DAVE CLARK (DC) INTERVIEW
DATE:AUGUST 22, 2014
INTERVIEWER: Dr. LEONARD KLEINROCK (LK)

DC: I'm David Clark, of course, and I got my Ph D. here in this lab in 1973. I came here as a grad student in 1966 and I worked on the Multics operating system which was this wondrously large machine. Multics itself never went anywhere, but we we joke that it spun off Unix, so we're OK. And after I got my Ph.D., I didn't know what I was going to do. I thought I might be a faculty member, but I got a job, today it would be called a postdoc, but we didn't call it that back then. I was working for Jerry Saltzer and he wanted me to manage the team that was writing the software for Multics to get it on to the ARPANET. So I got hurled instantly into a management position, for which I was totally untrained. Nobody in academia ever trained to be a manager, you see. And I supervised these two or three guys who were writing the code to do the NCP, the old ARPANET software. I got sort of interested in this, actually, my Ph D. thesis, lead me into this stuff. So, when DARPA started getting interested in the Internet, I was an obvious victim to get sucked in, and so I joined the team, and my initial job was to write the TCP code for the Multics system, which was challenging.

LK: What year was this?

I don't remember, I think I started working either ’75 or ’76, I can't remember exactly when it was, because we crept up on it. But you know, I had some initial conversations with Vint, and he said yeah, I want Multics hooked onto the Internet with the new protocols and so I started writing code. And that's what I did. I was part of the team that was trying to get the code working and also find the spec during the 1970s.

I was funded by ARPA from the time I got my Ph. D., until probably some time in the mid-90s, I don't actually remember when it was. But certainly all through the 80s and and well into the 90s. Their focus changed, and so I had to sort of realign what I was doing, but probably at least twenty years.

Right. I did not code the NCP, that was done by these guys who were working for me. But I actually wrote the TCP/IP, I mean, I wrote the Internet protocol stack. And I wrote it several times actually. Back then I could code. Although, I've recently started coding again that's another story. I wrote it first for Multics, and then I wrote it for the Xerox Alto. And then we ported that code and it became the first implementation for the IBM PC. So I've written it for a huge machine, which was totally unsuited, because it took twenty five milliseconds to service an interrupt. It had a completely separate I/O control which didn't really work very well and I've written it for a machine small enough, that we had enough buffering in the IBM PC for one packet.

LK: A whole packet?

DC: A whole packet! We could buffer one whole packet!

LK: So, in what areas were you working, and related to that, how did your interest in networking begin?

Oh, that goes even back further. We were trying to make Multics a multi-level secure system. And back then we had the orange book that described how you should…you know, there was B-level systems, A-level systems and so forth, and Multics was in fact an operational multi-level secure system in a way. They validated this in the U.S. Air Force. You know these these lieutenants and they read code. You know they audit in code. So we wanted to reduce the amount of code that had to be in the security kernel. And Jerry Salter said to me: "For your Ph D. thesis, why don't you see if you can take all of the input/output code and throw it out of the kernel and user space"… which nobody had ever done before. And I said, that's really interesting. And it was a hero task because… you know the code for Multics was written in PL/1 language most people don't remember now, but the true heroes programmed in assembly language, of course, but we had the separate I/O controller, which had an even more primitive programming language, so trying to program the I/O controller from user space was just a delightful tour de force of primitive programming. I don't know whether I'm allowed to tell anecdotes, but you can always remove them…

This was a separate computer, but it had no user interface. It had no way to communicate except through an interrupt channel and it queued messages. But of course, if your code was screwed up, it might not be servicing the message queue. So if the message queue from the I/O controller would fill up, well then it had a panic queue, but if your code wasn't servicing the message, then it might not be servicing the panic queue either, so when it filled, the panic queue would only had one last thing it could do, which is it rang a fire bell. This very loud bell. And so to ring the bell meant you'd really screwed up the code and. You come up to Multics and you'd mount your tape and you push the boot button and the bell would ring and it's like the button had been wired to the bell, and it was a really bad program so…so… I managed to throw all of the I/O code out of the kernel. I got the tape drivers out of the kernel, and the print drivers out of the kernel, and the teletype drivers on the kernel. And then I realized I couldn't get the network code out of the kernel. Because I had to do the multiplexing as particular the de-multipleing of the incoming packets in a multi user space. And I said, oh, networking is not just I/O, it's something else. But we don't know what it is. We didn't have a model of what networking was. But I had figured out from a structural point of view, from a code point of view, it wasn't I/O, it wasn't like a tape drive, it was something else. And so I said, what is it and this is when we got interested in what the structure of de-multiplexing was and how this interleaves with interrupt handlers and process scheduling, things like that. And I had a long history in the early part of my involvement with networking trying to understand how the structure of the operating system and the structure of the network code had to be mutually reorganized in order to run efficiently and, for example, the way people thought you might do it originally was while a packet comes in you'd..you do the IP layer and then maybe schedule a process and it would do the TCP layer and so that ran like a dog, because process scheduling took forever and ever and I… I wrote a paper after a while that was called "Up Calls", which basically said, run the whole thing in the interrupt handler basically, because it takes longer to schedule a process than it does to process the packet and people said, “oh the code to process the packet takes forever and ever and ever and ever” and I had to get this get this guy working for me named John Ramsey and I said, “Well let's see how many instructions it takes.” And so we actually compiled the code. Followed the common path through the code down to the instructions and I said “Oh it It takes nothing, it’s two hundred seventy instructions to process an incoming packet, it's nothing.” And I showed this to Van Jacobson, who said: “I can do it in fifty”, and he went off and came up with a fast pass strategy for doing it. The trick was, don't code it as if the probability… there are many things that could be wrong with a packet. You have to test this and test this and test... Don't code it as if all outcomes are equally probable. Code it as if the packet is going to be well formed and make all these other things, exceptions and so, code it so that you go… and if it's wrong, you don't care how long it takes to process if it's wrong, because you've screwed up, so you know you're going to close the connection. So when I started writing the TCP code., I was equally interested in whether TCP was well specified. And of course, that was the period in the late 70s, where some really smart people said: “No, TCP is one protocol and IP is another. In the original paper, they were all glommed up.

LK: You were critical in making that observation?

Well I wasn't…I wasn't the lead… I didn’t take the lead on that. Dave Reade and Danny Cohen. It was the guys who were trying to do voice. And people occasionally say to me now, well when did you first think you could do voice over the Internet, I say I think we did it in ‘79-something. We just had to wait for the speeds of the links to get fast enough to make its sense of what…we always knew we could do it, but they didn't want to use TCP, they wanted to use UDP, so we had to do that split. And so I was equally interested in whether the protocols were well structured, but I was also interested in whether the operating system was well structured. And it clearly wasn't, but we didn't know back then, what sort of basic operating system primitives and structures would make it effective to write network code. Timers, you have to have very efficient timers, you have to have very efficient process scheduling, ah, have to have very efficient dispatching on an incoming packet and in some cases, you have to write a process. That's waiting for multiple things and it sounds, today, obvious but it wasn't obvious at the time, we had primitives that said wait for an event, no, I'm waiting for one of three events. You know I'm waiting for a keyboard input or I'm waiting for a packet from the network, I'm waiting for a timer to go off and..and and you know today, that sounds sort of silly to say that that was a novel thought. But this is what had to be worked out. So I think we really spent the 70s and, I think, most of the 80s actually pushing on operating systems, so that they had the right infrastructure, so it was trivial to write this code,

LK: So your original work on Project MAC and MULTICS, how much was focused on Timesharing as opposed to networks? From what you just described, you probably mixed the two, similar problems arise…what was the influence of your thinking about Timesharing on your thinking about networking? Is there a path there or am I making that up?

DC: Let's sort of think that through and figure this out. Multics was the second time sharing system that MIT did, the first was called CTSS. It was very, very primitive, and… I can't remember the chronology of what was going on at MIT versus what was going on elsewhere. The idea of timesharing comes out of stuff that Engelbart was doing, I mean it comes up in several places, it was in the air. And I don't remember the chronology of when we started CTSS as compared to some of the stuff that was going on in the West Coast and so I can't answer that. But it's clear that there are a lot of fundamental problems that we had to deal with in Timesharing, and once we got past the basics of can we do it: Effective scheduling, effective page handling. And networks stressed all that. So there was clearly an interplay between how we dealt with those issues within the sort of primitives of the time sharing system in the way networking work packets are very small. And so they're very fine grained events and. There was a lot of overhead in scheduling Multics, it was a lot of swapping and stuff like that. Well you couldn't do all that swapping on a per packet basis, but you don't think about that now, because you have enough main memory, just leave everything in the main memory. Well, you know, who cares, you know. But back then these were really sensitive issues, as I said, when I worked on the IBM PC, I was rather astonished to discover that there was enough buffering for one packet and, as you well know, it doesn't work very well. You just turned it into a one packet response, I'll send you and X and I'll send you an X, send it back and I'll send you an X. And the…the simple stuff we were trying to do back then was incredibly hard. It doesn't sound hard now. Unless you're doing Internet of Things where you've got a ten cent thing, and we could come back to that later, but we were doing remote login, so when I programmed unit protocols for the IBM PC, we were trying to use it as a terminal. And we were using it as character time terminal, so what I had to do was write code. That was interactive enough that you could do character to time interaction with the mainframe, given the constraints of the size of buffers and so it was all about performance, but not at the performances of sort of…at any level of sophistication. It's about, how do you deal with these horrible implementation limitations that today sound, just so… you know, it must be back in the dark ages or something like that. I mean the IBM PC had 640 kilobytes of memory.

LK: But dealing with character echoing, if you will, in a Timesharing system, is very different from a network because of the latency issues. That came back to bite…the echoing across the network.

Well, it's interesting to think where those early latency problems came from, because it's not that we were dealing with the speed of light. I was dealing with a machine that was ten feet from the processor. It was all the struggle to get both operating systems to deal with these network events efficiently enough that the sum of the total was usable. Because you know I…I hit the key…well I could send a packet from the IBM OC really quickly, I mean, I mean, I know what the entire code path…I counted every instruction, there was no issue of banging the packets out. And then the other machine would receive the Internet, and it would schedule… receive the packet, schedule the processes, all the stuff would happen. And then you just had to make sure that all of that worked smoothly enough and consistently enough that you got the character back at the right time. And it wasn't latency that killed us. It was other stuff that killed us, for example, imagine that I hit a keystroke. And I'm going to get a whole line of text back. So now you're dealing with really ugly abstraction design. A lot of programs back then would write to the kernel to write to the file system one character at the time, they wouldn't hand a line buffer over, they would hand a character buffer over. So now I'm writing the TCP, and somebody hands me a character. Well, what do I do? I could just put the character in a packet and send it, never mind the fact that I've now sent a packet which was this huge and had one character. Or I could speculate as to whether he's going to give me another character. But if he doesn't, I better protect myself so I have to set a timer. And it takes twenty five milliseconds for the timer handler to go off. It took ten times as much code to service that interrupt handler to service the timer interrupt than it did to just send the packet. So the obvious thing to do is to throw all the inefficiency over the network to the other guy. And just send the packet. But Multics could send packets more quickly than the network could deliver them. And so we'd get a dropped packet and now I hear I am on my PC trying to reconstruct a lines worth of text to put on the screen. And I've got a dropped packet. Well I can't buffer the characters. I can't buffer the packets on the PC, because you can only buffer one packet, so I have to unpack all the packets. Build a character buffer. Then build a data structure, that keeps track of where the holes in the buffer are. But not in terms of the packet boundaries. And of course, what TCP does is if you have a timeout, it doesn't necessarily re-send the same packets it would actually send the whole string in one packet, ok? Because I now had all the text ,so when you got output from Multics what typically happened was you'd see the text appear on the screen when character time would go blodiblodiblop, and then it would be a long pause or go blam, it would fill the whole line in when I got the retransmission.

This is not what you would call performance. This is not sophisticated enough to need the tools that you bring to bear. Everyone talks about queuing theory. Here we're talking about much more elemental stuff which is just in compatibility in the systems were structured in order to make all this stuff work smoothly. It was primitive, it was. It was hanging on by our fingernails. We were really exploring unknown territory, we were…we were just out there doing stuff that hadn't been done before. And of course, they collided in interesting ways, right?

Well there's one story from my lab as a whole and there's another story for me working on the Internet so let me give you both stories and then you can see how they play together. Back then the relationship between my lab which was then called Project MAC, and was then called the Lab for Computer Science, and is now called CSAIL, Computer Science and Artificial Intelligence Laboratory. Our relationship with ARPA ,was very different, we wrote one big proposal that covered all of the activities that are funded within the lab. We wrote one proposal every three years. It was an activity which we took very seriously, we really, really polished each part. But if I wanted to work on networking or advanced operating systems, it was a part of that one proposal and it went in, and ARPA could come back to us and say, "You know, we're not really really interested in this stuff." But that guidance was very much informal and a lot of the discretion as to exactly how we expended the money was because it was within the scope of one large contract… was, was much more within the scope of the lab. My lab director was the one who made the final determination of well, exactly how much money went here and how much money went here, we had to shift the money around. So in that context, the lab as a whole, we were getting much less specific direction from ARPA than than one would today. First of all, there was one prime Program Manager that was responsible for ARPAs relationship with our lab.

LK: Who was that?

DC: I don't remember, that changed from time to time, I mean, you know the early heroes. But you know that…you know the early heroes who did the stuff like Licklider and so forth, but I don't remember who it was in the 70s and that's because I had a different relationship. But they didn't claim to be experts in all the work that was going on in the lab, they didn't claim to be equal experts in language design and, you know, robotics and vision and AI and networking and operating.... So they were much more working with the lab as a building full of smart people, not as a set of projects that they were individually monitoring. That doesn't mean that we ran open loop, we didn't run open loop at all, but I think there was much more delegation of the responsibility for carrying out high quality research to the lab itself. The lab director was the one who really was accountable, because he ultimately was allocating the money among different pots. And I don't remember when it was that we began to see the breakup of the large block funding to the small grants. So that now, my lab as a whole wasn't primarily facing one program manager inside ARPA, we were facing lots and lots of them, each of whom viewed himself as the expert in the area that we were working in, but that's an important structural change in terms of how we work with our panel. Well, eventually…I mean we went through this transition and again I don't quite remember when, but in the course of my career we've gone through this transition from the lab basically being funded by ARPA. And they were making a large bet on our lab and…and I remember, there was a wonderful conversation, it was somebody down here, some program manager, office director, he was even the head of ARPA. He had a hold of the buzzword of the day and he said, I want silver bullets. I want silver bullets, and Dertouzos just looked at him and said, we're a silver gun and you'll take any kind of bullet that comes out!

You know and…and that really was the way Michael dealt with ARPA, which is to say, look, we're competent. We're smart people. We're doing leading edge stuff. We know more about it than you do. And I'm ommitting some of the rude words that might have been in the sentence. And leave us alone and let us get on with it. And then you reach a point where for a variety of reasons, you see the shift in the…in the thinking inside ARPA which is basically to say: Now we we have to more closely manage this, we have to take some of that responsibility back. We have to break your proposal up into smaller proposals which are project-specific and now you're going to have a project manager working with you, who is only responsible for your little part of the work. And he's not a expert in how Project MAC functions. He's an expert in networking and robotic surgery. And so you end up with all these little proposals and it changes the tone of the way ARPA thinks as well. If you have this large block grant every three years, ARPA can be much smaller. Because they're basically saying there's one person and he's responsible, I mean, he just put a lot of his credibility on the line. There's a lot of money flowing into my lab. But if you're going to claim that we need we need somebody managing networking and somebody managing robotics and so forth and then we have multiple of them because it was just… it increases the size.

And it also shifts the presumption about how much of the invention of the direction…how much of the determination of how we're going to go about something is centered in the lab versus centered in ARPA and clearly we've seen a shift in that space where, in some sense, I think today inside ARPA, if you're Program Manager, you can't get a program started by saying there's a problem to solve. To get a project started…or program started, I guess is the correct term in ARPA terminilogy, you have to convince the management. Not only that there's a problem, we're solving, but that you have an approach. And you have to explain what the approach is. And so you know if you do robotics…you don't…you don't say, well, we want to solve the robotics problem. You say we want to solve a robotics problem, we think we'll solve it this way, would you like to work on this? Of course the community has a lot of input in that there's some things like that where the community can get together and try to persuade ARPA what the right problem is what the right approach is. But you now have program managers who think their job is to figure out how I should go about doing what I want to do.

LK: So did the number of visits, reports, interactions change over time? How was it in the early days versus the evolution…

DC: Well, as I said, my situation was special, so with the lab as a whole, every year we wrote a very substantial annual report. This was published. It was public, it was, you know, we printed…it was a big book. But it obviously was one of our reporting requirements, so we had a fairly large… carefully done, it was…it was important and you know, we did an annual report and now the program manager could show up any time he wanted and if he thought something was a little sick, he might bring in somebody and so forth, but the formal relationship was this annual report.

Now as I said my relationship was a little different, because after a couple years of working on the ARPA NCP… and at that point I was working for Jerry Salter, so if there was interaction with the sponsor, it came through Jerry and I didn't see it because I was this baby Ph.D… I started working on Internet and as soon as I started working on Internet, I started going to that working group meetings and there was a lot of very personal interaction with Vint. Now Vint was not the program manager responsible for giving money to my lab. But that was irrelevant in terms of the relationship I had with ARPA. But my relationship with ARPA was entirely based on interactions with Vint, and they were technically oriented, it was clear there was communication within the office in which Vint was saying, you know, this guy's doing good work…you have to fund the guy, you know and so forth, but… but that would come down through this grant structure.

So I never saw the sort of oversight structure that the lab might have seen, because what I saw was my engagement with Vint which is part of this team that Vint had built with people from BBN and SRI. You know, and your corner of the world, and so forth, and so that's a special case answer. Because…because we were trying to do this large collaborative, ambitious thing. And the management structure that had nothing to do with the management structure or the oversight structure that my lab….

LK: Were you in a unique position having been the recipient of funding in an environment run by Jerry Saltzer who himself went to ARPA, and you were below that, and later you moved into that position?

DC: Yeah.

LK: So the question is: How do you see the level of delegation and responsibility you received at the second level down, and how did you, when you reached the top level, delegate, pass, give the flexibility to your people?

DC: Well again I'm in a special case let me, let me sort of explain what I mean. I am not a faculty member. I am a research faculty member or a soft money faculty member. So I have a slightly different working structure here in the lab. We have oscillated and forth In this lab in terms of how we organize ourselves into groups. And we've had big groups with lots of faculty and we've had little groups, each run by one faculty and we sort of tried different structures. But we have to align ourselves against two objectives. And one of them is a structure that makes sense to the funding agencies and the other is a structure that makes sure that the careers of the young faculty are protected. Because we want to make sure that we protect the tenure and promotion case.

So if you're part of…if you're part of a large group which collectively goes after funding, this may actually relieve some of the funding burden from the young faculty. On the other hand, then at tenure and promotion time it may be a little harder to argue what they've done, so I think over time what we've seen in my lab is that we've gone to smaller and smaller groups, in which each faculty member is more and more responsible for bringing his own money covering his own grad students, covering his own postdocs. There are a few big projects in this lab but mostly the groups in this lab today correspond to one or at most two faculty members. So today, if you are a young faculty member, you get your Ph. D. and maybe you've finished a post-doc, and then you get your beginning faculty job. You may get a start-up package that protects you for year or two and then you're out there raising your own money.

I could survive a lot longer without having to face this challenge, both because I was in an era of these large block grants and also because the more senior faculty above me like Jerry protected me for a number of years. But it's important to note that because I'm a soft faculty member, I never had to jump the tenure hurdle. I didn't publish a paper for five years after I got my Ph D., because we were just all writing this code and there wasn't any paper to write, because we don't know what to do yet. And my… one of my department heads looked to me at one point and he said: You actually have accomplished something, haven't you? And I said, well, I like to think I have, and he said, that's really interesting, because I've looked at your C.V. and we never would have given you tenure. Because you didn't conform to the template of so many papers by so many years… you just went out and did good work and…And he went off, scratching his head.

But it hasn't changed anybody's theory about what it means to get tenure. So there's one conversation we could have about tenure but that's not what you want to talk about, you want to talk about ARPA. The point is that there are instruments today that will try to protect the careers of young faculty, but interestingly, I think they've moved from ARPA to other funding agencies. You get an assist career grant and you can get fellowships of various kinds to protect young faculty, so smart faculty no longer assume that the thing that will protect them or shelter them is a large grant from DARPA, within which they can work. It's a…it's a…it's other sources of funding. I don't…I don't actually know how that the interplay is between the ARPA structure and the university structure. I think…And it may be… It may be idiosyncratic, MIT other other schools may have different responses, I think. I think the shift in the group structure here is driven as much by the individual faculty members’ desire to control their destiny in a world where the larger funding structure doesn't work. But we've done some big things here recently.

LK: What do you mean, it doesn’t work?

Well, it's very hard to get the money. Now that's not always true, I mean some of the robotics stuff here, like the Robot Challenge or the autonomous vehicle challenge. The most of the faculty have come together to work on that. On the other hand, that's not a standing group, that's an ad hoc response by multiple faculty to say, Ah! Big challenge. Cool challenge. Big challenge, multiple skills, let's go, let's round up a team and do it. So… so it's…it's not something that is a fundamental part of our structure, it's something that's a response to a particular size of solicitation. So I think we've really decided that it's in response to the… I think, the extra funding structure, that we've really broken the lab up into individual faculty members, and then we come together in a very ad hoc way, ad hoc in the good sense of the word, to respond to a larger funding opportunity when we see it.

LK: So you’ve indicated this change in ARPA funding, where it went from large block grants to a more individual focus. What about the duration of the grants?

Well, that's again…there's two dimensions to this. When we got the block grants, it was three years. So once every three years we did a big deal. On the other hand, there was a lot of understanding inside DARPA, that we were doing things that took more than three years. So, Multics was a ten-year project. And we weren’t under a lot of pressure to take everything that we're doing and make it look brand new and different every three years. But we did have to go back to get money every three years. So I think there is… a part of the shift here is a sense of what the presumption is about how long projects take. And I remember there was a… I forget who it was, but there was a DARPA office director, who said to me: You know, part of my job is to be able to describe everything we fund in two incompatible ways to people who believe that everything has to be new and novel. I have to be able to describe everything as if it's brand new and novel. And the people who understand that really…transformative research takes a while, to have, to build, describe everything is building on what we did and so you have that problem writing proposals today. You have that problem at ARPA today. And you, have that problem with NSF. Let's say you're building infrastructure, like, let's say you want to build a measurement infrastructure to measure the Internet. And you've got boxes out there that are measuring the Internet. And you get a three year, great ,you put your boxes out ther. You really want that data to be gathered for a decade. So you go back to NSF three years later and say, will you fund my boxes, but we already did that. So how can I continue doing the work well? It needs to look new and different… I don’t want it to look different, I want the same research method for a decade, so I can get a consistent time series data. And the merit review process says: But there's nothing novel?

So our discipline, and this is an emergent phenomenon in our discipline, there’s nobody in charge, this is an emergent phenomenon in our discipline… We worship novelty, probably to an excess and so everything has to be new and different, so one question is how long does a grant last, but the other is, in what period of time can you get something done? And you know, we're not the only one to have this problem, when we were… When we were talking about trying to change the Internet and could we improve security posture, I think it was actually Al Gore who said this, he said, you understand, every part of the government has a time constant within which you can think. If you're, if you're elected to the House of Representatives you have to run every two years, so you'd really like to focus on things that you can take back to your constituents as having had an impact in two years. If you're Senator, you can look a little longer, but he said: There's no way this nation will ever put in place a plan to solve a problem, global warming, anything like that, which involves coherent activity for more than ten years.

Now, if you can break the project up into stages, such that every ten years there's a major payoff…So you, because you're really smart person, understand you're pursuing a thirty year program - but when you go to Washington, you describe it entirely in terms of things that have less than ten years - you have a chance of getting it done. But if I went to Washington and said I want a twenty year program to make the Internet the most secure place in the world, They'd say: Not going to happen while I'm alive. Why should I worry about that. Leave that to your kids. You’re going to break it up into little pieces and so the only question is how big are those pieces, and what’s your ration of the piece.

We've been doing this project recently with the National Science Foundation which I encouraged and I've been helping with, called Future Internet Architecture. And the question we asked the research community was: What do you think the Internet should look like in fifteen years? Don't make it a little better. Don't worry about migration, just say: What should it look like fifteen years out= And I'd like to remind people that the current transition we're worrying about the Internet, which is the transition from

IPv4 to IPv6. We called for that transition in a document that some of us wrote in 1991. So what, it was twentyfive years and so… When we started Future Internet Architecture, I got a phone call from the IPv6 guys saying: You're killing us, man! Because all of a sudden people are going to stop doing IPv6 and going to start doing whatever you're doing. I said no, no, no, you don't understand. I'm trying to figure out now what we're going to be arguing about twentyfive years from now. So the question was, did NSF have the continuity to do this, okay? And it had to constantly be resold inside NSF, but we've had a series of very understanding and thoughtful assistant directors for CISE, who understood that we had to have continuity here, so we had a period of three years where we did sort of small grants and it was a period of three years, where we did four substantial grants to four groups and then three of those got refunded for another two years, so this project's going to be almost ten years in duration. And, and some people in the community are outraged. Because we took money away from single PI grants. You know? But a lot of people say, wow, the NSF actually can do something whose duration is bigger and whose vision is bigger than a single PI grant or even a single expedition grant?

LK: So how does that compare to ARPA's behavior during the same period?

DC: Well, I think that DARPA's done some big things. The autonomous vehicle challenge. But they did that within a period of three years. But then of course they did a robot challenge after that so they sort of broken it up into chunks. I think you could say that that ARPA's had a fairly long commitment to autonomous vehicles. That's not my area, so I'm sort of looking over the fence but…but again, they've broken it into chunks which fit within the funding scope of what they would view as a traditional funding cycle or something like that. It has a whole different feel.

But I think that, I've always been living on a… almost all of the grants I've worked with from the government been three-year grants, I mean there was three year grants when I was a baby Ph.D.and there are three-year grants now ,they were bigger in size, but they were three years. And the real difference is how does that interplay with the presumption of how long a project is going to take.

LK: One point you raised I'd like to get back to is the security. MULTICS was considered security, and Project MAC was as well, even before the ARPANET was getting connected. And yet, the ARPANET didn't pay attention to security early on, and the Internet is now paying the price for that. So it's interesting that there was a clear focus on MULTICS, but it didn't rise to top priority in the networking?

DC: I think that may not be correct. It's not that it wasn't a priority. It's that we understood…we figured out very quickly, that we had no idea how to think about it. There's a…there's a book that describes it.. came out of an essay that describes how to build secure operating systems…. the Orange Book. And fairly early on, they said, OK we should write the same book for networking. And they don't know how to write it. I mean, they wrote one. It's about this thick and it's red. And if you go back and look at it, you realize that in the context of a distributed system, we simply didn't have the right fundamental model. It's a systems model, I'm looking for, this is not, I don't know how to do encryption… this is…I don't know how to put the system together, we didn't know how to put the system together. And the first few models we had of how to do it were just all wrong, okay? It's…It's clear that in a in a secure environment, you could have two machines that were secured by the same administration, so everybody in this machine trusted their we had shared keys. And they could put encrypted pipes between them and..and…We understood that, that was fine. And so the core of the network could be insecure because it's encrypted everything, except that we had no theory of availability, we had a theory of confidentiality, which is encrypted. And we had a theory of integrity, which is encrypted. And then, we said, but we need a theory of availability. And… and I spent a lot of time thinking about what a theory of availability is, but we didn't have one back then, I said, well the network can fail. So this machine really can't trust that one. There's a concept we came up with very early, I think I may have coined this phrase, I can't remember, but…but the concept was "fate sharing". Which is: What are the things in the system to whom your fate is linked? In other words, if that crashes, I'm going to crash, and since we all knew that all the software and all the computers was buggy, we put a tremendous effort in writing our TCPs, and to making sure that you could not send me a malformed packet that would cause me to crash and of course, this goes on today.

We didn't …we didn't call it this back then, but there's something now called a fuzz attack in which you just take a message I'm sending you and you just randomly manipulate, well we did fuzz TCPs. We turned random option bits on and send the packet in corrupt to see whether the other guy would crash. We did fuzz-testing on TCP and the amount of code that you put into your TCP to protect you from a malformed utterance, and this is true at every level f the protocol stick, swamps the amount of code that it takes to function under normal circumstances. OK, so you have these two machines and they trust each other, but they don't trust each other, because maybe they're malformed. And if this machine crashes, you want that one to crash, it's not what wanted, that was over there, this, that machine, I want…. OK, fine. So how do you think about the trust relationships and, and then we said, OK, but now get to the next step which is: I want to do something that originally….Wait a minute, let's talk about security. We had a very simple security model, which is: The world was divided into two kinds of people, the people I trust and the people I didn't trust. And I didn't want to talk to the people I didn't trust. That was foolish. And the other assumptions we made was: Well, the network can lose packets and crash and screw up because we know this, all kinds of this probabilistic packet forwarding and as you yourself proved, every once in a while you're just going to drop a packet because, like, the queue overflowed, right? We know what the other networks like, there's another story there, but that's for another…another question. And so we have to wrap code around the network that we trust, to recover from the errors. So we came up with this concept called end-to-end-principle Dave Read and Jerry Saltzer and I wrote it down, but that's…We didn't invent that idea, when we wrote the paper, we were trying to describe the design approach that had emerged. And the design approach that had emerged was the end nodes are trustworthy platforms for computing and. If two end-nodes are correctly operating, they can compensate for errors in the network. And if one of them isn't compensating, one can detect it and disconnect himself. So they protect themselves.

So what do we actually know now…what do we know now… is that the end-node is the most corrupted place, and the world, is full of malware. And you absolutely cannot trust the code that runs on the end node, and the network's so much more reliable than the end-node . The second thing we've learned is that most of the communication you do on the Internet is with people who you don't trust… you go to Web sites, but they could be full of malware and try to try to infect you. You get an email that could be malware, could be spam, could be phishing. So this assumption that you only want to talk to people that you trust was simply a socially incorrect assumption. And the assumption that the end-node was a valid place, was a trustworthy place to run code, was a technically… now notice, that that arises not because the Internet is not secure. But because of problems at… I would say… not, not the... Buffer overrun, sure, but we cured that. It has to do with the fact that we haven't figured out how to build application level code on the end-node, that can protect itself when it communicates with people that are untrustworthy.

So the end-node gets contaminated. And then we can't trust the communication. So the point is, I can now explain what the construct is that you have to get your head wrapped around in order to understand how to make the Internet experience… I mean that in the large, not the packet-bearer, I would make the packet-bearer service better or pah, that's not the issue, it has nothing to do with the insecurity of the internet experience. But I can now explain what the problem is and therefore you can begin to derive what structural responses one will have to do with redesign of applications and things like that. We didn't understand any of this. We tried to understand it. We worked on trying to understand. We just knew we didn't get it. And so it's not that we ignored security, "nah, we'll do it later" no no no no no. And just go look at the Red Book. This this is a network security book and you'll just see, we just didn't know how to think about it. We just didn't know we were…we weren't there. We had to let the system mature to understand what the threat model was, understand what the viulnerabilities were and so forth. So it's…it's complicated.

LK: Dave, given this history of the way the PIs interface with the funding agencies, I get the picture that the nature of unsolicited proposals, solicited proposals, specification of what the project should be, keeps changing hands as to who generates the ideas, who makes the requests, is there competition or not, can you talk about the world you lived through as those issues evolved?

DC: Well again my experience is very tailored because I've I've really only worked on one thing. And the problem we've dealing with in any decade has been different. You know the 1980s was the decade of trying to figure how to scale the internet up in the 1990s. For me it was the decade where we began to worry about quality of service and so forth and the 2000s we started dealing with social and political and regulatory and economic issues. But it seems to me that there's definitely been a shift in the space in the sense.

If you think about the traditional NSF grant, it's within a very broad scope. They don't call them BAAs, you know they call them solicitations, but they're broad, you know, write us a great proposal in networking. You know. Write us great proposal in artificial intelligence, like that. And if there's any conservatism in that system it's the conservatism built into the merit review process, which is again an emergent community driven phenomenon. And the merit review process is conservative, there's no doubt. Program managers are capable of being quite a bit more risk-taking than the merit review process is.

But DARPA has gone, I think, in two directions. They have these broad agency announcements which basically say: Tell us something cool. And clearly, smart program managers inside DARPA and other military organizations, you know, Air Force or Navy or Army Research, use BAAs as an escape path. So that I can come in with some cool idea that they've never thought of and sit down with the right program manager and say: Let me get you excited about this cool idea and I get him excited. And he says, yeah, well I've got this open BAA over here, so just write this thing in that context and so…

But I think there's another dimension of change here that's worth thinking about. When ARPA started working in the IT space, they were the only game in town, and NSF was not yet geared up to do this and the service labs like Army, Air Force and so forth, which I think are very good, very small, didn't have much funding firepower, so in some sense anything ARPA needed, it had to work through for itself, so if they needed fundamental research they had to do fundamental research.

But as the field has matured and there's more stuff going on, I think ARPAs sense of where they are in the in the sort of value chain of research production has changed, and I I forget which ARPA director, I was talking to at a conference. Tony Tether, maybe before that, I can't remember, but I was saying: You know there's work you do that sort of puts ideas on the shelf when you need them and then there's work to do that takes the ideas off the shelf when you need them, and he said: DARPA takes ideas off the shelf. We do not put ideas on the shelf.

And he said: We hope that there are other people in the ecosystem today that are putting ideas on the shelf, we hope that NSF is doing it. We hope the service labs are doing it. That's not our mission. Our mission is further down the value chain of production of procurable objects. So they had a lot of success with UAVs and things like that, and those really are taking ideas off the shelf. And I said but if you don't stock the shelf….? He said: Yes, if the ecosystem is, he didn't put it this way, but he said, if the ecosystem is out of balance, then the nation's in trouble, but it's not my job. So…. In the early days when I was getting many… in the 70s and 80s, the NSF program in that space was pretty feeble. And you went to NSF for a small grant, get a single PI grant, a couple of grad students, as a sort of a consolation path if you didn't get the money from DARPA. Because DARPA was where the exciting stuff happened, and I think one of the things that's happened over the.. say, the last twenty years, is NSF really became a substantial player in the space. So today, If I want to do something really innovative and really off the wall, I'd go to NSF, I wouldn't try go into

ARPA… and I would talk to the right program officer. But you know, if you… if you have a crazy idea and you want two years to see if you can develop a crazy idea, you can go to an NSF program manager and get…. What I guess they call today, an eager? They keep changing the names, they used to be called…but they keep changing the… but it's a…it's an exploratory grant. You know, a small….and they can just give it to ya. None of this merit review stuff they can just give it to you, at least give you money for a year. OK?

LK: How large are they?

Support. Faculty, a couple grad students for eighteen months you know. Hundred fifty thousand, I mean it's not a big deal, but it. But it says, OK, you need a year before you're at the point where you can write a proposal that will….It's like seed funding. You know it's like angel funding for bright ideas and they have program officers in NSF have, actually tremendous discretion. And most of them of course come from the community, we don't import Martians to run NSF, there are few lifers there, people who are permanent employees but most of them are rotators. And so, if I had a really bright idea today, something that was really crazy, I would go to DARPA, I wouldn’t be able to sell it.

LK: Would you have in the 70s?

Oh, of course that was the only place to go. And they knew that was the only place and they cared. But I think… So that's a question for the nation, which is why they were putting ideas on the shelf at the right rate to take them off the shelf? I think that that's a larger question than an ARPA question. It's much easier for the military to get Congress excited about something that it is for NSF to get Congress excited about something. You know… it's just… you know… Science in the abstract is hard to sell, in fact, I think NSF in the in the recent budget turmoil in Washington has done…CISE,the computer science part of NSF has done a pretty good job but I think it's because they can actually sell the potential outcome in terms of improving the posture of the nation or improving economic ,you know, innovation and economic growth and so forth. But…and at this… NSF at this point can have a much broader mission.

LK: Excellent. Okay, so let’s shift gears a little bit. As a P.I., as a researcher, there was clearly some level of collaboration with PIs at other institutions, other organizations, other institutions. P.I. meetings, for example, took place. How much cross-institution, collaboration did you experience, and what did you think of the P.I. meetings, if you were part of those?

DC: Well again, it was different in the 70s and the 80s and 90s. We didn't have P.I. meetings back then, the Internet working group had meetings. And that was absolutel inter-institution. I mean every, every operating system for which there was a Internet stack being implemented was being done by a different place. You know, so…Was Bob Braden at U.C.L.A. at the time? I think he was doing it for the 360? So he was, he was… so you guys were doing the stack for the 360 and somebody was doing the stack for Unix and somebody was doing the stack for this, now…so that was a team, but it was it was not united. We didn't write papers and presentd to each other, we were doing that, we were running code, it's that this code doesn't work dammit, you know, it was a very tightly knit group.

So all of my research… if you talk about the sort of fabric of my research, all of it was outward facing. I mean a couple grad students at MIT, but nobody else inside my lab was doing what I did. So I was entirely outward facing.

LK: Those meetings sounded focused, even though it was broad based across disciplines?

DC: Oh very very very very very focused, right.

LK: Were you involved with any cross-disciplinary activities which maybe seeded your thoughts in the field?

DC: No. when we got into the later stages of ARPA funding, ARPA was trying novel ideas in networking to see whether they'd fly, the most obvious example being active networking. They had active net P.I: meetings. And there were people who had funding inside active nets that were doing very different things. They weren't part of a coherent solution there. So we had active net PI meetings and we did present work to each other, because in fact we didn't know what each other were doing, because they were they were different...different threads that fit within that large scope.

Those were effective, I think, those are important. I think because you had people doing different things under one umbrella, it was useful to get them to talk to each other in the early stages of the Future Internet Project, when we did these small grants. We had meetings where we had those people talk, which… so we ran NSF P.I. meetings, if you will, within the scope of the fine grant, because these people…this guy's working on some novel routing thing and this guy's working on some security thing, in this case we're going to some management… we had to get them to talk to each other, so part of what I did was run….they wouldn't didn't call them P.I….well yeah, we did, we did call them P.I. meetings, we called them principal investigator meetings, and so the trick was to scope them, so that they're not a waste of people's time. And in fact, the way we ran these meetings was we always asked the community itself to nominate a planning committee. And help us organize the committee. Organize the meeting , so that it was the meeting the community wanted. It wasn’t like Darlene Fisher and I were going to tell these people what to do. We had planted every one of these meetings inside a planning committee, because we want to make sure… every.. after every meeting we said: Was it worth your while? You know we constantly check, you know, I don't want to waste your time, what do we do and what are we doing right, what do you want to do? W

We had student meetings as opposed to P.I meetings where we got the students from all these projects together, we're going to do that again in the spring. Students love it. We don't let the P.I.'s come. But the students talk to their peers. We make a student, well…sort of…We sit in the background and make sure it works out. But we make a student, make the projects, nominate students to become a planning committee and then we help them learn how to run a meeting. And then they run it.

LK: So a lot of these…the subject of the meetings was about networking

DC: Yeah.

LK: But how much was the network useful in providing the ability for communities to interact, how important was it that the network was there as a communication medium?

DC: Oh, it was critical. It was absolutely critical. But then again, that's a symbol of a larger thing that I think is important: we could not have done the network research if we didn't have a network to do… to facilitate the research at the time we were building the Internet. We had the scaffolding of the Arpanet and some of the old apps, we had email on top of the Arpanet and there was the day we cut over e-mail to run on top of TCP instead of NCP and so forth. We made ourselves, you know, in the current jargon, eat our own dog food. You know, But…but if we hadn't had the infrastructure, we couldn't have done this. But there's a there's a deeper issue here, which is I think our community is at its best when it uses its own tools, and if you look back to the early days of Timesharing, it was possible for a team that you could assemble inside a place like MIT, to build a complete system. You could build all the applications, you could build a word processor, you can build a word format or you can get when Xerox did the Alto inside a lab the size of PARC, they could build not only the system, they built the hardware the software and all the apps. So the Bravo editor and all this, you could build it and they, they used what they built, And today, an awful lot of the infrastructure we use is too big, too complicated, too opaque. And for reasons of pragmatics and productivity, a lot of us use tools that we could not reproduce in the lab.

You know I don't know whether you like Word, or you hate Word. But all of my collaborators today, sorry, eighty percent of my collaborators today use Word. So I have to take it. I'm a Word taker I'm not a Word maker. Right? So I can't do research in word processing and I don't want to, but if I wanted to do research in word processing, we're barely at the stage where an academic can write a new web browser, so in terms of exploring security issues or sandboxing or something like that, you can sort of start with something like the Mozilla core and sort of build a modified browser, but the world has gotten big enough, complicated, commercial, where it's harder for us to understand what we're building by using it. We built really good, multi-way audiovisual teleconferencing in the 1980s. It worked better than Skype. And it worked over... I'm sorry, Skype works very well, some of the time Skype craps out. I can't tell you why. I don't know what's wrong. But we owned every piece of the system, so when in the middle of a conference, it flaked out, we could figure out what happened, we knew where to go look, we could fix it. It was great and we had a shared whiteboard app that Van Jacobson built, it was fantastic. It was that was one of the best whiteboard apps I've ever seen, because multiple people could draw on the board. But it was always synchronous, so if you know if I was drawing and you were drawing at the same time we didn't… we didn't tangle open locks and so forth, you know. We could all just work, it was beautiful. He had a beautiful conception like my comment earlier about security, where you don't have the right framework, you can't… and he had the right framework for how to build an asynchronous, latency-tolerant shared collaboration space. So I think that we absolutely depended on these tools, but there's a deeper message in here, which is that we are at our best when we ourselves, are our own customers. And as we become disconnected from what it is that most people want to do with computing, we run the risk of misunderstanding what's important and if somebody says succinctly, said to us, said to me: Us ain't them. he was talking about, for example, our religion that, well it's great to push all the code to the edge because we've empowered the user. The typical user that does NOT want to be in power, have no clue what's going on, you know what the hell's a DLL? I mean, what's going on here, why do I have to do this. Somebody take this off of my hands. And we so… but if the code runs on your machine, you're in… and just…we just…you know…wrong, wrong. Different idea. And I use some of these tools and they're great, you know I use Dropbox and my colleagues will say, well… aaarrrh….guys, you know, it works!

LK: So there's a another set of activities in which you were involved, which I know have had some impact, one of them was the CSTB.

DC: Yeah.

LK: Do you want to talk about your experience and what that meant to you, what the impact was, what the conversation was?

DC: I have to say that the…the computer science and…. It was Computer Science and Telecommuications board… was an incredibly important thing in my career. It was transformative…because….and the first study I did for them, by the way, was on security. And it was in 1990.

LK: You were involved in 1988, in NREN?

LK: Right, so I was on the NREN report, but I chaired the security studies, so… but …I was on the first NREN study. Right. And then I got sucked in to being the chair of a security study in 1990. But, so first of all some of the studies were constituted, so that the composition was broader than the community I would normally encounter. And so I ran into economists, and librarians and activists and began to understand the power of talking to people that were clones of me and… and then in a while I got sucked into being chair of the CSTB. which of course broadened the agenda, because now we were responsible for trying to scope studies all across computer science in telecommunications, so we might do studies on, you know, the role of women in graduate school or robotics or technology in the arts. The board, or we created a committee that did a great study on the interplay of technological support for the creative arts and says all kinds of cool stuff, so it was an amazing opportunity to talk to really high caliber people who were not me.

And clearly my work has become much more interdisciplinary as I've understood that the constraints on the internet are not technical constraints, they're economic and cultural and social, regulatory. But I think, really, the… the delight in learning how to talk to people that…that were not clones of me, really was something I learned from participating in the CSTB, and it was just an incredibly timely and important thing for my career.

LK: So many of the people that populated those committees were relatively senior people. How do you protect a young faculty member from getting distracted from their focus, or tenure if you will, and yet enjoy the interdisciplinary nature of these meetings?

DC: Bluntly, you can't do it. Tenure, like anything else, is an emergent phenomenon, right? But it's pretty clear. Most universities today…what they look for in tenure is not breadth, but depth. And they want to see the hero papers. And, you just really, as a young faculty member, you don't dare get broad until you have tenure. I really think that's true. I got a wonderful quote from somebody that I was trying to persuade and I won't tell you what field they were in, 'cause this is somewhat snarky, but I was trying to persuade them to come play with me on something that was interdisciplinary, and she said to me: My field has fifty people in it and they stand in a circle so they can pat each other on the back. You step out of the circle, you don't get tenure. When I get tenure, I compete with you. So I think…. and I have a problem, because students look at what I do now which is very interdisciplinary and very broad and architectural, and they say: Well you know, you're really having a good time. And how do I sort of be like that? And the answer is: You can't do it when you're young and I didn't do it when I was young. You can't emulate, as a baby faculty member, what I do now. You've got to start out and establish absolutely impeccable credentials in an area. Prove your depth, and then go sideways.

LK: Seems like real defect in the system, because view of these deeply focused young people is not enhanced with the bigger picture that they could enjoy if they could investigate…

DC: Yes, but I think it's inevitable. I think that you do have to build an absolutely firm foundation in one field. I think if you are interdisciplinary at a young age, we don't… you don't know what you are, you're not quite this, you're not quite that, you're not quite the other thing, nobody knows what you are. I am, at heart, an Internet architect. So if I go talk to somebody about economics, they understand that I'm not an economist, but they understand what I am. But if I wasn't quite that I wasn't quite this, then why should I listen to this guy? OK, but because I bring impeccable credentials from one field, and I can talk to an economist who may have these impeccable credentials and we each understand how to calibrate the other, so I really think the question is not, how do we help these young students get broad early, or these young faculty members get broader. I think through… a question… the question is how do we help them understand that there's a point in their career where they should go through a transition and start looking sideways as well as down. I think in the early stages what the senior faculty have to do is protect the junior faculty. Not by getting funding for them, but by making sure that they don't take technology in the direction, where it's going to run afoul of some larger issue. But a lot of the younger faculty really deeply understand this. And let me explain why I say that you can be as focused as… you know, you can be laser focused on a technical problem. Until you try to start a company. And the instant you start a company you say, oh, is there a market? Is there a regulatory barrier? Can I bring it to market, at what cost, and of a sudden all these other issues falling on you. And so the fact that we are now in a position to talk to young faculty who would like to take a couple years off and go start a company. the fact that we now have, at MIT, we have mentoring programs for entrepreneurship. That's really where they're getting their breadth at a young age. It's not, it's not that their research has become interdisciplinary, it's that they are interested in this question of how their research can make a transition into the world. Whether it's a patent or a startup or something like that and then all the stuff is right in your face and you just have to think about it.

LK: You mentioned that you're an architect. And you've been heavily involved in the Internet Architecture Board. How did that play into the funding world, and what took place there of note?

Well we've, we've, we've reorganized. The sort of Internet…the operational technical management or, now people use the word governance, but that word's igot its own burd,n so in the beginning it was just a bunch of us and we got together, you know, and then we realized we needed a smaller group. And we hid… And we wanted smaller group and. Since we had no way to exclude people from it, we wanted to create a group that nobody would want to come to and so Vint had this idea of creating something, he called it the Internet Configuration Control Board, ICC. He gave it the most boring name you could think of.

And he was quite blunt about this, he said, you know, this is so the people don't think they have to come in here, because this is something boring going on, but then we can have a kitchen cabinet here, we did things. And then as we began to create working groups for specific areas. the first one was a routing working group which Dave Mills did and so forth. We realized that we had to sort of formalize a higher level group, but again we didn't want to use the word architecture because we thought that would mean everybody wanted to be in that room, so we just called it activities. OK? And that was again, it was a naming trick. And it really was a group that tried very hard. Few notable mistakes, but it tried very hard not to say: We're in charge or and so we're going to lead, but to have a somewhat step back, slightly contemplative view of what's going on, and see if it could exercise a little steering by nudging and pushing and so forth. And it really was a group that tried to..if it led, it led without power.

And in fact, there's..there's sometimes an advantage in leading without power, because the game you're playing..is..is clearer. I mean, I..I.. whatever… whatever we call this organization I was sort of in charge of in the 1980s and when Vint ran it in the 1970s, he was very technically competent, so there was no question when you were talking to him, and he, you know, he was your equal in the technical conversation, but you know he also controlled the purse strings. He could put his arm around you and walk you up behind the woodshed and say you know, that's a really interesting idea you have, and if you want to come to these meetings and try to integrate that idea into the way TCP is evolving, that's great, but if you think you're going to do your own little protocol because you don't want to do my protocol, you're going to do your protocol and get somebody else to pay for it. OK. And you did that. OK.

So when he decided to go to MCI and he asked me if I'd take this over. The question was: How are we going to do it, and I did not have the power of the purse. He said, well, I can do two things. And one of them is: I will give you a title, so I hereby deem you the Chief Protocol Architect of the Internet. What do you think you can do if all you have was a title. And he created a title for Jon Postel. So the two of us had titles.

And then the other thing he said is: Let me introduce you to…. There's a great scene in 'Crocodile Dundee' where they're out in the outback. And these guys from the city are way out of their depth and one is carrying a high powered rifle. He comes across one of Crocodile Dundee's friends and… and the guy isn't armed. He said, you need a gun, he said, I don't need a gun, I have me a Dink. And the guy said, what's a Dink? And this big guy comes up behind him and blasts him with a fist, like.. that's a Dink! And so I had me a Barry Leiner. OK? So I didn't need me no gun, I had me a Barry Leiner and one of the really important things here was that Barry and I got along. If we hadn't gotten along, it would have been terrible, but Barry controlled the purse. But he didn't want to be in my role. He didn't want that, really he wanted some other role, OK?

So Barry and I would go for a long walk. And we would talk about something like BBN and whether we should use the BBN Butterfly as a router. OK? And then he went off and made funding decisions. But he and I informally had a lot of conversations, so that's how we ran it in the 1980s, which is, we sort of tag-teamed the whole thing. But I think everybody understood that I didn't work for DARPA. I made no funding decisions. Exactly the same thing I'm doing with the Future Internet projecs right now. I don't work for N.S.F. I don't make funding decisions, I don't review proposals, that's entirely opaque to me.

If anybody who's working on the FIA project is considering submitting a proposal and want to send it to me, I can look at it. I'll tell them what I think. I have absolutely, I'd never sit on the merit review committees, I mean I precluded myself from all merit review. I deliberately said: I have no power. And they become disinterested.

I think some of the reasons Jon Poste was so effective in this place. Everybody just understood that Jon was not empire-building, he didn't want to be. He didn't want to sort of seize personal… he just wanted the Internet to be great. You know.

LK: It was interesting that the ARPA funding world was taking advice and delegating virtual power outside the ARPA formal structure…

Oh yeah, absolutely. Absolutely. And I think that may be somewhat uncommon. But NSF has done it with the Future Internet stuff. Very low key way. To a certain extent because I had some experience as to how to make this work. And so I could go to them and say, look if you do this, then I can do this for you, and I can do this and this, and I had to write a proposal for merit review like everyone else. But a tiny proposal. I made a promise to them: If you take me on as sort of a facilitator of this overall project, my promise in exchange is, I won't build my own empire. I will not try to be one of the FIA grant recipients. I will not fund an army of grad students to work on this. I will limit my role, so that everybody understands that I'm not competing with them. And it has to be managed, and it has to be that…then this has to come out of the community. I don't care what the… I mean Vint went to ARPA to do this, I mean, you know, It's not as if he came out of, you know, came out the chute with some sort of…he went to DARPA because he saw it as the best instrument to do what he wanted to do. And in that respect this is a community-driven activity, it's not that it didn't…well Bob was inside DARPA at the time but I mean, you know. So. So I think all of us who are trying to do something today, need to say OK, what's the best instrument to do it. And that has to do with how you position your own career as well. And a lot of people of course are somewhat ambitious. And they say, well, I'd like to build my empire at MIT to do this, or I want to build my empire at Stanford to do this, I'm going to have fifteen grad students. And you have to decide for the good of the community and the good for your career and so forth whether that's what is the best way to get something done. We all understood that the argument that had been used inside DARPA to sell this program at the time, to sell this program was what's now, I guess, called network centric warfare, that it was to say, what's automating command and control?

There's this misunderstanding that this network was built to be a post-nuclear network that really goes back to Baran's stuff when that was clearly what he was trying to do, but that was never on the table when we were doing our part and it was all in command of control. Nobody ever came to us and said you know there's this particular requirement because of command and control. Nobody ever said, well don't do that or do this. I think everybody understood that in some sense command and control looked a lot like office automation, looks a lot like a lot of other things, that if we could get good tools for collaboration and data, aggregation of data, dissemination, very basic skills, that we were building the right technology for automating command and control. So we knew that the selling story inside the military was command and control but I don't think that either constrained what we did or caused us to get what I might call quirky guidance or anything. I never, I don't remember anything like that, but I had no problem with making…with paying attention to the fact that the military was concerned about the military applicability of the stuff, I mean, that's fine.

And, you know, very early on, the intelligence community took on the challenge of understanding how to build a secure version of the Internet. And obviously they had clearances, and none of us did, so it was a one-way communication. I would sit in a room with our contractors and talk to them for an hour and at the end of the meeting, I'd leave. But they kept asking me to come back, so I guess we have a good one-way conversation so…so in some sense, the minute… the military, the intelligence community took on themselves the responsibility of how to tailor this technology to the particular set of problems they had. And only indirectly could I see what we talked about earlier, which was the emergence of the sort of negative result, which is they didn't know how to think about the…. They had, with the with the Orange Book they'd worked through what it meant. Maybe you think it's right, maybe think it's wrong, but they'd work through what it meant to build a secure system. They didn't know how to work through what it meant to build a secure, distributed system. They had to get there.

LK: In the early days, with so many great achievements, remarkable achievements. How would you describe the funding culture in that period that may have been important?

Well, it was clearly less competitive and more cooperative. I wasn't competing with somebody…there weren't two people funded….well, there was one case where this happened. But I never felt that I was in competition with somebody else to get my funding, there were actually two different groups funded to do a TCP for Unix. And there was a special reason for that. But… if you look at the current strategy where they do competing grants and then halfway through there's a demo and then they downselect. None of that. And I think that's corrosive. I think it's corrosive because if I know that there's a three-year grant but halfway through the program, I may get my contract canceled, you may get your contract canceled, the last thing I'm going to do is share my ideas with you. I'm going to hold every idea close. And I think that's corrosive. I think that's inappropriate and NSF made very clear in the Future Internet Architecture: We do not downselect. This is not a competition. Share your ideas. And I had to persuade, I had to take people aside and say: They actually mean it. You know, like they're not going to defund you in the middle of your grant, NSF does not defund people in the middle of their grant, they don't do that. So tell us what you're doing, so OK, you know, I can actually talk about it here. So. So I think there's a, I think there's a real issue here, about whether you set up research as a competition, which really motivates people. I mean look at the Grand Challenges like the like the Autonomous Vehicle. Or whether you set it up as something where you want to get the best by getting people to share their ideas along the way. I think there's a really deep question there about…about the best way to run the institution of research and how we share good ideas.

LK: So that was a funding question about what was behind ARPA's success. But in general, can you summarize what made ARPA so successful? Funding issues or beyond funding issues.

DC: Well, it was of course a creature of its time. I.T. was in its infancy. It was this wide open space. We didn't know what it was. Like almost everything you did, it was an exploration of the unknown. And I think they were very risk-tolerant, and they understood that certain number things were going to fail and that's the price of excitement. And they were risk-tolerant, and they bet on people that…they… It wasn't just well, have I written a good proposal this time, you know, it was what I think his track record is, what is he going to do? And as I said, the program managers had a lot of discretion inside DARPA. They weren't going out for merit review. So there has been some internal review but inside, the government, you know. So they had the scope to be risk-tolerant. Now I know there were some DARPA office managers who sort of… I would say sort of halfway through this trajectory we're talking about, you know, late 80s, early 90s, came in there with what I would say is the wrong mental model. They came in there very risk averse. And they said, you know, the metric of success for my funding is that all the projects come in on time and there are no failures. And everybody was yelling and screaming and saying nonono, thirty percent failure rate is necessary in order to get the. He said, I'm just not, , that's not right, that's not my job. My job is not to, my job…I was… I think I was supposed to be risk averse? So they were very risk-tolerant. And they bet on people and… and they, they understood both continuity and the importance of funding flyers that were going off in directions.

I think that there is more accountability inside ARPA now, which means there's an emergence of a conservatism, a sort of an emergent phenomenon in the space. I said the merit review process at NSF is conservative, that's an emergent phenomenon. Nobody said to NSF: Now, be conservative in what you fund. But the merit review process is scared that someday, it will fund something that fails. And there's this, you know, if you go get NSF money there's this art that everybody understands, which is, you have to do just enough of the research before you write the proposal, so that people believe you can do it, but it doesn't look like you've already done it. Because if you do too much, they won't fund you because you've already done it, but if you haven't done enough then they won't fund you because it might fail.

Nobody told… NSF never said to everyone: That's the way we do it. OK. That's emergent. But everybody said: "But NSF has no…" NSF doesn't! Those aren't the NSF rules. There's no rule that says you have to behave that way, NSF program officers can be tremendously entrepreneurial, I know that. This is an emergent phenomenon that emerges from the community itself in the merit review process. The exact same question of how polished does a paper have to be, to be in a leading conference? OK? We've…I'm going off on a tangent here, but I think it's relevant to this story. Our field does not have a lot of top-tier journals. We view top conferences as tenure quality. The equivalent of tenure publication. And in the networking area, the conference of record is SECON. So we've worked very hard in the top-tier schools to make the case, especially with respect to other departments, who don't have this kind of publication tradition that a publication presented at SECON is the equivalent of a journal publication in the quality of its peer review, OK? As soon as you successfully make that case, merit reviewing for SECON becomes conservative, because they don't want to have a paper that represents a failure, because then they might screw up the reputation that they developed, and so to protect the reputation, they become conservative. So you know why…What is the force inside NSF that makes a conservative, and there is one. There is a force inside in a sense that makes it conservative, and that's Congress. And you get this crazy congressman who goes and reads all the NSF Awards to find ones, you know the Golden Fleece Awards? OK. The Media Lab here got a Golden Fleece Award for making a digital video of the streets of Aspen, Colorado. And he made it sound like a fool's mission. They were doing the first experiments with street views and automated navigation. And it was amazingly good stuff. But he made it look silly. And you know…are you… is there good reason why you're off exploring the sex life of frogs? We have less trouble than other fields like biology and so forth… good. And it's really some anti-science guy in Congress and so there is a force inside NSF... by the way, the DoD can resist this, right? The military can say, military needs trump… and so we're going to take this fire, you know the reason we want to explore a sex life of frogs is that ten years from now, we can…we can solve this problem with terrorists. And they are… they salute, OK, because you said the word terrorist. In that sense NSF does not have the protection we have, to go back to a sort of a Vannevar Bush, which is getting a little long in the tooth and say, well, you know, science is good for the nation, so we need to know what the sex life of frogs and know… so….

But for all of that I think right now NSF is more daring than ARPA, which is too bad because I think that ARPA could use the… If they could wrap themselves in the flag of military need…and be a little more daring. But again, they tend to set the mission, they tend to say, well, we want to work on this problem and here's the approach. Not because they have to be as if they have a scapegoat and so forth. And again, program managers inside ARPA can be very entrepreneurial, so the question is what, what are… how are they evaluated inside the… inside DARPA. OK?

And you know, it becomes formulaic. I forget who it was, was it Hal Meyer who said, well, most people run out of good ideas after about three years. Somebody basically said, you know, you can stay at DARPA for a long time, because human beings run out of good ideas, which I think is stupid, but nonetheless, it may be true for some people, but not everybody. But everybody who comes into DARPA comes in with his own sense of the good idea that he wants to pursue. Well, how do you clear away space to push through your good idea? Well…you have to kill off the stuff of your predecessor. Because it's all funded, OK? So the idea that you're going to rotate, and not rotate them in to do merit review of a community driven process which is what they do at NSF. But you can rotate them in to extract their lifetime quota of good ideas in three years and spit them out again. Everything gets torn down as fast as you get them built up. So…so I think there are some… I think there are some corrosive consequences of some of the ways these systems are structured now, how are people evaluated? How much flexibility, I mean, that's… the National Science Foundation is a foundation for a reason, that was supposed to apply some independence from Congress, but you know, you know, a savage, snarky senator can still give you a lot of trouble, by going and reading all the awards and finding one that looks as if he can single it out as ludicrous expense, wasting the taxpayers' money and kill off the science stuff.

LK: So Dave, that was really exciting. Have we discussed any topics, you'd like to elaborate on or have we missed some topics, you'd like to comment on?

DC: You know, I think we've got the important topics, actually. This question of risk versus predictability conservatism. Big versus small, long versus short. Young and old careers. I think these are the important things. I guess the over… the overriding theme here is the field has grown up. And for people in our age, we just hit a golden time which was…it was a wide open field. Everything was blue sky, there were no fences in the West and…

and it was glorious, OK? And I think the students coming along today say, well you know, I'm really excited about this stuff but it's not quite as wide open and glorious and, you know… I was talking about the Internet to somebody and they said, you know, well, you know, you've been successful, you're obviously very good but a faculty member today cannot aspire to change the world… the..the system is…the ecosystem…You know, the people who study ecology say well, you know, ecosystems grow up and then they sort of rigidify because, as things have evolved in in in the mutually dependent ways the system sort of ossifies. And so you can create little niches, you know, you want to do a startup, you find a little, little hole in the space. And then you go in to occupy the whole.

And this is true, a natural selection is, two things can go in the finest holes, but we had to kill off the dinosaurs for the mammals to grow up and somebody very simply said, you know the dinosaurs didn't root for the mammals, so….the maturity the field means that to change the world is really hard now, so that's something… I mean Google changed the world. You know some of the startups are successful, but very few do, and furthermore, the students notice that the… that the people who have changed the world, like like the guys who founded Google or, you know Zuckerberg and Facebook, did not do it in an academic career, they did it by dropping out of an academic career. And so they say academics can't change the world anymore. We're support players. And our job is to train our successors, and send them out, maybe in the industry, one of them will change the world, so…so I think they… and I… you know, I say no, you know, you should be more ambitious than that. And they say, no, if I'm more ambitious than that, I won't get tenure.

LK: So that's the conundrum, not that you don't think you have the ability, but that it might endanger your tenure?

Well that's, that's one of the constraints, I mean the other constraints are just funding levels, right? I mean, when I'm really feeling cranky, I go look at the website that's run by the…. I think it's the AFL-CIO that runs a website where they track the total compensation of the CEOs in the Fortune 500. And they have an average compensation number for a Fortune 500 CEO, I think it's about fourteen million a year. And whatever the number is, I like to go to Washington and I like to see… some people talk about Iraq days and so forth, you know, I like to go to Washington and talk about funding levels in terms of what I call, you know, units of CEO, OK? So, NSF networking budget is about 3.5 CEOs and a typical startup today probably consumes, you know, ten or twenty CEOs before it goes public. OK? So if you say, well, at what point can I no longer look to research funding to develop an idea? You have to get off the federal research trajectory pretty early and get onto a private sector research trajectory. And that's why a lot of the students who are going to go off and do something really exciting, go off and start a company, because that's the only place to find enough money to do it. But if you go off and start a company, then you're under venture capital constraints and venture capital constraints are: No, we don't make bets to change the world. We make bets that you can find an ecological niche, and you can go live in it, and…. So a lot of people who think they like to change the world are very quickly disabused of that by the venture capital funding system, so…so the answer is, the only way I know to change the world today is to conceive of something big enough that you can't do it by yourself, but you can somehow build it up through some sort of academic collaboration, so that it takes people by surprise. You know other than that you're going to start a Google are you going to start a company and hope it's a Google. Going to start a company and hope it's a Facebook. You know. But I think we need, we do need role models for our time and we are not role models for our time. The students are correct in that respect.

LK: I could argue that trying to make a hit through technology is hard as you say. But you can move into a white space, take an interdisciplinary view, move that vector over a bit where there is a clearer running field. I often say that to students, because the students seem to be depressed with exactly the observation you made, it squelches a lot of it, you have to redirect that energy…

DC: Again it's an emergent phenomenon. I went…when we started the Future Internet Architecture project, I was worried about the careers of young faculty. And Darlene Fisher, who is the program officer in NSF that was doing that, she's one of the long term careers in herself. Program officers… Darlene and I, I think, collectively or together, spoke to senior faculty at most of the top ten universities. And we would say, if a student gets… or soory, young faculty gets involved in this Future Internet thing, which is going to be this larger collaborative activity, are we putting their promotion and tenure case at risk, because we don't want to do that.

If we're putting their....what's really interesting. At the top ten universities, the senior faculty were ecstatic about what we were doing. They would, you know…and they said, the junior faculty are more conservative than we want them to be. Taking intellectual risk is not career risk. But they don't believe that, because if you have a failure, you can't get the paper published and they think we're going to count papers, so we want to persuade them that intellectual risk is not career risk.

And I said to them: But how do you explain this to them, since a failure doesn't get a paper published and at the end of that, you read the papers and what do you do? And I also asked about collaboration and they said, you know the key to tenure for Top Ten university is that senior faculty like you and will write the letters. So the reason why junior faculty should participate in these projects is that they get to rub shoulders and get to know senior faculty who can then write very personal strong letters. And we can, we can look at those letters and say this is why this guy's career has gone the way it's went.

So now I went to the next ten universities, and they said we would advise our junior faculty at all cost to avoid this project. It would be career damaging, because we have not managed to work out, once the tenure case leaves our department and goes up to that level of the school or something like that, how to defend anything except a journal publication. And so as part of being a junior faculty in my department, I have to tell them to take all the SECON papers and turn them into journal articles in journals you've never heard of. And I said, that's why your department is not in the top ten and they said yes.

I mean there was no confusion about this conversation. So the better the department, the more the senior faculty wanted the junior faculty to be more risk takers than they are. And the less confident the department was of its own place within its university hierarchy, the more they told their own junior faculty they had to be conservative.

LK: And that became a self-propagating…

Absolutely right, this is all an emergent phenomenon. You know, MIT, we're forty percent of the whole school. You know, the whole university right? We can… if we want to put a tenure case forth to get a lot of firepower behind, we got a squarely tenure case, you know, they only wrote two papers, but one of them changed the world, you know, well they only wrote two papers, you have a look at the… we can get that case through. Other schools can't. You know, So. But you're right. Try it. I think it was Roger Needham who once said to me, I would rather do research with a shovel than a teaspoon. And he said, let's go find a field where nobody else is standing and go dig there. Get off the vector. Go sideways. And the trouble with the Internet is the Internet is so in your face, it's so there. And it's got so many problems and is so successful, has got so many problems that it's easy to decide that your role is to go fix those problems. And of course, as an academic you sort of have neutral standing. You're not representing Cisco's interests or Comcast's interests. On the other hand ,if you identify a problem that's really about to afflict industry, they have far more money than NSF or DARPA was going to give you to fix that problem they'll overrun you, they'l trample you from behind. Perhaps not with the solution you put forward but they'll trample you from behind with whatever solution comes out. And even the IETF is slow to get standards out so. One of the reasons we did Future Internet was not that I actually thought I was going to replace the Internet in ten years. I never thought that. I wanted these students and these young faculty to think out of the box, it doesn't have to look the way it does today. Think about something else. Think about… and the students are sort of looking up, so they're saying maybe I want to do the next Facebook, maybe I want to do the next Google I say yeah, but there's fun at the lower levels too. They don't have to be the way that… in fifteen years, they don't have to be the way they are now. Think about security, can we get security right? There everybody runs away. Think about management, can we get management right, everybody runs away twice as fast, this is you say management, God, talk about a topic nobody wants to talk about!

But this sort of take'em, shake'em kind of a thing, you know, saying don't be so conservative in what you do it as I said, the senior faculty in the top ten universities absolutely agreed with this. That could…they have protected the case of what I said. We haven't had…to the best of my knowledge, we have not had to have any tenure promotion problems as part of this project. I do not know of any problems. Who knows?

LK: Dave, this has been a wonderful interview. You're that David that never changes your ability to put forward ideas in a beautiful way, with a top-level view, with a global view. Thank you very much.

DC: Well you're very welcome. You've had a lot of opinions. But you know that's fine, this is an opinion moment.

LK: Well, you’ve always presented yourself this way. Each time we talk, I’m repeatedly impressed and amazed, your view of the world and how you put the pieces together, honestly…you’ve got a top-down view and a bottom-up view…

DC: Well I'm… Thank you. It's good!