



Interview With A Professional

Guest

Steve Turner, Ph.D.

Senior Director, Cumulus Networks.

Interviewer

Partha S. Ghosh

Course - ENGR-200W

Class Time — 29th Oct — 17th Dec, 2019, Tuesdays, 6.00 pm

December 1, 2019

Introduction

My guest for this interview is Steve Turner, Ph.D. Steve is a Senior Director of Engineering at Cumulus Networks, Mountain View, California. Prior to Cumulus, Steve was a senior director at Juniper Networks. Steve started his career as an individual contributor at SGI (Silicon Graphics, Inc). Steve's knowledge in computer systems, both hardware and software is profound. He is one of those unique leader who has the ability to mentor engineers. I was lucky to have Steve as my manager for a brief period of time at Cumulus networks. His style of communication is lucid and is greatly understated unless someone talks to him. He has the unique ability to explain anything too complicated in a few sentences. His leadership style is unique; he empowers the engineers to take responsibility and deliver. Steve has been part of some very successful company in the last 30 years. So I did not have to think too much before I decided to approach him for this interview.

A few weeks back I requested Steve for this interview, and he gladly accepted my invitation. We met at the Thai Basil restaurant in Sunnyvale downtown, on a warm afternoon over lunch. I asked Steve about his graduate education, his daily jobs, his transition into leadership positions, importance of communication. Steve was glad to share what he wanted in a graduate engineer while he is hiring one. Steve also shared his opinion on ethics and how complicated the subject of ethics could be. I took this opportunity to ask Steve a few question on the future of computing; some open-ended philosophical questions on the future of hardware and software.

Masters and Ph.D.

Partha : *Lets talk about your Masters and Ph.D. from University of Illinois Urbana-Champaign. Why did you decide to pursue a Masters and eventually a Ph.D. ?*

Steve : There were a couple reasons. At the time I graduated with a bachelor's degree, and it seemed difficult to find a job. I did apply for a few straight out of college jobs but the job market was a bit soft. I was literally like (not happy) when I got accepted working for AC Delco, who makes car parts. I talked to myself, what kind of job would that be, ... designing intermittent windshield wiper controllers. ... Having a bachelor's degree in electrical engineering and computers ... on hearing controllers ... but I would be doing be something more interesting, if I could get at Advanced degree and have a more sort of Juicy kind of rolled straight out of school.

Partha : *How do you think having a Master and Ph.D. helped you in your career ?*

Steve : It definitely open doors, it has definitely gotten me positions that I would not have gotten so easily, otherwise. People look at that and say all these are academic credentials. It has beyond the abilities that gave me the training (which I can talk about that too), but it's definitely one where it's allowed me to have a credibility to move into different kinds of roles that I think would (otherwise) have been more difficult, if I'd only had a bachelor's degree.

Partha : *Apart from your research, What were the important skills and learnings that you acquired during your Masters and Ph.D. ?*

Steve : The other aspect is that it (Masters and PhD) taught me a few key skills, e.g how to research things effectively, how to be self-motivated and self-driven, how to say no to your advisor (one of the key things when you're doing a PhD thesis, one of the key skills you learn or else you never finish, is to tell your advisor it's good enough, you're done, because most of us hanging on as long as they can and it's kind of up to the student that sort of one of the final test is, are you ready to say to your advisor "no I am finished now, this is enough". That takes a lot of like self-confidence and drive, that's a hard thing to do but the fact that I did and then it worked and then when I got like gave me the confidence to realize I can do that again). So you know both of you (advisor and Ph.D student), know the ability to do basic research and analytics kind of point of view that I got more in graduate school and undergraduate and also the recognition that I control my own fate and I have to be able to be in charge of my own thing,

over I think things I got out of grad school.

Partha : *When you finished your Masters and eventually Ph.D. what were your feelings ?*

Steve : Oh yes ! I was relieved yeah, it's the same feeling when you get a Bachelor's degree or like any of the other major credentials in your life. So this is recognition, it's like all those dreams, oh yes usually the other things ... I can go into play around now for forever ... ; no one could come back and say we made a mistake. This is recognition and also I did not have the dreams of studying for another test (a huge smile lit Steve's face).

Partha : *Are there professional organizations that you recommend joining, while being in university and what are their advantages ?*

Steve : I joined IEEE (Institute of Electrical and Electronics Engineers) about my senior year of undergraduate and graduate School and for several years afterwards. I was also a member of ACM (Association of Computing Machinery) I was early in my career as well this was a long time ago, it wasn't what it is now. It had a lot of scholarly articles, a lot of interesting research was published in those journals and associate professional organization. So having access to the journal publications was a big win. After I graduated, having some access to alumni organizations and events hosted by IEEE and in the Bay Area; that was helpful to get me introduced into different career opportunities. That's where I think there is some value in (joining these professional organization). I do think there is a need for a lot of the inherent value for the availability of these base information, now a lot of things are published just as academic white papers on University website and Google will find them for you rather than having to go to buy IEEE Explorer and look up Journal proceedings.

Career Path

Partha : *What was your first job ?*

Steve : My first job was working for the MIPS division of Silicon Graphics. My job was to write diagnostics code in verilog for RTL design. The part that I did, had to count cache misses, branch prediction, memory access exceptions etc

Partha : *Did you have a really smooth transition with respect to the things that you learned in University was getting directly applied in your job? Was*

it interesting?

Steve : Ah Well !! Um (A few moments of thought) It was interesting and it was at the time you know the processor design was an interesting thing, but that was not what I studied in school. In school, I was focused on compilers . . . compilers were somewhat related but my focus was mapping numerical problems on to highly parallel machine architecture like vector or MIMD machine or map the parallelism of applications into the parallelism in the hardware. That was what my thesis was effectively on. So it was kind of different to go into RTL design which was pretty low-level, but it was backed up by the digital design classes I had done in EE (Electrical Engineering), but I never really worked on verilog as much as I did in that job and that was what I ended up doing that at the first job. In the second job I did similar things. I worked for a microprocessor company writing these little verilog stubs. Before I left MIPS, the second year of working in MIPS, I was working for, what was called the architectural team, we were writing the simulators, writing these little stubs on which the simulators ran and simulating the next generation design, writing a pipeline simulator; it would be used to do the architecture studies for the next generation of the CPU. I was working on that when Intel's, what's called i-864, was code-named Merced, came at the time and it disrupted the microprocessor industry. MIPS along with DEC SUN and a lot of others had to cancel their next-generation processor plans, So a lot of the people that I was working with go laid off. So at that time what they did (SGI – Silicon Graphics, Inc) was really weird. They did a two prong thing they decided to sack about half the people there, which by the way lead by Bob Mansfield(who now leads Apple's Hardware design team . . .) and focus on incremental change the existing design instead of a next-gen design. They asked me to sign a contract with a 50% raise and I would have to commit for 2 years, kind of honor thing. I thought that they were kind of wanting to bribe me by offering a 50% raise and sacking 50% of my friends, that was not OK. Any way I did not need the money I was only making more money than I knew what to do with it, so I quit immediately. It kind of didn't work for them (SGI) to say something like I'm going to bribe you to stay. It actually had an opposite effect on me. Be careful with these kinds of situations.

Partha : *You have had made a few jobs changes in your career and at different stage. What were the key points that you thought while making those career changes ?*

Steve : Oh Boy, Well, I will say, I don't recommend it but it's truth if you want the truth, I've only made a few job changes with the idea of, "this is the career path I want to be on in this is like let me proactively go seek this

next opportunity”. Generally I made job changes when I have become unhappy in my current one and felt I need to make a change because I’m not enjoying what I’m doing currently. So rather than be so clever, for my own career advancement I need to change because that would be better for me. I have not been so smart in that way. I have generally done is if I’m not happy then I’m going to seek something new. I don’t like this and then I generally look for what will like. When I was a young undergraduate I had a lot of jobs that help fund my undergraduate education and I learned that I didn’t care nearly as much about what the job was, waiting tables, flopping computer tapes in a computer room, sweeping floors job ... was then especially for that kind of job that are not very mentally taxing job ...; it mattered whom I worked with ... then very early on like when I was still young 20 felt like, it mattered a lot to me with whom I worked with. So when I interviewed at places, I tried to get to know the culture, the people and deciding, is this a culture I’m going to feel good ? Am I going to enjoy being here ? Are these the people I want to spend my day with, because the job can be exciting as heck. I made a mistake I went to a company Transmeta, Phenomenal talent, unbelievably with interesting project. It was not fun because there are huge egos and huge hostile arguments, with pounding on the table and screaming ... so that was no fun at all. At the end of the day it has to be fun with the people I spend time with.

Partha : *What are the different challenges that you have observed as an individual contributor and being in Leadership ?*

Steve : I think the biggest one is developing a sense of satisfaction, purpose, accomplishment all those things out of indirect contribution. As an individual you write code, you design software, you craft something from nothing. In engineering the best thing that I found is to create something from nothing ...; this is creative process. There was nothing, then there was a requirement that I (the engineer) built something, using some tools, to meet the requirements, that like I created that, I usually see if I made these things and now that’s a thing that exists in the world because I made it. Once you’re a manager you don’t get to do that, you give that to other people. The people (engineers) get the satisfaction of taking nothing and making something, you (as a manager) help them do it. You (as a manager) don’t actually do very much at all. I (being a manager) miss (being an individual contributor) that and that took a lot in me, that I could live without that and be content with letting other people have that and I’m just facilitating that. That was a tough adjustment to make. When you are leading a team your job (as a leader) is to make your team productive. It’s not always about what’s the quickest way to get something

done, to the most maintainable way to get something done or something else like that is often how do I make sure that the team is as highly functioning as possible, and we're not going to have attrition, or we're not going to have an ability to understand and communicate or other things like that is what becomes your (managers) primary concern.

Leadership And Communication

Partha : *Does communication play a big role when you are in leadership?*

Steve : Communication plays a big role in you move into leadership. I often half jokingly say that I'm the manager and the only tool that I have to solve this problem is to call a meeting. (chuckle) As a manager when you can not solve a problem yourself it's best to get the people together for a meeting . . . , there can be other tools in the toolbox (manager's tool box), but to a first approximation facilitate communication among the local (geographically local) team. However, in a distributed team you would have to do it using emails. So facilitating a communication and communicating is key.

Partha : *Since you have been leading people for a long time what would be your advice to and engineer out of grad school what are the soft skills, communication skills the person should focus on in order to be more productive in order to be a good engineer?*

Steve : If you want to be in an engineering leadership it is about soft skills it is about (um thinking as how to phrase it) the people side I talked about you give up the creativity the sense of building something from nothing but the flip side of that is I think as hard as computer problems are pretty thorny and pretty hard. The hardest computer problem that I have ever dealt with, is easier than one of the simplest human problems that you deal with. It is because human beings are so phenomenally complicated and then when you get a group of them, interacting to try to understand how those interactions are going to get affected, as you make a change, as you allocate responsibility or whatever . . . , but it is really (really) rewarding when you get it right, but it is really challenging. Communication is obviously needed but what do you need to cultivate in yourself is empathy, being able to anticipate how that person receives this (message), what is the impact on them if we do it this way verses that way. You have to be able to put yourself in their shoes (at this point a truck whizzed passed us beside our dining table), to be able to say if we do it this way it will be okay.

Partha : *I think one of the words that you used that is very rarely used in professional field is the word "empathy".*

Steve : I think if you don't have that (empathy); how do you anticipate the impact of your decision on the individual ? At the end of the day product lives or dies but the team should be motivated and pulling it together. If you fail to pull that off (as a leader), if you fail to create a culture where everybody wants to be successful and wants to be in, no amount of micromanaging a browbeating driving goal etc You cannot scale that way you cannot drive, micro analyze little goals to everybody and motivate them in another way. An effective organization needs to delegate as much as possible down to the individuals and give them both authority and accountability to provide end results, so that they have the authority to make the decision, and they have the accountability to realize that they are making the right decisions. The more you pushed down (authority and accountability) the better it is for the organization. It is motivating and empowering ... , and people would want to rise up and do their best. If you find yourself having to shy away from that it may be a poor fit (for the team), that's a polite way to say somebody who is not working out in the team because they have checked out, or they have they're not fit for the goals (of the organization) or they have the wrong skill sets, that kind of thingsr, Else you have to drive somebody to weekly goals or are you done this week, what are you going to do next week and so on. That is a sign that the team is dysfunctional, the team needs to be able to say that this is not working out, or we will find out different ways to make this (teams) work.

Partha : *In general light from your experience, do you think company companies (leadership) value thinking and try to find out the leadership qualities in an engineer in work environment ?*

Steve : Um ... That's a good question. Not as much as they like to I believe that good leaders especially good exec (executive) leaders maybe. One side note for when I was in management (early days), I got sort of disillusioned. I felt that guy who is leading us, all he talks about this big (says big multiple times) vague broad brush dry topic; why doesn't he get around there. I came to realize over time it is appropriate at that level that don't think that (Steve's early thinking), engagement, empowerment, Innovation those are real. If he's (a senior leader) not got his eye on ... , how are we going to make those happen, and if he allows himself to get dragged into the quarter results, are we (lower management) on track with projects, make this partner happy ... than no one is going to be paying

attention to those big topics. These (innovation, engagement, empowerment etc) big topics really (really really) matters, so that is why leaders at the highest level of the organization do focus on (big ticket items) because somebody has to, and who better than the person, who cannot get into the weeds (daily routine of engineering works) to deal with it.

Now do companies value leadership and growing up leaders ? The guys at the top are smart guys ... at the top they absolutely and usually do. What happens between them and the individual on the ground; are either not wise enough or narrowly focussed, management (lower rung of management) chain will lose sight of that, and will choose to focus on the near-term deliverables and say you know what, I know you want to do that (innovative stuffs) but I don't have time now, I'm sorry ... So like I said a good executive would recognize the importance of the priorities but it can get lost in translation in the management chain and individual managers can feel helpless. First line managers feel accountable about there are deliverables, they don't have resources, so that they can allow the individual to grow. The individual engineer feels my manager doesn't care about this (innovation, empowerment etc), I (engineer) don't know what the organization cares because from what I can see they don't care at all ...; that layers of abstraction in between (management chain) loose (objectives of the upper management) and that's an organizational challenge that we all have to fight.

Partha : *so you brought the question of Ethics during your first job as a leader you always have ethical dilemmas how do you normally handle them?*

Steve : Oh perfect, of course (chuckling)

Partha : *Could you cite an example without naming any person or company.*

Steve : I would say that I have been lucky in my career and did not have individuals who work for me or near me who have been behaving unethically ...; that is a challenge but I think that has been pretty darn rare.

My belief is that ethical behavior is nowhere near the norm less than in like 1% of the population ... (unethical people) like so selfish, are sociopathic they don't care about the ethical norm, and they behave the way they want to behave. What I think is far more common people get caught in what is if I do bad, would that be bad or if I do this or would it be bad, if I do that, in what ways I'm not strictly ethical or I'm screwed either or—if screw the company what do I do ... those are the hard ones. Let me cite an example say in a sales meeting, this scenario is sort of made up but very real, you are in a meeting with the customer and you are the engineer who got

dragged to showcase the new product and you and the sales guy couldn't talk about it (before the presentation). The sales guy makes a claim that you know is false. So now you know he just told the customer something that is not going to happen, a feature that has not been delivered yet and doesn't really work, and he asserted that it works now. Is it ethical for you to say to the customer hey he is lying to you ? Or maybe more politely that you disagree with the sales person in front of the customer (because you might think that the company should be an ethical). On the other hand the salesperson has a great relationship with that account (with that customer) and he may have greater context about what he is saying to this individual right here, right now. The reality I know is that person is not going to take advantage (question) about that (feature) but if they hear any doubt about it, they can use it to push us (your company) out of that account. So if you do disagree, you could be ruining that sales person's career because now the customer is not going to trust him anymore because you just told them that you were lying and in the end it did not have any impact. The salesperson would make word play like this and it would have no effect on the deliverables. So you got to be really careful, unless you are fully aware you can think you're making an ethical choice. That's where the hard one comes, not that this is right or that is right do the right thing ...; but what is the right thing here, those are the ones in general are difficult. The principal I go by is telling the truth as much as possible is easier because the truth is much easier to keep track of. If you find yourself lying, it's hard to remember whom did you lie and what. In general, it's easier to tell the truth. The lies get multiplied and it's hard to keep track and for self-interested reason or nothing else tell the truth as much as you can. Sometimes when you are in thinking layoff coming, you can't tell individual employee that there is a layoff, it might be nice to tell them but it would be unethical to the other people (stakeholders) in the company who aren't going to know, the shareholders of the company who don't know There is a way this (layoff) news gets broken and there's a way this news (layoff) should not get broken. It seems unethical to have to sit down and work with someone somebody and not tell them that they are going to get fired next week but telling them would be even more unethical supposed the ones that gets hard you're not able to tell the truth for a very reason got a good reason and does that does happen and it happens more often in management.

Productivity and Hiring

Partha : *Lets dive a little into Steve's daily schedule. What keeps you busy in your daily job? How you how do you prioritize your goals how do you manage risks*

Steve : Goodness !!! On the first one there is an old book which is called the 7 Habits of Highly Effective People. There are a lot of similar books about how to organize your life how to properly prioritize your day most of them that I have seen involved being very conscious and mindful, how you spend your time either tracking your time and using a tool to manage and track your time or having a day planner online or physical form, where you write down these are the things I need to accomplish today and I'm not going to quit working till I get them done, that sort of thing, I imagine those are truly beneficial and helpful and I don't use them at all. My trade-offs on that and I'll use this as an excuse; I will use his reasoning (chuckle) as it applies to a few things in my career where I believe that there are things I could have done I could do even now would be beneficial to me and to my career which would be unpleasant for me so much so that I would do better but I hate doing that and so my life would not be fun I would not enjoy my day as much as I want to although I would be productive but I would have a worse career. I might not go as far otherwise, I might not do as well otherwise I'm not going to be organized in that way and that's a deliberate choice. I think you can weirdly mindful of not being mindful. That is one thing I have thought about, I have actually taken those classes because my previous company sponsored me to go to classes to be more organized. I went through, did it and spend 2 weeks afterwards tried that and I hate this and it really sucks. It's a chore, it makes me feel that I'm not having any dynamism in my day, I don't like this, and so I stopped. If I had kept doing it I would be more productive and there are feelings. I sometimes think you need to balance In general if I get to my career goals I'm a big believer make sure you know Have a sense of fulfillment from your work, Don't feel you are wasting your time That's really important But also make sure that you are having fun While you are doing it So that it's not only about Accomplishment at the end of the day It's also about how did you feel going about through the day. Because if all you are doing it's just having a sense of accomplishment And hating every minute on your way There then that's a horrible choice On the other hand enjoying every minute off your day I'm not having any accomplishment at the end of the day is also a horrible choice So you have to find a way to balance And enjoy the time you are spending at it Most of the time at least Majority of the time hopefully And at the end of the day

or the week looking back that I had quite a few things done Is a good use of time You have to balance those two things That's the way I did plan and prioritize And all that stuff.

Partha : *I like The phrase used mindful yet not being mindful*

Steve : Yeah

Partha : *One question on doing your everyday job. What is your secret formula to get things done ?*

Steve : (few seconds of a burst into laughter and then chuckle) Okay I got one trick, that is, here is one nugget I think I could sum it up in a phrase. I use this phrase sometime, "Authority is never given it's taken". People don't Grant you the authority to do something you can. Some executive can annoy you and say this person is in charge of whatever, the reality is if you acted like you are in charge or even if they didn't do that then you are in charge and even if they say that to you and you act like you are not in charge, then you are not in charge. So the person above you is not responsible whether you are an authority. You have to bring that to the road yourself and effectively assert yourself, as being the decision maker and authority and If you are in the right role in the organization you will get away with it and if you are not, then somebody somewhere in the organization will figure out why is this person pretending to be In charge, and will get him out of there. So you do it appropriately, do not make the mistake that a lot of people, in their carrier make, (they) assume they need to be told that they are in charge of something, before anybody will respect them for being in charge. One of the things that I learned early on as a manager (I became a manager of a small group) But I was put in charge dotted line effectively to a much larger group and no one really granted me that my company (Juniper Networks) when it was minimal, and it did not have formal lines of management. No one ever said that Steve was in charge of ... But Steve has got the hardware team doing this, Software and Manufacturing and marketing and others Kind of lineup ... I started telling them what needed to get done, and they all just did it. I realized that if you just assert, this is what needs to get done and you give a good reason, people listen, and they don't come back and say who the hell are you ? Why you are telling me what to do? ... If you do with a reason and politely, you will get things done. I think if something needs to be done to get done (laughter and chuckling) and they do it and it's amazing. Tell with Authority by just telling ... we need to do this let's do it now.

Partha : *You hire a lot of engineers. The first thing that you look for is the candidates resume. What are the things that you look for in the resume*

of an engineer ?

Steve : Generically there are things, I will say briefly ...; Of course, they have to have a rough skill set that matches. So if they got the wrong tools wrong areas of expertise, (I'm not going to hire an acoustic engineer for networking), so whatever you got to have a quick rough fit. Next higher orbit is what is their history ? Did they jump around a lot. Person being at a company for 9 months to a year another company for 18 months or 2 years. I see lots of those resumes where people have jumped from job to job (to job). If I'm going to bring you onboard then I'm investing on you a huge amount to train you up and get you started and if I can't get a couple of years then that's not good. Now there are exceptions there are good reasons for leaving within 6 months. But if they do it over and over and over (again) than it's a red flag and so a tenure of couple of years in general and there is stint or 2 where they have been there for a while. If you are hiring fresh out of school then obviously that is not the criteria. But if you are looking for an experienced engineer then you look for staying power. Beyond that on the resume there are particular skills like signal integrity where education is really (really) important. You are really not going to be good at that if you do not have the right education say math. In computers these days education (qualification) is not that important, I'm working with a couple of people who do not have a college degree at all and who are perfectly good at what they do(one in customer support one is in Quality Assurance), they are doing an excellent job. So I have learnt that you can be good at your job and not have a formal education and so that it's a ruling out thing. Education is important for some jobs that are specialized. Resumes are a fast like a rule in, rule out thing. But my habit is to have a call over the phone to see if the candidate can make the basic cut. I judge a lot in that phone call about how engaged the individual is in expressing, I'm looking for communication skills, I'm looking for signs for native intelligence, what are they thinking are they able to make sense of what we are talking to each other. In that phone screen can they convince me that they are somebody who can think in a phone screen in 20-30 minutes I do not get into deep detail about their background, their technical skills, but I'm asking instead what you are looking to do, what is the most interesting problem that you have tackled in the last 6 months, how did you attack that and not ask if you are a great fit for this role or not Because I screened that out from the resume and I will screen again if they make it through that hurdle. It's also remarkable how many people will pick a fight with me in phone and I'm talking to them for the first time and I will say you know, I will have the courtesy of saying hey I'm here, we are doing this, this is the kind of role we have. A remarkable number of people will say stuff like why

are you doing this ? Wait a minute I'm interviewing you and you are questioning whether we are having a good company or not ? You are making it easy I can hang up on you now. So it's surprising that how many people will immediately do an intellectual jousting with you. Don't do that. You are supplicating to them and everybody expects it. If I'm not listening from you that I really want this job, I'm going to be great at this job. If you do not blow your own horn, then you do not understand how the interview goes or you do not want this job at all. This comes along a lot 10-20

Partha : *You have been in college in the 80s. You have used tools in the 80s and 90s and you are using tools now. How do you think the software progression been ? Has it become more complex more user-friendly ?*

Steve : I think we are all limited by human brain power. Complexity has not changed shifted around a little bit. My perspective on software engineering into it after being years not into it (Steve worked as a hardware engineer for a long part in his career). I'm trying to make the shift at the moment where I'm. Most modern software engineering jobs in the world today, I'm not talking about networking industry it's not quite there yet and it's going to shift, if you look at the vast majority of software engineers on the planet the talent that you bring to bare solves the problem (differently) than it did 10 to 20 years ago. 20 years ago people used compilers, GDB, mostly c, if you are advanced c++, if you were using script it was perl, bourne shell scripting. Real programs were almost all c and the all Java; and those kinds of imperative language you would take a little bit of libraries and write a lot of custom code and write the application software that leverages the custom libraries with custom code. Now that is inverted, now the libraries do all the heavy lifting, you write a little code in comparison. You spend a lot of time to identify the appropriate library to use. So the complexity has shifted from how to structure my algorithm, to how do I find the library with the right algorithm and how do I integrate that library to make the whole thing to fit together in a very general sense. The problem has shifted from how to create something out of nothing line by line to how do I grab some pre-built stuff and make them all fit together. The interesting thing to me in software engineering now what is the community support for this ? How active is the community for this library, does it have a good test suite built in so that I can rely on it if I integrate it. What are the licensing terms. Like 20 years ago nobody worried about those terms. We used standard libraries and nothing much of anything else. Now you worry about what is the tool set that you go to bare about what do I need to assemble to make this problem solve. Today it's a different kind of complexity I think. Networking is still a kind of little backward a lot of application (made a move).

Partha : *Staying with productivity, what are the three or four tools that you would ask graduate students straight out of school to get a head start in the workplace ?*

Steve : Exactly to the point as I mentioned earlier, the tools have shifted what tools you need to know and in fact the nature of the problem has shifted. And so instead of knowing how to use GCC (Gnu C Compiler) and GDB and now you need to know how to use stack overflow and google and that kind of thing. Researching and identifying existing things, Github, not because it's GIT and how to do version control in GIT, and Github, so you know how to find out the package pypi, nodeJs identify what those are, and those are key tools. Also believe it or not linkedin and social media how to follow people who are thought leaders in technology direction that you are interested in going in. So you can see what are the advances that you are going to be up on I'm not going to give you a database like Redis, I don't know in four years it might be totally out of date. If you know who to follow so you know what is making redis out of date you need to know who to be listening to, whom to follow so that you know what are the things to be researching. How to follow the different communities open source platforms like Github, I can tell that's an active one, I can tell that has a good user base, I can tell that is well-supported. These are the main things that I would want in a new software engineer out of college.

Future of Computing

Partha : *Getting on to some topics which are more philosophical, Moore's law is nearing its end in 2025? How do we go from there ?*

Steve : (Chuckle). I don't know. (Chuckle). Moore's Law has been nearing the end for some time now. It depends on how you define Moore's law. Oh, Boy! Boy! These are open-ended question !! (We did a quick time check as both Steve and myself had to be done in the next 10 minutes).

Steve : Many years ago on a use net forum called comp.org, somebody asked a question asked, what are the computers would look like in the future ? This was like thirty years back. Some people gave the best answer that I have ever seen, never seen it better than that. Someone described the super computer of the future would look physically embodied like "Smoking Hairy Golf Ball". So wait each of those words are important. "Smoking" because at the end of the day Heat is the main problem. We have not yet figured out how to do entropy free computation. So there is always waste

heat in any computation you do and you are going to be limited by how much you can dissipate. So that little Golf ball is going to be hot, smoking hot. And then it's going to be connected to the outside world. It's physically contained because you want to have everything locally closed to get around physically speed of light limits and switching density very high, the smaller the feature is, you want it to be fast so that it's nearby to communicate and not have to change state it needs to be small and compact, but it needs to interface to the outside world so it's hairy and it's a golf ball because it's not going to be large. So it's going to be a tight little hot ball with a lot of physical connections coming out of it; once we get to that those Golf balls are going to be better. I'm not buying that Moore's law is going to end with multi chip module and build our way up. There is a saying that everything in computing IBM did it in the 50s. Multi chip modules is like that, IBM built them why back then it was SSI and not VLSI. So they went from SSI to LSI to MSI to VLSI, and then we stop saying that a long time ago. There were dozens of transistors in a device they built device to connect them to a package. This idea building devices and then connecting those devices with devices to make them a package has been ancient so it's going to come back again. So what will happen to Moore's law.

Partha : *Is the entry barrier into Linux Kernel Programming high compared to 10 years earlier ?*

Steve : I don't know if I'm very good at answering that. I don't think it's higher, it's been always higher. It's a tightly-knit community difficult to break in it has always been that way.

Partha : *What is your advice to computer science graduate ? Stick to the fundamental or chase something that is the flavour of the time like IOT (Internet of Things), Artificial Intelligence, Machine Learning ?*

Steve : You got to get a good grounding on the computer fundamentals. If you can't think about combinatorics, algorithmic complexities you are doomed and you are not going to learn that anywhere except in the school and so you got to have the foundation. But don't go into esoteric endless explanation it's a foundation and then pick something that you are passionate about something. It's ok you can shift focus you can chase something till you are not interested in it any more IOT or whatever you are passionate about.

Partha : *should we regulate algorithms ?*

Steve : oh like Facebook kind of algorithms.

Partha : *Yes*

Steve : Oh Wow !! That's an excellent question. Because there are

domains where it's an absolutely must like self-driving cars, perhaps social media certainly jobs, the one I saw most recently is very disturbing, companies are selling recruiting services that are ruling candidates in and out based on their facial recognition. They are a job screening company that says that you apply, and they will put your name in a company and the company subscribes, and they say that they will put in candidates. The company doing the screening is based on an algorithm based on the facial recognition, like does that face look like the other face which has been successful, which is a disaster. I think that is a disaster and that there are places where algorithms should be absolutely regulated like public safety, discrimination. That's difficult to do, but we have to do it absolutely.

Partha : *This is the last question, There are security breaches all over, credit card companies being breached, Facebook has been breached. Data breaches are all over the place! Where do you think the problem lies. Technology or process or people ?*

Steve : It's almost always the humans. It's almost always social engineering or people using a dumb default password. The Technology, yes there are always zero-day exploits. We need to do more about that but I think the 90-10 or 80-20 rules, most of the problems and less of an effort is on the human aspect of it where we should be focusing our design efforts is to not allow humans make a basic blunder, not allowing to plug a USB stick into the computer and automatically run anything. If you allow that then somebody foolish is going to take an unknown, untrusted device and stick it in. so Don't do that, don't let it to run no matter what it is. It might be convenient to do that but if you let them do then you have empowered the user to make a dumb mistake. So we got to build system that would make it humans to make it impossible to make a dumb mistake and allow the security breach to happen. There is Technology to help but it's mostly human and that is my point of view.

Partha : *Thank You, Steve, for your time.*

Steve : There you go.

Conclusion

I'm thankful to Steve for giving me an hour of his time for this interview despite his busy schedule. Interviewing Steve, provided me with a sense of purpose for career after masters. Steve's knowledge in the domain of software industry is wide. There are quite a few things that stands out from his interview. From an education perspective it's important that the basics of Computer science fundamentals should not be ignored. Courses in Systems software and algorithms are key for being successful in any area of software engineering. Steve's insight into the hiring process would be helpful for any engineer looking for a job. While talking about ethics, Steve pointed out a few cases where engineers need to be really careful. He advises to be truthful as much as possible because it's easy and one do not need to keep track of the lies. Steve is a part of leadership team, and he thinks and feels strongly that leadership team is always interested in finding out the next leaders from the group of engineers. He thinks that engineers need to step up and take responsibility in their work, but they should be mindful about their responsibilities. He too recognizes the conundrum that lower managers have in organizations with respect to division of time between delivering products and encouraging engineers to take risks. After spending time with Steve I felt that it's important to have a mentor early in career. It's important to develop work an ethic which builds up an engineers' credibility. Being ethical is important, however we should be tactful at time when we do not have understanding of impact or context. Software Engineering is a continuously evolving field with new techniques

and methods being developed for improved productivity and quality of product. Being an software engineer I should be keeping my self abreast with the latest trends.