

Olivier Bonaventure

Brandon: [00:04:27] Olivier, I'm really happy to have you on the podcast because there's a technology that I'm super-interested in, that you've been a part of for a really long time called MPTCP, or Multipath TCP. Welcome.

Olivier: Welcome.

Getting to Know Olivier

Brandon: [00:04:38] So let's get started. Uh, tell us a little bit about yourself. Who are you? What do you do?

Olivier: I'm professor, I'm computer science professor at the university, which is UC Louvain Belgium, and I mainly teach computer science, networking to bachelor and master degree students.

Brandon: Okay. And have you always been involved in networking? What did you do before the professorship?

Olivier: I started to do networking when I was 14 years old. I got a V.21 modem, which was 300 bits per second, which was really fun at the time. And so I was using the dial-up to connect to BBS, and then it evolved. I did an exam to do amateur radio, because when you are an amateur radio, you can do packet radio.

And so I did packet radio. And then I went to university and in university, then I specialized in networking and I did my PhD in networking. And then I went to industry for one year and I became professor there.

Brandon: Okay. So that's interesting. What motivated going from industry to academia? That was a short stint of one year.

Olivier: It was unexpected. So when I left university, I'd not yet finished my PhD, but I decided that, well, my first son was there and, uh, I wanted to do something else ~~in~~ in a company, to do something useful, and there was an opportunity to go to, uh, to go back to university and it was unexpected. So I applied and I got the position, so I came back.

Brandon: And that was the job you always wanted.

Olivier: Yes. So that's, I think it's, it's useful to train students. So there are funny parts, like all jobs. There are parts that are more interesting than others. So I like to teach, I like to do research, ...doing the administrative part is less interesting.

Intro to Multipath TCP

Brandon: [00:06:29] All right. Well, you'll get to teach us a little bit about MPTCP and I think that's a perfect point to roll into it. Can you describe Multipath TCP in a few sentences for our listeners?

Olivier: So that, when you exchange data over the network, you can lose information and you can have errors, and these errors and the lost information will be recovered by a protocol, which is called TCP, the transmission control protocol. And so TCP takes care of, uh, retransmitting lost data and, um, link congestion when there is congestion inside your network, but TCP was designed in the 1970s when, uh, the computers that were connected to the network had a single link to the network. And so all the, the mechanism that are used by TCP, assume that there is a single link from each computer to the network and all the protocols that we use today were designed with that in mind and Multipath TCP simply changes that it allows you, if you have multiple connections on your device, such as on your smartphone, where you have WiFi and cellular, it allows you to use [00:03:00] both. Cellular and WiFi to exchange data with a remote server. This means that we are adapting TCP to the modern devices that we have today.

Brandon: All right. So there's a problem that I have all the time, which is I'm on a Zoom call and I'm on my phone and I wanna walk around. I want to get some steps while I'm on that call. And my phone starts out connected to WiFi. It's got LTE and as I start walking out that signal degrades, but it takes a while for Zoom to figure it out, and there's an interruption. And it's exactly the kind of interruption where as I'm talking, I don't know if people can hear me. And so I know that if I'm going to walk outside, I'm going to have a problem. And so it sounds like this is exactly the kind of problem, one of them, that MPTCP would help with.

Examples of MPTCP, Today

Olivier: [00:08:06] So that's the kind of problem that MPTCP solved, but as of today, Zoom does not use MPTCP but there are other applications that use MPTCP and one example is Apple Siri. So Siri is the voice recognition application from Apple on the iPhones. And what Siri does is basically that, uh, your [00:04:00] iPhone. Captures your voice, sends it to a server and the server processes your voice and converts the voice to commands. And for this, it creates a connection to the network and it uses the cellular and the WiFi network. And with MPTCP it can switch from one to another and use both at the same time, because when you are walking you get away from your wifi network and you are not yet well-connected to the LTE network.

And so there are some periods of time where it's important to be able to send data over both paths so that they can reach the server without interruption.

Brandon: And that seems like a great application because as a user, you notice if there's any delay whatsoever.

Olivier: Yeah. The other example from Apple is Apple Music. So when you use Apple Music, you expect that the streaming of your audio of your song will start as quickly as possible. So you start to download over WiFi, but then you get away from your house and you want to continue to listen to your favorite song.

And when you switch from Wi-Fi to cellular, what happens is that, with MPTCP the data transfer continues without any interruption. And if the WiFi becomes too low, or if the cellular becomes too low, because you are inside the house and the cellular does not work

well, then you switch to the WiFi or you use both at the same time because the application knows when it will need to play back the next audio. And so if the next audio is not yet available, then it will get it from both WiFi and cellular so that you can get it as quickly as possible so that your song is not interrupted.

Brandon: So our listeners may already be using MPTCP and not have any idea under the covers that this is giving them a better experience.

Olivier: Yes. So, uh, when science works it's sometimes it's like magic.

Brandon: So you're a magician - you're helping create magic.

Lots of People Made This Happen

Olivier: [00:10:10] There are computer scientists that do magic, and this is one example, but in this case, this is uh, these are the Apple engineers and Christophe Paasch who did his PhD in my group, is one of the magicians there.

Who were the other people who made this possible?

There are many people who were involved in MPTCP So, um, Costin Raicu did the, one of the first implementations. Mark Handley was involved in the protocol design. Alan Ford developed the specification.

Sebastien Barre wrote the first implementation in the Linux Kernel Gregory Detal was involved as well. And there are many people.

The Original Motivation

Brandon: [00:10:46] Great. So Olivier, I wanted to go back to the motivations here because when we hear about the experience difference it already provides today, I have to think, "Why hasn't it always been there in the systems we use?" What was originally the thing that motivated it the most or the problem that led to your involvement and others' involvement?

Olivier: So the original idea was, um, in 2010, we were considering the preparation of a European project with several researchers in Europe and among them, there was Mark Handley and Mark Handley said, ah, we should solve the problem that TCP only uses one path and we should be able to use multiple paths and the problem should be simple because we only need to be able to do congestion control correctly over the different paths.

We wrote a project proposal and it was accepted by the European commission. And so we got funding for three years to solve this and other interesting problems. And then, so we developed the idea, Mark Handley proposed the first design and then we improved it. We went to the IETF and there was an interest within the IETF to develop it further.

And so we, it took three years to get the RFC published and the protocol finalized.

Derick: Standards bodies are famous for being slow-moving, but that was a years ago, right? That there's a huge lag time. I think, um, for this particular project from conception to seeing it in, like in the wild.

Surprisingly - Apple adopts it first!

Olivier: [00:12:10] So it started in 2010 the first research on that, and then the RFC was published in January, 2013. And then in [00:08:00] September, 2013, we were surprised because Apple used that on iOS 7 and so Apple adopted the technology nine months after the publication of the RFC, which was a surprise because you know, that Apple is a bit secret. And so they told us during the standardization that they were interested by MPTCP but they never really told us that they would use it for Siri in 2013. And since then, they have used it for Siri and they have tuned it and improved it for Siri and knowing the importance of Quality of Experience for Apple, they have improved the solution, and if they had some problems with MPTCP they would have stopped that and used another technology. And so it means that it answers, it solves the problem that they have, and it gives good Quality of Experience for the users.

Other use cases

Brandon: [00:13:18] So that was 2013. What are some other deployments that have happened or that are starting to happen?

Olivier: The other use case besides Apple is that there are smartphones in Korea using MPTCP to do bandwidth segregation between wifi and cellular, because you know that in Korea, it's important to have the fastest possible wireless network and they manage to get more than one gigabits per second of.

Of bandwidth before 5g. So by linking 4G and fast wifi, it was really important for them. There is another use case, which is being deployed in Europe and it looks at the other side of the, of the bandwidth. So there are many places in Europe, but also in the U S and elsewhere where the bandwidth that you can get from an ADSL link is very low.

So a few megabits per second. And if you only get a few megabits per second, there are many applications that you cannot use. So you cannot do streaming with Netflix and other applications that demand more bandwidth. And in these areas, there are network operators that are using MPTCP to combine DSL and 4G connectivity. This is used with MPTCP proxy. This is a bit detailed; we don't need to explain that in detail, but at least MPTCP allows us to sense the bandwidth that is available on the 4G network, and uses it. Typically what the network operators do is that they will prioritize the mobile device over the 4G network and let the rural area users at home use the leftover bandwidth in the 4G network.

So MPTCP can adjust the bandwidth that it uses over the 4G network based on the available capacity.

A protocol - making its own choices about a path? How it really works.

Brandon: [00:15:03] That's really interesting, in that the protocol is making the choices about which paths to use. And I could imagine some paths are very different in properties, maybe much more consistent bandwidth, but lower total bandwidth, versus maybe more variable, like wireless, but potentially more available and possibly higher latency.

So can you give some insight to our listeners as to how MPTCP makes these distinctions of how to send traffic across links at a high level?

Olivier: So, typically what MPTCP does is that it sends data over the two paths and when you send it out, you get acknowledgements. And by looking at the acknowledgements for the data that you sent, you can, uh, determine what's the roundtrip time over each of the paths and you can adjust the bandwidth based on the round trip time and try to send the data over the fastest path which is the one which will have the lowest round-trip time.

Because if you push more data over a DSL link, then data will be placed in a buffer. And so the roundtrip time will increase. And so then you can move to the 4G network and they have different, round-trip times and different bandwidth characteristics, but MPTCP adapts to that very easily.

Brandon: And it's mostly the stopwatch time, right? The how-long-it-takes-packets-to-come-back...that's driving the choice of which link to send them on in any given moment.

Olivier: Yeah, and the congestion control. So the packet losses that you observe over a given network will tell you that the path is full and that you should send data over the other path

Deploying at home routers in Europe

Brandon: [00:16:20] Got it. Uh, can you tell us about some European use cases you might be familiar with?

Olivier: So this is one of the European use cases. So the rural areas are deployed in Europe, by the company that we created, which is called Tessares, and so today there are more than half a million people who are using the technology to get better broadband services in rural areas. And this is expanding in other countries.

Brandon: But this is totally different than how Apple's deploying it - this is at the home it's being deployed. Can you tell us a little more about that?

Combining Wireless and Wired, at Home

Olivier: [00:17:00] So it's deployed at the home. So there is software which runs on the access router that you have at home.

And there is software which runs on the server in the ISP network. And so we use MPTCP between your wifi router at home, and one server in the ISP network so that we can combine the DSL and LTE connectivity.

Derick: So the router sort of becomes a TCP proxy at that point.

Olivier: Yes, so the Router is a TCP proxy. And well - in fact, it's a TCP-to-MPTCP proxy because it converts TCP into MPTCP.

Derick: That's interesting there, you know, the end-to-end principle back in the day said, you want to push all the complexity of managing communication between two end points out to the end points. And this is sort of like that. You're pushing the choice of these multiple paths out to the edge of the networks.

Olivier: Yes, but not not to your laptop or not to your smartphone because your smartphone and your laptop are not multipath-capable, and so your laptop is not able to see that there are two paths available. And so you need a device which is aware of the existence of the two paths to be able to use them.

Brandon: So these are home routers that have a DSL connection and an LTE connection?

Olivier: Yes, there are different deployments. So you can get a home router with both a DSL and a 4G antenna, or you have different devices that [00:14:00] are connected using an Ethernet link. And so you have, for example, your wifi on the DSL router and then the DSL router has an Ethernet link to the LTE gateway. The traffic first goes over the DSL and then when the DSL is full, it overloads data over the LTE network using the Ethernet link.

Derick: Wow.

Brandon: So this is a way for ISPs to, if there's a local congestion or outage event on the wired side, transparently, from a user perspective, switch over to the wireless that's available and use what's there.

Olivier: So there are many ISP s that provide backup services to the users, by telling them, yeah, please buy the access router which has LTE connectivity. And basically what they do is that you have a, either DSL or LTE with these types of devices, because you can not use both simultaneously, but with MPTCP you use both DSL and LTE.

And so it allows you to aggregate the bandwidth, which is not possible with the other deployments where there are mainly backup scenarios.

Derick: This sounds awesome - this technology. I, it has been, it was my experience that when you add bandwidth, um, even when people think you don't need more bandwidth, when you add it, somehow people find a way to fill your network, and, and this is, uh, this is providing that opportunity. I wonder if usage goes up when, in the presence of MPTCP, compared to without it.

Apps you couldn't run before

Olivier: [00:19:51] Uh, yes. So obviously it allows users to have, um, access to some applications that they could not do before. So if you have a two megabits DSL, and then you upgrade it to a hybrid network where you have DSL and LTE you get 10 or 20 megabits per second. And with that you are able to do Netflix. And so if you are in a situation where you could only do mail and surfing, and then you switch to mail surfing, plus Netflix obviously, you increase the bandwidth consumption.

Brandon: And not just that, but you've enabled someone to take, say a call from home and trust that that bandwidth is gonna be there, even if there's an issue with the wired side and I've, I've definitely seen enough coworkers that have had local internet outages this week, or a business outage recently that gets in the way - if you really trust that connectivity is always going to be there, you might take the call you otherwise wouldn't.

Olivier: You can do home-working You can do video conferencing. Uh, you can also be in a situation where you have a good downlink from the DSL, but not a good uplink. And so with the LTE you can get a better uplink. And so if you want to stream video from your home, because you want to do teleconferencing, then this is a use case as well.

MPTCP in the Kernel

Derick: [00:20:52] It seems like this should be in things like OpenWRT and every Linux distribution, but TCP is part of the kernel, like, is that, is that a barrier you think to deployment or adoption?

Olivier: So, TCP is part of the kernel and MPTCP is getting into the kernel as well. And so there are, there are two implementations of MPTCP now. So there is one that was started in 2011 in our research lab and this implementation has evolved since Linux 3.2 or something like that. And now it's available on Linux up to 5.4.

And so we have followed. The evolution of Linux and we have adapted MPTCP so that it can run some different versions of Linux. But this patch has always been off-tree and not an official one in parallel. There is an effort from, ah Apple, Tessares, Red Hat, Intel was trying to, uh, who are developing now an MPTCP implementation that will be included inside the official Linux kernel. So there are, I would say in the middle of this engineering effort, now they have MPTCP which is able to use a single path inside the official Linux kernel but it's not yet able to use multiple paths, but this will happen in the coming months and with the coming versions of Linux.

So expect that next year you will get, uh,, Linux with Multipath TCP in your favorite Linux distribution. And so it would become available on servers. It will become available on Android phones and so on.

Brandon: That's really exciting because that eliminates a massive point of friction to companies deploying this and then users getting to adopt once the companies add it to the products they use.

Olivier: Yeah.

A Long Journey to a Kernel Implementation

Brandon: [00:22:45] That's a really long journey here.

Olivier: So the first kernel implementation was in 2011 or 2012. And it was part of a research project. So, uh, we were in a European research project. And, uh, so we had this idea of doing multipath TCP and in the project, there were several large companies and there was a call among the project. Partners saying, Oh, look, who would be able to implement that in the Linux kernel?

And the company said, Oh, this is too complex for us. We are not able to do that. And one PhD student in my group said, I can do that. And then he started to develop the implementation and then the implementation evolved, but it was written by researchers. And I would say between brackets [initially for researchers].

So basically what Sebastien did is that he took the TCP code and then did lots of if-then-else to say that if this is single-path TCP you do that. If this is multipath TCP, you do that. And so in the end, we had a patch of about 10,000 lines. And when you send a batch of 10,000 lines to the Kernel maintainers, they are not very happy to review this kind of batch.

And so now the effort is to rewrite everything, but, uh, with a set of patches that are smaller and then can be reviewed by the kernel maintainers because they don't want to risk any change to TCP. Given the importance of TCP in the linux kernel.

Brandon: I remember this implementation. I think it was around 2012. We're trying to reproduce experiments and one of those was going to be one of the recently published-at-that-time, MPTCP experiments to show you could really use all this bandwidth, uh, over an emulated link that had some delay and drop characteristics that were a little different.

And at the time you couldn't run one instance of MPTCP on the same kernel with different namespaces and this is the tech, one of the technologies that makes uh, Docker possible and containers possible - it's that every container can see its own view of the network, and the kernel has to be made aware of this. It's called kernel-style virtualization and, uh, in particular network, namespace virtualization. So I remember submitting a patch upstream your implementation to try to get that to be a little smoother, but things have changed a lot if a decade later, it's going to be part of the actual kernel or at least on a path to that. Props to that that's a big deal.

Olivier: It will be part of the official kernel now. That's for sure. With the involvement of Red Hat, this will happen.

Unexpected Stuff

Brandon: [00:25:10] Great. So let's, let's kind of divert things a little bit back. I think you're one of those people who can tell some of the stories that are most unique here, about the roadblocks along the way to adoption and some of the surprises and stories that might've

come up. Can you tell us about some of those, what got in the way early that wasn't expected?

Olivier: So as I said earlier, Mark Handley said that it should be easy to implement MPTCP and the problem should be solved in one year to do the specification and to have something. And so we had an implementation of MPTCP that was running in the lab and this was the first design, the first specification.

So it was running in the lab. And we discussed with partners of the European project and there was a university in Finland. We said, if you send us your patch, we will install it on our servers, and you will be able to do experiments between Belgium and Finland. And this looked nice because we were able to do experiments on the Internet.

And we were surprised, because when we tried to open the first MPTCP connection outside the lab, it turned out that the connection was established, but it was impossible to send data. And so we went back to the kernel to try to understand what was going on. And in the end we found that in our university we had an old firewall, but it's, I think it's still there, and this old firewall was changing the TCP sequence number of the initial connection establishment packets, because there was a bug a long time ago in Windows 95, and so the firewall was fixing the Windows 95 bug by changing the sequence number of all the TCP connection that were ongoing outside of the university.

Brandon: You can't see it, but a bunch of us are shaking our heads right now.

Derick: Like, this is a lesson that apparently we never, we never, learned.

Olivier: This happened. So we fixed the problem by changing MPTCP to take that into account and Michio Honda was working with Mark Handley at that time and Costin Raiciu developed a measurement script to try to understand what was going on with middleboxes on the internet.

And he did measurements, from 100 locations in a full measure of measurements. And he found that when you send a TCP packet through the network, there are middle boxes that can change any field of the TCP header on the given path. So you can not trust any part of the information that you send in a TCP header because the information that you send in the client might not be the same as the one that the server receives.

So you know that there are NATs that will change IP addresses and ports, but you have firewalls and transparent proxies that change TCP sequence numbers. You have dealt with change acknowledgements. You have Middlebox that will remove TCP options, that will modify TCP options... you have some middlebox for example, NAT, when you run it with FTP, it might even change the payload of the TCP connection because you have to fix the FTP protocol that uses the next key representation of the IP addresses.

And so you change the payload, but you also change the sequence numbers and so on. This is completely ugly, but this is deployed.

The Internet Is Conspiring Against Any New Protocol!

Brandon: [00:28:06] So the deployed Internet is conspiring against any new protocol.

Olivier: So the deployed Internet is conspiring against MPTCP and against any evolution of TCP but it's deployed and MPTCP works well.

Derick: Along these lines, I heard that, um, you've learned something interesting about China's great firewall.

Yeah.

Olivier: Ah yes, it was funny. So there was a paper published by a researcher, I think from Australia who tried to map the usage of multiple TCP on servers. And for that, they just took the IPV4 addressing space and they tried to create a TCP connection on port 80 on all IP addresses of the IPV4 internet.

And they looked at the answers that they received. And so in the connection, establishment packet that they sent, they used the multipath TCP option to indicate that this was MPTCP and they just looked at whether the answer included Multipath TCP. And when the paper was published we were really surprised to see that there was a huge deployment of Multipath TCP in China.

And at the time we had not been contacted by people in China saying that we [00:25:00] are experimenting except a few students, or a few researchers were playing with the code. And so we tried to look ourselves at what's happening in China. So we sent TCP packets to China with the MPTCP option, and what we found was that when you send the packet to the Great Firewall of China it always replies that it supports the extension that you propose, even if they don't understand the option.

Olivier: So to address that we check in MPTCP that the response that we get from the server is not the same as the information that we sent from the client. So the option that the server returns contains some information, which is called the token in the first version of MPTCP, and this token must be different from the one that was sent by the client.

If it's the same, then this is strange because this should not happen. And so we fall back to regular TCP and we assume that MPTCP has not been negotiated.

And there are lots of other tricks that are used in an MPTCP implementation to be able to cope with those middle boxes and all those tricks, they increase the complexity of the protocol, but they also increase the deployability and the usability of the protocol across the internet.

Brandon: And so with these changes deployed, is it now deployable anywhere or are there any other edge cases you've come across with bugs you have to deal with?

Olivier: We did not receive recent reports saying that MPTCP does not work in our environment. And basically what happens is that if you are in an environment where, for example, you have a middle box that removes all MPTCP options then MPTCP falls back to regular TCP. So the assumption that we did in MPTCP is that we wanted to preserve the connectivity.

So if you try to be connected to a server, you prefer to fall back to TCP instead of having a failed MPTCP connection. And so MPTCP handles that in a graceful manner, so that you preserve the connectivity to the server,

Brandon: Do no harm.

Derick: Yeah. Do no harm.

Multipath QUIC?

Derick: [00:31:34] Could QUIC help out in this situation?

Olivier: So QUIC helps a bit in this situation. So QUIC is a new protocol, which is being finalized within the IETF. Basically it takes all the mechanisms that you have in TLS and in TCP. So all the reliability from TCP and all the security from TLS and it puts everything on top of UDP. And with QUIC all the data that you send, but also all the information from the packet that you sent is encrypted with a small exception.

And it means that it's impossible for a middle box to interfere with the QUIC packets as we had with MPTCP. But on the other end, there are middle boxes that simply block QUIC because they don't understand this protocol.

Brandon: But do you have to modify QUIC now, if it's built on UDP in the same kind of way and involves a lot of the same code.

Olivier: So to, uh, to modify QUIC, to be able to support multipath?

Brandon: Yeah.

Olivier: So there are discussions on that. And so we have proposed a multipath version of QUIC three years ago, and it has evolved in parallel with the space, with the standardization of QUIC and yesterday, there was a discussion in the IETF on whether the IETF will agree to work on Multipath QUIC. And this is not yet decided.

Brandon: Yesterday being October 22nd for our listeners, 2020.

Olivier: QUIC was designed with the web as a use case—it was mainly pushed by the web companies. And so their main use case is to be able to retrieve small objects as quickly as possible. And having a multipath capability makes sense when you are in rural areas or when you have limited connectivity.

And there are not yet fully convinced of the benefits of multipath in QUIC. And so we will see in the coming months, there will still be discussions on whether they agree to include multipath inside QUIC, but it's not yet guaranteed.

Satisfaction and where this really matters

Derick: [00:33:39] Overall, oh this is like more of a personal question. You know, you started this project so long ago and you've gone through so many obstacles that you've told us

about the standardization process, middle boxes, all these things, and, um, you're now you're starting to see that it's being adopted and now it's going to end up in the kernel, how satisfying is it for you to see something that you've been working on so hard actually on the precipice of becoming normal, like the normal way of doing things?

Olivier: I think it's positive. I would say it's mainly positive for the students who worked with me so they contributed to, um, designing and developing and then later deploying a technology that affects everyone. So people who are using iPhones - they can understand the benefits of the technology, uh, without understanding how it works. I'm very proud of, uh, what's happening in rural areas, because it helps people to get much better Internet connectivity. And in the current days, it's very important for people who have to do teleworking to have better connectivity by using this technology. So I'm very proud of this part.

I would say that for me, this is more important than the deployment from Apple, because the deployment from Apple helps a bit when you are moving, but when you discuss with people who have used the technology to combine DSL and LTE, they said that, well, this has changed their usage of the Internet. And so, uh, you have, um, grandmothers who by using the technology, they can do video conferencing with their grandkids that live further away.

And this is a change for them. And these days, this is very important.

Open Roaming

Brandon: [00:35:20] And did anything have to change to make this work really well across multiple wireless technologies versus wired and wireless?

Olivier: So, whether you combine multiple wireless it's mainly a business decision. What needs to change is the authentication in the wifi network. And so there are, uh, ongoing efforts to do open roaming so that you can roam from one WiFi network and be able to authenticate it on multiple WiFi networks. And this means that there will be many more, uh, WiFi networks that will become available, and so it will become easier to switch from cellular to wifi networks. And so this is an important evolution, but this is outside of MPTCP so it's making sure that a growing number of people will be able to use a growing number of WiFi networks.

Derick: So - single sign-on for wireless.

Olivier: So that's open roaming and, uh, well we know that in university for a long time, we have a solution which is called eduroam, and eduroam allows any student from any university to use the WiFi network of any university around the world. And basically open roaming is getting that in the commercial world as well.

Derick: Wow.

Textbooks Here! Runnable, too!

Derick: [00:36:35] This has been a great conversation. You know, this has been very informative for me. I've heard of MPTCP, but I didn't know that much about it. Now, it feels like, This is going to be the way it's always done probably very soon.

You do have though, some other very interesting projects going on, and I think it's worth mentioning them because they're really very high quality projects. And one of them is you have this book called Computer Networking Principles. It's a whole textbook about computer networking that's totally free. It's on GitHub and anyone can go in and look at it. Um, can you tell us a little about that and where it's going?

Olivier: Yeah. So, it started almost at the same time as MPTCP - I had the opportunity to go on sabbatical. Uh, I said that I wanted to do something else. I wanted to, um, give back to the open source community by writing a textbook.

And so it started in 2010 and now it has, it has evolved and it's used by different universities. And so everything is available. Publicly available. So it's open source and creative commons and every university can reuse it, and now what we are doing is to, we are improving it so that it can become interactive.

So we are adding to the textbook Interactive exercises where for example,

you can, uh, look at, uh, packets with, um, a kind of Wireshark trace. And then there are fields that are missing in the packet trace, and we ask the student to fill the missing information, and if they know the protocol by looking at the previous packet or the next packet, they can infer the information that should have been in the, in the previous one. We are trying now to develop new extensions that will allow students to connect to a Linux machine, through the web and to, uh, change the configuration or debug problems in an emulated network so that you can play with that remotely and solve problems.

Brandon: That is exciting - interactive exercises that anyone can run from anywhere .

Olivier: So what we do is that for research and for education, we try to make everything open source and everything freely available so that people can reproduce what we do, or reuse what we do.

Brandon: And I see there's a URL for this computer-networking.info.

Getting out there - runtime protocol programmability

Brandon: [00:38:45] Awesome. it out. out.

All right. There's one other project. I've heard you're involved in. Can we talk a little bit about protocol programmability and eBPF.

Olivier: Uh, yeah. So, you might have heard of BPF which is, um, a new technology that is part of the Linux kernel for three or four years. And it allows you to inject code, which can be executed inside the Linux kernel. And it allows you to implement policies and implement

different things in the kernel. And this is getting more and more important in the Linux kernel.

So, what we do is that if you look at the evolution of protocols, and I think Derek mentioned that earlier is that protocols within the IETF, they evolve very slowly, but the requirements from the network operators, they evolve quickly and many network operators would like to have more flexibility in their utilization of the network and the utilization of the protocols. So, what we do is that we take a protocol implementation. One example is BGP the inter-domain routing protocol. So we take BGP implementation and we modify those BGP implementations so that they expose one API and it's possible to inject eBPF code to modify the operation of the BGP implementation.

And this allows network operators to be able to tune the BGP implementation to their needs. And what we have done is that we have taken two very different BGP implementations, open source, FRRouting, and BIRD, and we have modified them so that they expose exactly the same API. And so if you have a network which uses different vendors it could be possible to run your own programs on these routers that modify the BGP protocol to your needs and adapt the BGP protocol to your needs, without having to discuss with Cisco Juniper and spend years to convince them that they should change one bit in BGP.

Derick: Wow, that is significant. Unless you're a very large service provider with significant annual spend, getting vendors to implement changes-people don't even bother it's almost impossible.

Olivier: Yeah. It's Mission Impossible. Yes, and so if, if you look at the IETF, for example, uh, for BGP, uh, to, to have, um, an RFC published for the BGP protocol, it needs to be implemented by two different vendors. And we looked at the delay between the first ID for a BGP extension, the publication of the RFCs and the median delay is 3.5 years. - But the worst case was 10 years. And it took 10 years for the IETF to adopt the BGP extension so that you could use 32 bits AS numbers instead of 16 bits AS numbers, which was a change that was required because the Internet was growing and we were getting out of AS numbers. So we had issues, but it took 10 years to adopt that.

Brandon: So this almost enables runnable RFCs by enabling routers to have plugins.

Olivier: Yes. So it brings programmability to the routing protocols; if you look at what SDN brought and with Nick McKeown and others. So with SDN, you had the idea of programming the switches - so programming the forwarding tables on the switches - and here, what we do is programming the routing protocols, so that you can tune the dissemination of the routes inside your network, and you can keep the routing protocol, the distributed routing protocols as they are, and update the part that you need to update. And the same approach can be used for OSPF, ISIS, or other protocols as well.

Brandon: And where can people learn more about this, if they're interested in trying it out?

Olivier: And so again, there is, um, a website which is called pluginized-protocols.org where we provide information, and we have done that for other protocols as well. So Derick

mentioned QUIC, earlier, so we did that for QUIC as well. We have a quick implementation that you can extend by using this kind of eBPF code.

And what's nice with QUIC is that it's possible for the server to send an extension as eBPF code to the client over the QUIC connection so that you extend the client for a specific connection. So for example, if you are on a QUIC server and you find that the client is connected to a lossy network, and you would like to add forward erasure corrections, so that you can correct the errors without doing retransmissions, it's possible with that to ship to the client, the code that the client needs to use to be able to support the forward erasure extension.

Brandon: I've never heard of anything like that. That's runtime-level protocol extension.

BGP Can do anything. Anything?

Brandon: [00:43:10] That's cool. Okay. Yeah. What is the craziest thing you've seen in a network?

Olivier: Uh, the craziest, see that I saw was, uh, in, in the protocol and it's in BGP as well. And so there were, um, two PhD students in my group, Stefano Vissicchio and Laurent Vanbever. They came to my office one time and they said, ah, we found something cool with BGP, because using BGP it's possible to implement, by using special configuration, you can implement one bit of memory and you can implement logic gates.

And I said, ah, this is cool. So you should write an RFC, uh, saying on April the 1st, that BGP can do lots of stuff and you can implement electronics using BGP. They told me, yeah, this is cool. But you know, if you can implement memory and you can implement logic gates, basically you have all the computing power of a computer.

And so they work together with Marco Chiesa who is now professor at KTH in, uh, in Sweden. And they prove that BGP is equivalent to a Turing machine, which means that BGP is a computer in fact. And this was unexpected. And so well, you can read the paper. It's a very theoretical paper because you have proof that something is Turing complete.

So there is lots of theory, but the nice application is that you can build your BGP configuration so that you can memorize some information. So BGP is more than you would have expected.

Derick: Olivier I have actual work and adult obligations in my life and I'm going to ignore all of them now so that I can implement some kind of logic using BGP inside of, uh, like Eve-NG or GNS - that's pretty awesome. That's - that's like extremely geeky stuff.

I got to find that paper.

Brandon: The internet is not just a series of tubes. The internet is a computer.

Closing Out

Derick: [00:45:15] I think we'll be able to attach to, um, the podcast for those that are listening, the various links we've talked about during the show, and we'll even link to this paper that turns BGP into a computer. If you haven't remembered or you're sitting on a train somewhere, don't worry about that - when you get where you're going, you'll be able to find these links and look at these things.

Brandon: All right. And before we step away, one question, whose story would you like to hear next on this podcast?

Olivier: Oh, I guess it would be interesting to listen to Mark Handley on his work on low-Earth-orbit satellites, and, uh, the performance that you can get from ~~this camp~~ of these types of networks.

Brandon: Oh, that sounds like a completely new area of networking. That would be really interesting to talk about. Awesome. Well, I want to thank our speaker Olivier Bonaventure for spending your time with us today, this was fabulous. This covered all kinds of areas. We talked about protocols running the internet and all the things you can do with it you didn't know - all the way to fundamental transport protocols that you've been working for almost a decade now to change that are going to be deployed ubiquitously within a year. That's pretty exciting.

Olivier: Yep. Thank you for the discussion.

Derick: Yeah, thank you.