Interview of Robert “Bob” Dolan

Interviewed by:
James Pelkey

Recorded: January 6, 1988
Menlo Park, California

CHM Reference number: X5671.2010

James Pelkey: When did you found ComDesign?

Bob Dolan: September of 1977.

Pelkey: And what did you do before that?

Dolan: Immediately previous to that I worked at ACC, which was at the time called Associated Computer Consultants, which is now Advanced Computer Communications. Worked for them for eight months. During that time, I worked on a project to build and interface from PDP-11 computers into a packet network that Citibank was building.

Pelkey: This was on a PDP-1I?

Dolan: It's what's called CitiNet. That was a private packet network. Most of the activities of ACC were related to private applications, almost exclusively government applications of private networking technology that came out of the Arpanet. And immediately prior to that, I worked for a company called the Speech Communications Research lab. That's noteworthy because, in fact, it did communications projects for ARPA, of Arpanet fame, and this particular project that was done there that I was involved with was to build a terminal front end based on PDP-11s to the Arpanet.

Pelkey: What year was this now?

Dolan: This was basically '74 through '76 at Speech Communications. What was noteworthy about that was, originally with the Arpanet, the major users, the consumers of the packet network technology were the large host mainframes, and as the experience grew, it turned out that a heavy part of the load of these hosts was actually processing terminal traffic into the packet network to be routed to other hosts, so it made sense to dedicate less expensive, less general purpose machines to the terminal processing part of the packet network. I don't think it was clear to people until probably the '74, '75 time frame that terminal usage, or actually, individual interaction with a packet network, as opposed to computer to computer interaction with the network, would turn out to be the primary use of data being transferred over a packet network.

Pelkey: When the RFP was let in '68 and awarded to Bolt, Beranek and Newman in '69, the RFP presumable called for minicomputers to be the node processor/servers interconnect devices.

Dolan: Well, yes, there was a minicomputer that was the Arpanet node, a Honeywell 316 or 516, which was a minicomputer of those days, but those did solely the packet switching at that time, and then the hosts connected into them. To the extent the terminals were connected in, that was a later development–

Pelkey: So the terminals were first connected to the host, and not directly to the minis.

Dolan: Yes, because I don't think anyone realized at that time that 95% of the traffic was going to an individual connecting through the packet network, then to a host facility, as opposed to host-to0host communications and applications of that sort. Probably still that would be the major usage of traffic going across networks.

Pelkey: I held a view that the architects of the Arpanet, the people who were involved, played a large role in the data communications industry. Is that accurate?

Dolan: Absolutely. One of the most interesting things is that packet switched networks actually got started in '69. Devices like multiplexers and so forth didn't come into common usage until considerably after packet networks were going, even though the statistical multiplexer, and even TDM’s, are obviously much simpler devices. A packet network is a much more gentle solution and really much nicer.

Pelkey: So you think that the genesis of the multiplex thing is out of the packet switching things, in contrast to coming out of the modem.

Dolan: I would say so, at least in terms of actually getting into widespread usage. You will find, actually, that there was a patent for a statistical multiplexer issued in I think it was 1976. I can probably find out.

Pelkey: Ok that would be helpful.

Dolan: Can we take this off line for a minute?

Tape Shut Off Temporarily

Pelkey: . . . Ken Krechmer's[1] comment earlier that there is a professor at UCLA who conceived the statmux, and you think the gentleman's name is Wesley Chu[2].

Dolan: Yes, at least from a patent point of view. What I think is interesting, and the reason we were off line, is because I didn't want to mention who discovered it. It is not commonly known that there is any patent on this area, so apparently it must never have been enforced, it probably was never enforced or checked out, but it really lays out the whole concept.

Pelkey: And the concepts came from the X-25 packet switching thinking processes?

Dolan: A statistical multiplexer is a relatively easy device, at least I thought a statistical multiplexer was a really easy device when I came up with -- found out that there was not, that whole area was not well exploited. When I started ComDesign, I didn't realize that. We were originally working on some local area network technology. I did not realize that simple end applications of the kind of technology used in packet switching, other communications just weren't available. You couldn't simply say: "Let's put X terminals at one spot, have it hooked to a host at another spot, and work that in a way that's deliverable to users." I'm sure that technology existed, but the question is did it exist in a commercial form that people could use in an everyday sense?

Pelkey: So when you first created ComDesign in 1977, it was really in a view of going after local area networking, as opposed to –

Dolan: That was the original concept. And it turned out that local area networks, and a whole lot of other things, were really much more complex than what was really needed for the market at the time. When I saw this little company, Micom, that was making a killing of 5 million and then $15 million fiscal years selling statmuxes, I thought: "That's kind of great, but it doesn’t do that much. It's kind of a trivial thing. They probably did it in a couple of days," because I came from a world where we were working with quite sophisticated technology, where there was no limit to the routing and the topology of the network, and it had redundancy and carry over features and things of that nature.

Pelkey: How did Micom -- were they the ones, to your knowledge, who really created the statmux?

Dolan: I think -- no, I don't think they were. I think certainly DCA had some technology that predated it where they were using minicomputers to do statistical multiplexing things, and also, if you go back, if you want to go into what a statistical multiplexer is, the front ends that were used by airlines and by other companies in the late '60s definitely did the same kind of thing, they just were never turned into general purpose things. They were developed for a particular problem here or there.

Pelkey: What do you think caused it to be productized as a general -purpose box?

[1] See Interview – Ken Krechmer

[2] See Interview – Wesley Chu

Dolan: The microprocessor helped make it something that could be dropped into an application very easily. Early they were done with minicomputers, and that kind of –

Pelkey: So the microprocessor was necessary for cost effective multiplexing?

Dolan: Right. If you looked at the Arpanet, every component was extremely expensive. They used 48-kilobit data lines, which actually used 12 voice-grade lines to create 48 kilobits with basically a half bay high modem, very expensive stuff. And then the Honeywell computer that was used was 50 or $75,000 at a minimum. BBN wrote that all in assembler. It was not terribly well architected by software engineering even in those days, at least in my humble opinion, and so it was quite unreliable. When the microprocessor came and while people got a better appreciation of software engineering it was possible to put it into a package where people didn't have to be tremendous experts to be able to use it. The Arpanet was not an approachable network, even though it did everything you wanted. It was not approachable if someone wanted to add something to it. A guy like me, a company like BBN, they could build code that would run in the host that would talk to this packet processor.

Pelkey: So the microprocessor made possible the statmux at some level.

Dolan: It made it possible to make it as a general-purpose product. There certainly were front-end processors that did exactly the same thing –

Pelkey: But they were expensive and –

Dolan: -- and certainly something applicable to minicomputer type of situations.

Pelkey: Did it require an eight bit?

Dolan: No, there was no special need for a particular type of microprocessor.

Pelkey: Because the four bit microprocessor was available in 1972

Dolan: Yeah, but I think in application, there are very few people that did anything more than a fairly trivial problem with the 4004 type processors, and the memories, you were really limited to -- basically at the same time, actually computer power comes from the amount of memory you have, more than the bit size of the memory, as you may know. The size of memories you could get for this microprocessor were 1K byte 2K byte type EPROMs, and for quite a bit of money, so you could maybe make a very simple game with a 4004, you could program a calculator, but you couldn't make a personal computer, you couldn't even make it to a very complex application. Certainly you couldn't even do anything as advanced as a sophisticated calculator, and although statmuxing is not, at the base level, is not that complicated, it certainly it stretched that a little too far.

Pelkey: So the memory, what 16-kilobit memory chips becoming available also was an important part.

Dolan: Four kilobit. Having an eight bit processor and just -- this application, the need existed, the technology existed, but no one got around to actually making it an easy to use drop-in product by itself, separate from the computer system. Prior to that time, the computer -- a lot of people that tried to solve these kinds of problems, always wrapped it into a computer system at the same time. In other words, the minicomputer would have some subsystem that would take a lot of computer power which would do the front end concentrating, as opposed to a communications only device as a separate device.

Pelkey: So it was the unbundling of that concentrating function from the minicomputer -- there was a real market to do that, so you could off-load the minicomputers so you could do application programs instead.

Dolan: Exactly. You certainly could have done that multiplexing; in fact, we had done it in several products in things we had done on PDP-11s. We had done that in -- at the university. We had done things like that where we multiplexed, but you don't have a general-purpose thing. You don't have that something you can just hook into any computer you see down that street and bring terminals on line.

Pelkey: Do you recall at that time whether DEC or Data General had concentrating functions in the software they provided with their machines? Or was it more that it could be done, such as what you did in ARPA?

Dolan: Actually, DEC, I think, had a PDP11 that was a concentrator front end for a PDP-10 in that range. That was done, but it was not done as a standard feature. In fact, the TC3 product that ComDesign had, where we plugged a board into a PDP-11 and off loaded the statmux basically just allowed us to have a general purpose product that would plug into at least every PDP-11, and make it look like we had multiplexing. We had directly connected terminals that were, in fact, run over a communications line.

Pelkey: That was a very successful product?

Dolan: It sure was.

Pelkey: And it was successful because of some of the reasons you just mentioned. Was there anything else that caused it to be a successful product? There weren't very many products like that on the market.

Dolan: Well, the idea of actually -- in my naiveté at the time, I think that we have had -- well, Micom's probably got statmuxes cornered, they have (unintelligible), I had known the PDP11. PDP11s probably represented 50% of the attached terminals to multi-user computer systems, and DEC didn't necessarily have a high percentage of the overall minicomputer market. There were a lot of players in there, especially back then, but when you take only the multi-user systems, which are the ones that are going to be heavy datacom users, DEC had at least 50% of the attached terminals to those minicomputers that were multi-user. So when we developed a product then that actually plugged into the bus, looked like a normal directly connected terminal, and allowed those terminals to be at another location, it was extremely easy for someone to make that decision. You didn't have to explain. Everyone in the statmux market had to, originally, up until probably about '82 or '83, had to explain to every customer what a statmux did. After about '82 or '83, they understood you could just drop it in. You really didn't need to know what it did and all this other stuff. It wasn't an issue. You got some terminals at one end, terminals at other ports or whatever, if you stick this device in, this will let you use one line.

Pelkey: Micom spent a lot of time doing these seminars, as I recall, in addition to the orange juice can ads.

Dolan: Even before they did the seminars, they still had to spend a lot of time. I'm not sure that they needed to explain that, by the way. I think that a lot of the reason so much time was spent on explaining what a statmux is simply because of naiveté in terms of the marketing. Why don't you just tell the people what problem it solves? No one was telling people what problems it solved in that, after you got beyond modems, nobody talked about solutions.

Pelkey: My opinion is that the orange juice can was one of the most effective advertising campaigns of any technology.

Dolan: That's true. That was very effective, but I think a lot of people bought statistical multiplexers that didn't even care whether it squeezed the data. All they needed to do was, they just had some terminals here and terminals there, and how fast they went was a second iteration problem, and they would have just bought it if they had known they could have just dropped it in and those terminals would be running. We had a lot of people that ran on a 9600-baud line or 4800-bit line, and they went for 1200-baud terminals or something like that. They clearly aren't using -- I mean the fact that it's more efficient or something really -- we know it and we feel good about it, because we have this great product, but I don't know that it sold any more systems.

Pelkey: So when you looked at -- how did Micom come to you attention?

Dolan: We had decided that TC3 was a real good product, a product that had a remote -- think about it as a remote distribution channel. That's one of the way we tried to sell it, a remote distribution channel for a PDP11. We had decided that it was a good product and we wanted to develop it. We didn't have much money. Well, we found out that universities needed a one-ended multiplexer, which basically -- Micom only had to develop one type of multiplexer; the same modem was used at both ends. In the case of the product we had, we had to develop two different things; the board that plugged into the PDP-11's and looked like a standard peripheral, and then the unit that sat remotely. We found out the University of California Santa Barbara needed to build a basic one-ended multiplexer, where they would have the 370 CPU, if you can believe it, do the de-multiplexing, to save the very expensive ports on the machine. So we bid that real low so that we could make sure that we could develop that, and that ended up being the far end of our TC3 development, and we built like ten of them for, I think, the bid was $20,000. Well, that was enough for us to develop, and complete the development, and ship that product to UCSB, and by then we then had a completed piece of hardware to put at the far end.

Pelkey: And did they need it for the ARPA application?

Dolan: No. After about '74 or '75, the ARPA people all moved into -- started to move away. They had gone into private industry. Basically, that real excitement, that real understanding of the network and all that just started to dissipate, and the people in the computing faculty, by and large, didn't understand the importance of that, and it was ARPA and military related, and in those times that wasn't necessarily a great thing, and basically, it just died down.

Pelkey: Who were some of the people that were in that ARPA group in the early '70?

Dolan: Jim White is the one that I think is probably the most striking guy. Jim White is now with Telenet, and he developed the X-400 standard for electronic mail, and I think he did a lot work on XNS also.

Pelkey: So he went to Xerox?

Dolan: He went to SRI and then to Xerox, and is now at GTE Telenet, I believe. Now he wrote the very complex program that's used to connect to the packet network. The host processor that connected to the Arpanet had to do lot of that packet network things don't have to do now, and it was very hard to do them on a batch oriented IBM 360 75 type computer. He wrote that entire program himself. So he was one of the prime movers.

Pelkey: And then that software was used everywhere else in the network?

Dolan: On the 370s.

Pelkey: UCSB had the responsibility of interfacing the IBM machine to the Arpanet?

Dolan: No, it was not really to -- back in those days, everyone kind of wrote their own software, and they didn't really automatically assume that they were going to just try to use someone else's. It was kind of a pride. "Sure, we'll do our own." Now, it turned out that that network control program was used in most of the IBM sites and BBN's was used -- BBN's that they developed for the PDP-10 was eventually used in most of the sites, as people understood how complex this was. I don't think anyone understood how complex it was going to be to write networking software. (unintelligible), but Jim White wrote that single-handedly. I think that was probably the only network control program that was written by anyone single-handedly or even without three or four people.

Pelkey: Who else was in that group?

Dolan: Jon Postel. Was it Jon Postel? John -- it's another guy who's really into the network. Anyway, there's another fellow that did a lot of the early networking standards from a software point of view that was in it at UCSB who is now, I think he went to Xerox also. Roland Bryant was the one who got all these contracts to UCSB, and so he was very instrumental in getting the support of the network, for UCSB to be one of the packet networks. There were originally only four nodes: UCSB, UCLA, BBN and I think University of Utah. Kind of an odd looking network topology, but enough to prove the concept, and it was rolled out after that.

Pelkey: Was there anybody else? The purpose of my question is to understand what people came into connection with what other people. I have a view that, in the communications industry, as I find out who came from where, there's going to be certain central groups of people who interacted with each other at one point in time that spread out, and that's how the knowledge got transferred, but it was limited to certain groups of people, and those certain groups of people had a certain view of the world, and other groups who developed another view of the world, when they came in contact with each other, then that made things richer. That's what I'm trying to track down.

Dolan: I think it was John Perkins. Anyway, I spent most of my interaction in the ARPA situation, I mainly interacted with Jim White and just spent a lot of time understanding what he was doing and so forth, so I learned a whole lot about that, but I was -- I moved into minicomputer communications shortly after I worked at both on 360s, and then spent the rest of my time developing timesharing type communications stuff on a PDP-11 at an early stage, so I moved out of that area early on. Roland Bryant would be a big source of information on that. That tree has just got to be -- is richer than (unintelligible).

Pelkey: You made the comment that people underestimated the complexity of networking software difficulties.

Dolan: I think it was probably two or three years after the Arpanet was on before people really realized that there was really something totally different about this kind of software than a typical application, and in fact, the Arpanet went through some very interesting problems. There was one Christmas about two or three years after it was up where it got into a tremendous deadlock situation, and they could not get the packet network back up for two or three days, if I remember correctly, because it kept getting -- the whole network -- each of the network nodes are designed to be able to reload, and so forth, and tell when there's problems and congestion, and it got itself into this loop so that, no matter what happened, it would get back into a deadlock situation, and it just -- it was just amazing. There were numerous problems of that sort, where the network would get congested. That was just the tip of the iceberg. There were topology problems, routing problems, tuning problems, making hosts talk to these, the number of states. This was back in the days when there were no real protocols that were complicated enough to get into problems. When people started -- I think, and it took a couple of years for them to understand that this was not just a problem maybe of poor software design but actually that there was something inherently very complex about everything that they were doing.

Pelkey: To come back to the story of ComDesign, you got this contract of doing this channel connected remote terminals. So you developed a remote unit for your -- and when you had that -- I had gotten to this in terms of asking how you became aware of Micom. Obviously becoming aware of Micom and what they were doing caused you to change your strategy some. Did the contract come before Micom?

Dolan: No, they needed someone to build this device that would de- multiplex and we just saw that as a way to get the TC3 developed.

Pelkey: Was the TC3 built with Micom in mind?

Dolan: No, knowing a little bit about Micom, and figured they already had a product that was end to end general purpose multiplexing, but I saw that there was a big need, and probably a lot higher value, i.e. gross margin, in a device for PDP11s that would do that job really well, so what we did in the process, we eliminated the need to buy the hardware at the other end to stick all those ports right back in the back of another device. So we were able to build a board that was cheaper than either of the two components that you needed on the computer end, and so we were able to build a device at 80% type of margins.

Pelkey: Who was doing statmuxes at that point?

Dolan: I think GDC, Micom, and DCA. I think Timeplex was doing something. When you say then, I'm not sure what you're talking about.

Pelkey: '77 -- '78.

Dolan: Well, we're talking about very early '79. We did not abandon the local area network, which is basically an improved version of a 3 Megabit Ethernet, Metcalfe's original paper. We were building an 8 Megabit Ethernet type device type product where all the communications would be transparent to the users of the device. In other words, there would be a board in a PDP11. No one on the network or -- the PDP11 would not know that it was not talking to directly connected devices, that it would actually be using a network. That's only a recently thought of idea in widespread use, that the user of the network, or the computers that use the network, shouldn't need to know whether the resource is local or remote.

Pelkey: So you spent –

Dolan: We spent about a year on that.

Pelkey: Most of '78 on that issue.

Dolan: Oh, yeah, in fact, Jim McGill and some of the guys there were some of our original consultants to help us, because I didn't understand some of the analog questions, in terms of driving the 8 Megabit bus.

Pelkey: So then Micom came to your attention somewhere along the line, and then this contract at UCSB?

Dolan: Well, no, the idea for the TC3 came as kind of a subset of the general technology of using a local area network, having transparent devices, in working with a customer of ours in Australia. I had determined by that time that it was just going to take too much work to market and sell the product, even once we got all the pieces done, so we were looking for a way to get the TC3 done, as just a problem, and I saw it as something relatively straight forward to do, based on what you could do with packet networks and so forth.

Pelkey: So then you got the TC3 and then obviously you got the TC5 - -

Dolan: TC5 was, basically, after we looked at it further and started selling 3s, we realized that Micom did, in fact, not have a total monopoly on multiplexers. We looked at the techniques used by Micom. They did not come from a systems orientation. In fact, if you look in Micom brochures in the '79, '80 time frame, you'll find that they very proudly say that they don't use any programmers.

Pelkey: I remember that well.

Dolan: They did not take a systems or a customer view of it. They actually kind of treated the microprocessor as a piece of the hardware and the bricks. So it had a very rudimentary control. It was controlled with dip switches and so forth. Once you got someone to buy it, then of course there was no problem moving the dip switches, but the problem was, it was a little hard for people to understand. Furthermore, at that time, there probably were 100 different kinds of ASCII terminal protocols that were being used, in terms of -- now we're kind of in a situation where people just use an X-on X-off type flow control, almost throughout to speed up and slow down terminals. There were so many different standards being used that a lot of flexibility was quite a marketing advantage for us, so that's why we decided to come out with the TC5, because it was really rather easy to do once we had done with the TC3.

Pelkey: And when did you introduce the TC5?

Dolan: Gee, '81, I think.

Pelkey: Now, the marketplace at that time, obviously, had grown. Micom was bigger than it was when it first came to your attention. What were some of the things in those days that you think that led to the success of the industry? To one company being successful over another? Was it technology? Was it distribution channels? Was it marketing?

Dolan: I think marketing and the distribution channels probably made the biggest difference. Micom eventually got to a point where they -- I think the orange juice was probably a turning point in terms of emphasizing a benefit. a solution to a problem and trying to get a little bit away from hardware, because something that's real important to understand about most communications products, at least those ones, is they are not being purchased by communications people: they are being purchased by computer people. They have a problem that happens to need communications to work, so you're talking the wrong language if you don't talk about what the problem is that they've got to solve. Now in these days when you've got voice, data, com managers and so forth, now the problem is, if you try to sell data product to a voice guy, or voice product to a data guy, you've got the same problem, but most of the data guys know what the computer guys are doing, and the computer guys at least have a rudimentary idea of what the datacom guys are doing, but you've still got some of the same bridges where there is a foreign language going on there, and basically, most people were appealing to the communications people. Well, the communications people didn't knows -- the communications people at that time were guys that managed the large IBM multi-drop networks, which is the way IBM networks operated in those days, with almost all multi- drop bisync terminals.

Tape Side Ends

Dolan: . . . was a problem because with bisync, which was the primary protocol in those days, you had an ability to have a string of terminals on one connection and you could address those one at a time and send data back and forth. It wasn't terribly efficient in the communications line and it was cumbersome and had a lot of problems, but it worked and IBM customers wouldn't have bought the IBM machine in the first place if they were concerned about how expensive it was. Non-IBM users, they had a real problem, because just even the hardware for the controller for the bisync communications controller was in many cases -- it would have been a substantial cost of a minicomputer, or even some of the other mainframe computers. In fact, some of those bisync type controllers did very little except for handle that string of terminal bytes allowing one to send, the next one to send.

Pelkey: I've heard that you expressed that some of the comes back from the earlier days of the first IBM terminals, like the Selectric typewriter, where they just did local echo, whereas teletypes did remote echo.

Dolan: Oh yes.

Pelkey: So what happens is teletype lent itself to the communications environment and IBM's view of the terminal was -- forced there to be a communications product there to deal with error correction and other problems, so you had have these communication controllers that handled the way they designed their terminals.

Dolan: The fact was there was no particular reason why you had to use a full duplex situation with minicomputers, but since that's the way the software is run, changing that -- you could have done what IBM did where all the echo is done locally. There's no real hardware cost either way, but the software needs to be designed one way or the other. Non-IBM non-mainframe computer systems were designed to take advantage of the fact that you didn't have to print the same thing that the person typed, so you could move the cursor, you could do something that was smart based on what the keystroke was, which might, in most cases, be just typing that same character back, but in other cases might be wrapping to another line or whatever, things that we see on micros all the time these days. The problem is, if that was the way the software was done, if you then wanted to take those terminals and put them in remote application, you could no longer use a protocol like bisync. A protocol like bisync would -- this is probably redundant review, but just so you know it -- at those times, you might be running on a 1200 or 2400 bit per second synchronous line, the controller might send out a line of text that might take a quarter or a half second to transmit to a selected terminal, and then go back and ask one terminal if it asked that terminal or another terminal if it had. Will this work him anything to send. That terminal would dominate the line now for another fraction of a second, and so forth, and in a round robin process. Well, clearly, if that was the kind of protocol you used, you would not be able to run the same applications you were running, because you need to echo much faster than bisync could handle. So the line speeds weren't -- now, with very high speed bisync you could start to do that, but modems didn't run that speed.

Pelkey: In early days, clearly the communications business was related to this mostly through the '70s to the rise of the minicomputer.

Dolan: The non-IBM dominated one. See, to some extent, all the communications functioned, beyond the modems, once you get above the level of functionality of the modem in the communications juncture, if you were an IBM computer user, you could buy a terminal and a communications controller from IBM. There basically was no room for other people to go in as independents and provide a plug -- a solution that you could just plug in and do it. When you went outside the IBM and most of the mainframes, those that copied what IBM was doing with bisync, all of a sudden you had quite a market, so most of the big timesharing customers -- the non-transaction people, i.e. the timesharing type people and everything else, but the guys that aren't going customer order entries or the airline type people, the rest of those people almost routinely either used minicomputers or mainframes that had the same characteristics as minicomputers, i.e. the PDP-10. The PDP10 had a very minicomputer flavor to it, very interactive, could support lots of users very easily. You get an interactive, dynamic kind of situation, then you have a whole different set of needs, and that's where the IBM type thing just was never adopted, and that gave birth to a whole range of higher level communications functions. The first was, letting you put more than one line on a single leased –

Pelkey: Which was a multiplexer?

Dolan: Yeah, multiplexer, packet switches and so forth. Packet switching still is not used highly in IBM environments.

Pelkey: So technology -- patents throughout this process, never seem to have played much of a role.

Dolan: No, in modems they -- a little bit, but no, more did standards play a role, outside of modems, because at that point, it was more a matter of whether you were compatible with other things, but you could use your own protocols, and they did, between your devices, and there was never any situation where your device would be called upon to talk to someone else's device. You bought them in pairs, because they always went on leased lines and, in fact, that situation exists in the modem business. When you're buying a leased line modem, you don't really care about standards nearly as much. It's with dial-up lines you have a problem. There's where Telebit's going to get themselves in deep shit. That's an editorial, by the way. But standards simply didn't have the role. They weren't needed to foster it, so that was not a barrier. Government regulations weren't a problem because once you connect to the modem, you were outside, and at least in the US and most of Europe, you were outside the bounds of where the PTTs tried to control you.

Pelkey: Modems are sort of line the DAA of earlier days.

Dolan: Exactly. The modem guys had to run that (unintelligible). Now, some of the guys that had statistical multiplexers, of course, were modem manufacturers.

Pelkey: But for the most part, the guys who dominated modems didn't become major factors in the multiplexer business until much later.

Dolan: No, and I think that was primarily because the people that bought whole wads of modems were IBM users. It was real easy to sell them because you talked to one guy and he'd buy a thousand modems.

Pelkey: Oh, ok. And some of the -- the modem business was growing so rapidly and consumed everybody's resources, in terms of people and capital and so on, that they didn't have money to go off and do other things, so they may have known about multiplexers or seen them as an opportunity, but changing priorities internally –

Dolan: Well, you've also got to look at what their business is. Modem people are not doing any mathematical transformations, they're not doing any computation. There's no software, at least there wasn't any software then in a modem. Now that you can do analog in software, essentially, with the digital signal processor it's changed, but basically modem companies didn't tend to have any software people, and it kind of takes software expertise and it takes more complex systems orientation, in other words, two systems talking to each other. You can basically take a scope and a modem and hook them together and at least the simpler modems, you were able to tell whether you modem was doing the right thing. You almost didn't have to hook two modems together to see whether they were going to work, because there weren't a lot of different states and complexities, so the kind of expertise it took was a lot different. It tended to come from the minicomputer guys, guys that knew minicomputers, guys that basically worked on communications protocols.

Pelkey: Minicomputers?

Dolan: Minicomputers, or, actually in some cases, mainframe guys, but still working on the communications end of it.

Pelkey: And anybody who was really doing that would have, more than likely, have had some relationship with Arpanet?

Dolan: No, I don't think so. That was a big set of people, but –

Pelkey: One last thing –

Dolan: I hope a lot of that is useful.

Pelkey: Oh absolutely. One last question before we finish this: local area networking. The guys who did multiplexing -- while you had started in local area networking, you're kind of a unique beast in that sense. There were not companies that were in modems and multiplexers and data PBXs that developed a local area network. That technology and those companies came from a completely different set of –

Dolan: I'm not sure that the people in the data com area, other than LANs, I don't think they see the need for local area networks, because local area network is -- yes, it's a communications product, but it's an intra-building communications product. It appears to solve a different class and different kind of problem, and it wouldn't have been considered data communications back then. I don't think people considered it data communications, or communications. They considered that as a way to hook -- more like a wiring scheme or -- they just did not consider that communications.

Pelkey: So in the early days, the data communicators really oriented themselves to the telephone.

Dolan: Yes.

Pelkey: The leased line, and not to pushing data from place to place.

Dolan: Right.

Pelkey: Thank you very much.

END OF INTERVIEW