TSVAREA@IETF98
Session 1 and Session 2
Monday, March 27, 2017

Chairs: Mirja Kuehlewind and Spencer Dawkins (TSV ADs)
Minutes provided by Phil Eardley, Dave Dolson, and some anonymous folks in Etherpad (please let us know that was you, and we'll add your names here).

Beneficial Functions of Middleboxes - Dave Dolson

David walked through the table of contents from his draft, and gave troubleshooting choppy video and verifying firewall policies as examples.

The only TCP header field David hasn't found a use for is Urgent Data. Operators use everything else.

Spencer noted that this presentation is for an operator to tell us what they are doing, not for us to tell an operator what he should be doing.

Ian Swett Google:

- Thank you for starting this work.
- It would be good to distinguish between what middleboxes are doing at the TCP level, what's at the IP level, what's also in UDP, for example.
- Do you care about bandwidth or how the bandwidth is being delivered?
- You need some data to back up some of the claims.
- What are the critical things to measure when managing a network, and what's not critical? We're facing this in QUIC now.

Lars Eggert:

- You're making an implicit assumption is that the only way this info can be provided to operator is through middleboxes. That's not true. For which of these metrics you've identified, what alternatives exist? e.g., telemetry. Look at SNMP or YANG, or inject traffic to help troubleshoot.
- Middleboxes may be necessary for some purposes, but we should at least identify those.

Mirja:

- To keep in scope, this is documenting current practice.

Lars:

- I disagree that this is all needed.

Mirja:

- No, the point is just to document existing practice.

Tom Herbert:
- How is packet delivery improved by using these features?
- From application point of view, when I send information, I don't know how those things are improving performance.
- I know I feel better and more secure when I encrypt everything.

Dave:
- You feel more secure, but you're not more confident that the network will deliver.

Matt Mathis:
- Passive measurement is valuable.
- Caution about performance enhancing proxies: they cover up TCP bugs and make the network stick at certain points in technology.
- With modern TCPs, proxies almost always hurt performance.

Dave:
- Yeah, proxies don't always behave properly, but buggy anything is buggy.

Matt:
- Key point is that end systems are what allows us to evolve the network.

Eric Rescorla:
- This was introduced as a descriptive document, but these seem like normative statements. Many of these are negative to the user experience.
- You should say what is done, not that there are benefits.

Dave:
- We often think we're doing this for the benefit of the user.

Eric:
- I'm sure you do, and I can disagree.

Gorry Fairhurst
- We need to recognize that PEPs covers a whole range of functions.
- As a transport guy, I want to transport packets. If you just want to move bits, that's one thing. If you want to do it well, you need some of these devices.
- My experience with an IPSec network was that encryption made the network very difficult to manage or to know if it was working properly.

Erik Kline:
- You mentioned network coding, but if we document what people do, "deny non-sane packets" is not universally agreed. The definition of sanity is not consistent.
- "Sanity" may also mean ossification.

Benjamin Schwartz:

- You need to document repressive censorship, as you're documenting what middleboxes do.
- If this is a value-free assessment of what currently exists, then it should include how these boxes are used for repressive censorship all over the world.

Jake Holland:
- If you're documenting current practices, it doesn't make sense to exclude some uses.

Mirja:
- Documenting problems is interesting - why operators are motivated to deploy middleboxes.

Aaron Falk:
- The PILC working group produced "Advice for subnetwork designers" - old rfc, RFC 3418.
- It seems like this doc going same way. The document is a way for operations to educate the IETF, and vice-versa.

Transport-Independent Path Layer State Management (draft-trammell-plus-statefulness) - Brian Trammell, remote

Jake Holland:
- In draft is the association implicitly assuming packets are part of the same bidirectional flow?

Brian:
- In draft, there is an abstraction about how to associate a packet with a flow. Traditionally, 5-tuple.

Jake:
- What about multicast, using IGMP?

Brian:
- Sure, should work.

Eric:

what's relationship with QUIC? should align, as this is the UDP transport being developed
Mirja - was earlier presented to QUIC, they aligned

Eric Rescorla:
- What's the relationship of this to QUIC?

Brian:
- QUIC already provides most of these signals. But it's generally useful for a UDP to get out of this short idle state. With QUIC, you can automatically resume, so it may not matter. On mobile, it may.

Eric:

- We're not designing a whole lot of new transport protocols, we're designing one. So weird to give guidance that is different than the one we're working on.

Brian:

- Two things discussing QUIC are how to give start/stop signals.

Eric:

- Is it okay for these mechanisms to disagree or must they align?

Brian:

- Must align eventually or it's silly.

Mirja:

- [sorry, got distracted] But the benefit is for QUIC to point to this draft.

Ian Swett:

- The FIN signal, silent close, intentional or not, exists.
- Does it need to be explicit?

Brian:

- Do you have data on how much TCP silent close you see? That would be useful to know.

Matt Mathis:

- Some of that data exists in some of the errors, and the reporting is oddly geographically distributed, suggesting that it's only specific manufacturers doing TCP silent close.

Tackling privacy and confidentiality of communications: EU's ePrivacy Regulation - Diego Naranjo, Remote

(European Digital Rights - https://edri.org)
Diego provided an update of recent EU privacy regulations.
This was for TSVAREA's benefit.
We had no questions or discussion afterwards.

Tor's Perspective on Privacy and Traffic Analysis Resistance for Encrypted Protocols - Mike Perry

Mike provided a deep dive into Tor technology description.

Ian:

- Clarification about clock rate?

Matt:

- How does Tor handle DNS?

Mike:

- That's mostly outside Tor

Eric:

What about Tor's support of latency sensitive traffic?

Mike:

Tor can do voice but not video - delays not too terrible

eXpress Data Path (XDP) in the Linux networking stack - Tom Herbert

Tom provided an update on implementing userspace protocols.

- Benefits include performance and ease of programming/error handling
- Hard to get a general solution; some implementations grab packets into user space and then re-inject into the kernel
- Maintaining two stacks is a pain
- More proprietary changes

What we need is programmability in the kernel and everywhere. Particularly an issue in the kernel.

We want an open, flexible solution for extending the networking stack. This was the original idea that led to BPF. BPF was limited in what it could do, and didn't get more fully used for years.

eBPF in 2016 to make it 64-bit, make it compile into safe bytecode, etc. The compiled BPF code that runs in the kernel disallows common errors and dangers. This gives the safety of userspace while still being in the kernel.

The eXpress data path runs eEBF very early in the stack to prevent attacks. It deals with the raw packets off of the network. It doesn't do anything fancy with memory allocation, so is quite lightweight. Drivers call directly into BPF program, where packets are dropped, forwarded, or received.

XDP is high-performance, programmable, not a kernel bypass and does not replace the stack, and works on generic hardware. Shows hardware-limited performance on Intel chips, etc.

Future plans are to enable more drivers and hardware; improve performance gap for DPDK.

Discussion on Congestion Control(Michael Welzl, Praveen Balasubramanian)

Michael provided background on the way the transport community has been evaluating congestion control mechanisms for some time.

ICCRG was created because there were many experimental algorithms, that didn't have time to go through TCPM. One goal was to get discussion on ideas that likely wouldn't work. They haven't completed many documents, but have published some RFCs.

Should CC work be spread around the IETF as it is, rather than being centralized? Perhaps it's important that the various groups (RMCAT, MPTCP) have their own work.

Jana:

- As another holder of the ICCRG trashcan, we have Cubic in TCPM, and BBR that's been presented in ICCRG a few times, and will need to land somewhere. QUIC also has congestion control. It feels like CC should be separate from protocols.
- Should we separate CUBIC for TCP and CUBIC for QUIC drafts? ICCRG should be protocol-agnostic, and we should consider how we do this in the future.

Matt Mathis:

- The IETF bought into "TCP friendly," but what we have is Reno, and the Internet has changed. We should be thinking "no longer relevant" instead of "TCP friendly."
- How do you say "reasonable behavior in congestion" and "no congestion collapse" and so many other things that should be here, without calling it "TCP-friendly"?
- We need to replace "TCP-friendly" dogma with some other protocol-agnostic description. Academic reviews reject papers which do not address the TCP-friendly issue, and we are thus losing out on that research.

Michael:

- We do have one document saying it's about avoiding starvation of flows.

Matt:

- Avoiding pathological behaviors is a better description and aspiration than "TCP-friendly," but academic reviewers are forcing people to address "TCP-friendly", instead of newer descriptions.

(Praveen presenting)

- Michael talked about operations, Praveen talked about implementers.
- Currently, Google/Apple use Cubic, and Windows does CTCP. There's also DCTCP/DCQCN, and BBR, Timely, being used more in more specific areas.
- Academia has PCC, Sprout, Remy.
- Trends go towards operators using AQM, but AQM is available.
- Especially given the movement towards QUIC and other user mode transports, there could be many CC algorithms sharing a link.
- Bufferbloat is a real problem, and RTTs go up with load.

Questions:

- How do we ensure fairness when there are multiple CC algorithms on a shared link?

- How do we prevent an arms race between CC algorithms? Cubic vs CPTCP, etc. What does it
- mean to be “TCP Friendly”?
- What is the role of the IETF? RFCs on Best Practices don’t seem to be enough.

Aaron Falk:
- You say informational RFCs are insufficient. What else should we do?

Praveen:
- We need TCP fairness, but other fairnesses as well.

Aaron Falk:
- One of the asks from IRTF to IETF when ICCRG was created, was for the IETF to come up with a way of fairly comparing algorithms. Our criteria has changed over the last couple of decades but this has not been well articulated.
- When we created ICCRG, we wanted a way to know that a CC algorithm was safe. People kept showing plots showing their algorithm worked great in isolation, and somebody else showed theirs working better than others in isolation. But many algorithms weren’t realistic for the Internet, and they can't all be best for all things.
- Our criteria for evaluating transport protocols has changed; we should tell the research community.

Michael:
- The TCP evaluation suite has been around for a long time but it's very old, has not been well maintained due to lack of cycles, and is now very outdated. Should be taken and updated, although there are no volunteers!

Michael:
- Also TCP evaluation suite, very old and not enough cycles to maintain it well.

Mirja:
- How do we provide updated guidance?

David Black:
- ICCRG as a central clearinghouse for CC issues is very valuable. We deal with congestion concerns in TSVWG; and have SCTP, UDP, etc, etc. I agree that this is worthwhile work, and we should continue it in ICCRG.

Mirja:
- The question isn’t whether CC evaluation should continue in ICCRG, but whether the IETF can provide ICCRG with a new ecommendation.
- The current recommendation is to compare against, and updating to use CUBIC will be our new recommendation when it's published as RFC. Because it is so widely deployed, we know CUBIC is safe, but it’s not state of the art.
- How can we update the IETF's recommendation?

Lars:

- The slide of questions from Praveen could have been from 10 years ago when Cubic was new.
- All papers or drafts needed to compare themselves to the other algorithms and benchmarks. It's boring work, but important. If you did not run your publication through the TCP evaluation suite it would not be accepted.
- As networks get faster, the simulations are not valid any more. We have no way of assessing large-scale data centre cases. We need more test cases and toolings to help researchers.
- QUIC has lots more information about the path which could be very valuable for researchers and open up new areas for research in ICCRG (not QUIC wg).

Michael W:

- This seems unfeasible due to the amount of work, and volunteers to do the benchmarking.

Christian Huitema:

- When you are designing a CC algorithm, you want the best utilization of the network. But there are times when you're competing with algorithms that are not nice at all. So you have to design software with nice mode and pig mode. (side chatter, and laughter; pig mode = CUBIC?) That gives you new problems, such as, how do you get out of pig mode, when pig is gone?
- We don't recognize it now, because we don't recognize that we have pigs.

Matt Mathis:

- Why be friendly to TCP when it won't return the favor?
- One of the things that stopped me from being so conservative about this stuff is that on commercial networks, it works out to "Which customer won't be able to get HDTV during prime time?" And it works out so that in most cases, the answer is "none" in the core, and all congestion is local edges.
- What you care about is having the same behavior at the local overload. But research community is still working on solving irrelevant problems. We don't have a good way of identifying whether a protocol is safe. It's a complicated problem, but it's what we care about.

Matt:

- Business decision - which of your customers do you want not to be able to run HTTP during prime time? And if that's not a concern, then congestion control is not a big deal for businesses because single flows can't make a big impact, local overload is all that happens, and that's where you need the sane behaviour.
- Lots of people are solving irrelevant problems still (i.e. irrelevant to today's Internet). Is a protocol safe? We do not have a good definition.

Mirja:

- Another reason work on test suites didn't proceed is that so much content is distributed on so many servers. If we make the test suite as realistic as possible, most flows are short flows and CC doesn't help with short flows.
- Nothing available could help in my implementation experience.
- RMCAT has some test cases, but they are simple and artificial. They figured, if you do these tests you might be safe enough to test on the Internet.
- We should use status: experimental to do experiments (in the real world) - it is not a recommendation.
- We should use Experimental RFCs to codify our experiments, and make it clear that these algorithms should not be used generally.

David:

- The IETF is looking for simple rules not to screw things up.
- Saying "safe" in context of CC means not getting in each other's way. In my world, what are the minimum guardrails that have to be put on non-CC'd traffic so it doesn't break CC'd traffic?
- We should have a basic set of guidelines of how to not screw things up.

Jana:

- Having something as easy to implement as RENO is actually valuable. Our basic recommendation should be simple to understand and implement. This is why Reno is sticking around for a long time--it's easy to implement and hard to get wrong. That's a healthy place to be. That's why we're going towards BBR, etc.
- Test suite just not valid in a few years down the line - don't feel it's valuable. This doesn't scale; it'll take 3 years to implement, and will be out of date by the time it's done.

Hannes Tschofenig:

- No one has time to analyze all of the crap people come up with; but luckily there aren't too many proposals. We want to say "no", but have a good standard to why the answer is "no".
- In practice, you have to be a big player who can deploy something and then come back later and say "it works pretty well". The discussion is a bit artificial for that reason.

Matt:

- It's trivial to write an application that causes congestion collapse. Anything that causes regenerative load can have a space that causes congestion collapse. DNS can cause collapse when the server is beyond the retransmission rate. It's not fixable in DNS, but it shows that it's very easy to have simple TCP implementations and cause collapse. These need to be viewed as transport issues.
- Should there be a simple test of this scenario that can be used to test protocols, instead of TCP-friendly?

- We can make a test suite to look for regenerative load. This could be a measurement-based thing, to analyze small issues in state machines. It may have a test that we can run against all algorithms in an RFC; or it may be an unbounded problem.

Lars:

- You should keep an RTT sample if you can, and if it gets kept at one message outstanding, that can help the UDP cases and similar cases.

Jana:

- CUBIC is being selected because it's being used by a lot of companies and the Internet hasn't broken. It qualifies as BCP—or perhaps MCP, “Most Current Practice”, since it isn’t the best. It should be documented, even if it isn’t the best thing out there.
- No CC work gets standardized—they all go to ICCRG to get documented. Why standardize? Just document.

Mirja:

- IETF consensus is good to define which things are safe.

Jana:

- But Cubic isn’t safe!

Mirja:

- It’s safe to the degree that the Internet didn’t melt.

Colin Perkins:

- There’s a lot of important folklore knowledge being shared in this room that isn’t written down anywhere. It should be published so it can be referenced.

Jana:

- We should not have a standard we recommend; things are moving so quickly.
- Why even standardize CC? It's always evolving: let it.
- Jana: There’s also the DCTCP draft that’s in TCPM. The CC algorithms have the same people working on them in all of the meetings, even if they are in different groups. I think it’s okay to leave CC where it is, which is an evolving place.

Matt:

- We now understand control frequency must not scale with bandwidth.
- CUBIC overshoots badly at times, and people competing with CUBIC get hit badly.
- The problem with CUBIC is that the control frequency does not scale with the data rate.
- CC is going to keep changing for a long time, and any time the ink dries on a draft, it will be out of date.

Tim Shepard:

- The audience of the RFCs, who want to understand how the internet work, as engineers, building a new TCP implementation; they should probably implement something for CC

other than what's in the original RFC 793 (which is nothing). So it's useful to have a default pointer to a SHOULD algorithm, just to have something.
- The fact that we point at Reno is not a terrible thing.

Qiaobing Xie:
- Looking at the slides that document the state of congestion control, it looks more like a social problem, of the companies deploying competing solutions.
- Discussions were evolving around collaboration in the old days; that's where "TCP friendly" came from. Reality now is that it's a competitive Internet, not as cooperative. It's harder to talk about being friendly." We know there will be pigs; what if we identify which CC will be most robust in the presence of pigs?

Mirja:
- We see very limited deployment of new algorithms because people don't know what to use!
- One recent new case is WebRTC since it's new; other case is big company able to do lots of experiments.
- If you have a big company, you have the resources to do the large scale experiments. If you're not in this situation, the best you go to is New Reno, and you realize that it sucks on today's Internet, and people think "TCP sucks" and want their own new protocol. We want a better guidance.

Lars:
- Who can write a new stack and affect the Internet? Almost nobody.
- There is no arms race since there's nobody who can start this and benefit. It's in everyone's interest not to have an arms race. If BitTorrent breaks Skype, BT gets the blame.
- We want to have something that people can start reading and learning. NewReno is not great but not terrible and probably best choice available.
- Compare transport to security. They say "no" to things. Cipher suites are deprecated, protocols are changed, etc.
- CC algorithms are like crypto algorithms. Some are more specialized, and they change what the best practices are, over time.

Matt
- If you're selling advertising, and you click on the ad, and it breaks your video or something, that looks very bad - so there's a penalty to being too aggressive here. People should and will be cautious.
- As for moving away from Cubic, we won't have fairness, but we never had fairness. It matters less and less.

Colin:

- It’s Important to note that there are some areas, e.g. WebRTC, where the world has changed and we don't know how it will work.
- Following up with Lars, I think the issue is less between TCP CC, and accidental interactions between TCP and multimedia CC. We don’t have a good idea of what’s happening there.

- Tim: One challenge of understanding the Internet is that it changes by the time you understand it. I am used to the fact that there is this change.
- “Apricot 2017” video by Geoff Huston, about the “Death of Transit and Beyond”. If you want to connect to the Internet, you connect through someone who gives you transit. In a few years, you may not care if your connection has transit or not.
- I recommend that people watch it as you are thinking about CC, and what you believe about the Internet.

Michael:

- We have a quiet ICCRG list, and there is clearly interest! Please send your thoughts!