

new faces

of FreeBSD

BY DRU LAVIGNE

This column aims to shine a spotlight on contributors who recently received their commit bit and to introduce them to the FreeBSD community.

In this installment, the spotlight is on [Nick O'Brien](#), who received his src bit in March, and [Richard Scheffenegger](#), who received his src bit in April.

.....
 Tell us a bit about yourself, your background, and your interests.

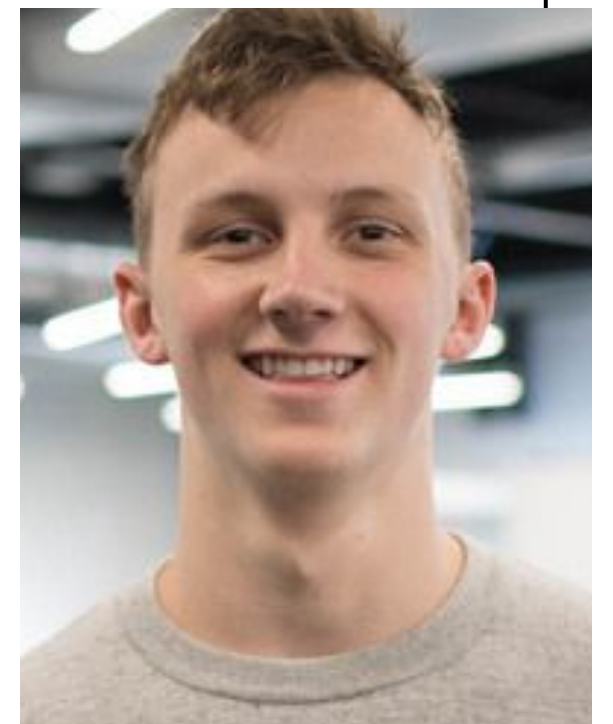
- **Nick:** I was born in and grew up in Seattle, Washington, but I've been living in San Jose, California, for the most part since late 2017. Currently, I work on embedded software at Axiado where we get to use FreeBSD to do RISC-V things.

Initially, I got into writing software more as a necessity than as a direct interest in writing code. Since the time I was in grade school, I have needed to write software for different personal projects and would just Google "How to" After years of miscellaneous software development, I began to realize that I really enjoyed the actual problem-solving aspect of engineering and got increasingly serious about the actual skill of writing software.

Some of my hobbies outside of computers include running, getting thrashed by waves when trying to surf, and strumming a guitar.

- **Richard:** I'm based in Vienna, Austria (commonly mangled up with Vienna, Virginia, here at my workplace), which is also my hometown. Some 15 years ago, I joined a Network Storage Company that was looking for networking experts to support their then new product, a caching web proxy. However, despite my skills in that area, my resume was turned down by a professional headhunter. I started my networking career by setting up a "networked" BBS (part of the TrekNet, based on FIDOnet technology) while in school, using mail and USENIX gateways before the general population even knew the Internet existed. This was well before the web became a thing. I remember that Gopher was all the rage when I visited a computer room at the university. Then I worked in a number of smaller companies, always related somehow to networking. At that time, around the turn of the millennium, I was the go-to expert in German-speaking countries for one vendor's special network operating mode, a way-too-early precursor to TRILL—where even the vendor's support people reached out to me numerous times. But before digressing too much, by chance, my resume was picked up by an intern working at the headhunter company and simply filed my profile for a similar job, in addition to the job I originally applied for.

After having been thoroughly tested and interviewed multiple times (in retrospect, I believe the people doing the technical part could not figure out how I responded so knowledgeably and quickly, while none of my answers could be found verbatim by any of the search engines



of the day), I started at Network Appliance, now known as NetApp, in 2005. My original office back then was the EMEA headquarters in Amsterdam, where I worked in a field related to my prior IT expertise. NetApp is one of a handful of enterprise-class vendors offering professional, large-scale NAS solutions. Most other large storage vendors emphasize SAN much more, so my ending up at NetApp was a good match after all. Eventually, I started working in an EMEA-wide role within the company and was allowed to pick my primary office. So, after having worked out of Germany, I eventually moved back to where I came from.

Also, being very interested in networking—in particular, transporting data from A to B—turns out to be quite an important expertise when your employer deals with devices that are supposed to provide data as quickly as possible across a variety of different protocols—NAS and SAN, and for the IP part of that, via NFS, SMB, and iSCSI (NVMe coming soon to IP too). And, as I had been trained as a chemist, a good grasp of complex systems interactions also helped.

Over the course of my tenure, I became almost obsessed with trying to eradicate all sources of excessive delay in IP protocols, which makes people often look for different technological solutions when it comes to storage (e.g., SAN, IB). I remember getting our CTO at that time upset, as I wanted to convince him in a prepared speech in an internal forum with a larger audience about the merits of reduced IP latency for our products—after having had a short chat in private with him, in which he agreed with me.

The interaction led me to engage with the IETF as an active participant, and I am still working with a heavy focus in the transport area and TCP, in particular. For a short while, I was also cochair of the (newer) AQM Working Group during its existence and worked to get more modern Queue management mechanisms than RED adopted. The reason for my involvement in AQM is that I had become quite interested in Explicit Congestion Notification, which is a way for TCP to adjust without loss to the (momentary) capabilities and capacities of the network. And it is generally understood that what most network devices implement—if they implement something other than drop-tail—is rather inadequate for the intended task.

However, as we're located in Europe quite far from our engineering, which is in the U.S. and India, the pace of improvements in our product in the space I cared about was slow. At some point, when I provided yet another fully worked-out bug report around TCP to address some issues I had observed with our customers running storage traffic across TCP/IP, I was told "Once there's a FreeBSD patch available for this behavior, we can pull it in."

So that is how I really got involved with the FreeBSD Project, by reaching out to my existing contacts and working to become first a contributor (providing patches with test cases, explanations about where the issues are, and also filing some bug reports) and then ultimately being accepted as a committer.

.....
How did you first learn about FreeBSD and what about FreeBSD interested you?

- **Nick:** I had heard about FreeBSD here and there for a while and probably even used it, but I didn't start really getting into FreeBSD until I joined Axiado in 2018, where we started with figuring out how to get FreeBSD booting on RISC-V hardware. I enjoy writing code for FreeBSD, but I also really like the dedicated, tight-knit community. It feels like it's a relatively small community that's making quite a large impact. Lastly, I couldn't pass up an opportunity to work with so many smart people.

- **Richard:** It is hard to say; I was certainly aware of the BSD family for at least the last 20 years. My first active exposure to FreeBSD was with what became <https://reviews.freebsd.org/rS206456>. I became interested in TCP ECN and found that the original ECN implementation in the BSD family suffered from an oversight (and not fully documented behavior in RFC3168) where it would not behave as expected. As FreeBSD is deeply ingrained in some of the commercial products of my employer, most notably the ONTAP data management software, it naturally followed that I grew increasingly interested in improving the situation there. That was particularly due to the fact that FreeBSD used to lack a number of important features (such as DSACK), which are important in interfacing with systems in the cloud, or had features which were not quite on production level (e.g., CUBIC). The latter is quite important for delivering good performance across longer distances. NetApp offers a solution built around synchronous data replication across wide distances called MetroCluster, which is my product expertise. Even while CUBIC has been available for some time, storage class traffic has its own set of challenges. For one, storage IO is very transactional—a client issues very detailed requests. Also, storage IO is often application limited—data is only requested at the pace the application can process it and a server can't transmit a large object as quickly as possible as it is only determined by TCP and network limits. This is quite unlike the typical architectural use case of TCP (infinite data and as fast as the network/receiver allow) and also is not encountered in a more web-dominated environment.

.....

How did you end up becoming a committer?

- **Nick:** Kristof Provost (kp@) and Philip Paeps (philip@) do FreeBSD consulting work at Axia-do. After working with them for some time and after I had interacted with the RISC-V community on Phabricator, they invited me to join my first developer conference in San Jose at Intel in October of 2019! It was a great experience, and that was really the first time I had seen a team come together like that for an open-source project. After a few more months on Phabricator, they proposed a commit bit for me. I was ecstatic!
- **Richard:** I became familiar with a number of committers to the FreeBSD Project during my IETF meetings. Working on improvements on TCP in general also entails implementing and deploying them eventually. So, I was already writing code to improve SACK loss recovery that was based on a suggestion of mine that got fully worked out in RFC6675, known as rescue retransmission. Most of that got stashed away in private builds and is not universally available. Working at a data company also exposes you to interfacing with appliances and software provided by many other vendors, some of which are also making use of FreeBSD, or parts thereof, in their software. I wanted to contribute bug fixes and improvements in the larger ecosystem for other users of FreeBSD—and ultimately also benefiting my employer and the mutual customers of those other vendors—by generally improving the behavior, stability, and interoperability in the still typical, complex, heterogeneous environments that are commercial IT architectures. After providing around 40 patches (mostly bug fixes and one major feature—DSACK—which resonated well with other contributors), I was invited to become a committer with rgrimes@ and tuexen@ as mentors. I'm still in awe at having that significant level of process and technical expertise to help me improve my contributions to the project.

.....

How has your experience been since joining the FreeBSD Project? Do you have any advice for readers who may be interested in also becoming a FreeBSD committer?

- **Nick:** It has been a really great experience, but it was also a slightly weird time to join with a lot of strange world events taking place. My favorite part so far is bouncing ideas back and forth with different FreeBSD developers all over the world. Everyone seems open to listening, so I feel like I really have a voice. I've also learned so much from my peers!

My advice to readers who would like to become committers is to understand that it's not too far out of reach. You don't need to get struck by lightning to get selected as a committer. Just start out by deciding what you'd like to contribute to, follow the ongoing upstream work on Phabricator, connect with others already working on those bits, and chip in where you can! When I told him I was interested in becoming a committer, I remember philip@ said something like, "It's not as far off as you think." At the time, it really didn't even feel like it was a possibility, but he was right; just a few months later I was a committer! We're all a team and everyone is here to help so just be grateful for those supporting you and try to make the most of it! If you're really trying to become a committer, trust me, someone will notice.

- **Richard:** Well, the core transport@ group (dealing mostly with TCP, some SCTP, and related aspects) holds conference calls every other week. I started participating in them when I began contributing proper diffs on reviews.freebsd.org. These conference calls are very helpful in discussing problems and solutions, as well as building a reputation. As you can imagine, messing around with something as critical as TCP comes with a very high bar to not introduce any regressions or performance degradations. Having said that, one of my earlier commits did have some unintended side effects, so I learned in a flash how to revert commits from HEAD. The review process there is very strict, and patches get discussed at length before being accepted. In short, I would not recommend anyone to pick the base TCP stack as their focus area if their interest is more about learning the processes of working in an open-source project. However, the peers and mentors I found are all excellent and give very good feedback. Counting my earliest, indirect contribution to the project with the ECN issue, it was about 10 years until I became a contributor. In my specific case, my employer has been supportive, as enhancing the status quo of transport in the networking space is important from a product perspective—even when it's a very slow process to make sure nothing breaks along the way. But the length of time it took me to become a committer is atypical and more about the area of my contributions. I know a few committers who started indirectly contributing to the FreeBSD Project while working on their master theses—when their professors were supportive—and they still contribute to this day. Since a master thesis hardly ever takes more than a year to complete, that would be more the norm.

DRU LAVIGNE is the author of *BSD Hacks* and *The Best of FreeBSD Basics*.