Interview of Dave Clark

Interviewed by:
James L. Pelkey

Recorded March 7, 1988
Cambridge, MA

CHM Reference number: X5671.2010

James Pelkey: How did you get involved with communications and networking and so on?

David Clark: Well, I got my PhD in '73, and the topic of the thesis was trying to -- it arose in the context of secure systems, as opposed to distributed or communication or anything else. We were trying to look at the kernel of Multics, and pull everything that was possible out of the kernel, so that we were left with a minimal thing that we could certify and verify. And my thesis was pulling the I/O system out of the kernel of Multics, which was a great messy thing which everybody assumed was sort of intrinsically part of what the kernel did, and I basically tried to make that a users based thing. Now, the hardest part of that, of course, and I didn't completely attack it in my thesis, were multiplex devices, and of course networks were the primary example, so I sort of had it in the back of my mind that networks were something I didn't know how to think about at the time I finished my thesis, and the Multics was in the process of being hooked onto the Arpanet right then. It was on the Arpanet, but we were still improving the code, and when I got my degree, I started working here, working for Jerry Saltzer, and the first thing that I did was to get involved in the ARPA networking, managing the project of getting Multics hooked onto the net and all this kind of stuff. And we had in mind, ultimately; we were going to try to do host-to-host stuff, that is to say distributed systems. And I guess, after I screwed around with networks for a couple of years, two things happened, and one of them was that we realized that the Arpanet made a really terrible local area net, because that was all we had. And the second thing that happened was that ARPA began its series of working meetings, what was called the Network Working Group, to plan the successor to the Arpanet protocols as an outgrowth of Vint and Bob's papers, which came out in '74, but of course had been written in '73, which is the same year that my thesis was done, so there were really two efforts that sort of got started at the same time. One was an effort for our group to develop a local area net that we could use to hook hosts together so we could do experiments in distributed systems, and the other was, we got sucked into the development of TCP/IP fairly early, so we had a network component to the project and a protocol component to the project, and is this sort of the context of the level that you wanted? Yeah, ok. We actually got involved in the TCP project because a fellow who works --

Pelkey: Now, this is '76?

Clark: This is '75, let's say, something like that, '76 or something like that. I could go back and look at my notes.

Pelkey: This is when Vint was at Stanford -- still at Stanford?

Clark: No, Vint was at -- I don't remember when Vint first came to DARPA, but certainly he was there by the time that I got involved in the Internet project. There was a fellow here named Dave Reed, who was on the faculty, and he looked at the initial proposal for TCP and said, you know, there are some things about this that are much too complicated, and he, with some contributions from me but it was primarily his proposal, generated an alternative to TCP, which he called the Data Stream Protocol, or DSP, and we wrote some memos on this and the folks down at ARPA were really excited, but they said: "Look, we really want to pull together on this and we all want to develop a common protocol, so we really hate to see parallel efforts getting started. How about joining the TCP effort, and folding some of your ideas into the standard we're developing." So we started going to the meetings and, in fact, although Dave was the one who wrote the DSP memos, I was the one that got interested in the meetings and started going. And during the remainder of that decade, a lot of the stuff that was going on here was the local net stuff, but I got more and more involved in the TCP stuff as an outsider. What I mean is outside MIT. And when Vint was preparing to, I don't know if he in fact knew he was leaving DARPA at that point, but when he was preparing for his eventual departure, he asked me if I would take over the TCP project. I guess, I can't remember, I think the first meeting I chaired was in September of '81, I think. I'd have to go back and figure it out, but I can sit down and figure it out, but it was a meeting in London, and after that I ran the Internet Working Group, as it was then called, and when Vint left DARPA, I took over sort of architectural responsibility for TCP/IP for DARPA, and I've done that ever since, although TCP/IP has grown up and gotten very big and mushy and hard to find and so forth, but there's still a steering board called the Internet Activities Board which I chair, which has some responsibility for trying to push forward TCP until the time when the ISO standards really displace it, which may or may not ever happen. That's a topic we can come back to if you care?

Pelkey: Yes.

Clark: Standards are obviously an important force in this whole business, which I think might actually get some focus as a specific topic because --

Pelkey: It does at the level below the usage.

Clark: Right.

Pelkey: You were then heavily involved in the process of converting from NCP to TCP/IP on Arpanet?

Clark: Oh, God, yes. I have one of the buttons that says: "I survived January 1, 1973." Absolutely.

Pelkey: Dan Lynch showed me one of those.

Clark: Oh, yeah, oh, yeah. I was in that up to my eyeballs.

Pelkey: You were a pretty popular person, I imagine, at that point. A lot of people didn't want to give up, and you had to force --

Clark: It was tough. It was tough. Just like getting the Arpanet installed. You know, Larry Roberts had to -- had to bash people up pretty heavily to get them interested. They said, you know: "I'm comfortable doing what I'm doing today." He devised some fairly cunning strategies for smashing people out of shape so that they would go install it in. But we had to do the same thing. Yeah, we had to legislate it, you know, and you look back now --

Pelkey: And it was the Internet Working Group that formulated those policies and caused it to happen?

Clark: Yeah.

Pelkey: And you were chairman of that during that -- up through '83?

Clark: No. I -- gee.

Pelkey: It was '82 when all the drama took place. In the beginning of '82 --

Clark: I was -- yeah, I guess I was, I guess I had just taken over chairmanship of the Internet Working Group then, but it wasn't primarily viewed as my baby, because in some sense, the transition had been coming down the pike for a long time and everybody knew about it and we -- I can remember the meeting before it happened and the meeting after it happened and, you know, in some sense, the process had been so architected that, in some sense, our steering group didn't have a lot of control over it. You sort of launch it about three years in the past and then it starts coming at you faster and faster and, you know, after a while nobody can do anything. It generated a lot of -- it generated a lot of short-term frustration. I think everybody understood why they were doing it. Their attitude was: "Yeah. We were happy to do this, and I just wish I didn't have to do it this year." You know, it's sort of this Alice in Wonderland: "I'd like my jam tomorrow, please." You know, it was for arbitrary values of tomorrow, and --

Pelkey: Was it converting all these application programs over to TCP/IP from NCP? Because TCP hadn't really been thought through, I gather, at some level relative to -- well there was incompatibilities between NCP and TCP/IP, which caused people who were running software --

Clark: There were certainly incompatibilities. Some of those were deliberate. There was no intention that TCP would absolutely mimic the behavior of NCP. In fact, for almost all of the applications, migration was not hard. That's my recollection looking back. In the important applications, well there was one upheaval. There was one serious upheaval that may not be relevant. We can talk about it. It doesn't

matter. Cut this off if it seems to be a blind alley. The remote login and file transfer were almost identical protocols. The only thing that was different about remote login was this strange idea that TCP has. Instead of having an “out of bound” flag, it has something called "the urgent pointer," which in fact is due to Dave Reed. It's the contribution that came in from DSP, from his memos, but the place where we rocked the boat at the application level was mail, and that, in fact, was not a strict outgrowth of migration from NCP to TCP. It was a recognition that mail had initially existed as a wart on the side of FTP, and it had become clear to us that mail was so important in its own right, that it needed to be re-architected, and so, and I don't remember the chronology of it, but there was a working group sort of parallel to TCP that was redrafting the entirety of mail, and we pulled mail out from FTP and we recast the transfer protocol to something called the Simple Mail Transport Protocol, SMTP, and we redid the document that describes what the header of a piece of mail should look like and -- (something breaks) ooh, sorry, careful -- and, I keep forgetting tall people shouldn't sit in that chair, but -- this is a great office to hide in but it's not good for having meetings. And a lot of people were pissed because applications are a big deal, you know. If you look at the bulk of the TCP, you get all scared, until you look at the bulk of the application. I draw the -- I draw the seven layer reference model as an upside-down pyramid. It looks like this, with physical layer down at the bottom, and TCP is really not a big deal compared to these applications. Well, rewriting mail and getting it to work right was a real problem. And there's an interesting aside -- I don't know whether its relevant to the topic of your book, but --

Pelkey: It is. It is.

Clark: It was a technical problem with mail, just as an aside. Whenever you build a layered architecture, there's this fantasy that your going to put in a layer that fixes errors, and that's just -- that's brain damage because you can't protect yourself from the failures of the peer logic at the other ends, but that gets hidden if you're building applications that human beings nurse along, because the low probability errors get corrected by the absolute top level, which is the human. You're doing a login and all of a sudden it seizes up. You log-out, you login again and, you know, you may or may not even bother to tell your manager. Mail was the first application that we built which was not in real time being nursed by a human. There are these spoolers, these background jobs that take mail and try to send it later. And what people discovered was that, in order to keep your mailer up, you had to do an order of magnitude more work, that's a loose estimate, but you have to do a great deal of work at the mail level, and they get all bent out of shape because they said, you know: "You gave us this reliable protocol to run over, and now I'm building all this stuff with timers, and all this kind of stuff, and why do I have to do that?" Well, it was the great -- it was the great scales falling from the eyes. You have to have the robustness at the level that you want the robustness. If you want your application to be robust, you got to put the shit in.

Pelkey: Right.

Clark: And, so there was a lot of pissing and moaning about mail because people were depending on it so heavily, and that's something we might come back and talk about later, but I think a lot of the transition was actually, you know, just as much as writing TCP, it was getting all this other stuff done. New naming architecture and mail and all this kind of stuff, so yeah, it was a hassle, there's no question.

Pelkey: Let's come back to that. That, and when you got involved with TCP/IP, how big was that group?

Clark: Well, it's -- it's a funny answer. What -- there's an inevitable process that goes on which is that whatever meeting you establish gets bigger, and eventually it gets so big as to be unworkable, and the only solution to this is to --

Pelkey: Subdivide.

Clark: -- is to create a new group. Let the old one nominally exist, just never convene it again and create it a new one. Now, we had something called the Internet Working Group, and I suspect that when I started going -- it was called the Arpanet Working Group, it changed it's name as part of it's attempt to disguise itself and hide and remain small. I suspect it was about 20 people. Vint chaired it. I could go back and find some early notes from that meeting if your interested.

Pelkey: Could you?

Clark: Yeah, we can do that, later, and then it got bigger and bigger and bigger, and there was a gigantic one. I guess it was about 1983 or something like that, '84. I can't remember. We held it at BBN. I can dig these dates up for you later. Just remember the things that you want me to find, where 95 people came, or something like this, and we said: "This is unworkable. We can't handle this." So we created a -- we created a subgroup, which we -- we gave ourselves a name which we hoped would be so obscure as nobody would ever want to come. We called it the Internet Configuration Control Board, and then we created a bunch of task forces to go off and do the real stuff, and the idea was that the task forces would -- I guess that idea came later. Initially, we just created this little subgroup, and then later - -

Pelkey: And you were chair of it?

Clark: I was chair of it. And then we created something else. We destroyed the ICCB and created something else called the Internet Activities Board. So we did this -- this changing the names so people couldn't find us --

Pelkey: And you were doing that consciously?

Clark: Oh yeah, absolutely. Conscious manipulation of the system because everybody wants to come. It's a sign of prestige, it's a sign of 'I'm a member of the club,' so what you have to do is figure out a way to expand the club, so what we did eventually, the structure that we came up with, and I think it's a very good one and it served us well for a bunch of years now, is we created a bunch of task forces, and the task forces did the work, and the Internet Activities Board is composed of two sets of people: the people that are chairmen of task forces, and they come so they can exchange ideas, and they say: "Well, I'm working on so and so," and somebody else says: "Well, I was working on so and so," at which point I would say, "Would you two go off in the corner and talk to each other and tell me what happened?" And we would also decide what task forces needed to be created or destroyed. If there was a new topic that we weren't paying much attention to, we would say: "You know, we really need a task force," and we'd scout around and see well who could chair it, sort of. So we were this slowly evolving body in which we would nominate new members as a side effect of creating new task forces.

Pelkey: And every member of the body is a chairman --

Clark: Well, there was another set of people who got invited to the IAB which would -- were the people who had some, how shall I say it, putting it crassly, they were the people with the money. But that's not quite it, because what we really wanted to invite were the people who had some need of seeing the Internet protocols evolve, because they were the ones who generated the requirements. So, you see, for example, the Department of Defense was the initial organization, and there was a representative of DARPA, but of course DARPA was the research organization, so the first thing we did was brought in an operations guy from the Defense Communications Agency, and we said: "Well, the two of you should sit there because you're the one who's driving research, but you're the one who has the operational requirements. Tell us what your requirements are." Then the next one we had was from the National Science Foundation because it set out this rather ambitious agenda to build a national network in support of research in all the sciences, and then recently we've added in the equivalent person from the Department of Energy and NASA and Health and Human Services -- not Human Services, that's what it's called, yeah, Health and Human Services, because each of these has a need for national networking, not in support of their own internal operations, but in support of their own research community, and what we're trying to do is facilitate -- what we're trying to do is facilitate the working between these people, and the latest thing we did, if you're interested in the organizational structure here, the latest thing we did, which we just did six months ago, is we took all of those people who represent the government agencies who are not the technical drivers, they are the people with the money and the requirements, and we separated them off from the IAB, and created something else with the ungainly name of the Federal Research Internet Coordinating Council, or something like that, otherwise pronounced FRICC, and then we did this for two reasons. One is that they need to constitute themselves as a separate body in order to

talk to an equivalent body which the Europeans had organized, and the other reason that we did it was that, by doing so, we could then ask them to resign from the IAB, which was critical to this. I was asking them for money, and the conflict of interest rules say you can't be on a board that you give money to, so they had to have them on a separate group, so now they come as observers.

Pelkey: Good.

Clark: Ok, so that's --

Pelkey: You -- MIT formed a TCP/IP group?

Clark: Yes.

Pelkey: Were you involved in that research?

Clark: Yes.

Pelkey: Because, it strikes me that, in terms of TCP/IP becoming, really, a protocol that was dominant, there was the event of it being distributed via Berkeley UNIX, which I want to discuss a little bit --

Clark: Right.

Pelkey: -- and also that, it being available, a standard version being available on the PC.

Clark: I think if you look at the chronology, the first thing, of course, that happened is that DARPA paid a bunch of people to make it available on a bunch of operating systems, and they paid money to make it available on a wide variety, but not as many as -- some of those had more impact than others, and I agree with you. The UNIX one was an officially chartered one. The PC was almost an accident. It's a funny story. If you want that, I can tell you that story.

Pelkey: Yeah.

Clark: It arose because, you know, if you're in charge of the architecture, you get a lot of funny hate mail. People send you messages in the middle in the night and say: "You know, this is a really sucky protocol. The spec is 12 pounds and it took me six years to read it and implement it, and it moves like molasses, and its a loser," and one of the messages that I got was: "You know, this is an incredibly complicated piece of protocol, and you could never get this to run on a small computer." And I said: "Bullshit. I can get it to run on a small computer." And at the time we had here a series of workstations from Xerox. We had Altos. We didn't have PCs yet, and I said: "I'm going to write the world's smallest TCP. I'm going to show you guys." And I sat down and I took the code from the Alto that was written in BCPL to run PUP, which was the protocol that came out of Xerox that was a, sort of a spin off of TCP, and I transliterated it backwards into TCP, and I set out as my goal only one ambition. I didn't want readable code. I didn't want portable code. I wanted small code, and I wrote the most sawed up piece of convoluted goo. I have been taking shit for that for the last decade. You'll understand why when I tell you what happened to this piece of code, but I wrote this piece of code, and it was an entire TCP. What I mean by that is that it successfully interacted with all the other TCPs across the Internet. I hadn't stripped anything out, except one thing. It could open a connection, but it couldn't listen for one. It was asymmetric. The reason I did it is I wanted to run User Telnet on it and to run User Telnet you have to open a connection but you don't have to listen to one, so I just left out a few states, but I could have put them in. And the first program, it was three and a half pages of code, in BCPL, and it was 1000 instructions. And I said: "I'm going to show those bastards I can write a small TCP." And the whole thing by the time you built the core image was tiny, you know, and it ran reasonably well, too. What I had done was, I optimized the service interface to the protocol, and that was an important insight, because what we realized was that, if you go and look at a typical TCP, that is to say if you dump the listings out and you attempted to read them, you can't find the state machine. You go rooting through the thing and finally in this bottom of one page someplace you find the state machine, all of the complexity of building a TCP arises, all is too strong a

word, but a vast majority of the complexity arises from the fact that you have to meet this rather broad service spec for a variety of clients. You have remote login that wants one character at a time. You have file transfer that wants huge blocks of data at a time, and you've got to be responsive to all of that. There is one protocol I audited, it wasn't a TCP, in which I literally found that of the 66 pages of listing, six of them were going into the protocol, and the rest of them were going into the interfaces to the layers. So by optimizing the relationship between TCP and the client above it, I threw out what I would assert to be as much as 9/10ths of the complexity. So, anyway, I wrote this little tiny TCP, and there was a guy working for me, a guy named Larry Allen, who said: "That's really great. We ran a Version VI UNIX. It didn't have a big kernel. There was no room in the kernel to put TCP in." He said: "I'm going to transliterate that into 'C'," and without knowing what he was doing, I mean he didn't understand the code, he just transliterated it from BCPL, which is of course a predecessor of C, he transliterated it into C and got it running in the user space of UNIX. And he said: "I can't understand this piece of code," because as I say I had written it to be smart. He said: "This had got some really weird stuff in it. I don't understand this at all, but I just transliterated it," and I had to go back and make it work a little bit because there were a couple of places where he had screwed up. And then there was another guy working for me, and I forget exactly who did what where, but I think the guy who actually did the initial port to the PC was John Romkey, I think, and John said: "I'm going to port this to the PC." And of course the PC has an absolutely unworkable programming and operating system environment, so we had to write our own little operating system environment for the PC. Basically, MS-DOS just doesn't know we're there. We just sit of to the side. We run our own timers and multi-tasking package and put the whole thing in there, and he ported it onto the PC, and we began to have a little fun with it. And then, sort of only after we had done this, did we really wake up to the fact that this stuff had the potential of being very popular, and then we began to get organized. And of course Jerry Saltzer --

Pelkey: Excuse me, what time was this? What time frame? When did you kind of get prodded into doing it?

Clark: I'm trying to think of a single event that I can hang it on. I lose track of time. I'm very poor at this. I have to go back and find some hook. (Groan). Romkey graduated in June of '85, and he started working for me as a freshman, so that means he came in September, let's see '84/'85, 4, 3, 2, 1, so he came in September of '81, so probably sometime around '83, I'm guessing. Maybe I could pin that down a little better in a minute, but let's say sometime around '83 we got organized. I think we --

Pelkey: So after the conversion?

Clark: Oh, it was well after the conversion, yes. There's no question.

Pelkey: So it was at least in '83?

Clark: Yeah. Even if it was in early '83, it was definitely after the conversion. It's odd, but there was a real battle going on here at MIT, and TCP penetrated MIT more slowly than almost any place else. There's a cultural aside, which again, you may not care about, but I'll give you the two-minute version of it. At the same time that Dave Reed went down to DARPA and proposed a alternative to TCP, a group in the artificial intelligence lab up on the ninth floor here, decided they wanted to get involved in local networking too, and they invented an alternative hardware base, and they invented an alternative software base, both of which went by the same name, which were the CHAOS System, and the CHAOS protocols evolved very rapidly at MIT. I think its inevitable that something that's not a standard and is therefore not trapped inside of a standards process can evolve fairly quickly, and the CHAOS protocols became very popular at MIT, but of course no place else. And so for TCP to become popular at MIT, it had to displace CHAOS, and we invested a tremendous amount of emotional energy here trying to resolve that problem without spilling too much blood on the floor, because it turned into a lab versus lab debate, and a lot of grumpiness ensued. So we actually didn't, I'm having trouble with the chronology here but, but at the time of the transition, you know, we cut Multics over and we cut a few things over but, if you looked inside the building and said: "What software is most of the machines running here?" And the answer is they weren't running NCP and they weren't running TCP. They were running CHAOS, and

we had to flush all that out of the system. And you can still find little piece of it every now and then. There's still machines up on the ninth floor with CHAOS protocols.

Pelkey: How did it get named CHAOS Net?

Clark: It's an Ethernet, and so the allocation mechanism is CSMACD, which they viewed as chaotic, so they called it CHAOS to describe the strategy for bandwidth allocation.

Pelkey: Ok. Now, you were going on a point about the PC. All of a sudden it came, this portation out to, that you had done the C -- I asked a diversion question -- but you're saying that something happened then, where it became this kind of standard, or it had a life of its own.

Clark: Well, we ported the stuff to the PC and we started playing with it, and there's a monthly newsletter, an on-line bulletin which catalogs the DARPA funded work and the Internet protocols, and I'm sure I put a monthly note in there saying: "Well, we have this protocol running," and a bunch of people sent us net mail saying: "Hey, can I have a copy of it?" You know, the first people who sent us that message, we sent them a floppy or something like that and, after the tenth person that sent us a piece of net mail, we woke up and said: "This is dumb. Why are we doing this?" And so then we got organized, and we actually had the microcomputer store over at MIT, for a while, they would send you a floppy for the price of a floppy, or they would send you all the sources on a UNIX tar-tape for $30, or whatever. We gave it away for a couple of years in there, and then -- did you get some of this stuff from Saltzer?

Pelkey: I don't think so.

Clark: Well, then, what we noticed is that various people -- we actually tried, yeah, this is an interesting story here -- we tried to figure out how to license it. We were unsure what to do with it, and we said: "Well, you know, maybe we can make this into a product of some kind," and we started screwing around with licensing, and we were talking to Bob Metcalfe out at 3Com, and Metcalfe said: "Well, you know, I want an exclusive license, because if you're going to license it to somebody else, what good is it to me?" And we said: "Well, you know, it's really hard for MIT to give exclusive licenses." --

Interruption from someone entering the office

-- So we went back and forth with Metcalfe, and Metcalfe said, you know: "I don't know whether I want to -- It's going to be a real pain to get in," and we, we tried 27 kind of licensing agreements, some of which were really creative. I'm very fond of one that I wrote. I wrote this reverse licensing agreement in which you could sell the first thousand copies free, the idea being that the overhead of going through all the license fight was such a barrier that --

Pelkey: You had to give some economic advantage to get in the door.

Clark: Yeah, right. And so, but he said: "Well, but you know, I have to go through the whole license," and I said: "Well, we'll write you a contract that says, you know, after you sell a thousand copies, then we have to stop and negotiate the stuff," and said: "Well, of course I can't do that, because by the time I've sold a thousand copies I'm committed to the trajectory, and I have to know where I'm going." And we went back and forth and back and forth and finally we decided: "Well, just screw it. We'll put it in the public domain, or this pseudo-public domain." We have this technique for giving software. It says: "You can do anything with this you please except take this copyright statement off." You can sell the product, you just have to leave the copyright statement on. And we've done that with a couple of products, to really good, good ends, the gateway being the other one. You know about the gateway product? 'Cause that's what Proteon sells -- that came out of my group too -- same way, same path. So we put it in the public domain, and what we discovered, after a year, was that three or four companies had picked it up and were trying to make it into a product, that is to say they were selling it for money, as opposed to our giving it away for free, but what you got when you bought it from them was the guarantee of support.

Pelkey: Right.

Clark: And the guy who was working for me, Romkey, he graduate then he worked for me as a staff member for a year or so, he got pissed and he said: "You know, this really makes me mad. If somebody's going to go make money on this, I'm going to make money on this." So he resigned from MIT and went and founded a company called FTP Software, and took the sources, just like everybody else had taken the sources, and commenced to sell it. And more or less as soon as he did that almost everybody else that was supporting it stopped it, because they knew that Romkey could blow them out of the water, because he understood the code ten times better than anybody else, and FTP's been pretty successful, although I think they consider the number of copies they've sold to be a proprietary fact, but its measured in the large numbers of thousands certainly. I'm not even sure, come to think of it, whether they consider that a proprietary fact, I don't know. It's hard to know, but they've been very successful. Whether they'll be successful as a company, I don't know. What I mean by that is by the number of copies that have gone out because of that, they have been very successful.

Pelkey: So what happened, this process of TCP/IP getting out to the PC community, plus it being out there and used through everybody getting it by buying Berkeley 4.1 and subsequent versions, and then all of a sudden it started to become a factor in the marketplace, TCP/IP -- the confluence of those three things all of a sudden kind of start pushing it over the edge where XNS disappeared --

Clark: XNS disappeared for two reasons. One reason was that it didn't have this external financial push behind it to give it away at first.

Pelkey: Good point.

Clark: There's another very specific comment about XNS, which is that they made a strategic -- if their goal had been widespread penetration -- they made a strategic error of the first magnitude, which is that they were very dutifully chugging along, and releasing the stuff in its public specs, and they were starting at the bottom and went bump, bump, bump. And when they got up to the top, all of a sudden they failed. They never released a spec for remote login. Never, ever, ever released a spec for remote login. Well, how can you have protocols released that don't have a spec for remote login? They never released a spec for mail. All they did was release a remote procedure call protocol and a file transfer, and then the protocol that you really wanted was Interpress. It was the protocol for how to get to a printer, because the reason you want to speak XNS to get to the print -- well it turns out that all protocols from Xerox had been coming out of PARC, the research group, and the spin off, which was SDD, and they had an attitude that openness was a good thing. But it turns out that Interpress belonged to the group inside Xerox that did Xerography, and their attitude from the beginning has been that trade secrets are the way to success, and so they declared Interpress to be a trade secret, and they wouldn't release it. And everybody in the whole world looked at Xerox and said: "You guys are wedged, because, if you're going to take a protocol suite, lure us half way in and then make part of it a trade secret, we're not going to talk to you. You're a bunch of assholes," and they wrote them off.

Pelkey: Now, this is even after XNS, earlier, much earlier than this, it was '70s, mid '70s, when there's XNS, they wouldn't let it out at all.

Clark: Well, there was, first there was PUP. Now, when did XNS really come along, as opposed to PUP?

Pelkey: That was post '76, when Metcalfe went --

Clark: There were some early release problems with XNS, but they got most of the stuff out pretty quickly, until you got up above transport. You know there are two great books that came out of Xerox, and one of them is Transport On Down and the next one was Courier, and then they dropped their voice, and nothing else came out. And we got into a real -- I remember talking to Bridge. Bridge started out using XNS, and I was talking to somebody out there, and I said: "Well, what remote login protocol do you use, because I'd like to talk to your boxes," and they said: "Well, we were caught between a rock and a hard place. Either we had to use the Xerox remote login protocol, in which we were running proprietary

protocols, and in which case I can't tell you about them, or we had to invent our own spec, in which case I can tell you what it was, but then it wouldn't be the same as anybody else's." And what they said is: "I can't tell you what we did." Which means it's obvious what they did, they used the Xerox proprietary protocol, but they couldn't tell me that's what they'd done, because then if I went and looked at wire, I could have figured out what the protocol was, you know. So you couldn't use a Bridge box to talk to anything else, and that caused a rather complex series of events in which I tried to persuade Bridge to put TCP in their boxes, and we had an anti-matter explosion, but that's irrelevant. Xerox shot themselves in the foot. I think they could have made it because XNS was widely viewed in the community as TCP with all the needlessly complex things stripped off. Now, in retrospect, not all of that was true, but that's the way it was viewed.

Pelkey: And still is.

Clark: And so if they could have captured enough of the market to have a critical mass, they might have been able to get someplace, but they just -- I think the -- there were a lot of vendors, and if you looked out at the West Coast, there were these two vendor schools. There were the vendors who were stealing it from Berkeley, and therefore they were getting TCP, and then there was the vendor school that was implementing XNS, and for a while they were almost even. And I really think that it wasn't just that Berkeley was being given away for free. I really think that it was, as much as that, when you started using XNS, there were pieces of it that were just missing. They were just plain missing.

Pelkey: All right. And that process did come from PARC and had been an earlier implementation --

Clark: Yeah.

Pelkey: -- and that TCP/IP, well TCP originally was meant to handle radio networks and other networks that just -- that were - -

Clark: It was not so LAN centric.

Pelkey: Not so LAN centric.

Clark: That's right.

Pelkey: Because ARPA had the need for this.

Clark: That's right.

Pelkey: It came out of Aloha and those experiences and --

Clark: The radio net, particularly --

Pelkey: -- and then Dave Reed's influence caused it to be a TCP/IP.

Clark: I don't know that -- no Dave wasn't the one that split off IP. That was more generic. There's a paper I'll give you, incidentally. There's a retrospective paper I've been writing on the Internet architecture. I'll give you a draft today, just remind me. I'm going to send it in to SIGCOMM for the summer but, it's not so good on chronology, but it lays down some of the original motivations of TCP. Why TCP came into existence and some of the evolution of it. Yeah, PUP -- PUP was prepared to sweep under the rug a few problems that TCP felt that it had to solve.

Pelkey: PUP was only for local area networking and it was only connecting their machines and it was only like two kinds of machines. It wasn't universal, in terms of different kinds of machines.

Clark: In fact, it wasn't -- I mean, you could build a PUP Internet. It had a two dimensional address space and it had some of the same architecture as TCP. As I say, I took an implementation of PUP, and in one week I turned it into an implementation of TCP. It was very similar.

Pelkey: Now why was it layered?

Clark: Why was what layered?

Pelkey: The protocols?

Clark: TCP or -- Well, TCP originally was not layered. There was no layering in TCP. Layering was a fantasy of ISO.

Pelkey: NCP was layered, right?

Clark: No, not particularly. Well, I mean, yeah, there's some layering. There was sort of one layer. I mean, it had the operating system layer. What I mean is that, there was that part that you thought of as being inside the kernel of your system and then there was that part that you thought of as being outside. So, generally speaking --

Pelkey: Plus you had mail, you had file transfer --

Clark: Well sure. That's what I was going to say. What I was going to say was, we had the one layer, which was transporting and applications. And that's not a surprising interface, because for those of use who have been in the operating system business, we had this idea of common services. We had this idea of the file system as a common service, so when we began to look at the networking, we began to say: "How am I going to take this basic packet service and wrap it in a more tractable abstraction?" The idea of a common service was an obvious thing for us to do. So if you look back at TCP, that's what I meant about sort of, I didn't mean kernel in the sense of secure, what I meant is common service. We had this two-layer model, really. We had NCP, which was the common service, and then we had the application, and that was the two-layer model. And then what you can do is you can look at the -- at the fracturing of those layers, and in fact, if you look at TCP, which descended from NCP, above the service layer, you don't have any layers. That is, today, we don't have presentation and session. We never saw any sense for that and still don't see any sense for that, said he, rashly. What I see is a service interface and then above that you build applications, ok, and all of the stuff that ISO's doing has always left me sort of un-moved. Ok. But, then I'm, I've never been too enthusiastic about the heavy-duty layering. I'm not a layer – I'm an anti-layerist. Now, below, initially there was this one layer, which was the Arpanet protocols. Now -- which was NCP. Now when you, if you look at the way the layers evolved in TCP, there were two --

Pelkey: And intellectual property rights, that was this network, NWG.

Clark: Yeah.

Pelkey: With Steve Crocker and you had the, business which I'm not going to remember -- the book, where people send in and say: "I think that this protocol should include this," that Jon Postel was a -- is the repository of it.

Clark: Well there were the things called the -- we started out with the Internet Experimental Notes, and then there was Requests for Comments, RFC --

Pelkey: RFC.

Clark: And Jon is the editor of the RFCs. Right. And the Network Working Group in some sense owned, because Jon was a member of that, owned the editorial control of the RFCs, and we still own that, to the extent that I am the chairman and chief architect of the Internet Activities Board, Jon is my deputy, and

through that we maintain editorial control over the RFCs, which is how we control the community. That's a controlled press, although controlled in a very open way.

Pelkey: I had a great conversation with Jon, by the way.

Clark: Yeah, Jon's an interesting guy. At the moment, he's the Internet reactionary, which is -- it's always good to have one. Right. But, if you looked at NCP, it wasn't layered internally. The first thing we recognized is that if you want to have operation of a variety of nets, then you need to have a lower level interface, where you have network specific goo, and then the thing up here.

Pelkey: Ok.

Clark: And then the second discovery was that the thing up here wanted to be cut into two halves, one of which was IP and one of which was TCP. At that point, we stopped layering, and the definition of what went in IP, we had a very clear rule. It's much clearer in fact, its much more easy to articulate than the ISO rule. Ok, the rule was, if the gateways were permitted to see the field, then it was in IP, and if the gateways were not permitted to see the field, then it was in TCP, or in the transport layer, whatever it was. So, the idea was that when a packet reached the gateway, it was not permitted to look above IP, and so anything that the gateway had to see had to be pulled down into that layer. Ok. And so our modularity principle was very simple, which was that the gateway builders only had to understand IP and its associated control protocol, and the only time you had to understand TCP was the end-node. That was the design principle, and that was the basis of our layering.

Pelkey: Good. Let me come back to the hardware side of the -- you were, as you said, sort of up front -- there was this need to do distributed computing. You had to start running things together.

Clark: Right.

Pelkey: Why did you -- and CHAOS net was going at that point in time -- why didn't you guys just take CHAOS net?

Clark: Well, at the time that we started out, I can't get the chronology of this right again, but the time that we reached the decision to get some local net technology in here, the CHAOS net guys were not ahead of us. The folks out at Xerox had done the 3 meg Ethernet, and we had gone out and looked at it, and the folks at AI Lab had gone out and looked at it, and so we had to decide whether or not we wanted to do research in the area, or whether we just wanted to import a commodity. And the AI Lab and we, more or less at the same time, reached a different conclusion, which was that their interest was in artificial intelligence, so they were going to import a commodity. Our interest was in systems research, and so we were going to do some research. And we looked at the Ethernet and we thought about it, and we said: "You know, fundamentally, we don't think that this is a good way to do it." And we went down to DARPA, and in fact the initial pressure to open ourselves to this line of thinking, came from DARPA. It was the same kind of pressure that said: "Look. Think about TCP rather than DSP." The folks down there said: "Look, we've paid Dave Farber a lot of money to develop this thing out at UC Irvine called the Distributed Computer Systems and we don't quite understand what it is, but if you're going to go off and develop a network, from a point of view of ARPA coherence, we'd like you to take the stuff that Farber did unless you have a reason not to. So either take the stuff that Farber did or tell us why not." And we went off and look at it for a long while, and in fact it appeared that it was easier to take that then it was to take the stuff from Xerox, because Farber had this thing plugged into Unibuses, which was the stuff we had to plug into. If you went out to Xerox, they had these funny cards that plugged into Altos, and they didn't do us any good. They had transceivers, but that wasn't the deal. We had to build these control cards. We looked at it and we actually said: "Hey, you know, we think the ring is intrinsically superior, and we actually think that if we took the Farber stuff we'd be further along." So right there is the seeds of the split between the AI lab and the LCS Lab, and the AI Lab turned out to have -- they have always had very good tools for hardware development up there, and they managed to get Unibus cards up quicker than we did. So there was a lot of pressure from each of us to use the other's technology, and that went on until it was OBE. I mean, we never resolved that. The thing that resolved it was Ethernet coming along

as a commercial standard, in which case, our use of rings and their use of CHAOS Net sort of faded out, I mean the hardware sense.

Pelkey: Now, let me try to be more precise at this moment in time, ARPA said MIT --

Clark: Not MIT, Lab for Computer Science.

Pelkey: Lab for Computer Science. We will pay you to do a study comparing what Farber's doing and Ethernet, in terms of performance characteristics and how it works and the properties of it and so on, and so therefore funding to go ahead and put this network in and to do work on a network was a function of doing a comparison study. They weren't going to pay you just to put a network in.

Clark: Nah. No that's not the way to put it. That's not right. DARPA was paid -- played very loose. You've got to remember that, if you compare DARPA and NSF, you know, the degree of constraint is very loose and, the way we did with DARPA is once every year we submit a formal proposal to them, and we describe everything we did in the last three years and it doesn't matter whether it went in the proposal, if it's really great stuff they say: "You're wonderful." And, so the discretion that I'm describing arose outside the context of a formal proposal at all. This is in the context of me going down and talking to a program manager and saying: "I want to do foo." And he'd say: "Done." and I think the entire conversation is more or less exactly as I put it to you. There wasn't a study or anything else. I went down there and I said: "Look, we really need some local net technology," and he said to me, it was either Kahn or Cerf, I don't remember which one, he said to me: "I don't want you going off and reinventing the wheel. If the ring net suits your needs as a starting point for your study, then use the ring net. If the ring net doesn't, come back and tell me." That was probably the full extent, it was probably about two sentences, and I said: "Yeah, that makes sense." So we went off and talked to Farber, and we said: "Yeah, ok." We actually sort of got interested in the ring. So we went back -- I'm sure we went back - I'm sure I've sent a note to Vint on the net at some point saying: "We're going to import -- " I'm sure we did, because Vint, because somebody had to turn around and tell Farber to deliver to us ten prototypes. Ok, and he delivered to us ten prototypes, so yeah, there was some DARPA communication in there, but it was never as formal as, “do a study.” I've never dealt with ARPA on that basis. You send a note on the net and you talk to them and, you know, we've always maintained a very good and evolving working relationship, and I've had a particularly good one because I've run, of course, this was actually before I was really seriously going into programs, so I guess that statement's out of time, but yeah, nonetheless, we've had a good and very informal working relationship in our program management, and I think that's the right way to do it.

Pelkey: Now at that point did you go back and try to look at what Newhall and Pierce had done?

Clark: Nah.

Pelkey: You had just taken their net and gotten the token ring, Farber's network, and then you started to say: "Wait a minute, Farber did some things wrong here."

Clark: Maybe we did, maybe we did. I mean I certainly read those patents. Now when's the first time I read it? Let me think? Sorry, I just can't answer that question. I don't remember when the first time it was that I went back and looked at those patents. I mean Farber, Farber was interested in them, but primarily our reasoning was that Farber had put some things into this technology. There were two things that we thought about it. One was he had this associative address matching in there, and it was incredib . . .

Tape Side Ends.

Clark: Today you wouldn't think of it as complicated, but back then it took about, it took a board space about that wide and about that -- it was just horrible, all this associative addressing. And we said: "You know, Dave was trying to build this very complicated distributed system in which you could broadcast your things and the right one would answer, and we think that's a bunch of bullshit. We'll just strip all that out."

And the second thing was we didn't like some of his strategies for the analog aspect of the network engineering: token regeneration and synchronization and that kind of stuff. So we basically -- I suspect we went back and looked at the patents. I think we did, but I couldn't say. Certainly we took as our engineering starting point what Dave did. Then we extensively reengineered it. The first prototype that we got came in with yellow wires, and by the time it was done, you couldn't find them for the blue wires. I mean, it was just completely rebuilt. It was also -- he had some bugs and things like that. And we experimented with figuring how to generate wire lists here and send them out to a wire rap shop, but finally screwed it. It was a real pain. It was actually a disaster.

Pelkey: How'd you pick Proteon? Proteon built the boards for you.

Clark: Want the true truth? The first version of the boards was -- we called it the version 1 LNI. V1 LNI was a one megabit token ring which was not a card, it was a 19 inch box with this flat thing in it. It was awful. And at the end of that we said: "You know, this is a good enough idea that it deserves a reengineering." But we knew, based on our experience with the one megabit, that we were not technically competent to do the kind of reengineering and stuff that would lead to a manufacturable board. We just knew we couldn't do it. So we were casting around for a -- we were casting around for somebody who could collaborate, and again, we didn't think there was any commercial potential here. We saw the Ethernet coming and we just didn't quite understand that we had much of a chance of doing something against the Ethernet. We didn't understand the IBM connection that might arise. Proteon came in for a very simple reason, which is we didn't think about it very hard. It turns out that Howard Salwen is an old classmate of the director of our lab, and it was a -- Why don't you call Howard? He's a -- he knows how to do this stuff. You see, Mike, our lab director, Mike Desutos knew that Howard's company was very involved in communications, and in particular that Howard was very strong in the analog engineering skill, and he knew that the crucial aspect of getting this ring to work right was to have somebody who is not just a digital electrical engineer, but who was an analog engineer, and so I think Mike was very anxious to make sure that we succeeded by finding somebody who would match our skills. And so Mike played matchmaker and basically said: "You know, I know Howard. I've known him for along time. He's a good guy. He runs this consulting company. We'll give him a contract for $100,000 to do a version 2, so called, a Version 2 LNI for the Unibus." And so we wrote him a contract. We didn't do any competitive selection or anything like that. It was sort of a -- we sat around a table one day and said: "Well, why don't you meet this guy and see if you like him." We went out and met this guy and said we liked him. Gave him a contract.

Pelkey: While this was happening, what, end of '78, beginning of '79?

Clark: I'd have to go find my notes. I don't know.

Pelkey: Do you remember the beginning of -- Bob Metcalfe came on as a consultant, or worked, had some -- MIT just allowed him to come here and --

Clark: When he was finishing his thesis at Harvard, they had no environment for him to sit in, so he came here and sat around here while he finished his thesis, and we published his Harvard degree as an MITR. And I think he was around a little bit from then on, but more or less after that he went to Xerox.

Pelkey: Right.

Clark: And then after that he went to 3Com. So he was only here, really, except for one brief visit back, he was only here during his PhD days.

Pelkey: Do you recall -- I'm led to believe that the first five months of '79, leading up to some kind of a conference that was going on at --

Clark: Yeah, you mentioned that conference before. I don't have any recollection of it. It's just gone.

Pelkey: Jerry remembered it.

Clark: Ok, fine. That's good.

Pelkey: He presented a paper there. It was the first local area networking conference, sponsored by MITRE.

Clark: Oh, wait a minute. Oh, wait, let me think a minute. MITRE. Is that the one that Lee LeBarre ran.

Pelkey: No, no. (unintelligible).

Clark: Oh, maybe that's the Lee LeBarre thing.

Pelkey: And there was debates about Ethernet and token ring and --

Clark: Yeah, that's lost in the mists of time, but, yeah, there was an early networking conference that MITRE ran here in Boston, I don't remember where it was. It was run by a guy name Lee LeBarre and you had to go to it, but obviously in my own mind it was not a pivotal thing, because it's completely gone. But if I went, I had to have proceedings, unless there weren't any.

Pelkey: Is he still at MITRE?

Clark: Oh, yeah, he's still at MITRE. L E B A R R E. First name is Lee. Lee LeBarre. I think he's the guy who did it. He certainly ran some workshop back there. Hang on a second, there's one other way I can do this.

Pelkey: This is all in September, 1977, all this stuff in terms of DSP and TCP/IP?

Clark: Yeah, right. The first memo in the series is November 15th, and then this memo by Dave Reed, which is the first memo, is November '76, that's on -- this describes DSP, I think. Yeah, proposed Data Stream Protocol, and then you have us noodling this through a series of memos here, into the first half of --

Pelkey: So, local network note, 4, 5, 6, 7 and so on here, dealing with DSP and TCP --

Clark: That's right. Revision of the DSP spec in June '77. Initial connection mechanism in, and then we get to September '77, and some other memos. This one on contention ring was September '77 from me. Some more status reports… Now here you come to a series of reports by Liba Svobodova, who was then a member of our faculty, and this is September '77, and its a comparative study between the Ethernet and the ring net, and then there's another one --

Pelkey: And its spelled L I B A S V O B O D O V A.

Clark: Right. And here's another one by the same woman, October 14. It's models for performance studies in the Ethernet and the ring net, and here's another one. “Comparative Study of the Ethernet and the Ring Net, Measurement Techniques and Tools.” This was us basically trying to look at how this ring net would work. Now, let me get some other chronology here for you. Liba currently worked, worked at MIT at that point. After that she left MIT and she went to work for INRIA for a year, and then she went to IBM Zurich. Now, Phil Janson wrote his PhD here in '76, and after this he went to IBM Zurich, J A N S O N --

Pelkey: Phillipe?

Clark: Phillipe Janson, now, so he knew in 1976 that we were looking at ring nets, and its my belief that when Zurich was rooting around to build a network, and they didn't want to build an Ethernet because they had a strong dose of "not invented here," that Janson was the one that suggested that they ought to build rings, based on the fact that we had exported a grad student. Now, the verification of that is to go

back and find a paper that Saltzer wrote called "The Eight Differences Between the Z-Ring and This Ring."

Pelkey: Oh, I didn't know -- that'd be a great to --

Clark: Yeah, well, I got to find it. This is "Differences Between the IBM Research Z-Ring and LCS Version 2 Ring." Recently, IBM Zurich lab published several papers. The paper emphasizes the differences between the two designs with an idea of tracing origins, discussing implications. Emphasis is quite misleading, however, because, in reality, the two designs are much more similar to each other than either is to, say, the Cambridge Ring, the Century Data Bus, IBM Series One Ring, or even the Token Passing Prime Net and Apollo Rings. So, you know, certainly when Jerry wrote this he had it in his mind that these two were the primary -- were closely related, and he gives credit in the back here to discussions with -- in Zurich with Karl Kuemmerle, Ernst Rothauser, Hans Mueller Harry Rudin and Phillipe Janson. Ok, so it was my belief that, that the lineage of this thing was that we, that the basic idea of the ring was exported to Zurich through Phillipe, and there was a limited amount of interaction with what we were doing, which caused them to pick up some of the basic ideas, and then they went off and pretty much did their own thing.

Pelkey: Could we ask Jerry if I can get a copy of this?

Clark: Sure. Let's give him a call when we're done here.

Pelkey: Great, great. Is Phillipe still at IBM do you know?

Clark: Absolutely.

Pelkey: In Zurich?

Clark: Yeah.

Pelkey: So he'd be a good person to ask that question to?

Clark: Oh sure, absolutely. No, he could give you a great deal of -- he could give you a lot of the history here.

Pelkey: Good, good.

Clark: I recently had to reconstruct some of this because I got a phone call from somebody inside IBM, Paul Green, I don't know whether you know Paul. Paul Green is one of the old timers -- you want an IBM perspective on networking and communications in general, Paul Green is a great guy to talk to.

Pelkey: Where will I locate him?

Clark: Well, he flits back and forth between being in research at Yorktown and being at the staff level for somebody down there. You know, he got various assignments down there. I could find his phone number for you, probably.

Pelkey: That would be great.

Clark: But, he asked me if I could explain the lineage of this whole stuff. It's not for publication, but what they were considering doing was giving . . .

Tape interrupted

Pelkey: Anyway, when did you become aware of Soderblom?

Clark: Only -- in the overall stream of things, very recently.

Pelkey: Right, so that wasn't an issue back then.

Clark: No, no, no, not at all. Nobody -- none of us --

Pelkey: Patents and all this kind of nonsense?

Clark: No none of us knew about Soderblom. We thought about whether there was something patentable in the ring and I think we concluded there wasn't.

Pelkey: Ok.

Clark: We knew about Farber and Newhall and that kind of stuff.

Pelkey: I want to come back to mail. When ARPA wrote its specifications, there was no mail. As I understand Ray Tomlinson and a guy who worked with him to do TENEX were the sources -- that was where mail came from.

Clark: There was, there was computer mail. There was mail -- there were mail systems running on one computer, but there was no distributed mail, and the initial vision of the Arpanet was as a resource sharing vehicle. In other words, I want to get to that computer over there, and what it actually started out being used for was remote login, as opposed to resource sharing. And then inter-computer mail caught on, and it was just this explosion.

Pelkey: Which was a great surprise.

Clark: Oh absolutely, yes.

Pelkey: Why -- why wasn't it thought of or projected, and why was it such a great explosion?

Clark: Beats me. I'm almost too young. You know, I was -- I was in a foxhole, not talking to anybody, doing my thesis in 1973 and then I came out. But I think by 1973, we already had --

Pelkey: Mail was around.

Clark: -- we had inter-computer mail, so I'm almost too young. Go back to -- go back. If Kahn can't tell you --

Pelkey: I have a lot of people's views on it.

Clark: Yeah, right.

Pelkey: Without allowing you to "X" out of this question because you're too young, why do you think -- why did the guys who were in the traditional data communication business, the modem guys, why did they miss local area networking?

Clark: Well I think there are two reason, and I'm fluffing that, I don't -- one of them is they didn't know anything about hosts. They only knew about terminals, and you know, I notice here you have this heading called "Terminal to Host Networks," and I think that's an inappropriate heading for that chapter, because I think that both Arpanet -- I think none of these four things here --

Pelkey: I think I've changed --

Clark: None of these four things had anything to do with terminals to hosts, they had to do with hosts to hosts, and the modem guys didn't know that hosts talked to each other, and you can see that in all sorts

of contexts, most obviously SNA. I mean if you ask what was the predominant vendor, who do you pay attention to, the answer is IBM. And you look at SNA. SNA is a 3270 concentration net for the first eight years of its life, and so the thing that IBM was doing was not persuading the modem vendors to look at host to host. I think the other reason the modem guys missed it was that the modem grew up thinking that their technological niche was optimizing the use of scarce bandwidth, and -- I had this wonderful conversation with the folks across the street. We have this theoretical network group across the street, that's the Laboratory for Information and Decision Systems, I guess I was talking to Bob Gallagher one day, and I said, this is back in the late '70s or something, I said: "There's this wonderful stuff," and of course those guys turned out to do a lot of consulting with Codex and they have ties into the traditional modem business, and I said: "There's this wonderful thing coming along, this local area networking," and I started explaining about how it all worked and I made some comment about the interesting design space that was opened up because, since bandwidth was cheap, you could now use bandwidth as a way to lower your system costs, as opposed to optimized. And Gallagher said: "Yeah, I understand that but, you know, unless what you're trying to do is optimize it, I have no tools to bring it to the table. My whole business is taking a scarce resource and optimizing it, and if what you're doing is what you might call a global system engineering in which bandwidth is a cheap thing, none of my tools are relevant. My entire bag of tricks has just gone out the window." And in fact, those guys dropped out of the business. And if you look at what they've been able to do, and I shouldn't point across the street because those are high class practitioners of the trade, but if you look at what the low class practitioners of the trade have been doing, they've been writing papers which say that if you were to change the allocation mechanism, you could increase CSMACD by 3%. That's a bunch of shit because you could also make a net 3% faster for zero dollars, and so those guys over there were smart enough to recognize that their tools had nothing to do with this, but if you go off in the modem place, I think that they have the same mindset, except they were in the product business, as opposed to the research business. They just missed it. They sort of looked at it and said: "Oh, local area nets. Cheap bandwidth. That's not what we do." And it only makes sense in the context of host to host, and so if you saw what they were doing, they got this terminal to host model, and you've got this optimized expensive bandwidth model, you've got two things going against them, I think it's easy to see you could end up fucking it up.

Pelkey: There was a real paradigm shift here in the jargon of the day, relative to -- in terms of data communications, in terms of predominance now of the upswing of the LAN companies and the increase in LAN companies and growth rates that, where the modem companies are, it's a flat business, it's a declining business and --

Clark: That's right.

Pelkey: And now there’s lots of mergers going on because the old datacom guys are trying to buy the new datacom guys.

Clark: Sure.

Pelkey: Micom is buying Interlan and DCA is buying Fox, and we're going to see more of this.

Clark: Sure, of course. They've got money, but they've got a moribund field. The -- you know the -- the PC -- if you try to go back and you look at the deep reasoning, there was also a real sleeper inside the PC. The thing that made local networking really happen was the PC.

Pelkey: Right.

Clark: But people's initial model, you might think people sort of suffered from intellectual over-shoot is what happened. They were suffering from bogged down grubby timesharing systems in which everything was slow.

Pelkey: Yes.

Clark: And then these guys came along, these kiddies came along and they build these toy personal computers and they sucked. I mean they had these lousy disks and 16k bytes of memory, but they had one good thing, which is they had this incredibly high bandwidth path to the display that could do this neat stuff, which of course caught the vision of the folks who did Software Arts. You know, and they said: "We can do this wonderful spreadsheet program." You know Frankston and Bricklin and all these guys, and they picked up on this early Apple, which of course was a dreadful machine in every respect except one. Of course if you go back and look at the Xerox Alto, it was a dreadful machine in every respect except one. It had a lousy address space and I thought I had re -- I thought I had gone back a decade and a half when I tried to program the Alto. I was thinking about overlays. I mean it was terrible. They put of every ounce of energy into this -- into this -- I mean they had to build something they could clone. I mean they had to have enough copies to give one to every researcher and so they said: "Well, we got to spend the money. We GOT to spend the money on the display." And so everything else, you know the 64k byte of address space and a slow disk and lousy I/O architecture and so forth, but boy did it have a fast display and a mouse, and that kind of shit.

Pelkey: Wasn't Xerox also in the process -- they were in the document business. The parent was funding them --

Yeah, do a sophisticated document on screen and they'll see bouncing graphics and --

Clark: Well, they had that, but they also had Allen Kay and the Dynabook fantasy, which has nothing to do with Xerography of documents or anything like that. I think they were driven as much by the Allen Kay Dynabook fantasy as they were by any product, but yes, of course, I don't know. You ought to go talk to somebody at Xerox. But talk to Bob Taylor.

Pelkey: Right.

Clark: Incidentally, have you read Bob's book? Everybody thought that when the personal computer came along, what he had done, is we had emancipated the user from the timesharing system and we had somehow made it possible for you to have your own computer, just as you have your own calculator, and they missed the fact that the essence of any real computing, as opposed to toy stuff, is access to information and information is not here, so you have to be able to reach out and get it, and so I think that everybody who built a personal computer missed the fact that networking was an inevitable component of the personal computer. They just didn't see. As a matter of fact, when the first products came out that were network products, they weren't information access things, they were expensive physical device sharing things. They were shared disks, not shared files, and only after people built these shared file systems did they recognize that what they really doing was not sharing this expensive disk, they were recreating what timesharing had given you for nothing. And, you know, I think we may see this thing swing back the other way. I think that we may actually see, at least in the research community, that we're going to go back to large centralized servers, and the thing that'll be sitting on my desk is a very sophisticated display manager, and it may be more powerful than a VT100, but none the less the shift of computing may go back toward where the information is. But it took us sort of a pretty hefty crystal ball to see that.

Pelkey: Yeah.

Clark: I mean we didn't see it. When we did TCP, you know we wrote this paper on the desktop computer as a component of an information system, or whatever the hell the title is, people didn't believe it.

Pelkey: Yeah

Clark: You know, they actually said: "No, no, no. You know, it's just not going to be that way." We were offering it as a capability that the machine could be. You know, even then we didn't have what it took to say: "It's got to be."

Pelkey: Right.

Clark: You know, so, if you're in the terminal business -- I'm still thinking about why the modem people missed it -- if your in the terminal business and you don't think about PCs and you don't think -- it's really easy to see that, you know, you just couldn't look through all of these options that everybody got wrong and recognize that local area nets were going to be such a critical component of the distributed environment, of the personal computer environment.

Pelkey: Yeah, yeah.

Clark: You had to understand about the economics of networking and cheap bandwidth and what personal computers were all about, which is not to say what computers were about. So, I really can't fault the modem people. You know communications and networking have turned out to be very different things and I, you know, I don't think they should be blamed for missing the local area network business, as if they were standing right there. I really don't think they were. I think they were standing someplace else.

Pelkey: Yeah. You've been incredibly helpful, and you've answered all the question on my list.

END OF THE INTERVIEW

