Tuesday, 19 March 2018

Chairs/ADs: Spencer Dawkins, Mirja Kuehlewind
Room: Sandringham
Note taker: Magnus Westerlund

* Administrativa - TSV ADs

- Note Well, Blue Sheets, Jabber Scribes, Agenda Bashing

No agenda was bashed ...

- TSV Overview and status

Thanks to TSV-ART, and to the TSV-ART triage team. These reviews really help - both Mirja and Spencer have balloted on documents starting with TSV-ART reviews in the past IETF cycle.

TSV Working groups Update

A lot of documents published, and a lot of working group drafts actively being revised, and thank everyone for the good quality.

We do have at least three working groups that are very close to concluding - ALTO, TCPINC, and TRAM.

- Related work:

  - draft-gont-6man-address-usage-recommendations

This draft is being chatted about in TAPS, because it's relevant to TAPS path selection (based on transport protocols supported, address families supported, security mechanisms supported, and other stuff), but should be of interest to a number of other working groups ("if you think you'll be choosing between IP addresses, you should probably be involved in this discussion")

* Experience with TCP encapsulation for IKE - Tommy Pauly

The ADs asked Tommy to talk about this, because we're seeing more people using TCP as an encapsulation/tunneling mechanism (not just UDP, these days)

Apple is investigating this for NAT traversal, because UDP encapsulation usually works, but they see 5-8 percent complete IKE failure when using UDP. Other people are using something like this, but nothing is standardized. Not just the enterprise remote access/VPN use case, but also captive portals in hotels, hotspots, etc. so more visible to the community. The more you look like HTTPS, the better things work.

They're using TCP to encapsulate both IKE and ESP, multiplexed on the same TCP connection. A magic number is used at the beginning of the connection to avoid collisions with non-standardized usage of the same TCP port number (usually 4500, but can be anything)/

TCP encapsulation isn't a perfect plan, because you're introducing head-of-line blocking if you encounter losses and reordering, and you can get large bursts of traffic, and if you're encapsulating TCP traffic (that was using TCP with ESP), loss recovery at two layers of the protocol stack can lead to interesting window size interactions, like doing slow start with a traffic stream that is, itself, doing slow start.

TCP encapsulation works, and that's better than no connectivity. If we do more of this, it would be interesting to experiment with TCP mechanisms on the outer packet headers.

Vladimir Olteanu of University Politehnica of Bucharest: Have had certain experience running TCP with ECN markings within the tunnel when congestion is encountered on the outer TCP connection. That's not perfect, but it helps.

Tom Herbert: Was worried this was sticking a TCP header on something that didn't behave like TCP, and using it as a tunnel. But this is sending the packets on the byte stream created by the outer TCP connection.

Did you see problems with head-of-line blocking? Not really, but we didn't have many parallel streams in the TCP tunnel - it would be worse if more streams were being blocked.

Stuart Cheshire: Jana Iyenger's Minion supported better out of order performance, but it didn't give much performance boost. This was because most applications require data in order, like video streams, spreadsheet etc. Thus, there is little benefit.

A lot of thinking about this goes back a long way. Without fast retransmit, you take many RTTs to recover, but with fast retransmit only a single RTT is really needed for repair with many applications, and a one-RTT buffer isn't asking too much.

Lars Eggert: have you seen improvement in the UDP connectivity? QUICs deployment may have forced improvement in this area over the past three years.

Tommy: we really haven't seen a change from the UDP blockage numbers we heard about in QUIC discussions three years ago.

Luigi Iannone from Telecom ParisTech: If a protocol is designed to use UDP, there is usually a reason, right? Are there any negative impact on the protocol from using TCP as an encapsulation?

Tommy: Yes. All of what we're talking about here, would be potential impacts on a UDP-based application protocol.

Luigi: Would there be a benefit of turning off congestion control for this purpose of getting through firewalls using a TCP header without TCP behaviors?

Tommy: well that has lead to this cases of sticking a TCP header on a packet with .

* Discussion: Using TCP as encapsulating to pass through various middleboxes

Mirja: we're seeing more and more proposals to use TCP encapsulation for tunneling - not only IKE, but TRILL, and CoAP, and some others. Is TCP encapsulation the right approach forward? Do we need to provide further guidance?

Tommy Pauly, this work done in Sec area. It would have been good to have a guidance document. We write a consideration section, it would have been good to have RFC to point to.

Brian Tramell. No, TCP encapsulation isn't the right approach, but Yes, we should give people advice about what happens when they do that. You've already done most of the work. Can we use your document as starting point to a general guidance document? If we say No, then people needing it will do even worse things. We should do this.

Tommy: yes, this could be a starting point.

Mirja: if UDP encapsulation doesn't work and we recommend against using TCP encapsulation, what's the alternative?

Kyle Rose: What does running over TCP indicate to the inner protocol? Like it is hiding the packet loss, thus hiding the congestion signal. Can one provide the signal?

Tommy: this has some relation to the ECN in tunnels document.

Lars: TCP info does exist in some APIs, but it requires polling. It can provide the info. But how does one expose it?

Jake Holland: One wonders why middleboxes are blocking UDP. If TCP encapsulation becomes the workaround, will we be required to do this as TLS over TCP to prevent ossification?

Tommy: Hopefully not.

Jake: is there a way to provide suggestions for standards to middlebox vendors? (nervous chuckles in the room)

Mirja: yes, but you don't know if anyone will respond to what you write in an RFC.

Chunsan Xiong: In performance parameters you only look at the loss rate on throughput, how did this affect the delay? There are some applications where latency matters. How does different ways of providing the information or responding to impairments. Is there some way to provide real-time applications over TCP encapsulation? That is not obvious.

Tommy: yes, agree that in certain cases, the delays you get from TCP recovery would be visible.

Spencer Dawkins as AD for TAPS: If we're headed this way anyway, when we chartered TAPS, we thought it was close to research, but they've made progress. Now TAPS has rechartered to look at things like transport security when doing path selection. As the responsible AD for TAPS, I wonder when I'll see a recharter request for TAPS saying that we might need to look at additional information like loss rates to select a path.

Spencer as individual: I really appreciate the insight that running reliable transport protocols independently at multiple layers of the stack probably isn't as bad an idea now as it was when people were running TCP without Fast Retransmit over x.25. And since QUIC has state-of-the-art congestion recovery behavior and runs over UDP, I'm waiting for the day when someone tries to run QUIC over UDP in a TCP encapsulated tunnel and seeing how that that will work. (Tommy whimpers quietly into the microphone, and then gathers his strength to continue)

Stuart Cheshire: It's important to remember why we have short timeouts for UDP. The middleboxes are providing real services, like blocking unsolicited traffic to your mobile so random hosts on the Internet can't drain your battery by sending junk. For TCP the FIN bit does state cleanup, so the middlebox can rely on that and use longer timeouts, but for UDP there is nothing. I wonder whether QUIC would provide a FIN bit?

Brian Trammell: it has been discussed in the QUIC working group, and there's an issue filed, but what the applicability draft says now is "you're going to see short timeouts, so if you don't have anything to say for 30 seconds, that's what the QUIC PING packet is for". This is not seen a significant problem.

Kyle: what if you used the state machine for TCP without the TCP recovery mechanisms?

Stuart: Minion showed that anything claiming to be TCP needs to be TCP. You have to mimic TCP so closely that there was no real benefit. Middleboxes know TCP doesn't have gaps in its sequence numbers, so when Minion decided not to retransmit stale segments, the middleboxes saw that as bogus and tore down the connections. TCP recovers within an RTT now, so you might as well be TCP.

Jake Holland: There was a proposal for PLUS, that got a mixed reception. Do we need to revisit that proposal, based on this feedback?

Spencer: "Mixed reception" is a fair statement.

Tommy shrugs and smiles, but says nothing.

Mirja: I would love to, but that has a certain deployment effort, and won't happen tomorrow.

Lars Eggert: 0-RTT the re-establish is fast, but there is an issue when you want a connection to be open for receiving. PING drains battery. An improvement would take a long to deploy. Doing a discovery will take time.

Tommy: Right, our IKE over TCP work happened because we wanted to receive incoming requests. We have MOBIKE for people who want to send requests - that establishes a path quickly enough.

Lars: you have the problem of knowing what keepalive values you should be using on a path, and there's no way to discover those except by observing what works and what fails.

Spencer: This would be a fine topic for a HotRFC lightning talk …

Brian: for the deployment lifetime of HTTP/2 over-QUIC v1, a FIN bit may not be the right thing. For future application over QUIC it may matter more.

Mirja thanked the room for not freaking out when we put "TCP Encapsulation" on the agenda.

Tommy was volunteered to start looking on TCP encapsulation and Brian Trammell promised to help out.

* Open mic

Nothing brought up. Apparently, everyone is delighted with what happens in TSV.

The ADs wished participants safe travels home and safe travels to Montreal.