Oral History of Pradeep Sindhu

Interviewed by:
Marguerite Gong Hancock and Marc Weber

Recorded: April 25, 2016
Mountain View, CA

CHM Reference number: X7781.2016

Marc Weber: So, I'm Marc Weber of the Computer History Museum, and I'm here today with Marguerite Hancock, and we are interviewing Pradeep Sindhu, the Founder of Juniper and a networking and semiconductor pioneer. So thank you so much for doing this, and welcome!

Pradeep Sindhu: Thank you, Marc. Thank you, Marguerite.

Weber: So tell me what is your full name and when and where were you born?

Sindhu: My full name is Pradeep Sindhu. Pradeep Singh Sindhu to be complete. I was born in Bombay in either 1952 or 1953, I don't know exactly which. But that's where I was born.

Weber: And talk a little bit about your childhood and what was your neighborhood like, what were your interests?

Sindhu: So I have to preface by saying that my dad was in the Army. And since he was in the Army, we actually moved every couple of years. And we generally moved from one place where the Army posted him to another place. So I never made fast friends, because I never actually was in one place for too long. And so I was, as a kid, mostly kept to myself. I was told that I was a shy child, and I spent most of my time actually making toys, making things with my hands, and generally just having fun as a kid.

Weber: And was your father interested in technical things? Did he work in technical things?

Sindhu: My father was not, I would say, interested in technical things. But both my parents actually encouraged me to do whatever it is that I wanted to do. Always.

Weber: And did you have brothers or sisters?

Sindhu: My elder brother-- so I have three siblings. An elder brother, elder sister, and a younger sister. My elder brother also followed my father in his footsteps. He was a Major General in the Indian Army. And my elder sister is a doctor. She emigrated to the US. And my younger sister is a child psychologist in India.

Weber: And your mother? Did she--

Sindhu: My mother passed, and my father's also passed now.

Weber: But did she work?

Sindhu: She did not work. She was a homemaker.

Weber: And so you say you like to make things. So can you talk what sorts of things?

Sindhu: Well, it depends on the age. But what I remember is if I had toys, I would take them apart. I would love to make things with mechano sets, when I could get my hands on them. And when I couldn't, I would make things with wood. As I graduated to being maybe nine, ten, eleven, I was playing with electricity, and making things with electromagnets. And I remember making a rudimentary sort of

communication system to communicate from one room to another. Discovered that iron is actually not a very good conductor. <laughter> So little things like that. It was all great fun. I was having fun.

Marguerite Gong Hancock: Well, thank you for telling us about your early years in the home and with your family. I'd like to now turn to your experiences in school. Could you tell us a little bit about your grade school experience? What kind of classes or teachers? What were some of your favorite topics, or least favorite?

Sindhu: Right. So I remember quite vividly all the way back to when I was maybe three years, three-and-a-half years old. And the class sizes were always small, because these were schools, Indian Army schools. Up until the point that we were in one particular place for a longer time. And here is where I can remember starting to become interested in science, and to become curious about, you know, why things were the way they were. So I've always been curious about the why part. My favorite subjects, I'd say science, physics and math, chemistry. Probably my least favorite subjects were history and the languages, especially grammar. I liked that the least. And in India in those days, you know, we used to be taught grammar, English grammar, Indian grammar and then the literature side. So there was a lot of emphasis on language. Which is a good thing, but they weren't my favorite subjects.

Hancock: Do you remember a particular person or experience or even perhaps an event that really crystallized for your interest in science?

Sindhu: I was in high school, probably ninth grade, tenth grade. There were two teachers, in particular, one a math teacher, the other a physics teacher. And the thing that struck me-- I can still remember this, it's a question of wave particle duality of light. And I can still remember that very clearly. Because it was a puzzle, and it still is a puzzle. Not just for me, for everyone.

Hancock: Thank you. In this time period, was that the same time period, or a different time when you then connected to computing, or so the digital part.

Sindhu: So, on the digital side, you know, I didn't do anything with computers, or computing, until my undergraduate, because I never had access to them. I think I probably knew that they existed, but didn't really have any experience with them. So when I went to my undergraduate school, which is IIT Kanpur, Indian Institute of Technology in Kanpur, they had actually a pretty decent computing center. They had a very powerful computer, they also had computers that the students could lay hands-on. And also the way we programmed them was by submitting computer cards, so card decks. I know I'm dating myself a little bit, but we also had a computer called a IBM-1620, which we got hands-on programming experience with. And that was a kick. And there was also a PDP-1 built by DEC, which had been transported to IIT Kanpur, and then we got to assemble that, as well. So that's my first experience with computers.

Weber: Going either then, or further back, what did you think you would be when you grew up?

Sindhu: You know, I don't remember thinking about what I would be as much as what I would like to do. And I liked to make things, invent new things. And to the extent that I thought about which area I would go in, I remember very clearly thinking when it came upon high school, that electronics, as it was called then,

or electrical engineering, which is a broader subject, were always the thing that interested me. Primarily because I sensed somehow, even at that age, that that's where the growth was going to be. Mechanical engineering, even though I loved mechanical things, interested me less. And this is a little bit strange, because I loved building things, mostly mechanical, because that's all I had access to. But nonetheless, what attracted me in terms of what I wanted to study, not necessarily what I wanted to become, was electrical engineering.

Hancock: This early recognition of both your interests and capacity, but also where things were going is interesting. Can you say-- where were you looking? Or what were your sources for identifying these areas of, as you said, things that you could make, things where there's innovation, areas of growth? What kinds of sources of information or people were you looking to? Or were they just coming out of your-- bubbling up out of your own kind of awareness as a young teen?

Sindhu: Honestly, I cannot put my finger on where the sources of information were. I was an Army brat. I do remember that at this time I was in New Delhi, and I was obviously interacting with other kids. And actually I had friends, because we had been in Delhi for about three years. The first time we'd been in one place for more than a couple of years. So I got to make stable friends. I read a lot. And you know, the subject matter of what I was learning was interesting. So probably those were the sources. Because I don't remember actually talking to anybody in the field at all. At least that's my recollection. So whatever it was, it kind of sort of came from inside, and/or maybe subliminal. I was very interested in radio, by the way. Fascinated by radio, because that's the only electronic thing. We didn't have a telephone at home, but we did have a radio. And the fact that you could listen to voices, signals from very far away was utterly fascinating to me.

Weber: So did you build radios or take them apart?

Sindhu: I certainly took them apart. I wasn't as successful in putting them back together, because they were all vacuum-tube based. When I was in college, I actually did make a transistor radio, pretty sophisticated one. But that was later, and not before college.

Weber: And your habit of filling notebooks with ideas, is that from later, or was this already going on?

Sindhu: It was not a habit that I had back then. Back then it was just curiosity and building on fact, building things. And trying to figure out how they worked, and seeing what else could be done.

Weber: And so in your studies, I mean, it follows pretty clearly from your interest, but did you change directions at all as an undergraduate?

Sindhu: As an undergraduate, so as you may know, the IIT has a pretty competitive exam to get into them. And they ask you to select what subject you want. And I had asked for electronics, particular at IIT Kanpur, because I knew that that was a very good school. And whether you actually got that subject or not depended strictly on your ranking on the exam and the entry--

<overlapping conversation>

Weber: So you did you get--

Sindhu: -- process.

Weber: And you gather the nuts.

Sindhu: I did get it and I did go to IIT Kanpur.

Weber: Talk a little bit about your university life and what did you like to do, or important people.

Sindhu: Well, university life was, you know, it was kind of a formative period, because when I went there I was 16, or 15. As I said, there's this ambiguity, at least in my mind. And we were there five years. It was a five-year program. And many of the kids that came were all very young, you know? Because nobody was 17, I can remember. And we were really kids, and we were learning together. And these kids, some of them were very, very bright. So interacting with them was fantastic. And that's typically how we learned; we learned from each other. But the teachers were also phenomenal. Really, really, really good. Especially the teachers in the EE courses towards the end. Some of whom I've kept in contact with up until now.

Weber: Yeah, any particular ones that really had an influence?

Sindhu: Sure. Two of them. There was a Dr. Kiar Surma [ph?]. And a Dr. Vicoise [ph?], Arun Vicoise. Both of whom I actually met recently a couple of months ago when I went to India. So both those guys had an impact. And also I would say Professor Cianar Rau. He was a Chemistry Professor at IIT Kanpur. The way he sort of connected people to natural phenomena was utterly amazing. For example, I still remember the little experiment he did in class where he poured liquid helium, and you could see it climbing up a slope, and then he explained why. And then there was also a Professor of Computer Science, Mushu Krishnan [ph?], and he had a pretty strong influence on me, because of his love for programming. And that, you know, I guess that influence is subliminal because it came out afterwards.

Weber: And obviously, because you then did switch to computing something clicked, but later, because you continued in electrical engineering in Hawaii, right?

Sindhu: Yeah, so in Hawaii, so this actually quite a funny coincidence. There's connection with PARC here, which came later. So I had gone to the University of Hawaii to study Communication Theory. And this is back in my interest in radio and all this stuff. And Communication Theory is all about how to communicate over distance, typically using radio waves. And you know, and coding information onto radio waves. So most of what you study is stochastic processes, probability, how to signal, encoding, so on. But during this time, I was also taking classes. And it turned out that they had-- University of Hawaii had been donated a computer called the BCC-500, Berkeley Computer Corporation. Well, BCC was founded by a guy called Butler Lampson, famous guy from PARC. I think he was either Chief Scientist at PARC, or certainly very high up at PARC. And BCC had folded, but that computer actually sat at the University of Hawaii in the EE Department, and was babysat by a Professor called Wayne Lichtenberger [ph?], who later was part of the founding tam of Cisco. So there's lots of connections back and forth. And Wayne was-- so I took a lot of courses in computing at that time. But the thing that was really awesome

was the fact that you actually got to program this machine. This was honest-to-God timesharing machine, where you could interact with the machine, you could try programs. Very different experience than punched cards, right? And so, this is the first time I actually got to play with what I would call the real computer that was somehow personal in nature - even though it was timeshared. But usually, I'd be the only person on the machine. So all the programming was in assembly, but it didn't matter, because I was having fun. Liked the idea of-- not that I didn't know about procedure call and stuff like that, but I really learned how to program there. And I learned about computer architecture there, because I got to take pieces of the machine apart, put them back together, got them to work, and so on. And I saw how things were actually built inside it.

Weber: And this is while you're still an EE major--

<overlapping conversation>

Sindhu: I would say this was the 1½ year period when I was in Hawaii.

Weber: Did you now Norm Abramson?

Sindhu: Of course, I knew Norman.

Weber: Okay, because he was Communications. Although not your--

Sindhu: Norm taught me a couple of courses. Thomas Gaarder taught me other courses. I knew David Slepian, who has sadly passed. He was one of the very influential guys in coding theory. Hawaii, by the way, had five or six of the top coding theorists in the world. And I learned a lot from those guys.

Weber: And Frank Kuo was there.

Sindhu: Frank Kuo was there. Abramson was there. Levin [ph?] was there, Lyn Shue [ph?] was there, and as I said David Slepian was there.

Weber: So how did you go-- why did you go from IIT to Hawaii specifically?

Sindhu: I had applied to a bunch of colleges. I could not have afforded to come without a scholarship. The only place where I got a scholarship was the University of Hawaii. So that was a specific choice, but it also helped that there was both computing and communications, which was heavily advertised in the University of Hawaii brochures, and that was attractive to me.

Hancock: I would like to then turn to your next step, which was at CMU.

Sindhu: Next step was CMU.

Hancock: So what led from there, in terms of focus as well as institution?

Sindhu: Well, so in discussions with Lichtenberger, you know, it was pretty clear to me that I was being pulled by computing, strongly. And so naturally led to discussions about what I wanted to do next. I

wanted to study it, get my PhD in computing, and you know, Wayne told me that it's basically Berkeley, MIT, CMU. And I got-- I don't know if I applied to MIT or not. I don't remember. But I got accepted at Berkeley as well as CMU. I didn't get a scholarship at Berkeley. Though some promise of, "Maybe you'll get it in the second semester," or whatever. But I couldn't afford to go, so I went to CMU. But CMU was also-- Wayne told me it is a better school. And I had a blast at Carnegie. It was the early days of computing, and Herb Simon and Al Newell and Alan Perlis, were the three key guys. And, man, you know, I thought I knew a little bit about computing, I really learned computing there formally.

Weber: And you switched your focus at that point.

Sindhu: I switched my focus and eventually I got a PhD in Computer Science, with a specialization in Distributed Operating Systems. And you know, CMU had a focus that it had a very strong systems focus. And systems means the grad students actually had to build stuff and get it to work, and then do research. And when you did research in systems, you had to build stuff, and then write papers on it. And my specialization was an operating system on the CM-Star multiprocessor, multi-microprocessor machine. And I worked closely with a grad student called John Ousterhout, who was, I think, one senior to me, or maybe two years. One year, I think. And I would say that that operating system was-- it was initiated by John, and the overall architecture was done by John, but I contributed significantly to the success of Madusa, the operating system on CM-Star. So that was the focus of my thesis. It was, you know, "How do you make distributed operating systems reliable?" That was my thesis.

Hancock: So you said you had a blast.

Sindhu: I did.

Hancock: And you found your field. You had your first sort of breakthrough in your own operating systems development. Tell us more about why it was so important for you at the time. What made it so significant?

Sindhu: You know, that's hard to describe. So I never liked hyper-specialization. What I like to be able to do is to understand how things work, but in working on machine architecture, which is what CM-Star was, as well as in the operating systems, I was able to span pretty wide spectrum in terms of depth. So some people specialized, you know, people often ask me, "Are you an expert in hardware or software in Silicon?" And I look at them, "I don't know what you're talking about." I learned about the entire stack all the way down to transistors, because that was my formation in a sense, EE. But somewhere along the line at CMU, I switched from asking questions of, "Why were things put together the way they were put?" To asking questions of, "Given the laws of physics, what could one do? What could one build that doesn't exist yet?" And I don't know why that transformation happened, but it happened at CMU. And then when I got my first job, which was at PARC, it got cemented, because I think experience at CMU almost certainly helped. Because you know, you could see that you can create things that never existed before, which were very, very interesting, and that had interesting behaviors. And so that just-- it was fascinating to me. So the switch from understanding natural systems, which requires the ability to analyze, with some formal machinery called mathematics, to a different way of thinking, which is synthesis, which is making things that didn't exist before. And probably the making things that didn't exist before reached back into my early

childhood. So that's-- I'm guessing here. You know, I don't remember consciously doing this. But in retrospect, there was a clear shift to analyzing things to actually building things.

Weber: And you saw people at CMU doing that.

Sindhu: Absolutely, it was the-- you know, the environment in which you work and live and study is phenomenally important. If you can call that culture, culture matters a lot. Because you're constantly coming in contact with people who are thinking that way. So whatever tendencies you have, they either get amplified, or they get crushed. Meaning that there were many people who joined the department-- it was already a small department. But their interest and delight-- and, you know, typical thing was class sizes were half by the time class graduated. It certainly was in my class. There were, I think, 16 or 17 of us, and about half of us graduated.

Hancock: So those numbers, that 50 percent sort of persisted, how do you explain your successful completion? What kept you moving forward and--

Sindhu: Sheer doggedness. And nothing more. I wouldn't attribute brilliance to it, it's just plough into it.

Hancock: Probably being modest, but can you can describe to us as you've moved from PARC, you mentioned, how did you get started? Your first role, your first people that you worked with? First project that you were on.

Weber: And why PARC?

Sindhu: Why PARC--

Hancock: And why PARC?

Sindhu: So I interviewed quite widely. IBM Yorktown; IBM San Jose; Bell Labs, both Holmdel and Murray Hill. And a few universities, UCLA, because of Kleinrock. Obviously, I'd read his books and thought very highly of him. Bell Labs just didn't attract me. Even back then it had the smell of death. I don't know if you want to put that on tape, but it was-- it seemed to me like a place where people had very high egos. But they weren't-- didn't seem to be doing very much. It was all in the past. And when I came to PARC, it was very different. It was the project that people were working on it was called the Dragon Project. And this is 1984, I believe. And the goal was to build a 100 MIPS personal computer at a time when I think machines were finding it hard to crack three or four MIPS. And the idea, which I think came from Lampson and Thacker and folks, was that if you put a lot of computer power in front of people, they will use it do very, very interesting things, and progress really fast when you do that. And the key idea was to make computers personal, and even back then it was hard to see how to do that with a single processor. And the fact that I had a multiprocessor background actually was coincidental, but a happy coincidence. And so that project was very interesting to me. And so I actually joined the project, and the project leader at the time was Ed McCreight. Actually a bunch of the original PARC folks were at PARC in the Computer Science Lab, which is where I was. There were three labs. There a was a System Science Lab and then there was a-- and I forget the name of the third, maybe ISL, which was focused more on AI and Lisp and so on. Lampson, Thacker and a bunch of the original guys had left to go to DEC.

Weber: Right, with Bob Taylor.

Sindhu: With Bob Taylor, correct. And so the atmosphere when I came was a little bit depressed, because these guys had just left, you know, a month ago. But I had the time of my life there! Because I learned a lot. We were building the architecture of this machine from ground up, silicon, software, microprocessor design, caches, there were lots of things to be invented. The design tools to build them. So I was interested in how one could simply build more complicated systems. Not simply because you wanted to build complicated systems, but usually it's tools, having good tools that help you build or conquer higher level of complexity. You obviously want to make things as simple as possible, but sometimes it's not possible. And so then what limits you are the tools. So I worked simultaneously on design tools on the one hand. And worked on the design of the machine, and the particular aspect that I chose was cache coherency. Because this was a multiprocessor machine, and because-- by the way, even back then this was-- the strand was completely in place-- but the speed at which microprocessors were improving, the clock speed, was going much faster than the cycle time of memory. Which meant that memory was a precious resource. You had to have caches. Each CPU had to have a cache, and then you had this cache coherency problem. And there was no good solution at the time. And so the-- what I worked on was cache coherency algorithms. And I think it might even be some of the first published work on cache coherency. And the particular protocols that I invented, some variant of those are still in use. Because it's a general problem. So snoopy protocols, pipeline busses, cache coherency are the areas that I worked in.

Weber: And how many processors were you generally when you--

Sindhu: I think the design was either 12 or 16. Memory fails me right now, but something like that. It wasn't hundreds.

Weber: And had you looked, at all, when you were looking where to work at a-- BBN was working on multiprocessor stuff.

Sindhu: BBN was working on multiprocessor stuff, but they were doing a butterfly-based design, if I remember correctly.

Weber: Yeah.

Sindhu: But we didn't look at that. I think the BBN machine probably did not have caches. What they did was they had processors on one side of that switch matrix, and memory modules on the other, but I don't know that they had caches, because if they did have caches, you had-- they had incoherent caches.

Weber: And you weren't look-- when you were looking for where to work, you didn't yet know you wanted to do the multiprocessor, right?

Sindhu: No.

Weber: BBN didn't appear on your radar for that.

Sindhu: BBN was not on my radar actually when I interviewed, not at all.

Weber: So back to-- so at PARC.

Sindhu: So back to PARC. So you know, I said the budget was ambitious. We actually did build, I think it was a set of three or four chips and got them to work. But building that much silicon, writing the software and then boarding the entire stack using CEDAR, the computing environment, on top of this, was a very tall order, and required lots and lots of money. And by happenstance, Sun Microsystems at this time was-- it had workstations, but it was poking around on servers. And the servers that they had were actually pretty small in scale. Two-process servers and so on. And the particular thing that they were missing, they had the SPARC instruction set, and microprocessors based on SPARC, but they did not have the memory system figured out. So it was a very happy coincidence. And there was a guy called Wayne Rosing.

Weber: Oh, yeah. Who went to Google much later.

Sindhu: Perhaps. He was actually-- he got it instantly. And so the proposal, long story short, was we made a proposal to them that, "Hey, let's take this design that we have, and we'll remove the Dragon Processor and substitute it with the SPARC processor and build the machine based on this using the algorithms that we had developed. And that project-- so Sun eventually agreed, and that project led to the development of the Sun Dragon machine. Which I think in the-- which took two-and-a-half years or so. I think 1988 to 1990. And I think in '91, the machine shipped. And if I recall, the first year the machine did close to two billion dollars. So it was a successful machine. It was a first large multiprocessor server machine that Sun did. And it was very rapidly followed by a half-size machine. So two machines, bam/bam, you know, within six months of one another. And then Sun went on to build other machines based on that technology. The team of 11 PARC researchers came back to PARC and what we decided to do, we made a proposal to Xerox, said, "Hey, you know, the work that we did at Sun, perhaps we can use it to directly benefit Xerox." The problem that Xerox Corporation was trying to solve was Xerox was a paper and toner company, when it needed to be a silicon and software company. And that was a big mismatch. And by the way, probably the reason why these guys left PARC in the first place. So we thought, naïvely that we could help by taking the technology we had developed to use at the engine inside, document processing machines. It made some sense, but in retrospect doesn't make that much sense. And the reason it doesn't make sense in retrospect was that while we were able to secure the funding for doing this little startup inside of Xerox, we were treated as a foreign entity inside the company. For example, you know, we went in thinking that most of the document processing algorithms were already figured out, and that all we had to do was implement them using this machinery, but it wasn't the case. And so I remember spending along with my colleagues, some year-and-a-half, two years, developing image processing algorithms. And then adapting them to run on a general-purpose machine. This lengthy time, along with the fact that each of the Xerox divisions wanted to do their own development, eventually, you know, I figured out that this was not going to work. And the experience I'd had with Sun, or at Sun, was that everybody's focused on solving the same problem, and it's a win-win situation. That's what got me thinking about a startup.

Weber: And before we go to that, what was the name of the venture within PARC?

Sindhu: The venture was called Xerox Impact.

Weber: Okay. And did you know Larry Masinter and System 33 in that period?

Sindhu: Larry Masinter definitely rings a bell, but...

Weber: Okay, no, that was document-related at PARC.

Sindhu: Right, right, right. There was a whole bunch of work being done at-- in El Segundo led by a guy called Ron Rider. And Ron Rider was actually instrumental in helping this Impact Division get funded.

Weber: And you were funded by who, and for how much?

Sindhu: I don't remember the amount anymore.

Weber: I mean, roughly. Was it a big effort. It was a big effort. We were some 30 to 40 people for a couple of years. That's serious money.

Hancock: You talked a little earlier about the sort of impact of the products coming out of that, and the monetization. Is that right?

Sindhu: Yes.

Hancock: Can you say that again, what--

Sindhu: You mean the impact of the monetization at Sun? Or within Xerox?

Hancock: Within Xerox, either one. We haven't even--

Sindhu: Well, within Sun, I mean, those machines were quite successful, and I think-- you know, at least my impression is that had Sun not gone into servers when it did on the strength of these machines that Sun would have had a shorter life than it ended up doing. Because workstations were getting commoditized very fast. And there was religion back then which went something like, "Multiprocessors are bullshit." Which turned out to be wrong, but hey! All of us make mistakes.

Weber: Where at?

Sindhu: At Sun.

Weber: Okay.

Sindhu: That was religion at the time. But fortunately, Andy Bechtolsheim was very supportive of the project, and you know, Andy's been a colleague of mine for a long time. He was at CMU for a very short time with me, working on the same project.

Weber: So there was internal competition to do the multiprocessors.

Sindhu: No. There was the belief that multiprocessors are somehow evil. Because there was too much faith on "Moore's Law."

Weber: You didn't need the complexity of multi-processer, you could wait for the hardware to catch up.

Sindhu: Absolutely. And at the time, it was a fair bet that-- and by the way, I will be the first one to say that if you can build a single processor machine that continually improves exponentially in performance, that's what you should be doing. It is simpler. All this multi-process stuff is complexity, but today, there is no alternative. And even back then, there was no alternative. It's complexity, but necessary complexity. And that's the part that was not appreciated throughout the corporation.

Weber: Butther people wanted Sun to go more-- to go into servers, but single-processor servers waiting for--

Sindhu: Well single-process servers were not competitive. That's the problem. Because the whole idea of servers was that-- workstations were for a single individual, possibly running hundreds of processes. But the focus was one individual. But servers typically were running many, many, many jobs in parallel. So you didn't need it to be able to run a single job on a dozen microprocessors. So what mattered in servers was throughput. You didn't necessarily need to break a single task into multiple pieces. That's complex.

Weber: But I guess what I'm asking is where is it the multiprocessor-- the success of that that led Sun down the path of doing more servers? Or were there-- were you fighting against a more traditional way of doing servers with the Sun processor?

Sindhu: Actually, once this project started to be successful, there was no fight. But you know, I left there in 1991. As far as I can tell that product and its--

Weber: Iterations.

Sindhu: Its follow-ons were very successful in the marketplace. And Sun eventually acquired Cray also, I believe, or some piece of Cray. And they built bigger and bigger machines. The largest one, I remember, was the 64-processor machine. They got into databases and so on and so forth.

Hancock: While we're focused on PARC, I want to make sure I understand your evolving role. Because we talked about some of your early, early projects, but over time, tell us about the other roles and responsibilities that you led.

Sindhu: Well, I got promoted. And I think I held the title of Principal Scientist at the end. But I didn't want management responsibility, because I didn't think I was a good manager. And I wanted to focus on doing technical things. And you know, that's been my focus pretty much throughout.

Hancock: When we interviewed you earlier, you talked about having your notebooks begin, I believe, during your PARC days. Is that right?

Sindhu: Absolutely.

Hancock: Technical ideas.

Sindhu: My habit of writing things down, writing ideas down, fleshing them out, keeping notebooks, actually engineering notebooks dates back from that time.

Hancock: Was there a particular antecedent or inspiration, or did you just begin?

Weber: Or did they order you to keep--

Hancock: Or were you requested like Fairchild, where you had to--

Sindhu: No, no, no, there was a-- PARC was-- you know, the one thing I will say about PARC was I have not encountered a better environment in which to create. It was truly fantastic! And one misses environments like that. I think Bell Labs was a place like that a long, long time ago. And it's-- maybe Google is a place like that today. But there was at least one-- there was a book on Bell Labs, I think "The Innovation Factory" book, where the author makes the point that perhaps only monopolies can throw off enough money to support labs where interesting work can be done." And there's something to that! Because otherwise you're into this quarter-by-quarter making revenue and making profits and so on. And you can only think about incremental things. And not all things can be done incrementally.

Weber: And you need a long time horizon as well.

Sindhu: You need a longer time horizon, exactly.

Weber: What made you think you're not either good at managing, or interested in managing?

Sindhu: Honestly, sometimes people are pain to deal with, and unpredictable. And it's just not a preference that I have. I've done it. And people tell me I've done it reasonably well, but if given the choice, I would stick to managing fewer people. So for example, I've never liked huge teams. And I've not liked huge teams because as teams get bigger, politics becomes a bigger piece of the puzzle compared to when you're smaller. And the problem of communication within the team becomes a hard problem. And I don't have a great solution for that, so I stick to things that where I think I can apply my competence. So it's more a preference than an absolute, "I won't do it."

Hancock: So I'd like to now turn-- did you want to talk about Sun, or should we-- I guess we've covered that.

Weber: I think Sun you've covered pretty well.

Hancock: We've covered that. So that brings us--

Sindhu: That was also super, okay? That time was very, very good! Very productive time.

Hancock: Say more about that, you know, it's hard for institutions to be really-- to support really creative and innovative people. Would you say something about what made those places so fertile for you and others that were working at the time?

Sindhu: Well, so Sun, in particular, it was a little rough. You could imagine Sun, a very successful company at that time, that here's these 11 people from PARC that are going to teach Sun how to make a computer/server. Excuse me? How does that work? <laughter> So we were called a paper production factory, because we wanted to document things, because when something's complicated. But I think that Andy's insistence that this was an important project was absolutely instrumental in getting the right management in place and giving us the cover to actually execute. And we did execute. But there was always this sort of feeling that, "Oh, no. We should be able to handle it ourselves. And who are these guys from PARC telling us what to do?" But I think once the teams started to work together, people had respect for each other's abilities. And I think after the first six months, I'd say things went much, much better. Because then people had the respect that, "These guys aren't really bozos, they're trying to help us!"

Weber: And you were fulltime. I mean, I assumed you were--

Sindhu: We were full resident. We were in all respects like Sun employees, we had badges. Except we weren't paid by Sun. We were paid by Xerox.

Weber: And what was in that for Xerox?

Sindhu: So, if I remember the deal was that Xerox would get some kind of a better discount on machines that they would buy from Sun. I'm not sure it was a great deal for Xerox, but you know, Xerox did get something out of it.

Weber: And this may or may not be the-- maybe it comes later, but the two things, I wonder when your interest in networking began, because obviously Juniper brought you squarely into that.

Sindhu: Sure.

Weber: And then you seem to have a long-standing interest in exploring physical and scaling limits. Where did that-- where and how did that start?

Sindhu: That's actually a very perceptive remark that I'm interested in scaling and pushing boundaries of performance and scale.

Weber: And that's not from me. That's from someone who knows you better. <laughter>

Sindhu: So I don't know where that come from really, but it's been there for as long as I can remember. And probably comes from, okay, it comes from, actually, I think a combination of two things. I've always wanted to understand the limits of things. Meaning when you're trying to analyze something which is given a natural system, you try to explore it to the point that you're actually satisfied with a level of understanding. I've always had that aspect of wanting to know how things work. And to reduce things

down to its atomic pieces. Once I flipped over to making things rather than analyzing things that existed, I think you can see naturally that you're trying to explore, "Well, how far can I push this thing? Can I make it ten? Can I make it 100? Can I make it 1,000?" And there were-- as long as I can remember, I'm used to-- you know, whenever I test out ideas, I'll poke the limits, and goes to infinity, what happens. Or if things become very small, what happens? Because you find very interesting things at the limits.

Weber: Like what?

Sindhu: Maybe that's where it comes from?

Weber: And what do you find at the limits that attracts you? When you push it to the limit.

Sindhu: I think you get a deeper understanding of what can and cannot be done in principle. And you know, maybe this is fast forwarding to today, but I think when you build things, there're kind of three types of laws that you need to base your work on. One of these are the laws of physics, don't try to violate them, because they'll bite you. Second laws are the laws of economics. Really, really important, because you can't violate those either. I try not to violate them. And a third are, I don't know whether to call them laws of programming or laws of manufacturing, or a combination of the two. Which is when you put together something, whether it's a piece of hardware or software, whether it-- making it modular and making sure that the interfaces between these parts are clean and long-lived is absolutely essential. And to me, it's almost a law. Doesn't come-- it isn't as crisp as the law of physics, but I don't think you're going to get any more crisp in this domain, because it's a little bit fuzzy, but if you violate those things, when you're building hardware, you get slapped on your wrist by Mother Nature right away. In software, you don't get slapped right away. But three or four years down the line, people won't be able to maintain the software.

Weber: Right, it's maintainability is the--

Hancock: Sustainability.

Sindhu: Exactly. So those are some of the reasons that I like exploring limits.

Weber: And was that part of what led to Juniper? Or were you already thinking in these terms?

Sindhu: No, what led to Juniper was essentially frustration. And many people at PARC, much more capable than myself, have reflected on this. They tried to work within the system of Xerox, but were not able to. And we thought, perhaps, we could do better. But it didn't work out at the end. And the conclusion I came to after spending two years at Impact, Xerox Impact was that the only environment in which one can do things that I was interested in doing, which is building systems, was in the context of a startup. And so then the question was, "Okay, so what kinds of systems?" And this is why I left PARC, left Xerox. Took a leave of absence in I think June of '95, to spend time to think about what I wanted to do next. And I looked-- I explored three sort of different areas. One was in computing. Okay. I'd done image processing, and prior to that, computing at Sun. In computing, I had no good ideas. And not to say there weren't good ideas, I just-- I couldn't find them. I looked at graphics. There were a whole bunch of graphics companies. Silicon Graphics was a good example. It seemed to me to be a crowded space.

Then there was networking. And networking broke down into sort of wide area networking, and Ethernet switching, so the local area network. The Ethernet switching space was overcrowded. And I also felt instinctively that the barrier to entry was smaller. So there were two tests for me. One was it had to be a domain where I had some expertise to bring, otherwise why do a startup? And secondly the barrier to entry had to be high. And third, the problem had to be really large and interesting.

And so that kind of naturally led me to look at the wide area. And I saw three things that caught my attention. I remember actually before going on vacation, I'd taken a bunch of *Red Herring* magazines. I don't know if that magazine is still around, but at that time it was very interesting every time. So there were three things that I saw which were going on. One was that no matter which way you looked at it, the bandwidth on the network was exploding. Whether you measure it terms of number of servers, number of plans, the number of bits per second, number of websites. No matter what you took, it was exploding. And numbers I remember-- I may be off a little bit, because it's been a long time-- I think it was doubling every four months or so. So very fast! And here again so my fundamentalist-based thinking kicked in and I said, "Well, if it's exponentially increasing, why is that?" And remember my background from PARC. So what I put together was I said, "You know, the internet protocol has been around for a long time. Packet switching has been around for a long time. People have been using email for a long time. But why do things explode right now?"

And my analysis was very simple. It was the invention of the World Wide Web, which made ordinary mortals, by pointing and clicking, able to use these pretty complicated things underneath. I mean, I was just using FTP at the lab and stuff from my days at University of Hawaii. So I knew that these things had been around, but all of a sudden they were becoming much more used. And so that was my analysis. That, by the way, combined with the work-- fundamental work that was done at PARC in bitmap computing. And providing easier to use interfaces for people. So all of that stuff was coming together. And so I said, "Okay, that makes sense." That means that this exponential increase is here to stay for some time.

The next one was I knew that the cost of long-distance bandwidth was dropping like a stone because of advances in optical fibers. So there's this thing called the bandwidth distance product of any transmission medium. And copper was the medium that was in use. And there's an enormous difference in the bandwidth distance product of copper versus glass. And glass was getting better and better at exponential rates. So I figured that, "Okay, so the cost of the wires to connect Point A to Point B is going to go down, is going down. What else would hold this thing back?" So it was very natural to look at the technology used for switching. And there were three tech-- four technologies around at that time. There was IP. There was ATM [Asynchronous Transfer Mode] . There was frame relay. And there was time division multiplexing. TM was sort of the old technology. Didn't do packet switching. Also, I'd been exposed to internet and IP deeply at PARC.

Weber: Oh, okay.

Hancock: That's important.

Sindhu: Because Ethernet was invented at PARC.

Hancock: Sure.

Weber: Because that's what, yeah, I mean, we can come back to it in a second. But I did want to ask you about your-- how networking suddenly became such a big thing. But there was more of the backstory then.

Sindhu: There's more of a backstory. It was in the air. It was taken for granted. And it's stuff that I learned while I was at PARC.

Weber: So you knew, you used Ethernet, but--

Sindhu: But I was also curious. I knew-- I wanted to know how it worked. And you know, the programming environments at PARC were such that you could explore all the way down to the drivers. You could see how it worked.

Weber: And you had mentioned being interested in, for instance, UCLA, because of Len Kleinrock's work in queueing theory.

Sindhu: Very interested in packet switching.

Weber: So that's already very hardcore networking.

Hancock: It's in part of their--

Sindhu: It was in the background.

Weber: But you had never actually sort of thought of yourself as a networking guy.

Sindhu: No, no, no, no. I would never call myself a networking expert, no. I was much more interested in computing. But all these things came together, and I said, "Well, ATM I rejected simply because ATM made it easier to build the devices. When we tried to build a network out of ATM, it was much more difficult." And I also figured that ATM was on the wrong side of probability. And let me describe that, because it's important. IP is a connectionless protocol.

So if I give you a network, an IP network and I drop a packet, IP packet anywhere into the network, it'll find its way through the grace of routing, to the destination. The way an ATM network worked was that the ATM noes were switches. And if A were to communicate with B over an ATM network, then A would have to initiate a fast setup between A and B through some other machinery. And then you would establish that path, and once the path was set up, now pack-- cells, ATM cells would float really fast. Well, if you actually did the math and you said, "You know, how quickly can you do this file setup and teardown?" Not very fast. People have tried valiantly to make that go fast, but there's certain laws of physics that come in the way, because you know, the round trip times are in the 80/100 milliseconds. And you can't do very high file setup and teardown rates. And my final conclusion was that, you know, this way of communicating which was this three-phase, where you set up a communication, then you actually do the communication, and then tear it down, this was a holdover from telephony, because that's how telephone

switches worked. And even back then, what was clear to me is that most of the communication that would happen would be from computer to computer, and only indirectly between people. So in other words, even though people might be the end users, they were communicating through computers. So the essential thing to figure out was, "How can I make computers communicate more efficiently with each other?" And then everything else would follow. And so this clearly pointed to IP. And--

Weber: But I mean at the time, ATM was popular partly because people thought setting up real-time video streaming, things like that, it makes it easier, right?

Sindhu: Yes, and there were all kinds of religious arguments against IP. "Oh, it was too expensive. Routers are harder to build than switches," blah-blah-blah. Well, I knew a thing or two about VLSI. I knew that you could put more and more logic into silicon over time. And even though nobody had actually put the forwarding path of a router in silicon, I knew enough about the IP protocol, to know that, well, you could do it. And so-- but what took me down this path was the scalability of IP. So I said why-- what ATM did was it set up a circuit. Well, for the entire discussion to be-- the communication to be successful, once you've selected the path, all of the machines on the path had to work perfectly for the duration of the call. Well, let's look at IP. For an IP communication to fail to work, all paths between A and B have to be cut. Even if there's one path existing, the communication will still work. This is why, fundamentally, I said, "No, ATM is dead." There were two reasons actually. One was the call setup rate.

Weber: Time <inaudible>.

Sindhu: And the second is it was on the wrong side of probability. So that said, "Okay, it's IP." Because its the only scalable technology, and networks matter only at large scale. Small scale networks are not useful. The most useful networks are networks at worldwide scale. And in my conception, only IP was capable of doing that. So then the question was, "How are IP routers built?" Because I had no knowledge. So I was very excited when I came back. Actually took a look at BIA routers, Cisco routers and so on. And I saw that, man, the hardware technology that was used to build them was ancient. Ancient!

Compared to the stuff I'd done at Sun two-and-a-half years prior to that. The machine was running at 20 megahertz, for example, using microprocessors. And the entire forwarding path, that is, moving packets, done in software. And so that's when the light bulb went off and said, "Okay, let's build a machine where you split the functionality of a router into two pieces. The so-called forwarding piece, which touches every packet. And that needs to be in silicon, because it's simple enough that you can put it in silicon." And then the other piece which is the management of the routers and the control plane. Well, the control plane is the set of algorithms that figure out what the topology of an existing network is. By sending “hellos” to each other, and you can literally think that each router makes a little map of the universe around it, in terms of who's connected to whom. And once you have that map, it actually shoves that map down into the hardware, so that when packets come in, you know, based on the destination where to send them out. That's it! So that was the idea of Juniper.

Hancock: So with that, how did you take this idea that you've put together, you've evaluated the principles, looked at the idea, and how did that turn into-- from a germ of an idea to a company?

Sindhu: Well, before it turned into a company, I had to satisfy myself that you could actually build something like this. And I did that by working out the problem to a certain scale, and did a design, some of which is actually reflected in that notebook on the table. And once I was satisfied that, yeah, pretty much you could do it, and it would be substantially faster than what existed. And the number I had in my head was about 20 times faster. In technology, you don't get 20 times faster very often. And so this was very exciting. So I shopped the idea around I asked some people, friends at PARC, "Hey, do you have contact at VCs and so on?" And they said, "You know, don't make cold calls, because you won't get anywhere." Well, I didn't listen. I made cold calls, and I didn't get anywhere. <laughter> So and I didn't get anywhere until I talked with Vinod Khosla at Kleiner Perkins. And Vinod, obviously, so he gave me an hour. And he called for a reference, Andy Bechtolsheim. And Andy must have given him a good recommendation, because Vinod set up follow-on meetings with other partners at KP. And these guys were pretty well-satisfied that, A) there is a market; and B) there's something very interesting to be done by putting together a team. And so we, you know, Vinod agreed to start a company. I think this was maybe December of '95. The exact date, I don't remember. But the company was incorporated in February of 1996. And I remember Vinod then asked me, "Don't compromise on the people. Hire the best people you can find." And he was exactly right.

Weber: And why-- you have this idea. You've at least drawn it out. Did you have a prototype of it?

Sindhu: No, it's on the notebook.

Weber: Just a pure drawing.

Sindhu: It's all on paper and in my mind.

Weber: So but why start a company rather than-- did you think of shopping it around to Sun, for instance, or--

Sindhu: No.

Weber: What made you want to do it on your own?

Sindhu: <laughs> The experience of having worked at Xerox, and tried to-- so you know, all during my years at PARC, I had plenty of ideas. Very few of them saw the light of day, except for one, which is the one at Sun. And the experience of actually seeing machines that you've designed being used, it's very, very nice. Very satisfying. And so what the Sun experience told me was that a good way-- perhaps not the only way-- was to do it in a startup setting. This was the example of Andy Bechtolsheim. The experience at Xerox also told me a poor way is to quote/unquote "shop" the idea. And by the way, I did go and talk to some folks at Bay Networks. And it's the usual thing, right? "We have some projects going on. This, that and the other." And I said, "Okay, enough. Thank you very much."

Hancock: I'd like to hear a little bit more about that first conversation, sort of "Aha!" when you connected with Vinod. Do you remember some of the elements of your basic pitch to him, and then how did that evolve?

Sindhu: Sure, sure, I remember. I was actually-- I think I was on a trip somewhere. Because right after that discussion-- it was a one-hour discussion-- I had to take the plane. And I didn't think much of it, because the past six meetings had not been very successful, so I wasn't expecting anything. But Vinod asked a lot of questions. And he asked me, you know, why do you think this could work? So I told him that, "I have a design. And I'm pretty confident it can be done. There's no logical difficulty. I's just a question of hiring the right people, putting the money, and getting it done." And he asked me why I thought what I was about to do-- so I gave him all the three elements that I told you about, which is the exponential increases in bandwidth; I told him why I thought things were exploding, because people had graphical user interfaces, they could now point and click rather than writing arcane commands on teletypes. And he got all that. Plus, I think KP had two investments. One in @Home, which was a user of Roberts. And the second one was a company called Concentric Networks. I don't think it's around anymore. So, the guys at KP understood networking from the user perspective. So, I think that's why I got a positive hearing.

Hancock: And that advice that you had-- so you saw this as critical to get the right people.

Sindhu: Yes.

Hancock: And to get the very best people.

Sindhu: Yes.

Hancock: Tell us about the team that you assembled and recruited to work with you.

Sindhu: So, I had a list of the top twelve routing protocol experts on the planet. And I can probably name them right now, but I won't. Out of those, ten of them ended up eventually working for Juniper and the first guy I reached out to was a guy called Dennis Ferguson. And he became my co-founder and the story of Dennis is interesting just because he's a very interesting guy. So, Dennis was at the time working at a company called Epsilon was trying to build something like router using a team [ph?] switches. And Dennis was quite frustrated there although he didn't tell me when we first met. So, I described what I wanted to do and Dennis being Dennis pokes a huge hole in my thinking, because I was trying to build a router using caching routes and Dennis goes, "Well, this doesn't cache very well." So, I said, "Dennis, why doesn't it cache very well?" So, he said, "Because you have lots and lots of packets from many sources, many destinations and you would need a very big cache and it won't work very well. He said, "But! I have this algorithm that I have developed. I just don't know how to make it go fast." And, so, I said, "Dennis, let's talk about it." I said, "At my home." And, so, he described me his tree-based lookup algorithm, which essentially takes IP addresses-- before are 32 bits-- and they need to do what's called a longest prefix match. So, you start from the high-end bit and you start looking and you have to create a data structure such that you pick the longest prefix in the table that matches, not the shortest one. Because you want the most precise match. Right? So, Dennis described me this algorithm and what I said to him is "Hey, why don't we try pipelining this? Because we have plenty of packets coming per second and so we'll do this one bit at a time." So, I sketched out a design for him and Dennis' eyes just lit up. Because prior to this what he said was, "Hey, Pradeep, here's a couple of diskettes with the algorithm description. You can have them. Good-bye." I still have those diskettes at home, by the way.

<overlapping conversation>

Sindhu: Those would be interesting artifacts.

Hancock: Those are very interesting artifacts.

Sindhu: So, when I described this to him Dennis was hooked. He said, “You know I always thought I could make this thing go fast, but I didn’t know how.” And we computed on that day that we could do forty million packets per second, which is hugely larger than anything that could be done up to there. And today it sounds--

Hancock: Right at that time you had written--

Sindhu: Right at that time on the back of an envelope, “Forty million packets per second.” The stunning thing is that M40 he did 39.6 million.

<laughter>

Marc Weber: Do you have the envelope?

Hancock: Not bad: six tenths of a--

Sindhu: It’d be a miracle if I find it, because it was in a piece of paper somewhere. But I remember it was in my living room and we sat and discussed and after that we went for a walk. And Dennis said, “I’m in.”

Marc Weber: Because you had been trying to get him in and he was at first--

Sindhu: Yeah, no, I'd had lunch with him before and this was the second meeting with Dennis. And then the other guy I recruited was a guy called Bjorn Liencres. I’d worked with Bjorn at Sun on the cache subsystem. He had done one part; I had done the other part. And Bjorn was the best hardware guy I knew. Dennis was one of the top guys in routing protocols. He had also built routers. So, these two guys were my co-founders.

Marc Weber: And, so, how did you choose the name?

Sindhu: The name was interesting. Let’s see. So, I had made up a list of names. And I was insistent that the name had the letters IP together-- Internet protocol-- somewhere in the name. And Judy O’Brien, who was the general counsel, with _____ had asked me to contact Wilson Sonsini and particularly Larry [Sonsini], who was very helpful and he tagged Judy O’Brien as being the counsel. And, so, Judy called me-- I think it was maybe the fourth or the fifth. She said, “Do you have a name? Because we’re going to incorporate.” I said, “No, I have a list of names.” She said, “That won’t do.”

<laughter>

Sindhu: “Lists don’t work. You have to have one name.” So, I had names like IP Comm and IP Switch and half a dozen other made-up names and then there was Juniper. And, so, I said, “How do I decide?” I

showed the list to my kids. Of course, none of the other names made sense to them. And, so, they said, "Hey! We like this one." That was it.

Marc Weber: And, so, you incorporated on--

Sindhu: February the 6th.

Hancock: Yeah.

Sindhu: You know, there's a lot of other meanings we put on Juniper after the fact. For example, the logo of the company-- the first logo that I designed was a silicon wafer with an image of a juniper tree embossed on it. And that was actually particularly meaningful because one of the key algorithms that we had invented thanks to Dennis and myself was a route lookup algorithm, which is tree-based, which is a special kind of a tree structure. It was the first time that you could do very, very high speed route lookups. What was significant was that when we came out with the M40, the fastest router was significantly faster than the fastest Ethernet switch, which people thought was impossible.

Marc Weber: Yeah.

Hancock: Mm.

Sindhu: I mean, we proved them wrong.

Hancock: I wanted to ask about the-- little bit more about the financing, because there's such an important-- often-- relationship with your--

Sindhu: Absolutely.

Hancock: You've talked about the lawyer and your VCs, tell us a little bit more about the relationship and the roles--

Sindhu: So, about the financing…the way it worked was I had zero experience with this stuff, so I'd gone to Vinod and I trusted him fully. And, you know, what he said was that "Look, the way we'll do this is I will give you a loan of two hundred thousand dollars in exchange for a promise to fund the company at a certain valuation." And it sounded fair to me. And right after KP funded Juniper to the tune of two million dollars, I think was the first round, there was intense interest by, I'd say, a dozen other VC firms to want to put money in a second round. And I think we were in an envious or enviable position of being able to select the next four VCs in addition to KP. So, we did Series B literally a month and a half after Series A. And that was-- memory's fading now-- probably it was ten million. So, the second round went very fast. First round was painless. You know, my experience with KP and with Vinod and Will Hearst were the two people involved was it was a relationship of trust. And I don't think I ever disappointed them. Things may not have gone as smoothly if I had. I had a fantastic relationship with them, with all the VCs, particularly Andy Rachleff of Benchmark. They were all really, really supportive of the company. All helping to build the company.

Marc Weber: The Sun connection obviously helped get you established at the very beginning.

Sindhu: It helped--

Marc Weber: --establish trust.

Sindhu: --give me credibility. The other person was extremely helpful was Dick Kramlich from NEA. And Geoffrey Yang from IVP.

Marc Weber: Oh, okay.

Sindhu: And Seth Neiman from Crosspoint. Those were the VC companies.

Marc Weber: And how about the board? How did you--

Sindhu: So, the board was-- I think on the board-- there were two board members from Kleiner, one from Benchmark. Geoffrey Yang almost never came to the board meetings and Seth Neiman used to attend some of the board meetings. So, it was naturally that sort of people and Dick was on the board. And I told the VCs pretty early on that, even though I was the founding CEO, that I didn't want to be the CEO long-term, that we should initiate a search right away. And we did actually in April-May, started to talk to candidates. And we talked to, I would say, half a dozen. Maybe eight people. And I think I met with Scott Kriens in August, I think right before he was going on vacation or right after he came back from vacation. And the thing I liked about Scott was that his competency was complementary to mine. He was not a technologist. He was a sales guy. Furthermore, he understood the market to which we were going to sell service providers. So, he had been VP of sales at Stratacom, which had been sold to Cisco some time ago. I think a year ago or so. And Scott had been taking a sabbatical, some time off. So, I really, really liked Scott and so did all the VCs. And, actually, at the time there were, I think, sixteen people at Juniper, maybe seventeen-- all of them interviewed Scott, which is something I insisted on. And they all liked him. And, so, we made the choice and we had a CEO and I took a back seat as chief technology officer and vice chairman.

Marc Weber: And what were the roles of your co-founders?

Sindhu: So, Bjorn-- so, we were not big into titles at all. And Dennis had no aspirations to manage. He was-- he wrote the first implementation of kernel software, which was written on BSD Unix. He also worked on the routing protocols and generally contributed to the hardware design as well. Dennis is one of those guys who's extremely versatile. And we were very lucky to have him at the company, because the company would not have made it very likely without Dennis. And the reasons were that Dennis had two views into a router. One was he had built one before-- an IP router. Secondly, he'd actually been at a network operator using them. And that was priceless. So, we didn't need a VP of marketing to tell us what features to put in, what not to put in. He just knew. And Bjorn's role was, essentially, as the VP of hardware engineering. So, the design of-- the physical design of the machine, the layout of the boards, roughly what the chips were and my role was as essentially as overall architect of the machine.

Marc Weber: And you felt confident bringing in the CEO that you weren't worried about losing control or--

Sindhu: No.

Marc Weber: --having it go another way.

Sindhu: No. Not at all. And, you know, this has been true for every CEO that Juniper has had since Scott, that people have given me freedom to do whatever I do best, generally without interfering.

Hancock: While we're focused on the founding and early growth can you talk to us about the road to going public?

Sindhu: Sure. So, you know, the first machine came out in 1998, August. So, if you count from February when the company was incorporated and we were three people-- we got started seriously probably in March or April. So, it's two years and six months. Or two years and five months. And lots of people said what you're trying to do is impossible. Because it'd never been done before. But I think the reason we were able to do what we did was the team was really experienced and we didn't make any big mistakes in the hardware bits. Because if you make mistakes in the hardware bits, it's very hard to recover. And if you have good people, lower order mistakes can be easily corrected. And, as it is, had we chosen a machine which was two times bigger than what we planned to do, we would've failed. Had we chosen a machine half the size it wouldn't have been as competitive. So, the choice of the size of the machine was pretty spot on, close to the limit of what was doable.

Hancock: Back to that being close to the limit, was there debate forming that size?

Sindhu: Absolutely. There was-- there were some people saying, "Oh, you need to make it ten times bigger." Or the other people are going, "We could even do this!" And, you know, making these kinds of decisions I've always found that debate is healthy. When you're exploring some complicated issue no single person's perspective in arriving at a robust decision. This is my experience. And in this case we actually chose a particularly technology. I think it was-- I don't remember the name of the IBM-- IBM was our _____. That technology turned out to not be sufficient. We had to switch horses to the next node in silicon. As it is we barely squeaked by.

Marc Weber: And how far in was that in time?

Sindhu: I think that decision was made-- I think close to the end of '96. So, about months in. Maybe even early '97.

Marc Weber: Maybe this is the logical time-- one of the questions was to go through development of the M40 and also to take a kind of…. The M40 seems like the most obvious example, but sort of through the lifecycle of the development process from concept to execution. So, maybe just talk sequentially about the M40 and how-- what were the ups and downs and pitfalls.

Sindhu: Sure. What I would say there-- the only ups and downs that were there were team dynamics, because we had people from three different backgrounds. We had people from the computing background. We had software people from the routing protocol and router software background. And then we also had some people from the building machines for service providers, the telecom background. And

these are very different cultures. And there was friction initially, but what I felt is that since we were working on a hard problem if were successful in solving this problem, then the team would actually bind together and indeed that's what happened. But in terms of solving the problems and getting good solutions to them we actually did it. I mean, I'll tell you all the risks that were there. We use a new signaling technology for sending signals in the machine from one place to the other, which was inspired by work that I'd been involved with at PARC. This was called GTL, Gunning transistor logic, which was invented by Bill Gunning, who's now long passed. We used GTL to do high speed signaling, never been done before on a machine of this scale. And we're putting the forwarding path of an entire router in silicon-- never been done before. We were separating the control plane and management from the forwarding plane-- never been done before. We chose as the general purpose computer engine X86, which no one had ever done before. People were using Motorola PowerPCs and other machines, but not X86. I had felt very strongly that pick the best general-purpose engine you can find, and the X86 was the best general purpose engine. One of the things that we did in development that turned out in hindsight to be absolutely critical was we had split the problem into two parts - the software and the hardware. We could actually independently develop software from the hardware. So, in fact, the first prototype of the routing protocol software was in the Internet functioning-- I think-- by the middle of 1997--

Marc Weber: But without har--

Sindhu: --on a PC!

Marc Weber: Right. Without hardware to test it out.

Sindhu: Without hardware to test it out. So, we had a PC with an Ethernet interface, then connected-- and we distributed this machine to various places on the Internet. And that's how we debugged the software: completely in parallel.

Marc Weber: 'Cause you had basically a virtual hardware that you were using with the software.

Sindhu: People love to talk about virtualization these days, but that's how we did it back then.

Hancock: Where was this physically? Help us put this in place and space. Where were you actually working?

Sindhu: Ah! In the basement of various friends' houses--

Marc Weber: Ah!

Sindhu: People who were working for Internet service providers and had connectivity to them.

Marc Weber: And, so, this was a secret test, I presume.

Sindhu: It was a secret test. Completely.

Marc Weber: And, I mean, some of the things you were doing-- obviously, people began to realize some of this-- were you not worried about Cisco or others simply copying--

Sindhu: We were just too busy to be worried about other people. We were in stealth mode. We didn't talk a lot about if you go back the Way Back Time Machine on the Internet you'll find a website which says, "We build Internet infrastructure." That's about it. I mean, that doesn't say very much at all.

Marc Weber: And where were your offices at this point or the lab?

Sindhu: So, the first real office was at 101 University Avenue in Palo Alto. We stayed there for a short time. Then we moved to Mountain View. Oh, no. I'm sorry. Not Mountain View. Yeah, we moved to Scott Boulevard in Sunnyvale. Then we moved to Mountain View on Rivendell [ph?] and then we moved to close to where we are today on Mathilda Avenue. So, that was kinda the sequence.

Marc Weber: But the M40 development was mostly--

Sindhu: It was mostly done at Scott Boulevard.

Marc Weber: Okay.

Sindhu: I don't remember whether that's in Santa Clara or where. It's kind of close to where Intel is. Intel's on the other side of 101. We were on this side.

Marc Weber: And when was the first hardware prototype though?

Sindhu: The hardware prototypes, the boards-- so, the silicon-- all this stuff was going on in parallel. The board development was in parallel with the silicon, was parallel with the software. So, the board started to come back quite early and the chips were on purpose staggered and we developed a-- made a development plan where we could bring them up even without having all of them together. So, there was no big bang.

Marc Weber: Oh, so you would have software fill in for the ones that weren't ready.

Sindhu: Not really software, but we could test it independently.

Hancock: Interesting.

Sindhu: I think the first chip must have come back in March of '98. The last chip came back I think May, maybe, of '98. And then in four months we shipped the machines.

Hancock: So, that coming together happened very quickly.

Sindhu: Very fast.

Marc Weber: And how many custom chips?

Sindhu: There were four custom chips. We were quite silly with names. So, the names were A,B,C and D.

<laughter>

Sindhu: Not very inventive.

Hancock: How did you choose your first customers?

Sindhu: Oh, that was easy. It was the customers that had the biggest problem with the core of the network, which is what-- you know, the core was collapsing. And by the way, '97-- I think it was-- maybe it was Bob Metcalfe who mentioned that the Internet is gonna collapse and he was actually not wrong. Because if-- there's one particular thing to say about the way machines were designed earlier, which is quite relevant and it's one of the main problems we fixed. So, routers used to be built using single CPUs. And this CPU was responsible for doing three things: management; running the control plane, which is computing the routes; and, thirdly, passing packets back and forth, based on that-- on those routes. Occasionally, what would happen is that links would go down and when a link went down the router on either side of the link would say, "Oh! I can't see that guy anymore. Let me re-compute." So, the moment a link went down, these things would start re-computing and sometimes links would flap. Meaning, that it went down but it didn't stay down, came back up. Well, so, when a link went down the routers on either side became very busy. Because routing protocol computation was a very computer-intensive process. When these routers became really busy they couldn't pass packets. So, their neighbors thought that the link was down and, in fact, the link was still up. And now you can see that this is an exponentially expanding process. And you have total network collapse. And this was happening more and more often. Now the fact that we separated the forwarding of packets from the control plane completely got rid of this failure mode. And, so, our machine was way, way more reliable.

Marc Weber: And the more machines of your design were out there the more it would increase reliability of the network. Because packets, as long as they were not going to the particular router that was down, they would move normally.

Sindhu: Yes.

Marc Weber: It was only the ones that were destined for that.

Sindhu: Correct. So, that was a very, very significant architectural change. In fact, today everybody does it that way.

Marc Weber: And what were the four chips that you designed yourselves?

Sindhu: So, one of the-- let me start with one of the last chip-- the D-chip was a SONET interface chip. SONET was the particular optical protocol that was used in that day. It's now sort of more or less dead. But it was a way of encapsulating bits on--

Marc Weber: Fiber optic.

Sindhu: Fiber optics. So, one of the chips implemented in the SONET protocol. The second chip was-- which went right next to the D-chip was a B-chip, which was responsible for memory management and buffering. So, when packets come in, you know, the contents of a packet was sprayed out to memory and then it was re-collected on its way out. So, this was a "one write, one read" machine. So, kind of a minimum energy consumption machine. So, packets came in and went straight to memory in parallel with the deroute lookup and when-- once you determined where the packet was gonna go out, you read the packet from memory and sent it out. So, it was kind of a textbook example of how you might build a router, except that it was really, really fast. And then there were two other chips, one was an A-chip, which was a switching chip. All it did was when packets came in on an interface it would actually divide the packet into cells and spray it amongst the eight memories on the eight-line cards on the input side. And on the output side, there was another switch which collected them back together. And between these two switch chips was the route lookup chip. So, the route lookup chip didn't get the entire packet. It just got the packet header. And, so, it did the lookup based on the packet header. So, those are the four chips.

Marc Weber: And you weren't-- by putting things in hardware were you worried about updates or-- I mean, once these are there, they're there.

Sindhu: Yeah, once they're there, they're there and that was a concern. And, so, we actually made the hardware quite flexible. So, especially the C-chip. The A-chip was relatively straightforward in that it was an internal switch and, so, it's functionality did not need to evolve. And D-chip didn't need to evolve either because the SONET protocol was fixed. But the B- and the C-chips were made quite flexible purposely.

Marc Weber: Flexible themselves or just ease to replace?

Sindhu: No, no. Flexible in their-- they were literally-- they were configurable and the-- each had their kinda instructions. It was not quite like programming them, but it was not quite micro-instructions, but it was flexible pipelines. Let's put it that way.

Hancock: So, this story is really remarkable. You have the essential design, the technology idea, your company, and then you execute it so closely to what sounds like your original conception. Is that true or was it--?

Sindhu: It was--

Hancock: Or is there more to the story? Was there more zig-zagging and experimentation?

Sindhu: No, there was no zig-zagging. I mean, I have to say we were very fortunate in that we never changed our business plan despite pressure from some of the consultants that KP had hired. I remember one of the consultants-- shall go nameless-- suggested that we put a whole bunch of other protocols other than IP, like IPX, SNA, DECnet-- you know, so on and so forth. And we said no. Too complex. Sorry. We're gonna focus only on the right area because these other protocols, they're all dead now. <laughs> They don't exist.

Marc Weber: Cisco had done multi-protocol routers but--

Sindhu: That's why--

Marc Weber: --in an earlier era.

Sindhu: That's why this guy, he said, "But look at Cisco." I said, "No, thank you. We don't want to look at Cisco."

Marc Weber: Did you have an earlier articulation of that business model?

Sindhu: Yeah, yeah, yeah.

Marc Weber: Do you still have that?

Sindhu: Of course.

Marc Weber: Yeah, that would be interesting--

Hancock: That's very interesting-- to see that as well.

Sindhu: Haven't looked at it in eighteen years, but it's there somewhere.

Hancock: So we were--

Sindhu: Where were we?

Hancock: We were-- well, we were just finishing the forty-piece and then we wanted to, as I said, kinda move forward into--

Marc Weber: Towards the IPO.

Hancock: And beyond to the crash and post-crash and re-invention, sort of "What's next?" to bring us forward.

Sindhu: Let's maybe touch on the machine was finished--

<overlapping conversation>

Hancock: --this piece.

Sindhu: No, not at all. The machine was finished in August of '98 and I think the first company we sold it to was Cable and Wireless. And I think the check-- Juniper still has that check of six million dollars. The very first quarter. And the next one was to UUNET, next set of machines. And UUNET is particularly interesting because the CTO of UUNET was Mike O'Dell and I met with Mike in March of '96 itself. So, he had called the Juniper offices because he knew Dennis very well. And, so, Dennis comes to me and goes-- typical Dennis, he says, "Uh.. Pradeep, you should probably answer the email that's sitting in your inbox. It's from Mike O'Dell." So, I looked in my email and, you know, I replied to him and said, "Sure,

Mike, we'll come and visit you." And, so, all three of us, the entire company, we went Back East to visit Mike and when I described to Mike the machine his response was very simple. He says, "Pradeep, just build it. That's all. All you have to do is make it work." So, you know, that kind of feedback was absolutely priceless, because it said that what we were working on was important. And, not only important, it was fundamental to the Internet.

Marc Weber: And, so, I mean, there was a huge pent-up demand when it was--

Sindhu: There was a huge demand and there was also this-- I mean, the machine literally sold itself, because when people installed it in the network it just made things much more reliable and much faster. And it made the life of the operators much simpler. So--

Marc Weber: In what way?

Sindhu: Because things weren't crashing all the time.

Marc Weber: Okay.

Sindhu: So, these guys made money when they provide connectivity. And they didn't make money if the network kept crashing. So, in the first year, you know, I may have misstated that the first Cable and Wireless check was for six million. It was six million in the first year, starting in August through December. The next year the revenue was a hundred million.

Marc Weber: Mm.

Hancock: Mm.

Sindhu: The year after that was four hundred million. And I don't remember when we cracked a billion, but in between there was the crash. But it was extremely rapid uptake and right on the heels of the M40. I think right after we finished the machine I was thinking of the next thing to do and rather than re-design the new machine what we figured out was a way to build a machine four times faster in about a year and a half. And we did that by essentially making a multi-processor version of M40. So, we took four slices of M40 and essentially sprayed packets over four M40s and put them back together again. And that machine was completed very, very fast and it was the very first 10-gigabit per second interface-based router. And I remember the interfaces-- the optical interfaces themselves were about $200,000 each, just the optical interfaces, and...

Weber: And what were the price points for the M40 and its successor roughly?

Sindhu: A fully loaded M40 must've been in the high hundreds of thousands of dollars.

Weber: And the basic?

Sindhu: So the minimum configuration was one line card, but the maximum configuration was eight line cards, and I think a fully loaded machine is what I was talking about was probably $300,000, $400,000.

Weber: And the successor?

Sindhu: So the history of this thing is that as the capacity went up the price roughly stayed constant, so we've driven down the cost per gigabit per second exponentially, and it continues to this day.

Weber: And the successor was-- I mean, the chips were all the same. It was...

Sindhu: The chips for the M160-- there was one new chip, but the software was the same. The only new chip was a chip that sprayed traffic and the chip that collected it at the other end, so that one was actually very satisfying to do. After the M160 we actually changed the design, because this particular way of building it had kind of reach its limit, because in these machines it was a cell-based fabric internally, but it was a middle-buffered machine, so it was switch and then put switch in the output, memory in the middle, so that allowed you to do a single-write, single-read, and all of the machines following this one were memory in the input, memory in the output, switch in the middle, because the previous one was not scalable beyond what we had done.

Weber: When the M40 came out you had preorders when people knew this was coming up?

Sindhu: We did. We did. We did. We did.

Weber: So you had a big...

Sindhu: We had a big backlog.

Weber: Did you show it in any big show?

Sindhu: No, because one other thing on the go-to-market side-- the number of customers that Juniper had was small. This was very fortunate, because otherwise the go-to-market machinery would've needed to be much more sophisticated. As it was we had a handful of ISPs, Internet service providers, that were growing very fast, and eventually these ISPs were bought by the large telcos. The telcos initially were very unfriendly to packet switching. They wanted to continue to do ATM until they realized that it wouldn't actually work, and then they wholesale adopted IP.

Hancock: On the technology roadmap, it's not easy or automatic to have successive innovations in generations, and yet you've talked about this sort of march forward that's not only matched but extended and multiplied the impact of Moore's law and other things. Can you talk about the evolution and pull back the lens in a broader scope after the M40, the 160 and further?

Sindhu: Absolutely. I think one thing we have done very successfully at Juniper is if you look at the machine that we did back in '98 it was shipped, it was 20 gigabits per second full duplex. The machines that we're doing now are on the order of 40 terabits per second full duplex, so that's a factor of 2,000. We take the 18th root of 2,000, and I think you'll end up-- I can't do that mental calculation, but it's probably 1.5X every year. That's faster than "Moore's law," because Moore's law is about 1.4, 1.41 per year, because it's a doubling every 18 months.

I do want to say something about Moore's law, which is it's misunderstood quite a lot. I think what Gordon Moore said was that the number of transistors on a chip tends to double every 18 months to two years. I think what people have then done is gone from that and said "Well, performance doubles every 18 months," and that ain't true at all, because the maximum potential performance might double, but it takes architectural cleverness to actually make it double. Just because you plonk down double the number of transistors every 18 months doesn't at all guarantee that the performance will be twice, so that's one point about Moore's law.

A second thing to remember is that there's no such law. It's an observation, and I think if you look today at the density of chips the interval between lithographic improvements is actually getting extended in time, and people are having fabrication difficulties building 10 and seven nanometer technologies, and so we are coming to a pretty difficult stage, and I think that architectural cleverness will become more and more important. In other words, extracting more performance from the same transistors is going to become more important. Life was actually very good between the 1970s and I'd say 2004, 2005, because the clock speed at which we were building chips actually was increasing very nicely.

But building chips that run significantly faster than three gigahertz is actually really, really hard. What's worse is that the power consumption also goes up dramatically as you increase power, non-linearly in fact. So I predict that we're going to see a flattening of the performance curve. It's already happened for general-purpose computing. It happened around 2005, 2006. In other words, the performance per core has almost completely flattened. The only thing that is actually increasing is the number of cores, but the benefit you get by putting more cores-- you don't get the full benefit, because you're limited by memory bandwidth. So whichever place you turn you've got a flattening happening. I think that there'll be a similar flattening happening for networking also. It hasn't quite happened yet. So networking seems to be maybe four years, five years, maybe even more behind computing in the sense that we haven't yet seen that flattening, but it's coming, I can tell you.

Weber: Back when your first products came out suddenly you were in the sights of a number of competitors. Cisco is the best known. Who were those competitors? What was your thinking about that? Speed was your big advantage obviously, but...

Sindhu: Actually it was not just speed. It was primarily extremely reliable functionality, and this was especially important in the core, because all the world's traffic went through the cores of these ISPs. If the core is constantly crashing, life is very bad, because you certainly cannot increase capacity. So it was the fact that these machines were much more reliable, that they performed a lot better for roughly the same amount of money. I think the other innovation that we did was the interface that we provided to the operator was much more programmable than what the other vendors did at the time. But I would say that the main competitor was Cisco. There was not a second competitor. It was primarily Cisco, and Cisco was of course in the beginning infinitely larger than Juniper, because we had zero revenue, and Cisco has remained a competitor all along.

Weber: In those early years suddenly you've gone from research and development in a very, very hot competition with a huge company, so what was that like?

Sindhu: Well, I never thought of Juniper as a huge company.

Weber: No, but you were competing with Cisco.

Sindhu: We were competing with a huge company many, many, many times our size.

Weber: You're David and Goliath suddenly.

Sindhu: It was the David and Goliath story, and predictably lots and lots of people wrote about that, and we got a lot of good press as a result of that. I think people wanted us to succeed, because I think in the networking industry, actually in any industry, if you have a monopoly, innovation stops because there's no incentive really. Once you have 100 percent market share, innovation stops. And I think it was actually very good for the networking industry to have competition, and if you talk to people from Cisco they will say the same thing. They will say that the emergence of Juniper as a competitor actually made Cisco better, and we actually showed a better way to build a fundamental part of the network, which is routers.

People talk about switches and routers and this and that and the other, and essentially I would say that--and people also talk about optical gear and optical switching and so on and so forth, and it turns out that in the fullness of time you will only have one kind of device that makes switching decisions, and that device will be much closer to a router than anything else. And here I'm specifically referring to a router as being a device which has two characteristics. It switches packets based on the IP header, maybe additional information, number one, so it's switching packets based on IP. Number two, it is doing dynamic topology discovery of the networks that surround it, and so it's a very automated way to perform the routing function, because prior to the existence of routers, the machinery was not able to respond to changes in the network because of failures, because I installed new machines and so on, so that had to be configured, and so routers were a very, very, very important, I mean, fundamental to the existence of the Internet, and not many people realize this, because they don't come in contact with routers, but routers are fundamental to networking, not just the Internet, to networking period. And at the end of the day you'll end up with networks that look like graphs, are graphs, where the nodes are routers and the links are optical. That's it. Nothing else will be there.

Weber: Do you want to think about leading up to the IPO?

Hancock: I think we talked a little bit about that.

Sindhu: So I can say a little bit about the IPO.

Hancock: Let's talk about the IPO.

Sindhu: Yeah, so I think Juniper did an IPO in 2000.

Weber: '99.

Sindhu: '99. Sorry. I'm sorry. '99.

Hancock: It was perfect timing.

Weber: Why then?

Hancock: Why then?

Sindhu: Well, because we were on a good revenue ramp, and we saw that the company could grow rapidly, and we also needed to raise more money. So we talked about the venture rounds, but there was one other round right after Scott [Kriens] came in. One of the things that Scott absolutely wanted to make sure was that the company never ran out of money, and so what he did, which was absolutely fantastic, was he got many of our customers lined up as investors, and so we did a $36 million round on the C round where I think UUNET and Level 3, 3Com, Ericsson, Lucent-- I don't remember the sixth one-- Qwest invested, and that was a master stroke, because these companies were many, many, many times Juniper's size. We didn't have a product, but these companies also recognized the importance of the technology we were working on, and they wanted to make sure that we succeeded, so we were well-capitalized, and then after that it was time to go IPO to grow more, and that's what we did.

Hancock: So you had this meteoric rise, and then the crash came and then a reevaluation. Walk us through that.

Sindhu: Yeah, so I've had a lot of time to think about the crash and why it happened and so on, so forth, right? So I think what happened was that the Internet has seen exponential growth in usage and in the number of applications and so on. I think people don't deal very well with projecting exponentials, because most of our projections are linear, so when the Internet was growing very fast what people thought was "Oh, everything's going to move online instantly," and so the expectations that were placed on the Internet could never have been realized in hindsight, and when they weren't realized, well, there was disappointment, and people said "Oh, this is just a fad. It's a passing fad." They were wrong about that too.

And so you saw the rise, you saw the disappointment, you saw the fall, but the thing that I always consoled myself was that we are building technology that is fundamental for network infrastructure, and Juniper's tagline really is "Connect everything, empower everyone," and I want to take a moment to say what that really means. I have this deep belief that networks are really important to all kinds of systems, and you see this phenomenon of connect everything, empower everyone occurring in nature also. And so roughly the idea behind this is that when I have a system that consists of things and connections between things-- these things might be people, might be neurons, it might be fundamental particles, it might be computers, it might be something else-- you have a network, and it turns out that the behavior of the system that is the network is determined far more by the topology of the network when you are operating at scale than by the power of the individual elements.

Now, this observation, I don't think anybody has ever made it before, but to me it's fascinating. It's fascinating because it's surprising, and you see the same thing-- for example, take a look at an ant colony or a bee colony. Individual insects can't do very much, but collectively they do farming, they thrive, they move around. Look at the human brain. Individual neurons are really, really very weak when it comes to

their computing power. They fire at a maximum rate of about 200 times a second, but there's a lot of them, and I would suggest that most of the magic is in the connectivity between the neurons, and you can also see why most of the magic is in the connectivity. If you look at a graph the number of connections is exponential. The number of nodes doesn't grow very fast, so you can see why the behavior of something might be much richer deriving from the particular topology that you've used or the particular interconnection that you use in the communication. So my point is that as networks get bigger and bigger the way in which you connect things becomes more and more important to the behavior of the system, and in fact the behavior of the system is determined far more by the network than it is by the individual elements. And the other thing which is utterly fascinating to me is you see as in Moore's law individual nodes in a network that we build, either computers or routers or whatever, they're going to reach a limit. This is sure because of technology. It turns out that when these nodes reach a limit I can always build a higher-level system by connecting these things together to make a more powerful thing, and it really looks like there's no real limit to how many times you can pull off this trick, so there's a recursive nature to this. To give you an example...

Weber: But you're saying add connections to clusters of things that are already connected.

Hancock: And there's this multiplying effect there.

Sindhu: Yes. If I take a computer and I look inside it its components are today multiple processors or multiple chips inside which there are actually multiple cores. You have the memory system, you have IO devices. These are all connected together in a nice way. That ends up with the server. If you look at now how do I use multiples of these together, well, I take a network, I connect these servers, and I make a data center. That's the next level. A single data center ain't enough, so now what I do is I take multiple data centers, I connect them together, so all I'm saying is that the network gives you an escape hatch.

And what's absolutely fascinating is that nature seems to have discovered this trick too, this recursion of applying the network over and over again at different levels, so that's the point. So the point being that connect everything, empower everyone just is saying that when you build a network that has the power of being able to connect anything to anything else the beneficiaries are the things that are connected. That's the empower part at various levels.

Weber: And you're empowering the next level by making-- yeah.

Sindhu: For example.

Weber: When the crash came did you have a confidence that there's some underlying process that's going to continue growing completely independent of...

Sindhu: Well, obviously I had no certainty, but I knew that the technology we were working on was absolutely critical to the building of the Internet, A, and, B, that the Internet was going to be one of the most important creations of man, and as it is turned out it is the most fantastic platform for innovation that humans have created. It's empowered so many things, and I think we are at the beginning.

Weber: You want to...

Hancock: Move to our last sections or...?

Weber: For the linear narrative, after the crash how did Juniper survive?

Sindhu: Well, so the stock price of Juniper went from I think $241 or whatever it was at its height down to somewhere between $4 and $5. It was obviously very, very disheartening to the employees and so on, and it took a lot of work to keep things together and to tell people that it's not the end of the world, that we are building things of importance, and eventually the importance will be recognized and the company will come back. And we did.

Weber: Did you have to lay off many people?

Sindhu: I think it was sometime after the crash that we bought a company called Unisphere, and this is for edge routing. What we built initially were core routers, and Unisphere was building edge routers to connect consumers, mostly with broadband residential access. And I think when we bought Unisphere we did have to let some employees go. I don't remember the exact numbers, but we did have to let some people go, and that was extremely painful, but we recovered from there.

Hancock: I'd like to take a step back and think about Juniper in the Valley. Juniper is the heart of the story of being a Valley company. Could it have happened anywhere else, or do feel like it was essential that it was here? What role do you think being placed in the Valley has?

Sindhu: I think it would be very difficult to have something like a Juniper be born somewhere else at least equally able to contribute, and I think it's precisely because of the network effect between people. So I cannot think of another place where if from here you drew a circle of maybe 10 miles you can find experts in almost any field that you want in high technology, and I'm not just talking about any expert, the world expert. There's that. There are great universities in the area, so young talent. There is the willingness of people to risk money. It's critical, and personally for me it was really important that the idea of a startup wasn't just an abstraction. I had an example in Andy Bechtolsheim trying…somebody I knew that had been successful, and I think that's extremely empowering when you see examples of people that you know that "Oh yeah, this is actually possible." So take Boston as an example, right, on the east coast. There have been interesting high-tech companies there, but it's nowhere near Silicon Valley, and I think it's because this place is unique in the talent that it has, and we should cherish it.

Hancock: Cherish it and sustain it.

Sindhu: And sustain it, and not do anything to screw it up.

Hancock: Are there vulnerabilities that you see, risks that you're concerned about for the Valley at large?

Sindhu: I will tell you one thing, that doing a startup today is way more onerous than it was when I started Juniper in terms of the paperwork that the government wants, the legal stuff. I mean, for example, you just look at the hiring paperwork. It's at least five to six times more. So I think maybe the government is trying to help us with creating more laws. I think they're hurting us and then nothing else. You do not legislate honesty. I actually have very strong opinions about that, that creating more laws actually screws

things up, because every new law that you create you're putting one more constraint on the system, and eventually you get a system that is too stiff, and it's going to break. And I think the guys over in Washington have never built a company, they don't understand this, and if somebody undoes this beautiful machine it'll be them.

Hancock: That's a powerful statement about the importance for the institutions and policies to support this flexible and dynamic change.

Sindhu: Exactly.

Hancock: What impact, if any, has Juniper had on the Valley?

Sindhu: I think that that is for other people to judge, not for me to say. I just like to think of Juniper as a company that is a high-technology company doing its very best to advance the state of the art in networking and security, and I hope we've done a good job.

Weber: Shall we move to the end?

Hancock: Please.

Weber: What suggestions would you offer to young people who are maybe starting a career thinking of high-tech?

Sindhu: Several things. Pay attention to the fundamentals. Learn how to read and write. Learn mathematics. Learn about the laws of nature. I have no idea what jobs will be around. I don't even know if 50 years from now we will have jobs for people. I worry about that, right? I don't think anyone knows. Living in a world where there are very few jobs would be very scary. I think the way that young people can arm themselves-- the best way to arm themselves and the best weapon that they have is a flexible and open mind, and if young people would spend more time in developing their mind rather than being interested in things that fade away much faster we would all be much better off as a society.

This advice is easy to give, very hard to follow. I mean, I consider myself really fortunate that I grew up in India. I didn't grow up here. Why do I say this? Because there are so many distractions, and every day there are 17 more distractions that are coming. Our devices are constantly going beep-beep-beep or tweets and so many forms of advertising that are trying to intrude on our time. I believe that nothing of value can be done without focus and concentration. Nothing. And I challenge anyone who says that "Oh, my kid does multitasking very well" and because he or she plays video games well that they'll be successful. Bullshit. If you want to create something it requires concentration. If you want to create something exceptional it requires exceptional concentration, and people who don't have that ability won't make anything significant.

And I think the one thing that machines can't do yet-- I'm not going to say that they never can-- is to create things that never existed before. They might play games very well, they might do more and more mundane tasks very well, but creating things that did not exist seems to be for the moment uniquely human, and I hope it stays that way. Let me just put it that way, because otherwise if there are no jobs for

people to do life is not good. That's the advice I would give to young people. Develop the capability to learn, develop the capability to focus, understand fundamentals and be less interested in the fad of the day and be less interested in what other people think about you. Don't worry about it. It doesn't matter.

Sindhu: So what we're looking at is the original notebooks from late November 1995 when I was sketching out the design for the M40, and the pages I'm looking at actually were the first thoughts on the memory system for a router. And as I read some of these things the design started out by saying "Hey, what if I made the simplest possible thing, which is a single FIFO logically, and then how would the memory system look like to make this single FIFO and then implement the single FIFO as a circular buffer, which was being written by many people and read by many people?" And that's the exploration that I do in these pages, and as I read some of the notes it says that, well, why not follow this design, because it's logically very, very simple, and so here what I'm doing is I'm exploring how to actually implement it. And the original scheme that I had actually kept reference counts in the memory buffers, which are fixed-size objects, but the reference counting scheme was too complicated. So I came up with even a simpler scheme where there was no reference counting. But essentially the maximum size of the buffer was 80 milliseconds, and you just made sure that the guys who were writing them were over the guys who were reading, and that design was simple and it was beautiful.

And on this page what I'm describing is how that design might work, and as it turns out there were 16 input ports here, each one 2.488 gigabits per second, and there were 16 banks, because what I wanted to do was I wanted to distribute the memory amongst the line cards that contained the 16 ports. As it turns out, we eventually build an eight by OC48 or 2.488 gigabits per second router, because it wasn't technologically possible to build a 16-port one at the time.

Now of course things are very different. And then this just describes how the packets as they came in would be broken up into fixed-size cells of 64 bytes and then how would the memory be managed. And, of course, one of the problems that comes immediately is if packets are going to be broken up into cells, how do you stitch those cells to each other, so this page actually describes the machinery to do that.

The next page actually looks at what would be the dynamics of a system like this. In other words, if you got lots and lots of traffic coming from every port, all going to the same output port, well, now this buffer is going to start filling very fast, and what happens then? Now looking at the dynamics of that in terms of the fill rate and the drain rate, so there's some calculations based on that.

And so, this notebook kind of describes my style of working, so these thoughts were well before Juniper was formed. I think at this time I was looking for money. I had an idea of how to build a machine, and, as I said earlier, there was one big thing that I missed, which my founding partner, Dennis Ferguson, filled-in, which was how does one do route lookup efficiently.

END OF INTERVIEW

