevalent do you think it was for people to have at-home access to the host system?

KLINE

Very rare. This predated the Silent 700, some of the first portable terminals that people—this was a pretty hefty device. But there obviously were people who had terminals at home. They had Teletypes, and this predated the IMP-size and the Trash-80s, and so you couldn't just easily use one of those with a modem.

FIDLER

What kind of work would you do from home, just anything that you would do from UCLA?

KLINE

Well, I could check on the status of the system. I could type. If I had homework to do, I could type in the homework. I could send email and I could just play. There was a pretty good chess program at MIT, and there was some other things. And I could show off. People would come by the house, or apartment at the time, and I could show off what I was doing.

FIDLER

So it sounds like you could have even been online like a reasonable amount of time. In the early seventies, for example, you add up work and home, you're on the ARPANET quite a bit.

KLINE

Yeah.

FIDLER

Were there ever terminal rooms at UCLA where you'd have—

KLINE

Yeah.

FIDLER

Do you know how early those came in?

KLINE

The computer facility had some terminal rooms in—they had two types. There were some they used for the staff. I don't remember if they ever had any for the students. Late sixties, they had the IBM 2260 terminals and later the 3270 terminals. We didn't really have any terminal rooms for our facility. There were a few terminals in the computer room, in 3420, and by the early seventies, some of us had terminals in our offices.

FIDLER

From those terminal rooms, would you have been able to get access to a host machine connected to the IMP to get on the ARPANET through them, or was it just separate systems?

KLINE

I would get from those terminal rooms—the terminal rooms for the computing facility went to the 360, and not everybody from the 360 could get out to the IMP. That was somewhat restricted for a long time. If you had access to the SIGMA 7, you could get out to the ARPANET, but not everybody had access to the SIGMA 7. Now, there were some people who got access to the SIGMA 7. There were some students from a local junior high school that came by and wanted to see if they could learn how to program. They had a little club. They called themselves the Resistors, and we let them use the SIGMA 7. Whether we should have or shouldn't have, I don't know. And some of them have gone on to bigger and better things. One of them is pretty famous; his name is Steve Kirsch. Steve Kirsch, he'd gone to MIT. I don't know if he ever graduated, but he started a company called Mouse Systems. He invented the first optical mouse. Then he started a company called Frame Technology and developed FrameMaker, which was eventually bought by Adobe. He did a company called Infoseek, which was a precursor to Yahoo and those kind of things.

FIDLER

I remember using that.

KLINE

It was one of the early search engines. He's done several other companies since then, and so he made several hundred million dollars, but he started his programming on our computer at UCLA. [laughs]

FIDLER

And let's say I am faculty or a student at UCLA and I've got access to a terminal room in the early seventies. If I wanted to get access to the ARPANET, do I have to go through someone at 3420? Is there a process that I apply? Is it informal? How does that work?

KLINE

If you even had heard of the ARPANET, you would have probably had to come by and talk to Len or somebody, because, first of all, most people would have never heard of it, and other than the staff at the computing facility that knew about the ARPANET were maybe working on trying to get their software for it working, or our people, you couldn't get at the—they weren't going to let—and ARPA wasn't just going to let anybody jump on the ARPANET.

FIDLER



I think it was probably the early eighties or late seventies, I'm not sure, but Elizabeth Feinler was talking about how they'd have people—you know, by that time when it's better known, they'd have people who would do any job at all in a computing facility just as a side benefit, which was their main reason, they'd get ARPANET access. I don't know if that ever showed up by the time you were leaving.

KLINE

I don't remember seeing that event at UCLA, but then again, I always had ARPANET back then, so I never had to think about that. [laughs]

FIDLER

Were you ever called an Arpanaut?

KLINE

Not that I know of.

KLINE

Briefly, you mentioned using the IMP to interconnect computers. Did you get a sense that that was going on at other sites? One of the insights we have into that is the amount of zero hop traffic that was detected on the IMPs.

KLINE

Well, there were clearly places that—let's take ISI. They had several PDP-10s, some of which they used locally, some of which they actually—the main reason they had was for people on the ARPANET to use. Basically, ARPA said, "Set up a bunch of PDP-10s. We're going to use them." In fact, at one point when they were basically stopping funding the SIGMA 7 and were putting in the ANT system, they said, "You guys don't need a computer anymore. You can just use the PDP-10s at ISI." There was ISI-A, ISI-B, ISI-C. Those were all PDP-10s. And so clearly those were only on one or two IMPs. I think they only had one. Maybe they got a second one at some point. So there must have been a lot of local internal traffic there.

FIDLER

Let's jump to ANT. I don't know if this is too specific, but before, we mentioned the local mail as compared to the email. Compared to your email use where you'd be talking to people at other sites, would you be spending much time exchanging messages with people that were either on the SIGMA 7 or, say, the 360 once it was connected to the IMP?

KLINE

Not much time prior to—probably by about '73 or '74, our groups were exchanging—we were spending a lot of time emailing each other stuff, but pretty much only for the purposes of—well, there were a couple different purposes. We were emailing—of course, if we had a paper to submit somewhere, we'd send somebody a copy of the paper that way, but just if we were going to have a meeting, you know, "I have the following idea. What do you think?" We'd send that by email, whether it was single system or multiple system.

FIDLER

Did social functions or informal stuff ever work into that?

KLINE

Did social functions ever work into that. Well, I would certainly communicate with my wife that way. There were clearly times when we would say, "Hey, you want to go out to dinner?" You know, somebody might send an email to that effect. When I was at CISCO

in the late nineties, we all had pagers. They were alphanumeric pagers. They didn't have a keyboard, but they would display not just a phone number, but alphanumeric message, and you could send a message to the pager from a computer by sending the right email string, or you could call this 800-number and tell an operator what you wanted to send. So I remember I was at a conference and we were paging each other. "You want to go to dinner?" And I'd get this text message on my little pager. This is pretty much pre-cell phones with messaging, but we were using it like that. "Yeah." I'd call the 800-number. "Yeah, let's go to dinner. Where do you want to go?" And so he'd get a thing, "Sure. Let's go to dinner. Where do you want to go?" And then I'd get a little thing, "How about we meet at the hotel." [laughs] So I definitely did that in the seventies and eighties, but I'm not sure whether you want to call that a social function. UUCP, which was a sort of separate development, that was UNIX UNIX COPY, started out as copy files between one UNIX system and another over just dial-up telephone lines, evolved into being able to send mail and messages that way, evolved into generating Usenet mailing groups, where there were lists and people could submit messages into a list, and other people who were getting those messages would see them. And those lists covered everything from topics on computer science to all kinds of stuff, a lot of porn, but they ended up—there were certainly lists on restaurants and on dating and this and that. Those got used. That's not an ARPANET-specific or Internet-specific thing. That was sort of a separate development that evolved into a way of communicating that you could sort of say was a social thing.

FIDLER

I guess that would have been really early on in its development when you were leaving UCLA. Did you ever participate in that?

KLINE

In UUCP?

FIDLER

Yeah.

KLINE

Yeah. I used it, yeah.

FIDLER

Following up on how you'd use the ARPANET, let's say between 1970 and the late 1970s, in particular with communicating with people, you've mentioned using it to talk to your wife. Obviously you're using it for professional reasons. Was there a change in the composition between, for example, professional and personal over that decade?

KLINE

It was almost all professional. In terms of using the net, it was pretty much the ARPANET, was both for the actual measurements or communications of protocol questions or exchanging documents on protocols or whatever, drafts of papers. By the late seventies, we were submitting drafts of papers for review by ARPANET. I was doing some consulting, and because some of the people I was consulting with had ARPANET access, we could exchange drafts of documents or notes that way. But in the seventies, other than what I was doing specifically on the ARPANET for the research on the ARPANET, like testing NCP protocols or whatever, it was mostly email. And in fact, people say, "What's the killer app for this device," or that device or whatever, the killer

app for the ARPANET throughout the seventies and into the eighties was email. You've probably heard that before.

FIDLER

Did your experience, do you think, line up with those around you in terms of how you used it?

KLINE

Uh-huh.

FIDLER

Did you ever use the online system at SRI, or were there other sites whose services you made particular use of?

KLINE

Used the online service at SRI partly for fun. It was kind of a fun toy to play with, partly because that's where we gradually were moving all the online documentation for the ARPANET. The RFCs we were gradually moving there. Of course, that was my first experience with a mouse. We got a couple of devices called Imlac PDS-1s. Imlacs were—it's spelled I-m-l-a-c. PDS stood for, I think, Program Display Station, Model 1, and this was a device about the size of a desk which had a display on it, a keyboard, and a mouse, and we had plugged into it a five-finger keyboard. And software from SRI, so we basically had a replica of the terminals they had at SRI with software that could connect to the ARPANET so that we could log into the online system and see the display the way you would there. And for fun, it was kind of fun to try to type using the five-finger keyboard than to take your hands up and actually use the real keyboard. [laughs] Because between the three buttons on the mouse and the five buttons on that five-finger keyboard pad, you had eight buttons, which means you could type all 256 codes. It would be kind of awkward to do so. Some people got pretty good at it. There were people who actually could type twenty or thirty words a minute that way. The coding was—if you know binary, one, two, three, four, five, six. A was one, B was two, C was three, D was four, E was five, and so on. And then the three buttons on the mouse were shift buttons, for uppercase or to go into the numerics or whatever. Anyhow, it was kind of fun to do that, but I didn't use it a lot for that. The nice thing about the online system was, of course, it had hypertext, so you could be in a document, you could link to some other document.

FIDLER

Did you also have the same tonal feedback that they had at SRI on that local system?

KLINE

I don't remember. So I used those, but it was just a couple more terminals for us, and it was kind of fun to use, and it gave us a little better experience of what the online system at SRI was like. I used Multics, but mostly just to learn about Multics. And the same with the PDP-10s, DEC-10, DEC-20s, I used those mostly just to learn about TENEX. They did have some good programs. This is pre-UNIX days. They had good email programs. Realize you didn't have anything like Outlook or Eudora, any of the email programs you're used to today. You had character-based email programs designed for character-oriented terminals, where you'd type a command and it would show you the headers, type out the headers of the messages. Even if you were on a display, you'd see the lines with the headers of the messages. Then you could type a number to show you that message, you know, pre-mouse. So, you might message one and its header, message two and its header, and message three. So if you wanted to look at message three, you'd type a three,

and it would type out that message, and if you wanted to replay, you'd probably an R or something like that. Some of the email systems on TENEX were a little easier to use until UNIX gradually evolved its set of email. There was a text editor that had been built for one of the PDP-10 operating systems at MIT called TECO; stood for Text Editor and Corrector. That was kind of a fun little tool to use to type, and we, in fact, eventually got the source for that and ported it to our PC. So I had a TECO on my PC that I could use to type documents, and I found it more convenient to use TECO than any of the text editors on the PC at the time.

FIDLER

Did it help to have local contacts at these other sites? For example, when you'd be learning TENEX or Multics, was there sufficient documentation to get you started or did you really have to know people there?

KLINE

There was sufficient documentation to get started. Most of the commands had a help function, and they were pretty intuitive. If I really got stuck, I could contact somebody at those sites, but I tried to avoid doing that, because I knew they didn't want to be bothered with—

FIDLER

About what years are these that we're talking about?

KLINE

Well, using the PDP-10s would be mid-seventies. By the late seventies, we had UNIX.

FIDLER

How about the Multics experimentation you were doing?

KLINE

That would have been early seventies.

FIDLER

And did you have to do anything to get local access at this, like an account?

KLINE

Had to get an account, but because we were developing the ARPANET, the same guys who were developing the NCP or the ARPANET stuff at MIT or at ISI or whatever, would automatically just sort of give us an account if you wanted one and you were in the community. Whenever I use that word, I think of these old movies where they referred to the intelligence community. You know, "The person's got to be a member of the community." There was a movie with Robert Redford, Three Days of the Condor, and he's talking to the guy from the CIA, and the guy said, "Well, the guy's got to be a member of the community." He says, "Community? You guys are kind to yourselves." [laughs] But I mean, it was a community, so if you were sort of a member of the community and you had any justifiable reason at all, they'd give you an account.

FIDLER

Did you see that changing over time when you were there? I know, for example—well, there were experiments with access controls, at least to the TIPs in the mid-seventies. By the early eighties, the NIC was looking to implement more TIK and then TAC access. Did you see any precursors to this with even just hosts?

KLINE

Well, at ISI they had PDP-10s that ARPA was funding specifically for people to use, but you had to have an account. We didn't have a TIP because we had either our own system

or we got this ANTS computer, which was basically a TIP. It was a PDP-11 connected to our IMP with a bunch of terminals on it. So you could think of it as a TIP, except it wasn't too much longer after that that we started running UNIX on it and using that to get—and that's because ARPA didn't want to fund our SIGMA 7. They saw it as a dead-end machine, which they were right. They didn't want to spend the money. They didn't see why they should buy us a PDP-10. They didn't see that we had any need for it. They knew we were doing interesting research, but they didn't see that our research needed a large local computer. Yeah, we needed access to send email and to type documents and other such things, but they didn't see any reason why we should be reinventing those tools.

FIDLER

Bigger picture on the same theme. If you wanted access at UCLA to either the computing resources or then more specifically the ARPANET throughout the 1970s, how would you do that? Presumably you'd need to start with an account on a host.

KLINE

To get access to computing resources, if you were most students, the only computer resources you could get to would be the ones at the main computing facility, and you got those because you had a class or because you joined the Computer Club, which got you the ability to run a couple of jobs a day remotely on that. Over time, we got from ARPA half a dozen VAX-11/750s running UNIX, and we started to use those partly for our research in distributing UNIX systems, but partly a couple of them were used in the computer science department, not so much for classes, but for some students. So if you had a justifiable reason, you could get access to them. And in terms of getting out to the ARPANET, we sort of tried to restrict that to the extent that there sort of needed to be a good reason. Didn't have to be a very good reason, but we didn't want to get ourselves in trouble that we're letting people just play around on the ARPANET too much that really had no reason to be there.

FIDLER

So if there wasn't a micromanagement by ARPA or later the DCA or NIC of exactly who could be on it, it does sound like there was some kind of final accountability where if you just let everyone one, someone would get in trouble.

KLINE

I'm not sure in trouble, but we just didn't want to worry about it. So pretty much anybody who could use the SIGMA 7 could get to it, but there weren't that many people that could use the SIGMA 7. Later on when we had the VAXs in the late seventies—I'm trying to think—not all of them were connected to the ARPANET, and I don't remember what access control, if any, we did to it.

FIDLER

Was there a kind of guardianship, though? There is a sense that you couldn't just let anyone on, even if those decisions weren't being directly monitored.

KLINE

They weren't being directly monitored, but we just sort of felt that, you know, we want these systems to have a certain level of performance for whatever purpose the computer was for. So, for example, on some of the VAXs, we were booting them up and down a lot because we were building an operating system on them, so we were crashing them randomly. So we didn't want people depending on those that—and then we had some we

kept all the time so we could actually do our own work and do our email. And those computers weren't all that powerful. You know, a VAX-750 could handle five to ten people, but it could get pretty slow. So I think it was less getting in trouble as much as we just wanted to keep the computers that were for our purpose usable by us. [laughs] If you have a car that you expect you can use all the time, but you leave the keys and anybody who wants to use it can use it, and you go out to use it and it's gone— [laughs]

FIDLER

Did that relate in any way to keeping standards for who would be a part of a certain community? For example, by—well, this is way forward, but in the early nineties, you had September when the new students would come on, and eventually that condition became permanent. But when there's the influx of users who weren't familiar with the standards, couldn't really be trusted to behave themselves, they have to be resocialized, that would take a while. Was there anything analogous to that back in the seventies?

KLINE

Not too much, because the community was small and—I'm trying to think. I can't specifically think of an analogous thing in the seventies. Of course, by the late eighties, you start to have PCs and things, and there wasn't a big problem with viruses and malicious behavior in the seventies or even eighties, and you didn't really have—other than those of us who knew about the networks and could actually get to them, most people's experience with the network was things like AOL, CompuServe, or Prodigy. Prodigy and AOL—CompuServe eventually died. AOL's still around, but only in a vestige of itself.

FIDLER

So getting the ARPANET in the seventies, then, you'd be doing work specific to those fields, would knowledge of it also be transferred informally? Like if you had a friend in a different department, maybe you'd tell them about it?

KLINE

If I had a friend, I mean, I might brag a little. Somebody might come by and, you know, "Want to go to lunch?" "Sure." "Let me show you thing we're doing." Or there would be, if I felt there was somebody that, "Oh, you could really get some benefit by communicating with such-and-such. Maybe I can get you an account on our system." Or in my case, at least on the SIGMA 7, I could just create an account. [laughs] But it didn't happen all that often. At AT&T they had developed some text-to-speech software, and there was a device that you could buy called a Votrax, which if connected to the PDP-11 with this text-to-speech software—it plugged in where a terminal would go, okay—it would speak at you, okay, and it could be plugged into a modem. And we got a special kind of modem that recognized touch tones, so one of the guys wrote a little program and we had this one phone line with this on it. So I could call the PDP-11 on this line, and it would answer and it would say, "UCLA," whatever, "log in," and I couldn't speak to it. I could type from a touch tone, okay, and the way it worked is you typed one of the letter buttons and then a 1 or a 2 or a 3. Nowadays, people when they text, you just push the button a bunch of times, but that depends on timing, and you might not do it fast, whereas, let's say there's A, B, C, if you push that button and then a 1, that would be an A, that button, and a 2 would be a B, that button, and a 3 would be a C and so on. And by combinations of other characters, I could do shifts and things, so I could type anything that way. So I could log in and I could type C, S, K and it would say C, S, K. Actually, I

think it may have said CSK. I'm not sure. And then it would say "password." When you logged into UNIX, what you got a pound sign, I think it was. It would say "#," or something like that. So it would be awkward to do much, but I could pick up my email anywhere in the world. [laughs] This is late seventies. I could pick up my email if I was willing to spend the phone charges from any touch-tone phone anywhere in the world.

FIDLER

Was there a relationship between projects like this that sound like—maybe I'm misreading this, but sound like kind of a pet project or a hack?

KLINE

This was a hack that they were taking advantage of software that had been built by AT&T and was built into UNIX, and a hack that a guy wrote and put on our—so that just for the fun of it, you know, I'd like to show that one off at parties sometimes. "I think I'll pick up my email. Anybody got a phone?" [laughter] And this was pre-cell phone. I mean, because we're all used to picking up our email on our cell phone today, but in the 1970s I could pick up—first of all, people didn't even know what email was, but in the late 1970s, I could pick up my email from any touch-tone phone anywhere in the world. [laughs]

FIDLER

That's a really good party trick.

KLINE

Yeah. [laughs]

FIDLER

Were there instances of similar hacks where it would spin off official work or maybe eventually be added to a project that someone was doing, like as part of their formal research?

KLINE

I can't think of anything that we did that way, but I know there were things at other places where somebody decided on a graphics thing, and then that became part of work they were doing on graphics, or maybe somebody did something so he could draw something, and that became an add-on to a word-processing package so they could add diagrams to a—but I'm trying to think if there was anything we did that—not that I can think of.

FIDLER

We've talked a bit about email. I notice that in the early seventies, all the UCLA people had ISI email addresses.

KLINE

Right.

FIDLER

Was that because it was a better system at ISI?

KLINE

Right. I mean, we really didn't have a decent email system with convenient features. The ISI, the PDP-10 email system—actually, I should say the TENEX email system, had a convenient set of commands to look at your email, to respond, to file it away, to save them. We didn't have all that capability. Maybe even had spelling correction; I don't remember. So we had a pretty rudimentary email system on the SIGMA 7, so it was just a lot easier and better to use the tool on ISI, so we all got accounts. And by the late

seventies, before the PDP-11s were really connected to the Internet, but we had the ANTS system or the PDP-11 later, the whole intent by ARPA was that we would use ISI for our computing access.

FIDLER

So when people talk about—you mentioned this before—UCLA gets PDP-11s instead of 10s because you're supposed to use PDP-10s on the network, was it ISI in particular?

KLINE

Yeah. I may have had accounts at BB&N, but it was only just for testing purposes. Pretty much there were only a few sites that ARPA was basically paying to provide computing resources for other people, and one of them was ISI. ISI had, oh, maybe five or ten PDP-10s, of which several of them were just for their own use, for their own projects, and several of them were for the purposes of providing resources to the ARPANET community, including—I wouldn't be surprised if the ARPANET project managers probably had their accounts at ISI. I mean, ISI, for all practical purposes, was, and maybe it still is, a wholly-owned subsidiary of ARPA. You know the history of ISI, right?

FIDLER

It was created at USC in the—was the late—

KLINE

Well, there was a couple of guys, Keith Uncapher, Bob Balzer, some other—these are guys from Rand and maybe STC. I don't remember. And they wanted to set up a research lab, and ARPA was really interested in funding them, but ARPA said, "We can't just give you money to a new startup. The government bureaucracy won't let us do that. You've got to be affiliated with somebody." They came to UCLA and said, "We'd like to be affiliated with UCLA." UCLA said, "Well, this is a big enough deal." We're talking about, like, \$5 million a year or \$10 million. It was a big amount of money. That kind of a research grant and contract, that's going to require Regents' approval and whatever. We'll have to see what the Regents say." It was going to take six months or whatever to get approved.

FIDLER

Sounds like UCLA.

KLINE

Well, it's more the UC, whereas USC, being a private college, was perfectly happy to get an institute paid for by somebody else with world-class computer scientists. And you've seen the buildings at Marina del Rey. I assume that you've driven by them, if nothing else.

FIDLER

Yes.

KLINE

So those were new buildings at the time, and they took several floors of these new buildings, and so they got some pretty nice office space in a good location. So USC says, "Sure." So it became the USC Information Sciences Institute rather than the UCLA Information Sciences Institute because UCLA couldn't move fast enough. It wasn't because USC courted this. It wasn't because they wanted to be affiliated with USC. It was UCLA was going to take six months or a year to make it happen, and that's more due to the UC bureaucracy.

FIDLER



On the topic of this advanced research, in the early eighties, one of the reasons that CSNET, for example, was being developed is that there was a sense that people on the ARPANET were gaining all this advantage from being able to collaborate easily and communicate, and, in fact, they had started to become their own community. It was even said that they were starting to collaborate and work less with people that weren't online, that weren't on the ARPANET. Now, I don't know how much of those things are true, but I'm curious if you observed anything like that.

KLINE

I don't specifically know that that was true. I don't know that it wasn't true. Certainly most of the people I collaborated with were on the ARPANET one way or another. By the late seventies or early eighties, Jerry Popek was consulting up here in the Bay Area a lot, and then when he got a couple big projects, he needed some help, so he asked me to come and help with that. Well, the people that we were consulting with were guys from Stanford, and they were all on the ARPANET, so we were able to communicate via the ARPANET for our consulting stuff, at least for sending emails and stuff, and most of the people I needed to interact with for one reason or another were connected to the ARPANET. Now, if you weren't in computer science, you probably weren't connected to the ARPANET, but even if you were in computer science, related to computer science, not every site—you know, how many universities are there and how many were connected to the ARPANET? Not that many. And then even within those, there weren't people who were connected. So I'm not sure exactly whether there was a formal policy at ARPA. If there was, it never got communicated to me, you know, "Here's our rules. Don't let anybody on except for—." But clearly, only those machines that were connected to the ARPANET would have people connected to the ARPANET, and the machines had restrictions in the sense of they only had so many cycles. So people tended not to let a lot of people use their machines if they didn't have a good reason to be using their machine, because they wanted to do their own work. So there probably was some of that. I didn't notice it, but I believe it probably was true.

FIDLER

You've mentioned a few times that a lack of computing power was something that influenced how many people would be let on to a particular computer and thus on to the ARPANET. I hadn't heard that before.

KLINE

Well, you have to understand in the sixties, a large computer did about a million instructions a second. Our SIGMA 7 did about a half a million instructions a second. By the seventies, the big large—I'm talking about the \$10-million computers, did two or three million instructions a second. Well, my cell phone does like a billion instructions a second. And we're talking about things that are a thousand times faster. It doesn't take much work to use up all the computing power. I mean, when the system was idle, no one else was on, I could type my stuff and do some things. When there were enough people on the system, five or ten people doing it, I could type a line and maybe have to wait thirty seconds before it responded to me. And if I was using one of the programs where the echoing was done by the software, I could type a character and have to wait a couple of seconds before I saw the character. So not having enough computing resources was a big deal. We're all spoiled now because we're used to fast PCs, fast mainframes, fast webs. Generally speaking, we're frustrated. We go to a website and we put in a

transaction, like, to see our bank balance, and we're frustrated because it isn't instant. It takes a whole second and a half, and we say, "God, this thing is so slow." But you've never seen slow if you haven't had to use some of the computers that some of us old-timers had, where you were happy if you got a response in a minute. One of the things that we used to do when we were first testing Telnet, we'd see if we could Telnet to a site. So we'd log in at that site. Then we'd go from there. We'd use its Telnet to Telnet to another site. And you might go to four or five or six sites that way, from A to B, from B to C. You'd type a character and it's going from each of these sites, and it might take five or ten or fifteen seconds before you'd see the character, because it'd have to go to each of these sites and be processed. And finally get to the last one, who's the one that's echoing back the character, and then it'd work its way back, and you'd see the effect, and you could see the effect if the sites were busy. So, yeah, computing resources—sometimes I'll be using my PC, which this particular one is a uniprocessor. It's not a dual processor. It's only a single core, but it's 3 gigahertz, so it's reasonably fast. But, you know, sometimes I'm doing something and it just seems like it's so slow. We're just so used to everything being lightning-fast, but that wasn't always the case. Imagine the freeways when they were being built in L.A. and the 405 these days. [laughs]

FIDLER

I wonder if the difference in response times based on the time of day or, more specifically, how many people are on the system would have impacted your or other people's decisions when to use the computers.

KLINE

Yes. In fact, you'll read stories about famous researchers working in the middle of the night because that's the only time they could get computing time because it was busy the other times or because it was too slow to use other times. For example, there's a famous mathematician, His name was Edward O. Thorp. He wrote a famous book called Beat the Dealer, where he did the mathematics of Blackjack, the first guy to really do the mathematics of Blackjack. I think he was at MIT. I'm trying to remember. But he used to have to use the computer at night to get time when he could use the computer to run all his calculations of what the actual odds of Blackjack were. Yeah, there were times when I'd say, "Well, I need to do this. It's going to take about three hours to compile that, so I'll come in and start at eleven and come back at two a.m. and get the results."

FIDLER

Do you think people ever did that as a courtesy to the rest of the users?

KLINE

Some people did it as a courtesy. Others did it because it was the only time they figured they could get—and then, of course, if you were doing things that might crash the system, then you often had to do it or let people know that you might crash the system.

FIDLER

And with the Blackjack calculations, that sounds like a batch machine. Are you comparing that to time-share—

KLINE

He was running a program. He may have been doing it like you'd think of as a batch, not on a terminal, but he was sitting there waiting for the results while, you know, he was the only one using the computer. I mean, there are machines today. People have to schedule

time on a gene sequencer or whatever, because it might be busy and they want to do something that's going to take a long time.

FIDLER

Let's keep going on hardware and move to ANTS, the ARPA Network Terminal System.

KLINE

Right. It was developed at University of Illinois and, again, the purpose was that ARPANET said, "There's a lot of sites that we really don't need to keep having them develop their own operating systems, their own things. They don't really need a big computer. Let's standardize mostly on PDP-10s writing TENEX. We'll just give them accounts. So they need a terminal, and they can either—which is more cost-effective, putting in a TIP or putting in an ANTS?" And I think they decided, especially if the place already had an IMP, that it may have made more sense to buy a PDP-11, which could be bought really pretty cheaply. I don't remember the exact dollars. And running this software from University of Illinois, and plug half a dozen or a dozen terminals into it, and use it to Telnet FTP, and put a printer on it and—

FIDLER

Was the idea to have more terminal functions than the TIP?

KLINE

Yeah. For example, under ANTS, I think you could have multiple sessions open at one time. I think you could have a connection to this site and a connection to that site and switch back and forth. I think on a TIP if you open a connection, Telnet to one site and you wanted to go to another site, you'd have to close that one. Also, I don't think a TIP did FTP and some of those things, whereas this did.

FIDLER

In the BBN completion report, which I think was published in about '76, but going from '69, well, a little before that, until '75, they talked about the ANTS as being maybe too ambitious and ultimately something that didn't move forward.

KLINE

It wasn't very successful. There were some number of them, but it didn't have all that much more capability than a TIP. It could have, but it didn't. And by the late seventies when UNIX was beginning to get popular at least at universities, because AT&T would license it to universities for basically nothing, they weren't commercially licensing it, but they would license to a university, basically nothing. Then ARPA paid Berkeley to develop a version for the Berkeley BSD UNIX on the VAX. Then that seemed like a more cost-effective—you could do more stuff. You could handle more users. They had more capability. They could handle more printers. They could do things besides just be a dumb terminal. So it didn't quite make it. The TIP made a lot of sense at its time. If you look at 1973 or '74 when the TIP came out—I don't remember the exact year.

FIDLER

I thought it was even earlier than that. I think '71.

KLINE

Don't think the TIP came out in '71.

FIDLER

Well, you'd know, not me.

KLINE

I mean, because the ARPANET, the first IMP was '69, and then the next bunch were in '70, and I don't think they had the TIP in '71. They had the 316 IMP, which eventually became the TIP, but it was a cheaper IMP, didn't handle quite as many hardware interfaces. I think it only handled two or three phone lines instead of four, and I think it only handled two or three hosts instead of four, but the 316 was lower-cost hardware and probably just about as fast as a 516. You can look up when the TIP came out, but it's I'm guessing '73. For places like ARPA, where all they needed was some terminals and didn't want to have any kind of computer, I mean, even an ANTS machine required power, required air conditioning, required a little more than a TIP did. TIP needed power and air conditioning, but not as much as a PDP-11 did. I don't know if you've seen it. They have one at the Computer History Museum.

FIDLER

An ANTS?

KLINE

No, a kitchen computer.

FIDLER

Oh, I've seen a photo of their kitchen computer.

KLINE

Yeah. It's a 316. It's a Honeywell 316 sold by Neiman-Marcus as a kitchen computer for like \$150,000 or something. I don't know if Neiman-Marcus sold any of them. I don't know. In theory, you could keep recipes on it and this and that, and it didn't go anywhere, but the point is that— [laughs]

FIDLER

Didn't the kitchen computer require like a two-week course or something before you could use it?

KLINE

Something like that. [laughter] I mean, it was some incredible thing, but I always found it funny because it was a Honeywell 316. Have you been to the Computer History Museum?

FIDLER

Yes.

KLINE

Oh, that's right. I know you've been there, because you've met with what's-his-face from there.

FIDLER

I met with Marc Weber—

KLINE

Marc Weber, yeah.

FIDLER

—and I actually had the pleasure of meeting John Hollar, the CEO, and I finally got to talk to Liz Feinler in person.

KLINE

Yeah. It's funny, because I always knew her as Jake. I knew her name was Elizabeth, but it was always Elizabeth Jake Feinler, and everybody called her Jake.

FIDLER

I think it's been a source of confusion for a lot of researchers.

KLINE

Yeah. And the first CEO of the Computer History Museum out here was John Toole, who was an ARPA program manager, and last I knew, he was at Google. I'm trying to remember if—do I have him in my book? I might have him in here. Let's see. Contacts, Toole. Then John Hollar took over and has really been expanding it. John Toole is—I guess he's not at Google anymore because—let's see. John Toole appears to have left Google, was CEO of the Computer History Museum, was research program manager at Google. Okay. He left the Computer History Museum, I'm guessing, around 2006, '07, something like that. It's kind of a fun place, especially for people like me and my generation, because we used half the things that are there. [laughs] I mean, it's not just history. I mean, they've also restored a few things. They have working 1401, and the 1401 was one of the first transistor computers that IBM sold, and it had several different purposes. But one of the things it was sold a lot for was you had these big number-cruncher mainframes, the 7090 Series, 7094, that were batch machines, and it didn't make sense to put a card reader and a card punch and a printer on those. So you needed a separate machine to take decks of cards and put them on tape, and then take the tapes over to the 7090, which were fast because they could be read faster and processed faster, and then you'd put the output on a different tape. Then you'd take that tape back to the 1401 and use it to print the listings and maybe punch any decks that needed to be punched. So a lot of 1401s were sold merely as frontends for the 7090s. In fact, they eventually made a switch, so rather than physically moving the tape, you'd put the tape up, you'd load the stuff onto the tape on the 1401, then you push this big switches which would switch the tape from being connected to this computer to that computer, a big, clunky switch with—you know, there's like fifty contacts or a hundred contacts in there that were switching.

FIDLER

Speaking of the specifics of using hardware, using an ANT versus using a host versus using a TIP, I'm curious about your experiences between the three, and I'm also, as a side point, curious if the ANTS were ever used for remote access the way the TIPS were.

KLINE

Access in what sense?

FIDLER

Dialing in from the road, for example.

KLINE

Yeah. ANTS you would either connect terminals to or connect modems to, as we did—the advantage of using a host was that, number one, if I needed to write a program for some function that wasn't built into the TIP or built into the ANTS, I could do that, but also I could store files locally and FTP them locally and maybe print them locally or do something else with them. I had a little more capability. So as long as the software was as convenient, I'd prefer host, but the hosts typically were a lot more money to maintain. It was a lot cheaper to build a machine that could handle sixteen terminals that was just doing remote access than it was to run a machine that could actually run software of the users for the same sixteen terminals. That changed over time as prices came down, but we're talking about the seventies.

FIDLER

And at UCLA, at least, was there remote access over modems, for example, to the SIGMA 7?

KLINE

Yeah. We had only a few. We had three or four dial-in lines plus this—yeah, three or four dial-in lines, yeah.

FIDLER

And the kind of regulation of access that we talked about before, would that have applied to the host machine and then the ANTS at UCLA? Is that getting access to dial into there would have just been the same process as getting access period?

KLINE

Those computers, the SIGMA 7 and the ANTS machines, were controlled by the computer science department, in particular by our contracts, so it's different than a student getting access to the mainframe or whatever, different than a student getting access to the—you know, any student could get access, if they had a justifiable need, to the computing facility, whereas these were our computers for our work. We could do what we wanted with them. Now, that building, Boelter Hall, was built—several of the floors were built to actually be labs, so they had built-in under-floor air conditioning at the level of the room, rather than having—if you'd been to computer rooms at typical companies, where the computer room was, you actually had a ramp that you had to go up, because the under-floor air conditioning took this much space, and the building hadn't been planned with that in mind. A long time ago, all the heating and air conditioning for the entire campus was provided from a plant on Weyburn or whatever, called the steam plant. They had a plant that pumped steam underground to steam tunnels all over the campus, which then was used to power heating or air conditioning.

FIDLER

There's still some kind of steamy building around the Western Boulevard-Strathmore area.

KLINE

Yeah. Across from Parking Lot 8, there used to be a big steam plant there that used to have big towers and steam always coming out of it, and that was the plant that provided steam for most of the buildings on the campus. That's all been replaced by more efficient systems and whatever. And it was always an engineering prank to break into the steam tunnels and crawl around. [laughs]

FIDLER

That's still an activity today.

KLINE

I'm sure people get in trouble for it, but—

FIDLER

So that was an Engineering College—

KLINE

That was a common engineering prank, yeah. Things like that, I think, went on in engineering departments at every university. [laughs]

FIDLER

Somebody at UCLA in the 1980s actually plotted with a computer all the access points throughout all the buildings on campus, with the room numbers, even.

KLINE

Yeah. And then other engineering pranks, at UCLA, at least, were making master keys so that we didn't—like I had a master key that somebody had made for me so I didn't have to carry around a big pile of keys for all the different rooms I had access to. [laughs] And then another thing was making master card keys so you could get into lots you weren't supposed to. Now, you still could get a ticket if you didn't have the right decal on your car, but it made it a lot more convenient sometimes. [laughs]

FIDLER

Before we segue to your security work, though, one of the last things I just wanted to ask. You mentioned processor cycles as something that impacted the time that you would use the systems, also just the fact that you're interested and frequently there. I know, for example, there was backups that were done at the end of every single day. I think Anita Coley handled at least some of that. Were there other very specific tasks that needed to be done based on system requirements that would impact your schedule?

KLINE

I was the one who started making sure we created backups. I set that all up, and then I had Anita doing backups. I don't remember what time they were done, but we would do backups onto tape. When I had LOCUS computing, we'd also do backups. We'd send those tapes offsite to a storage repository, because as we used to put it, "If the building burned down, well, it's going to take us a month or two to get some more computers and get a new place and all that, maybe three months, so we'll be out of business for three months." If we lost all this data, we were out of business, period. We weren't coming back. Anyhow, but the other thing is we also kept—you'd be surprised the number of times somebody would delete something that they wanted, and then they'd say, "I deleted this. Can you get it back for me from a backup?" [laughs]

FIDLER

So were there other activities that were also considered important that would demand a certain—

KLINE

Well, there was system maintenance from the hardware, and that would get scheduled. Sometimes it would happen out of non-schedule if the computer broke and you had to call the repair guys. But there was hardware maintenance. I'm sure there was, but I don't recall. Generally, you could assume the system was probably going to be up unless somebody was doing some kind of an experiment, and this was whether it was the SIGMA 7 or the PDP-11s later or the VAXs after that. By the time we had the VAXs, we had enough of them that things were loosening up a little bit.

FIDLER

What influenced your move over to security? Was this '73, '74?

KLINE

What happened was somebody was telling me that you couldn't build a secure computer system, and I said, "I don't see why not." "Well, you just can't." I said, "It shouldn't be that hard to build a secure computer system. Our SIGMA 7 system is pretty secure." It probably wasn't as secure as I thought it was, but it was actually pretty good, probably because it was small enough that I'd read every line of code in it. That doesn't mean there weren't any bugs. So I was just sort of interested in the field of what would it take to build a secure system. So Jerry Popek comes to UCLA, and I guess Dick Muntz knew I was interested in building secure software, and he said, "We've got this new professor,"

and his Ph.D. research was on computer security. I said, "Oh." So I met him, and we started talking about how hard it would build a secure—he said, "Well, it's really hard." I said, "Well, why? Why can't you?" And we talked, "Well, how do you know if it's secure?" And then there's the issue, well, how can you verify that it's secure? So we got into discussing what would it take to build a secure computer system, secure operating system., and the first question is what does it mean to be secure? Well, our definition was that, "Here's a set of access rules. You're supposed to be able to access this file and that file and that file, but not to be able access this file, that file. Can we be assured that the system guarantees that?" Granted, somebody else may be able to log in with your user name and password, but if they can't get through that authentication stage. Is there some bug in the system where they can trick the system into letting them have access to things? There's been a lot of bugs in UNIX and things where that's been possible and in various operating systems where that's been possible. So then we were interested in how would you design such a system and different models of how one might design such a system, and the research that had already been done on computer security, and Jerry was all up on all that. He had read all that research. I hadn't read it all at that time. So we started working on that, and ARPA was getting a little interested in that, and so we eventually got ARPA to fund us to do research in computer security. And there were people beginning to come, not only taking Jerry's class on computer security, but also there were some people beginning to come to UCLA because they had heard about Jerry, and Jerry was there teaching about computer security. So we got some grad students who were interested in computer security and started this project to build a secure UNIX system, which we got mostly done, but not completely. One of things we were actually hoping to be able to do was to be able to actually prove mathematically that the system was secure for some definition of security. We never quite got there, but we learned a lot in the process. Now, during that whole period of time, there were other people working on computer security, and other people were beginning to work on network security, in particular, encryption, and public-key encryption got invented and other such things. And we were interested in how could you use encryption both inside of a computer system but also on networks securely, and so we researched both of those issues and did a lot of research on that, published a lot of papers on that whole subject. Then at some point, as distributed computing started happening more, we got interested in how would you build a distributed system. And ARPA seemed to be less interested in funding computer security research, at least at that time, and more interested in distributed computing research. To the extent that they're funding security research now, it's partly due to the fact that there's all this stuff about cyber security and that. And also, like what Erik's working on, among other things, is if you wanted to guarantee that somebody in some third-world country can communicate and that the government can't block their communications, how would you do that, and how would you do in a way so the government can't track them? And so there's research on that that's he been working on for ISI. Those weren't big deals yet, because the world wasn't on the network in the seventies. [laughs]

FIDLER

Now, in 1979, you published with Popek, "Encryption and Secure Computer Networks," a paper, and in that paper, as potential security issues you mention line tapping, spurious messages, retransmission of previously valid messages, and disruptions as threats. And



I'm curious how much of this was hypothetical and how much of it was a concern that was maybe noticed or transmitted to you from above.

KLINE

Well, if you start looking at all the ways that—what can somebody do to a communication? Well, if you want to keep it secret, maybe they can tap the line and listen in on what you're doing. Some of these go back to military communications issues for hundreds of years or at least dozens of years. Another issue is maybe they even encrypted the message, maybe they can't read what you're saying, but maybe they can take a message you've sent and send it again. So they recognize, "Oh, this must be the message. We don't know what the actual text is, but we can tell this is the message you used to recall that launch, so we'll just transmit it again," hoping that it's—okay. Another issue is supposed they just want to take you down. Well, they'll just flood your network with traffic. Can you recognize the flooded traffic from the—it's called a denial-of-service attack. There was tapping, there was retransmission of valid, there was—

FIDLER

Spurious.

KLINE

—spurious, sending—you need to be able to test to detect stuff that's not valid, okay. There are ways to—the hardest one of those to deal with is denial-of-service attack. If someone is flooding your network with traffic, it's hard to get your message through. You might be able to detect the stuff that's not valid, but it's hard to get your—I mean, a classic example of this is some website that starts getting flooded with attempts to open connections. They don't know which ones of these are valid and which ones are not, so all their servers are being busy—I mean, Google gets those kind of attacks every day, and they have all kinds of systems to try to detect which traffic is valid, and they have complicated algorithms, like, gee, the Google China people probably should be getting most of their Google stuff from the Google China. Messages from China that are coming to Google U.S. have a higher likelihood of not being valid, so we should examine them more before we pass them on to our servers to clog up our servers. [laughs]

FIDLER

These potential issues that you list, was this largely coming from thinking about military command and control, or was it a broader concern just about disrupting networks, period?

KLINE

It's a broader concern about networks. We weren't the only ones to have come up with that list. There were numerous papers about what are the problems in communications security. Well, if I said to you, "What can people do to your communications?" You're going to talk on the phone to somebody. Well, they might listen in. They might steal the conversation in the middle and fake my voice and tell the person that, yeah, go ahead and charge that to my account. They might cut my communications so I can't communicate. I mean, it's not too hard to think about the kinds of attacks. Then the question becomes what can you do to protect against those attacks.

FIDLER

Okay. Now, at that time, by '75, BBN's making it possible to create these logical subnetworks. Some of them are encrypted. They have these private-line interfaces. There's key generator units.

KLINE

Right.

FIDLER

Was your work related to that in any way?

KLINE

Only peripherally, in that we knew of it and we knew the theory of generating key generators and private lines for the purposes of—I mean, the NSA had private lines and other military agencies had private lines between one office and another office with encryption unions, hardware encryption units, and there was issues of how [unclear] done. Do you do it manually, physically by having a courier? In the fifties, sixties, that was often done by couriers. Couriers literally showed up with the keys and put them into the device. But you wanted to be able to do that automatically, you wanted to be updated automatically, and yet be sure that the protocols for doing that couldn't be spoofed. I don't know if you've ever seen Dr. Strangelove or Fail-Safe, but you wouldn't want somebody that'll figure out how to spoof the fail-safe box. So there was a direct military need for communications security. Easier for ARPA to justify the need for such research. However, most such research was done by the NSA for their own internal use. They wanted to know how to break into communications encryption of other countries and how to protect their own communications, although they weren't so forthcoming about how to protect communications to the other agencies. They were more interested in how do we break into other countries' communications because that was really their primary focus. I have a friend who's pretty famous, Whitfield Diffie. You may or may not know the name. Among other things, there's an algorithm called the Diffie-Hellman key exchange algorithm. He was one of the inventors of public-key cryptography, lives in the Bay Area here, and over the years he's consulted for the NSA and whatever, but he's also pretty critical of them and a lot of the things they did. There was always speculation that the—at some point there was a need for a commercial encryption chip, so there was a standard created called the Data Encryption Standard, DES, and there were chips made for that. And there was always speculation that the NSA had weakened that standard. It's a complicated standard that involves taking blocks of things, feeding them into this algorithm. There are these various complicated lookup tables and translations, and there was claims, well, the NSA modified these tables. Did they weaken it or didn't they? And nobody really knows. Most people who've analyzed it said they actually may have strengthened it. Then there was going to be a chip later on, a next-generation called the clipper chip. The clipper chip the NSA helped design, and it was pretty clear the NSA had put back doors into the clipper chip so that they could read the traffic even—and that pretty clearly died. That chip never really made it.

FIDLER

Their involvement on the ARPANET's interesting in that, for example, they worked with BBN to develop the PLI technology. They were a sponsor, I think, at least by '78 or so, which means they were funding a lot of the operation of the ARPANET, and I'm curious—except maybe they don't get credit for it. I'm curious how much, you know, people, for example, at UCLA were aware of their presence.

KLINE

I wasn't aware of that. Jerry may have been. Jerry had done some consulting for the government, so he may have been aware that some of the funding in security research or encryption research for the ARPANET was indirectly coming via NSA. I don't know. I

didn't hear that, but it doesn't surprise me. I would expect it was the other way around, in that they may have been funding, but it may have been so they could get their own access to it, not only to use for their own purposes, but to make sure that they could always read whatever traffic was going on on it. You never know.

FIDLER

Where do you think your researched traveled? So once you're generating these publications, there's work on security, there's work on secure networks. Where do you think it was then used?

KLINE

The secure computing stuff largely died. Eventually there were so many bugs in, like, Windows and things like that, that people learned lessons about how to write better code, and there's books on how to write better code and secure code, and good coding practices for security, some of which we talked about, some of which were learned other ways. But in terms of trying to build a secure computer system, I think that's pretty much died for now. No one's really working on how do I build a really secure system. In terms of the encryption stuff, we did a lot of fundamental stuff which sort of became fundamental base literature that people study when they're learning about encryption and networking. So that was positive. We probably don't get much credit for that, but that's all right. I'll take money but I don't care much about credit. [laughter] It's funny, though, sometimes when you see things where you see people haven't learned fundamental lessons, and there's things that I'm constantly surprised about. By the time we were doing our research in the seventies, there was already the beginnings of discussions about digital signatures and that you could send a message and encrypt it in a way that only the proper sender could have encrypted it that way, and you can check that it really was encrypted properly and, therefore, you can know cryptographically that only Charley Kline could have sent this message, or at least his computer. If he's stupid enough to keep everything on that computer, then it would require his computer. Somebody else couldn't fake it on another computer. And the protocols were pretty straightforward. There were a couple of companies that built implementations of these. It's actually built into Outlook, even. If you get yourself an encryption key, you can send encrypted messages. But what surprised me was it never really happened, meaning that I would have thought all messages would be sent signed. Everything you sent would automatically be signed, and you'd be able to tell, yep, this really did come from Brad or at least someone with Brad's authentication, someone who knows whatever it takes to convince the computer system that he's Brad, and that, for example, when you get these things, the "from" line on an email, as you well know, I can put anything there. Nothing checks it. So I can send you a message and say this is from Len Kleinrock, and you'd be hard-pressed to tell that it really wasn't from Len Kleinrock. "Brad, I want you to come here because I think you're doing lousy work," or great work or whatever. Okay. So I've constantly been surprised that that just hasn't become the norm. Now, there is an issue in that it's beginning to be used, and the NSA and whatever are upset about that, not so much the signature as much as the encryption, because there's two things: signing, which means you can tell that I really sent it, and encrypting, which means only you can read it. And if you use both, then only you can read it and you can tell that I—if you know about public and private key encryption, I encrypt it using my private key. You can use my public key to decrypt it and you know, therefore, only I could have sent it. But I also take that and I encrypt that with your public

key. Only you can decrypt that with your private key. So first you decrypt it with your private key, then you see that there's an encrypted message in there supposedly from Charley, then you decrypt that with my public key, and you say, yep, it really did come from Charley. So I know that only you can read it and only I could have sent it, and that could be built into everything, but it isn't. But the NSA isn't so happy about that, because if people started using that, they'd have a lot harder time reading emails and things of the bad guys. Now, the problem is they read more than just the bad guys' emails, but there's a whole policy issue, but separate from the policy issue, there's a technical issue. And we sort of want the NSA being able to read Al-Qaeda's emails, and we'd sort of like to be able to know that Al-Qaeda is setting a plot to blow up a plane or launch a dirty bomb or something. We'd love them to find those. We just don't want them sending, "Gee, Brad, I'm sending you this thing because you should fire such-and-such who's working for you. He's actually stealing your stuff." You don't necessarily want that email being read by anybody other than—you know. Or the one you're sending to your girlfriends—you have two girlfriends, and you don't want to send it to the wrong girlfriend and say— [laughs]

FIDLER

Let's release this interview ten years from now.

KLINE

Yeah. [laughter]

FIDLER

Speaking of trust and speaking of privacy, over the seventies, for example, there was, among other things, a transfer of the operation of the ARPANET to the Defense Communication Agency.

KLINE

Right.

FIDLER

This is at least an official nod to the network becoming operational.

KLINE

Right.

FIDLER

By '83 or the beginning of the MILNET split, what did you notice in terms of broader changes in how the network operationalized and became more used for ongoing research and communication than experimentation? And furthermore, there's a lot of discussion about this early trusting community. Had that been destabilized somewhat by the end of the seventies?

KLINE

I didn't notice it, but I really wasn't—by the end of the seventies, I was so focused on distributed computing and trying to start companies and consulting I was doing, that I really didn't pay much attention. I mean, I knew it had transitioned. I knew about the MILNET thing. I knew that NSFNET was closing at that point or happening—I don't remember the years.

FIDLER

It kicked up around '86.

KLINE

And all these various things, but I didn't care if DCA was in charge. As far as I was concerned, that was just a funding issue and making sure that the wires were paid for and

maybe somebody deciding, “Oh, we need another circuit from here to there, so I guess we’ve got to fund another circuit.” Because at the beginning, there were only a couple of cross-country circuits, and then over time, more got added. So I didn’t notice it, but I wasn’t spending much time focusing on the network by the late seventies, early eighties. I was focusing on other things. If you’d asked me about local area networks and the technology there and the tradeoffs between this kind of Ethernet or that kind of Ethernet, that I cared about a lot then. And I was spending a lot of time worrying about which processor chip made sense to you because I was consulting on that. And some people wanted to use the Motorola 68000 chip, which was what was used in the Sun. Some people wanted to use the Intel chip because it was being used in the PC. Some people wanted to use the Zilog chips because they had relationships with Zilog and so on. And there were all these different chips that were potential processor competitors. So I was interested in all that stuff, but I really wasn’t paying too much attention to how DCA was managing the net. All I knew is it was working fine for me. [laughs]

FIDLER

Now, for the big picture, there’s a lot said about—and this, you know, you can draw on your time as a co-PI, for example, which we haven’t had time to talk about yet, but, you know, in the early eighties, there’s a lot said about there being—and we’ve talked a bit about this—a unique management style, a culture. There’s particular habits and practices that characterize places like UCLA and also places that were funded by the IPTO. And people often compare that in different ways to contemporary management styles in, for example, network research, computing research funded by the present-day IPTO. What kind of perspectives do you have on what may have been unique, what might have been valuable, less valuable, and how that changed over time?

KLINE

Well, I think it’s very valuable that if you’re trying to do something that’s new and never been done before, that you keep an open mind and you’re looking for the best solutions, and you recognize you’re probably not going to get it right the first time. Not only was the iPhone not the first smartphone; it wasn’t Apple’s first smartphone either. And a management style which says we’re not going to sit there and spend forever making specs and lock them in, then build the product and be done with it is a better way to try to do something really new and unique. However, that does run the risk of never getting your product out. So a research institution, that works great for. For a product company, that may not work great because you have to eventually make some decisions and decide what you’re going to build and what you’re going to sell. I think most universities give a lot of freedom to their professors to research in anything they want and try to come up with new stuff and try to get students involved in things. The problem, of course, is they don’t necessarily have a lot of funding to do things that require funding. It’s one thing if you’ve got a history major and you want the history major to go to—he wants to go off, and you’re a professor, and he wants to go research some unique thing about John Adams that people haven’t researched before, no problem. But if he needs to have a million dollars in funding to go fly all over the place and dig up archives that haven’t been opened in years that he’s got to get special permission for, well, there’s a lot of things, whether it’s pharmaceutical research or whether it’s computer research, sometimes those things require a lot of funding. So I think for something like the ARPANET, where they could recognize there was a value in sharing resources and interconnecting these

computers, nobody knew how to do it, the big players weren't interested. It wasn't in their business plans to interconnect their computers and certainly to interconnect their computers with anybody else's computers, that it made sense to have a pretty open style about what is reasonable to accomplish and how can we get it done and when can we get it done, and yet let's not just throw money at something and hope it works. You're likely to get to a better result, but it may take longer and you may get to no result, whereas if you have a hard deadline—you know, you're trying to put a man on the moon and you want to do it in ten years, you've got a lot of constraints. You have some budgetary constraints. You've got some serious research issues that aren't solved, and it gets pretty tough. We succeeded in doing it, but with a lot of risks. So they needed to be open-minded about what's possible, but then they eventually had to say, "Okay, we've got to make some decisions and actually build something." At Google you're supposed to spend something like 10 percent or 20 percent—I forget what the number is—of your time doing whatever you want on new ideas, research, whatever. Every employee is supposed to do that. Even though a lot of them are worked so hard they don't have any time to do that, but they're supposed to have 10 percent of their time, or whatever the number is, inventing things, doing new things, trying to come up with new ideas. And I think that's important in a company that wants to be research-focused, and since ARPA is research-focused, that's pretty important. With ARPA, one of the critical things is they've got to constantly bring in new talent as program managers, people that are really bright, that have good vision, that don't necessarily have to build things themselves, but can recognize good stuff and can recognize talent.

FIDLER

In retrospect, you were at some times at UCLA managed at least marginally and also were managing people. Compared to, for example, your time in the private sector later on, were there unique aspects to that, like particular ways of doing things, philosophies?

KLINE

Well, one of the things I learned at UCLA which I applied in the private sector is good people will make up for a lot of sins in management, so get the best people you can get. Second thing was if people trust you and respect you, and if you listen to them and try to make an open and inclusive environment, you're going to have a better working relationship and maybe come out with a better product. You've got to be careful not to get in this trap where you never get anything done, but if you can do the combination. So the people who worked with me and for me at LOCUS say that was some of their best years that they ever worked anywhere because both in what we were doing and how good the people were and how good our products were. And, yeah, we occasionally had some deadlines, we occasionally pulled some all-nighters, we had a lot of deadlines, but we were able to make the place fun and challenging and exciting, and if you can do that—now, not every situation has the option to do that. You can't always get the best people. You still have to do the work. So on a scale of one to ten, if you can't get all tens, you may have to make do with sixes or sevens, because you've still got to get the work done, which means it'll take more management time because you're going to have to look at what they're doing in more detail than if you know you can just trust them to do it right.

FIDLER

And so there'll need to be some kind of mechanism of attracting the tens.

KLINE

Right.

FIDLER

And this is something ARPA has excelled at traditionally.

KLINE

They've certainly excelled at getting some really good people as their program managers, and they've excelled at recognizing the places they should fund where there are tens doing the work. Then there those people, for whatever reason, attract their—I mean, Len was world-class queueing and networking, so he attracted world-class queueing and networking people to come there. I don't know the current generation of people, but from what I can tell from what I read about papers I see and things, they've got some pretty good stars now in the UCLA computer science department and in AI and some other things. I'm not going to make it, but there's a memorial service for Mike Melkanoff coming up. I'm not going to get there. But I studied with Mike. I don't think he was on any of my committees. If I remember correctly, his name is actually Michel Melkanoff, M-i-c-h-e-l, but he went by Mike. I'm trying to remember. Anyhow, it's funny, all these guys were professors when I was an undergraduate, so they're ten to twenty years older than I am, but when you're fifty or sixty and they're seventy or eighty, they don't seem like they're these old guys and I'm this young kid anymore. [laughs] They're colleagues. But when I'm a freshman or a sophomore and here's this guy who's ten or twenty years older than I am, and he's the established, knowledgeable guy, famous guy.

FIDLER

While we're on the topic of retrospective analysis, you have an interesting position in terms of perspective on the Internet because, for example, there weren't a lot of people sending email in 1971. Sounds like you were one of them. You accessed email from parties via one of those hacks. Presumably your perspective is informed by a *longue durée* and longer trend lines, for example, than, for example, mine, and I'm wondering if there's things you think you notice or perspectives that you have based on this longer history with networks that might not be widespread today.

KLINE

I'm trying to think. People are often asking me where do I think the Internet's going to go, and I don't really have a good answer to that. There's the obvious things. We're going to keep getting more bandwidth. We're going to get more power in our pockets. Things are going to get easier to use. More things will be able to be done by voice. But in terms of things that aren't quite so obvious, I think there'll be more security eventually, but I've been saying that for a long time. There's some technical things that'll eventually be solved so that it won't matter what your IP address is. People have talked about mobile computing and mobile IP so that no matter where you are, whatever technical IP address is needed so that you can be addressed, so that your device can physically get packets, it should still look the same like I'm sitting in my own home on my own local network, even if I happen to be in Japan, okay. I shouldn't have to set anything up. That'll eventually happen. Things that just weren't practical because of computing power, like Voice-Over-IP, that's happening. Everybody uses Skype. It's not that great, but it's happening. Eventually, as bandwidths keep going up and delays keep coming down, Skype will get pretty decent or in equivalence. I mean, when you start seeing people Skyping in the TV news program and it sort of works—

FIDLER

We are doing this in person after all, after forty-something years.

KLINE

Well, there are still some things that don't come through over Skype. It's a little hard to get inflection of the voice. Maybe you can. All the visual cues that you get besides just face, hands, everything else, I mean, it's hard to get a Skype that really gives you that same perspective. I think it's probably not likely to happen in our lifetime, but direct brain-to-brain connection, so I'll actually feel your feeling. [laughs]

FIDLER

Terrifying.

KLINE

Let's see. And I was going to say it's not that many years ago that we would have thought I'll be able to have pictures and video and Internet communication on my telephone? Except that you've seen the Nicola Tesla quote.

FIDLER

Yes. In fact, there's a series of predictive quotations, I think, although none of them were talking about distributed and adaptive routing algorithm or anything.

KLINE

They weren't talking about that, but the Tesla quote, if I can bring it up—I think I have it. I think I have the Tesla quote. Let me see if I can find it.

FIDLER

Were any of these future visions—I think there was a Wells, Tesla, there's a few others. Did these ever show up on your minds when you were—

KLINE

Not to me. Possibly Len, but I'm not even so sure I believe that.

FIDLER

Incidentally, was there a particular point when you realized that this is going to be ubiquitous and this is going to be dominant? Like a stage in your use of the ARPANET where you figured out, "Okay, this is happening now. This is going to be a part of our way of life so long as we've got a civilization"?

FIDLER

Certainly by the eighties, I was convinced that networking was going to be a major—everybody was going to be networking at some point. By the time AOL started happening in the early nineties, it was pretty clear that people were moving online, but it didn't really happen until '94 when the web happened. I remember coming up here for—I was with a guy for—we were visiting some companies, and I was driving around. There were billboards that had www-dot-something. It's the first time I'd seen www-dot-something on a billboard or in a newspaper, and thinking, "Wow. This is changing." [laughs] Well, I don't find—I have that Tesla quote somewhere. It could be in one other place. Let me see if I can find Tesla quote. I don't have it in front of me, but it's the quote where he basically says there'll be a time not too distant future where you'll be able to talk with anyone anywhere in the world, and you'll be able to pull up on a device not much bigger than a wallet or something, you know, pictures and sound of anything and look at libraries and whatever. This was a hundred years ago. [laughs] And he wasn't that far off.

FIDLER



Something we actually touched on but then didn't get into, there was the French CYCLADES network, and later there were these European networks, there's something up in Canada. Were you in contact with anyone from there? Were you thinking about these—

KLINE

I wasn't, other than peripherally, maybe, at some conference or whatever. Jon Postel certainly was because he was interacting with all these people. A lot of people were contacting him because he was sort of the repository of all knowledge ARPANET/Internet. And later on in ISI when he was creating IANA, the Internet Assigned Numbers Authority, and all that stuff, a lot of people were in contact with him. They weren't doing anything that was so unique that I really paid much attention to it, other than they were yet another network doing interesting networking stuff, but it wasn't that different or unique. More than that, it's not just us in the U.S. and in ARPA doing all the good stuff. There were other people thinking about packets, thinking about communications, thinking about reliable networks, thinking about communication technologies, thinking about protocols. While I think ARPA was the most advanced for a long time, we weren't the only ones. On the other hand, without ARPA's funding, it would have taken a lot longer for it to happen. And there really was a risk of it ending up being a bunch of incompatible networks in the same way that the Verizon phones don't work on the AT&T phones, don't work on the—IBM, CDC, DEC, they all had their—DECNET. IBM's was SNA, System Network Architecture or whatever, and they were mostly things to make either terminals on their systems work with other of their systems or remote job entry or some file sharing among their own systems. No attempts really to build generic general-purpose protocols that could be used and interact with other people, and that was the one thing that ARPA forced and one of the few really critical decisions. You can argue about could it have been done with a different management style, blah, blah, blah, but if it didn't force that decision that said, "We're going to make everything work together no matter what kind of hardware you have, and you're going to work on this network because we're funding you and you're going to make it work," we could very well be in a different place today. The first of the consumer networks was Prodigy. Then I think it was—I'm trying to remember if it was AOL next or CompuServe. I think maybe CompuServe and then AOL. They were incompatible. They didn't have web browsers at first; they had their own content. CompuServe was best if you wanted to follow stocks and things. They had better stock access. AOL had the better email and chat rooms. Prodigy—I forget what they had, but they were forced to add a browser as the web started happening. And pretty soon Prodigy and CompuServe died, and then AOL discovered that too much of their money was tied to the access and that they really needed to be an ISP that did more than just provide dial-up access. But we could have ended up in a real mess, where nothing was compatible.

FIDLER

Do you locate these decisions even prior to the move to TCP/IP, like in open compatibility for that, or is this in particular about which TCP/IP to use?

KLINE

It was partly the fact that ARPA said, "You guys, you're all going to connect," so it wasn't going to be specific to a particular brand. Then it was partly when TCP/IP came along, said, "It doesn't have to be just the people in the ARPA community." But TCP/IP

had to win due to its quality or its iniquitousness, not due to the fact that ARPA was forcing it, because nobody was forcing. ARPA and DCL were forcing all the government people to connect using TCP/IP, NSF to switch NSF to switch to TCP/IP, but nobody was forcing [unclear] or anybody else to go to TCP/IP, but it was pretty clear that, well, if we try to do it from scratch, we're going to have a big work factor to do, and it's not clear we're going to come up with anything better. So it made sense just to jump on that bandwagon. Now, once of the things that happened, just like with the phone system, is now that we have one sort of ubiquitous thing, TCP/IP, it's hard to fix problems with it because we all should be on IPv6, but we're not. We all should have security built into the protocols, but they're not there. The only issue is really conversion, not that it's too technically difficult to implement. It's all implemented. Most of the phones have TCP/IP v6 in them. They may not be turned on, but it's there. Windows and Mac have TCP/IP v6 in them, but most of the ISPs aren't providing TCP. If you want it, you usually have to pay extra for it. If you tell Comcast, "I want a TCP/IP v6 connection," they'll say, "Why?" And they'll say, "Well, gee, I'm not sure we provide that in home service. We do in business service."

FIDLER

From what we've discussed, is there anything I've missed? Well, there's lots of topics we haven't spoken about, but for what we have, is there anything you'd like to add?

KLINE

Well, let's see. Well, you're going to see some interesting transitions in the marketplace. When Stanford built the Stanford University workstation, which became the Sun workstation and then TCP/IP people realized, well, you need a router, and I think the first router was built at Stanford and then they commercialized and turned into Cisco. Then some of the people from Cisco spun off to build their own whatever. Building routers was sort of a black art for a long time. You either knew how to do it or you didn't. Now building router technology is pretty easy, so you're seeing a lot of new players in it. There's software defined networks and router protocols are changing some, and you've got Chinese competitors, Y\_\_\_\_\_ and others sort of coming, so you're going to see some transitions in the marketplace. It may not have any significant visibility to us, but it may have some significant disruption in which companies are the leaders in generating routing technology. Let's see. The next generation of phones is going to be TCP/IP-based, and there's some issues regarding reliability in the face of power outage and emergencies. The original phones from a hundred years ago, your phone in your home had no power. It got all its power from the phone lines, 24 volt and 48 volt singles on the lines, which is still there. If you put a dumb phone in, it doesn't have any power in the phone. And the phone company mostly runs off of batteries. It's got backup batteries for everything. I used to live in Sherman Oaks a mile from the backup emergency center for L.A. It was on Ventura Boulevard and Kester. I was a one-mile straight shot from there, so earthquakes, power outages, whatever, my phones worked. Everything worked, okay, except for that phone that you plug in, that you walk around the house with. Because there's no power, it doesn't work. But you had a dumb phone left over for that. Well, the problem is that they're switching to IP-based phone systems, and even without the IP-based phone systems, they're using optical and other carriers to pedestals that you may not see out on the street somewhere, which need power to convert that stuff to the wires that then go to your house. So in emergencies, the phones don't necessarily work, and

that's made worse with IP-based phone systems. So in terms of voice quality and all this and features, nothing wrong with IP-based phones. In terms of reliability in an emergency, and the FCC just made a thing about they want the phone companies to switch to—the phone companies are happy about that because they're hoping it's going to get them out from under the must-carry rules and rules that they have, that they have to provide Lifeline service for people who live in the middle of nowhere. So we're going to see that, but that's an indirect artifact of the technology. The technology is good enough that we can replace the old phone system, which makes phone companies happy, but has at least one negative consequence. What else? I'm trying to think of—I don't really have any good predictions of what's going to happen to the network. Somebody who's more up on that kind of stuff, who does that sort of as a living, is Vint Cerf.

FIDLER

Okay.

KLINE

If you ever get a chance to talk to Vint, I don't know if you get a chance to talk to Vint or have had a chance to talk to Vint.

FIDLER

Not yet.

KLINE

He works for Google, but he lives in—is it Virginia, I think. Reston, Virginia, or something. He's not out here too often. But his perspective is always interesting. He's Vint@google.com. Not gmail, but google.com. As a Google employee, he can get a four-character email. Let's see. Contacts. Cerf. Vint. I don't think he lives in Virginia. Yes, it's Vint@google.com, and it's McLean, Virginia. He's up as much as anybody on where things are likely to go on the Internet. It was interesting, one day when they were setting up a revolution or one of those things at the Computer Museum, I think Mark wanted some—did I have any pictures of Vint Cerf or Steve Crocker, whatever, from that vintage, and I said, “Well, I can search for current pictures of them, but I'm not sure I can search for old pictures of them.” So I called up Al Spector, who's the head of research for Google, and I said, “Al, is there any way to search for a picture of somebody twenty or thirty years ago? Given a current picture, can you match that with pictures that you have in your collection?” Because if you drag a picture into the search box, it'll search for that picture. So if you drag my picture, it'll probably come up and say “Charley Kline.” But if you had a random old picture of me, it isn't going to match it because it doesn't have it. He said, “You know, I have technology that can do that, but I can never think of a good use for it.” [laughs]

FIDLER

There's one.

KLINE

To be able to do a comparison of pictures that you're looking for what would this person look like twenty years younger. [laughs] And then see if I have a picture that matches that. Anyhow, things like that may happen. That could be good or bad. [laughs] I mean, the police would love it if they could say, “Well, we have a picture of this bad guy from thirty years ago. What would he look like today? And can we match it with a picture?”

What else can I do for you? What else can I answer for you?

FIDLER

I think we've got your time in the private sector for a future interview.  
KLINE

Okay. We can do some of it in person by Skype or something over the phone.  
FIDLER

I look forward to setting it up. [End of interview]

---

[Parent Institution](#) | [TEI](#) | [Search](#) | [Feedback](#)

---

*Date:*

*This page is copyrighted*