Interview of Dan Lynch

Interviewed by:
James Pelkey

Recorded: February 16, 1988
Cupertino, California

CHM Reference number: X5671.2010

James Pelkey: Why did you start ConneXions?

Dan Lynch: Well –

Ole Jacobson: I think he wants to know why we started the company.

Lynch: It's all the same. Well, no, not really. I started this business -- when did I really get the idea? Summer of '85, I looked around and I said: "Gee, whiz. The world is really jumping on this TCP/IP stuff, it appears to me, you know, vendors and users and all that jazz, and they just grabbed the technology and they're running it in local area net environments," you know just on Ethernet cables and broadband cables, and it works for them. It's providing them some multi-vendor inter-operability, ship some files around and things like that, and I said: "but these people don't know where they're headed." They grab just the “teeny tiniest” part, ok, and they don't know what they're headed for, which is internetworking. I mean, that's what TCP/IP was built to address, and I knew -- I said to myself: " Gee whiz, these vendors," I know I was looking at the vendor side of things, in the sense they're the suppliers, and I said: "they don't know what -- they don't know what they've grabbed. I know they don't," cause I've talk with some of them and I've done some consulting for some of them and they were technically adept at a lot of things, but they didn't have the marketing picture, really, you know, as to what this stuff was all for. And I knew that they would end up being in trouble, because internetworking is a lot harder than networking, and you will lose your shirt, financially, right. You will have unhappy customers and you will end up losing money trying to support the product unless you build it right and or charge for it right. You know, you can sell someone a complex product, in fact, if they pay enough so you can support.

Pelkey: That's right.

Lynch: And, well, so I went around trying to get consulting jobs with various vendors, with that as my sort of introduction, and I didn't do well. I did NOT do well saying: "Hire me. You're headed, you know --

Pelkey: in the wrong direction.

Lynch: -- in the wrong direction, and I will fix things for you, ok?" And it was real hard to convince them of that when they're making money hand over fist at the moment, right, doing the simple part, skimming the cream, right. I mean I had already been down that whole road, I mean, I'd been 15 years in networking development, and, you know, 15 years or whatever it was, and so, you know, I sort of knew where they were headed and they didn't. And I didn't get very far with that, and so I said: "What I got to do is get all the people that develop all this stuff to come out of their ivory towers and explain to these guys, ok, and then they'll believe me, and then they'll think I'm smart, and then they'll hire me to be a consultant and on and on, and I'll go help and develop, you know, specific product marketing plans and all this sort of jazz." So that was the, you know -- things don't end up, you know, probably, where you aim. And so that was my plan. And so how am I going to do that? Well, I got to get these 12 apostles, you know, of TCP to come out of their ivory towers, and they're all friends of mine, people I've worked with over the years, I mean -- I mean my job in the R&D community, when I was really deeply in for like ten years, from '73 to '83, was to make it work. I was not an R&D’er, ok, particularly. I ran the computer centers for the R&D, for the computer science researchers, so I basically ran the boiler room for the boiler room designers.

Pelkey: Right, right.

Lynch: A dangerous place to be.

Pelkey: Just make it work, don't bother me --

Lynch: Or change this, change that, boom, boom! Ok, here's a new pipe -- here's, you know, new software and new hardware, integrate it. Fine, don't -- and don't let it crash and save all our files and all that sort of jazz.

Pelkey: And what centers were you --

Lynch: Well, I was at SRI, Stanford Research Institute, ok, for -- in the '70s, and ran the artificial intelligence computer center there, and then -- and then in a fit of megalomania, decided that I could run ALL of computing for SRI, ok, and I set upon a campaign to create that political reality. Ok, it took 18 months. I actually succeeded. Ok, and it was a pyrrhic victory. I didn't want that job. It was a terrible nightmare.

Pelkey: Back in the masochistic days.

Lynch: Yeah, right, you know, sometimes, you know, you can actually get what you aim for and you might not want it. And then, in the early '80s, I was down at University of Southern California Information Science Institute, it was called ISI, you know, USC-ISI, and which was the biggest node on the Arpanet, and it was the place that essentially ran a bunch of big timesharing systems that supported thousands of remote users to get them to use the Arpanet technology, and -- so I was the manager of that facility, and it was in that era when the decision was really made, ok, we're really going to convert everything to TCP and get rid of this NCP stuff, and so I was the guy that was the manager of the transition, to get all that to happen for ARPA. I mean, there were two pieces of it. One of them was to get the backbone running, and that was BBN's contract, ok. And then there was this other small part called get all the software running on all the systems, on all the hosts, all the end systems everywhere, and that was my job, was to make that happen for ARPA.

Pelkey: And, if I recall from what Vint was telling me, you had to go to some great lengths, in terms of, turn the system off for first a day, and tell everybody: "If you don't get over to TCP - - "

Lynch: We had to use the carrot and the stick approach.

Pelkey: Do you recall those dates or when that was? And then you had to turn that off -- eventually you turned it off for three days. I guess there was even some buttons -- someone created some buttons "I survived TCP -- "

Lynch: Oh yeah: "I survived the TCP transition 1/1/83 buttons." yeah, right, I got it.

Pelkey: I have before me "I survived the TCP transition 1/1/83," in red, and Dan Lynch made up 500 of them out of his own money, so I --

Lynch: That's right.

Pelkey: Was that the final last day of it all?

Lynch: That was the day that they switched, was the absolute cut- over day, ok. Now only a bureaucrat would pick New Year's Eve, you know, and so you got a hundred system programmers half happy, right, or unhappy or whatever, trying to throw the switch, you know, around midnight, and that was the day the final switch was thrown. I mean, there were actually, you know as Vint told you, a number of prior dates in the previous year where we did test cases and things, ok.

Pelkey: So during '82?

Lynch: Yeah, during '82 is when we --

Pelkey: Do you remember the first one, when you turned it off the first day, roughly?

Lynch: No. I mean, I think we fooled around in the early part of the years with just -- well, see, the subnet, the Arpanet had the ability, its the thing that passes all the datagrams around, you know, all these hosts hanging off the edges, and of course, you know, what we did with a lot of the hosts was, you know, bi-lateral or multi-lateral agreements. You know, hey you and I, we're going to speak TCP only for the next couple of hours, right, and you didn't need anyone's cooperation to do that. And, in fact, that's always true, but in order to coerce people into this corral, you know, BBN put special code in the little packet switches. It could tell a TCP packet from an NCP packet, ok, and it just, I mean, oh NCP? No. Ignore, ignore. Just say no, ok? And so, you know, those packets stopped. Now I don't remember the exact dates. I mean, we did something in the spring of '82 and then we started getting serious in September. I remember when we really started getting serious in September and had sort of TCP only half days and then whole days and then there was -- and there was sort of a monthly event and then I think we did a three day one around Thanksgiving, and what we kept discovering, I mean what we discovered is, lots of problems. I mean, there are two things that go on when you do a thing like that. One of them is that you find bugs everywhere, and that's great. I mean, that's really what it's for, and there's a third reason which is: "It really is coming, folks," I mean that's --

Pelkey: Now, did everybody do their own -- all the nodes do their own implementation of TCP/IP from a spec that had been approved by this networking group?

Lynch: Ok, no, let me finish for a second, ok. And the other aspect of it, besides the -- oh, besides the finding bugs thing was: "Sorry folks. You got this system, and you just -- " that moves files around and let's people log in and sends mail and all that jazz and now you've broke it for a day or two, ok, and you tried to use this other system to carry the load, and it carried some of the load, but back in the hosts and everything, you were unable to -- you know, if your host was sort of semi-broken and couldn't send stuff via TCP or the TCP was slow or whatever, the hosts started to fill up with requests, with files, you know, all this sort of jazz, and all the host NCP software would break, because it was now seeing limits of load on it that it had never seen before. You know, artificially induced limits, ok. So now you're sitting here going: "God damn it. I got to sit here and debug programs that are going to be thrown away in less than a year? I mean -- "

Pelkey: Why are you doing this to me?

Lynch: Why are you doing this to me? Exactly right, you know, so with that in mind, we would -- you know, we would just try to limp the stuff along, right, you know, but we would have to -- it was amazing all the little funny things that we discovered in the NCP software that were, you know, queuing type bugs or assumptions about offered load and type of service and all that jazz, and most of us ended up fixing the TCP implementations rather than the NCP implementations, I mean, let's make sure that the future is good, cause, you know, I mean, we've learned some lessons, but let's don't bother applying them, let's just live through it.

Pelkey: That must have been a painful process.

Lynch: That was tough, ok.

Pelkey: And someone said that people not being able to get there mail messages --

Lynch: They get very angry.

Pelkey: They got very angry.

Lynch: Oh, yes. Electronic mail is the amazing discovery, right? I mean, we built the damned Arpanet, not for electronic mail.

Pelkey: Right.

Lynch: We built it for remote login and file transfer, ok, access to remote resources that was its absolute thing. E-mail was an accident. A happy accident, but you know, I remember talking to a gal on an airplane about a year ago, just a random seat mate, coming back from Washington or something, and she -- you know, just BS'ing: "What do you do?" "Oh, I play with computers, networks and things." She was a geologist and she was just moving to Stanford doing a post-doc, that's right she was from Columbia, and so I asked her how the field was, what kind of geology and it was sort of moon rock geology and extra terrestrial geology and all that sort of jazz, real interesting. And then she just volunteered, she said: "Electronic mail. It has completely changed our field." I said: "What do you mean?" She says: "We're no longer working on old problems or the same problems. We have E-mail all over the world, you know all the geologists in my specialty, and we all know each other and we all communicate and we're just working on new problems instead of old problems." Isn't that fabulous?

Pelkey: Oh, what a great thing to come back with --

Lynch: What a great thing to come back and realize: "Golly!" I mean, that was part of the idea of this whole thing, communication. You know, just get people together that have a common base of knowledge and then they can work on other things, rather than the same old things.

Pelkey: I had an interesting conversation with Paul Baran at Telebit. He refers to Arpanet as an experiment that went bad.

Lynch: A successful disaster.

Pelkey: It got too successful. It was meant to be just an experiment. It got a life of its own and it wasn't really supposed to have a life of its own.

Lynch: I know. Now you asked earlier about, did we have one common spec or something like that. The answer to that is, yeah there was always a spec, I mean, the spec actually got written pretty tightly in the mid '70s.

Pelkey: It went through, what, four versions. It was the fourth version or whatever that --

Lynch: The fourth version is the one that settled out, yeah, right. And versions 1, 2, & 3, you know, version 1 was the first one and it hung around for quite a while, till we got it to work, sort of right, you know, and then versions 2 and 3 came and went rapidly, because they were just sort of minor improvements, or something like that, you know, and then we started calling them version 2 and version 3, and that was simply because of the changing packet format. So the only reason to have a version number is you change the packet format and you don't want some poor receiver out there seeing something and not knowing what to do with that, so you just say: "Versions number. Oh, I don't speak that. I don't trust the rest of this data." You know, it just preserved researchers from killing each other.

Pelkey: Now, some of -- if I understand correctly, that TCP/IP, what Joy did over at Berkeley, getting it into 4.2 --

Jacobson: Well, 4.1A or something.

Lynch: Well, one and two.

Pelkey: Which is, do you recall what year that was?

Lynch: Oh I know that whole story. That was 1982.

Pelkey: That was a really seminal event in TCP/IP. Everybody got --

Lynch: Everybody got the ducks.

Pelkey: Before that was kind of in the ARPA community.

Lynch: It was in the ARPA community.

Pelkey: There was a lot of local area network playing around with it, but there was no market.

Lynch: There was no market. I mean, there was just the government saying: "You got to do it, God damn it," so Sperry was out slogging along trying to make an implementation, you know, and CDC. You know, the classical --

Pelkey: DCA was --

Lynch: And DCA, you know, was saying -- whipping that, you know, drawing that whip and they let a bunch of contracts, seven contracts or whatever to various vendors, to do a TCP implementation, get into the government marketplace, and everybody was kind of: "Oh, well, shit." And they'll do it, and they did it, and some people took money from the government and some didn't. I mean, IBM said: "We'll doodle, but not take the money cause, you know, we want freedom." And so anyway, Bill was building Berkeley UNIX. I was a member of the Berkeley UNIX steering committee, so I have some knowledge of the politics and all that jazz, right.

Pelkey: Now were you --

Lynch: I was at ISI.

Pelkey: But you were on the Berkeley steering committee.

Lynch: That's actually an oxymoron. The Berkeley UNIX steering committee!

Pelkey: Who else was on that committee?

Lynch: Oh, Vint and Dwayne Adams from ARPA. Dennis Ritchie from Bell Labs. Rob Gurwitz from BBN. I'll explain more about him in a minute. Jerry Popek from UCLA. You know, a few other people, three or four other people. I mean, Bob Fabry who was really the faculty member who sort of headed the whole thing. Steering Bill Joy, the words don't come out well.

Pelkey: The government forced the issue? Putting TCP/IP in --

Lynch: No, no. There had already been a TCP/IP for VAXs. Bob Gurwitz had done it at BBN under contract to ARPA, and they had done the official blessed ARPA version, but Bill didn't like it, and the government said: "Here, take this code. You're building Berkeley UNIX, but take this special stuff, this networking stuff, and slam it in, guys. Here it comes from BBN. Safe drugs, right?" And basically Bill worked with the code and got the idea and said: "I'd rather do it myself, mother." He can type faster than anybody and most of the characters are correct, and so he essentially insisted upon doing it his way, rather than the BBN way, ok. And a battle ensued as to who was going to get it. What code was really going to go in Berkeley UNIX? Bill Joy won; it's as simple as that.

(Telephone message interruption)

So Bill won that battle, and so we have, you know, that brand of TCP, instead of the BBN brand of TCP in the Berkeley UNIX. And the truth is, Bill's code was a little buggy. I mean, the stuff that came from BBN was better, better debugged. And the world has suffered a little as a result of that, you know, because people will read the spec and build something and then bang it against the most popular system, Berkeley UNIX, and --

Pelkey: Which is broken.

Lynch: -- which is broken, slightly, and you have to adjust. So, pain in the neck, but it certainly was a success, good old Berkeley UNIX, and it spread TCP far and wide. I mean, then Bill bailed out of there and went to Sun and said: "I don't think I need a PhD."

Pelkey: The same thing kind of happened when DEC started shipping Ethernet with their VAXs. They just decided: "We're going to just ship an Ethernet board in the VAX and -- So anyway, the government, at that point in time, really just said that they wanted to create a version of Berkeley UNIX -- they wanted Berkeley to do the UNIX so they could get out into the community, have one standard form of UNIX.

Lynch: Yeah, they wanted an improved version of UNIX. I mean, they created Berkeley UNIX because UNIX coming from AT&T was limp. And it needed the features of TENEX, which is a system that TCP was originally developed on. TENEX was a PDP-10 timesharing operating system, ok, that came from BBN, and it was a virtual memory version of the DEC 10. It was an excellent system. My license plate says: X-TENEX, so I'm very biased, but it's an excellent system technically in the marketplace that didn't win. Hey, life is full of those, you know. But what ARPA wanted to do was to take UNIX and take the best of TENEX and slap it into UNIX, and then spread that around, and that's what they did with Berkeley UNIX.

Pelkey: So you had, at that point in time, you had TCP/IP that was in the kind of ARPA defense community, that the government was kind of forcing --

Lynch: It was running under Multix and was running on DEC systems and it was --

Pelkey: If you wanted to supply the government, you had to have it running on a system that ran for the government.

Lynch: Right.

Pelkey: And then, all of a sudden, it became available on this Berkeley version of UNIX, which started to get out into the commercial sector.

Lynch: That's right.

Pelkey: And then --

Jacobson: Got ported to thousands of boxes, you know UNIX based boxes.

Pelkey: And a large amount of those went to this engineering workstation market.

Lynch: Right.

Pelkey: And at the same time, it got on top of Ethernet, so I gather.

Lynch: Well, yeah, that's because --

Pelkey: Ethernet TCP/IP and UNIX became so -- and 16000s and all

Lynch: Its funny, when I see people write stories and they go: "Oh, TCP/IP, that's the UNIX connectivity. It came from UNIX." Well, the first order, they're right. I mean, you know, technically they're wrong, ok, but they're not wrong. Who popularized it? I mean, take a look at the mouse. Doug Englebart built that sucker back at SRI in the '60s, ok, everyone on earth thinks that its Apple Macintosh, and they're right, you know, and Doug knows it. They're the one that made it popular.

Pelkey: I asked Paul Baran about packet switching. He said: "Well, maybe I created it but Larry Roberts is the one who really made something --

Lynch: Who made it happen, yeah, who made something out of it, exactly right. So, you have, well the technical foundations and then there's the market breakthrough.

Pelkey: What happened next after getting Berkeley out? Was there any other kind of a big event that came along in the life of TCP/IP that really brought life to it? Or was it just kind of --

Lynch: Oh yeah. HA! No. It's funny. Okay.

Pelkey: It wasn't 3Com at first.

Lynch: No, you see, all these -- all the Zilog spin-outs, derivatives, the Sytek's, well they came from Ford Aerospace but, all the line companies basically were looking for a protocol, and they chose Xerox's XNS, or variants of it, right, and because that in fact was what Ethernet ran.

Pelkey: That was a de facto standard at that point.

Lynch: It was pretty de facto, right? And so they ran with XNS to ship packets around. And, for shipping packets, XNS was perfectly fine, but if you're doing anything other than remote login, you actually have to have some reasonably higher level protocols running, you know like file transfer or mail or shared files or something like that, and Xerox would not release the specs for the application layer. And so Xerox --

Pelkey: Typical of their Altos language group.

Lynch: Yeah, correct, so Xerox just fumbled it for itself. I mean technically XNS, for local area netting, is betting than TCP, not significantly but you point to four or five features and say: "Yeah, its better, shit. Why are we using TCP? XNS is a better solution." And the answer is that these guys went out, the local area net companies went out and used XNS to do terminal-to-host services, basic terminal multiplexers on a cable, and do a variety of hosts. That beat going through one Gandalf or Micom switch, because you could actually have a variety of hosts hanging on a cable and a variety of terminals --

Pelkey: Terminal concentrators could only support so many….

Lynch: That's right.

Pelkey: So give me a cable.

Lynch: So give me a cable and somehow this is better.

Pelkey: What about Ungermann-Bass? Ungermann-Bass's product line was nothing but terminals-to-multiple hosts.

Lynch: Milking machines. Image is clear. Very accurate, you know, and then those vendors, Ungermann-Bass's and the Sytek's and the 3Com's and who else, Bridge, Excelan, started noticing that there was more money to be made -- I mean there was additional money to be made, in providing host level services, and they couldn't build it on top of XNS, cause Xerox just kept the lid on, and what's available? TCP was sitting there all along. So they just basically go one down to TCP.

Pelkey: So everybody -- where did they go get the resident knowledge, the experts to do -- there wasn't existing third party software you went and bought and ported at that point in time.

Lynch: No, you got Berkeley UNIX is what you did. I mean, if you were a vendor, the way you get started is real simple. You spend the $32,000 or whatever the hell it is, to buy a Berkeley source license.

Pelkey: Ah, and then you just took that code.

Lynch: Then you take that code and you start. It's code. It works. I mean, buggy a little, but it's 99% of the way, ok, and that's the documentation. I mean there's a 2700 page set of documents you can get, you know, but try to describe what's going on inside --

Pelkey: This is the RFCs?

Lynch: RFCs. And that's all --

Pelkey: Good, but who needs it. Just give me the code that works.

Lynch: Give me the code that works, and so that's how people got their kick-start. I mean, it's an excellent way.

Pelkey: Did they, at the same time, go raid talent? Or did they --

Lynch: There's very little talent to raid, is the facts. I mean, the dozen to two dozen who actually built all this stuff, getting back to where I started a while ago, all these people whom I know, ok, and I went to them and said: "You guys have failed. You built this beautiful thing, and the world is starting to use it, and they're abusing it, and you have failed to communicate to them what its real potency is, where it's really headed, what problems it's really headed to solve, and they're just using it for these little small things, and you got to go awaken them," and they did. They loved that. They said: "Sure." So I put together a conference.

Pelkey: Now this is in --

Lynch: August of '86, right.

Pelkey: In Monterey?

Lynch: In Monterey. I put together a conference, invitation only. I mean, I didn't know what I was doing, I just said: "Hey, we got to get some guys together."

Pelkey: And who were the apostles at this point.

Lynch: Well, the apostles were Jon Postel, and Vint Cerf and Dave Clark and Dave Mills and Paul Mockapetris and --

Jacobson: Some of the Berkeley people.

Lynch: Mike Carels, Greg Minshall, Jeff Mogul from Stanford, he was a grad student there at the time, Mark Crispin, the crispy critter from Stanford.

Pelkey: Do you have a list of those people anyplace that I could get so I could make sure the names are spelled correctly?

Lynch: Oh, yeah. I've got --

Pelkey: And that was a successful conference?

Lynch: That was outrageously successful.

Pelkey: That was two days --

Lynch: It was a three-day conference and I

Jacobson: Zero of the conferences.

Lynch: It was the beginnings of TCP/IP conferences, because it was vendors only, by invitation only. I mean, I literally just sat here and called a hundred different companies and said: "Do you want to come to this thing for three days? It's not going to cost you very much cause I got actually -- "

Pelkey: I remember Steve Holmgren. I'm on the CMC board. I helped found CMC. I remember Steve going up to the conference.

Lynch: Right, oh-oh. This looks like a suspicious . . .

Tape is shut off, then turned back on

Lynch: Has this thing been running the whole time, because (laughter) --

Pelkey: So clearly this process of wanting to have more functionality, and be able to get the source code to TCP/IP through Berkeley, and that Berkeley release came out in what year, '82?

Lynch: '82, yeah right.

Pelkey: You've inferred a couple of times -- I'd like to have you comment, if you would, on two things. One is this issue of why mail took off the way it did, because, and let me try to explain why that's an important issue to me, and the second issue is, you talked about inter-operability, internetworking, in that there's a lot more to TCP/IP than just passing some files around and doing some mail or doing remote login, and maybe you could just share that kind of vision as to the bigger or better relative to where it's going.

Lynch: Well, I like to describe it as it took us 15 years to develop the technology to ship datagrams from any place to any place, and we kind of know how to do that now, and now we have to concentrate on -- so the experiment succeeded, now we have to spend the next 15 years stopping the datagrams from going anywhere. We have to civilize them. We have to now take this technology, which we now know how to do, to connect networks in England with networks in America and arbitrary satellite and whatever, and TCP can do all that. It has the addressing structure and the architecture to deal with all that. We have to figure out how to stop data from flowing, not because that's what the world is made out of, it's made out of borders, ok, and things go across borders by agreements. And the data's been flowing just whoosh, as an open lab. Very -- no attention paid to security and, I'm not talking government security, I'm just talking commerce, you know, commercial security, and now the really hard part is going to be to do it…

Tape Side Ends

Lynch: . . . So I put some generals on and I put some sergeants on and, you know, some secretaries and scientists and all that jazz, and just started playing E-mail, and they were sort of blown away by the fact that there was no chain of command in all this.

Pelkey: There was no status.

Lynch: There was no status, that's right. We didn't even allow you to have Gen. Soso in your name, it was just Smith, damn it, right, Joe Smith, you know. General or what, sergeant? It doesn't matter. And they saw that -- the way E-mail was being built it was absolutely flat space, ok, and -- of who can talk to whom, so you have to develop a different set of protocols for when can I send a message to "X." I mean, a different ethic comes into existence there. It's fascinating to watch, and see who will send messages to whom and why and not and --

Pelkey: And why did E-mail take off? Was it just desire? Was it -?

Jacobson: Can I answer that? I think it was because it suddenly was there, and it was real easy for people to communicate sort of informally in a way. People didn't bother to correct their typos and structure their documents when they sent them to each other, because they knew it took three minutes or less to get the message there, and if there was any kind of confusion, the guy would send you a message back and say: "Hey, what did you mean?" So this whole new style of communicating developed, I think, in the Internet research community.

Lynch: The thing that's interesting to me about this is -- I said this years ago, and I haven't had this thought in a long time, cause I haven't been asked the question, what's different about computerized electronic mail? And what's different is: There is no pre-defined length of the message. It is not a piece of paper this size, it is not -- you understand? You can type a one-character message; you can type a million-character message. You're not -- there's no expectation of how long the message is by the physical-ness of the sheet of paper that's in front of you.

Pelkey: Interesting.

Lynch: I mean, it's just simple little things like that.

Pelkey: There is something very simple about it, because it was unexpected. I mean, the TIP wasn't even in the original specs and E-mail was kind of added on, and then it took off.

Lynch: E-mail was created by the two guys that wrote TENEX, wrote the TENEX operating system. I know why E-mail got created. I have the original code sitting in my garage, here.

Pelkey: Who wrote the original --?

Lynch: Ray Tomlinson and Dan Murphy.

Pelkey: Did they really?

Lynch: Yes. And they wrote it because they had a PDP-10 to themselves to do development of TENEX, right.

Pelkey: At BBN.

Lynch: At BBN. They had one machine, a BIG thing, right. Two guys, ok, one machine and they did it on shifts; I mean they basically worked 12 hours. 12 on 12 off, and sometimes they'd overlap, physically, and sometimes they wouldn't, and as soon as they got the file system to the point were it actually would stay together, they started writing notes to each other and leaving them in a place on the disk, and, you know: "I did this. This now works." Just little notes back and forth and it kept the history, right, nice -- and invented a trivial little format. In the beginning there was no to and from, right.

Pelkey: There was only to.

Lynch: The guys, they knew. It was subject, right. And it just grew like that. I mean that was the original E-mail, you know, two guys --

Pelkey: Now, other people came out and wrote other E-mails. Someone told me Larry Roberts wrote a version.

Lynch: Oh, yeah. Oh, he wrote a TECO hack, that's what it's called. Yeah, what Larry wrote was a little program to take these E-mail files, which were just appended, one message after the other, ok, and parsed it into something nice so that when you came in in the morning, it said: "Mail." It would say: "Subject, to, from," you know, just a summary sheet of your mail, I mean your envelopes, right, and just present you with the envelopes, and you sort of picked which one to type, and that sort of -- gave you some ordering.

Pelkey: Ah, so he put an envelope around it.

Lynch: He put and envelope, that's right. He managed. He put some management discipline on top of just these raw letters coming in, and he did it all in this language called TECO which is this incredible little silly language, Text Editor and Corrector, is what it was called. I mean TECO is fascinating as a command language. Every string is some command. Oh, shit.

Jacobson: What we used to do was to type your name and try and guess what it was --

Lynch: Guess what it will do. But anyway, so Ray and Dan did that.

Pelkey: Do you recall what year that was?

Lynch: That was '70.

Pelkey: '73?

Lynch: '70. '70 and '71, and then they evolved it -- then they got to where they had to build the Arpanet onto this thing, I mean glued in -- they were also doing all this putting this TENEX machine on the net, and they said: "Oh, well, let's do it so it works from one machine to another," I mean, you know they quickly generalized their local hack, one machine only hack to a network. I mean they made that leap because they were building -- they were writing the network code too. They were doing a lot, ok, I mean they were building an entire operating system, and entire virtual memory operating system from scratch, ok, designing the hardware, designing the software --

Pelkey: Did they work for Frank Herbert at that point?

Lynch: No. Frank Heart was the guy's name. He was at -- he was on the -- no, they worked for Bert Sutherland of, you know Bert?

Pelkey: Sutherland Evans.

Lynch: Well, he's Ivan's brother. Ivan Sutherland, Dave Evans, ok, and Bert Sutherland, ok, and then Bert eventually ended up at Xerox PARC as manager of the guys that built all -- a bunch of the Alto stuff. And Bert got fired by Xerox; one of the only junior VPs EVER to be fired by Xerox, and he got fired for telling the truth. The world -- scientists were trying to get Altos and Dorados and all that shit out into the world. "Come on, Xerox. Let go of that stuff. Sell it to us," and Bert finally stood up in front of an AI -- a big AI meeting one day and said, and admitted publicly, that they had the equipment, which was his crime, ok, and that they weren't ready to sell it yet. And that got him canned, for telling, you know --

Pelkey: What's he doing now?

Lynch: Well, he's consulting and investing, and all that sort of jazz. I mean, he's a -- Sand Hill address, ok. "I'm doing fine, thank you very much." They saved his ass, saved him from being some stupid $200K executive at Xerox for the rest of his life.

Pelkey: Isn't that something?

Lynch: So they worked for Bert, ok, who was building this $10k box, and then the network side -- Frank was building the network on one side, ok, and so these guys said: "Oh, we got to have our thing hook into the network too," so adding the network -- and see they were building an operating system from scratch at the time, and with the network in mind, and that's neat.

Pelkey: Oh my goodness.

Lynch: Very few times in the history do you get that opportunity, and they did, and that was the system that they built, called TENEX, and it was really neat, ok. It was a really wonderful system, but DEC sold the VAXs instead, I mean, and that's cool. God, I remember a sad story, Dan Murphy eventually went from BBN to DEC to port the technology to DEC and bring it up on a system that DEC called TOPS-20, and sold a thousand or so machines of, you know, big machines, over the years, and then when Dan finished doing that for the PDP-10 for DEC, they asked him to go work on the VAX software development team, because he had such excellent taste and knowledge and all this sort of jazz, ok. And VAX was the enemy, right, and he wouldn't go to work to make the VAX, I mean he refused to. He said: "Fuck you. I'm not going to do it." And God, Dan, and I know that story is true cause it's from Dan to me, I mean, and we're friends, you know. And he doesn't really care, cause he owns a jazz radio station in Boston, anyway, and that's what he really loves to do.

Pelkey: Two other questions and I regrettably must leave, but maybe as I get more into this, if I could bother you for some more --

Lynch: Sure.

Pelkey: At SRI, SRI was obviously one of the first Arpanet nodes, and was heavily involved, even before that, with some of this stuff, in terms of packet switching, some of the early meetings where, I forget the gentleman's name at SRI who in fact was present at some of the meetings with Baran and Roberts and these guys who was from SRI.

Lynch: Jeff Rulifson?

Pelkey: No.

Lynch: Jon Postel?

Pelkey: No.

Jacobson: How about Steve Crocker?

Lynch: No, he wasn't at SRI then.

Pelkey: This was in early '65; the meeting took place.

Lynch: Oh, Englebart?

Pelkey: It was. Or Uncapher?

Lynch: No, Keith Uncapher.

Pelkey: Ok, then it was Englebart, Doug Englebart. Who in SRI came through and came through the kind of communications group and came in contact with either ARPA or what was happening around those issues that then went off and either created little companies or went and got involved in other companies from your memory?

Lynch: A guy named Bob Daly.

Pelkey: Where did Bob Daly end up?

Lynch: Both those guys wandered off and did something with SONAR shit.

Pelkey: What about data communications in the commercial sector?

Lynch: Tom McGill went to Stanford Telecommunications here in the valley. Some of those guys came out of the telecommunications science group at SRI.

Pelkey: Clearly were a number of people went to Xerox PARC out of SRI group.

Lynch: Yeah, right. See, there actually were separate groups. Groups at SRI are --

Pelkey: Meaningful.

Lynch: Yes, it's very meaningful. Very boundary oriented, ok. Group means: "We're all on the same fucking contract, together, so we sink or swim together, right." That's group.

Pelkey: I don't mean group that way, I mean more in the general sense.

Lynch: Yeah, right, and they all lived inside this thing called the engineering division, or something, that you're talking about. Doug Englebart was a leader of one group and, you know, the NLS group that founded -- it's his group where tons of people went -- (the mouse and all that jazz that went to Xerox PARC) half of them went to Xerox PARC in '72 when they got started, and that's where Jeff Rulifson came from and Bill English was actually -- Bill English was the number two guy in Englebart's group, and he was the senior guy that went over to Xerox and really started doing heavy recruiting and building the team. They got Allen Kay, but he didn't come from -- he came from Utah directly.

Pelkey: One other question, ISO, and OSI.

Jacobson: Talk to Marshall Rose.

Lynch: No. Talk to me. I've been trying real hard for a year and a half to be, well I have succeeded in being even handed, and saying: "OSI is what people want. It's the automatic transmission networking. TCP/IP is the stick shift version. It's the prototype, and it works pretty well, but it's stick. If you want a comfortable existence you really want the spiffied up one." And if you sort of look at the specs of what they're trying to do and all that jazz, you sort of get the feeling: "Yeah, they're really trying to do it right and add a lot more functionality." There's a very -- well, that's another story. So OSI is the big hope for all of us. It's the internationalization of the stuff that just happened in the United States, and the international rules are such that you don't ever adopt any one nation's standards cause it just won't work, I mean, it's simple, politically. And so TCP is dead, because OSI is going to be the winner. Well, that's probably still all true, but not this century.

Pelkey: I've heard that.

Lynch: They're stumbling. I mean, the best example of it is a guy at Hewlett Packard, the guy in charge of protocols for Hewlett Packard, I mean, its his job, he's a big time manager. Roger Chung is his name, and he and I were together at a Congressional committee meeting about a year ago and we were trying to design this whole new networking infrastructure stuff and everything, and he stood up on OSI and he said: "OSI! I work for Hewlett Packard. We have a problem with OSI. We test our products before we ship them. There's no way in HELL we can DREAM of testing this. There are too many choices, too many options; this is not a protocol. This is the inclusive smorgasbord of everything. There's no WAY you can figure out how to test that, much less -- " So it's a hopeless mess, and it's going to take five or ten years of duking it out in the marketplace to select a subset that will eventually be the winner, and that's TCP already, in some sense. TCP is probably too minimal, ok, but it's interesting to note that the TCP community built these TCP things, ok, and essentially left them functionally untouched for over a decade. We built all this mechanism in to do remote procedure call and cooperative computing amongst the actual subroutines -- you know, not humans driving it but programs calling each other. There aren't any meaningful applications out there in the world that use that functionality. Why? I mean; is it not needed?

Is it not cost effective? Those are all possible answers you can think of. Maybe the real answer is: "Heck, what we've given humanity is good enough, you know what I mean." We're living on the leading edge of the technology and we always can think of the next thing to have. Maybe just E-mail and file transfer --

Pelkey: You mean live on E-mail.

Lynch: You can do this for a long darn time before you need Sushi.

Pelkey: With that, I thank you both for your time . . .

END OF INTERVIEW