TCP Encapsulation Considerations

draft-pauly-tsvwg-tcp-encapsulation-00

Tommy Pauly & Eric Kinnear
TSVWG
IETF 102, July 2018, Montreal
Document Motivation

Presented strategy for TCP encapsulation of IKE and ESP in TSVAREA meeting at IETF 101

Interest expressed for having a document to codify the pitfalls and concerns with TCP encapsulation

Document is meant to be an informational reference for other specifications that support tunneling over TCP
Non-Goals

This document is not intended to:

• Encourage TCP encapsulation for generic use cases

• Define modifications to TCP to add unreliability, or anything beyond tuning

• Define a specific encapsulation format
Motivations for Encapsulation

Why does anyone want to do this?

• Networks blocking UDP traffic
• NATs timing out UDP mappings
Encapsulation Formats

How should messages be encapsulated over a stream?

Suggests a Length-Value format:
Encapsulation Formats

After length, may add a type or demux value to handle heterogenous datagrams

Format should be generic across stream protocols (TLS/TCP) not just raw TCP

TCP encapsulated protocols will need to allocate a port; suggests parallel port from UDP when available
Performance Concerns

Document discusses concerns and mitigations

Adding TCP as an encapsulation or tunneling layer can introduce many performance concerns

- UDP-in-TCP adds unneeded reliability
- TCP-in-TCP (for tunnels) adds additional concerns
Performance Concerns

TCP-in-TCP

“Outer” tunnel

“Inner” flows

1. Inner TCP experiences Outer losses as delay
2. Congestion window interactions and Bufferbloat
3. Head-of-Line blocking
Performance Concerns
Loss Recovery

Concern
Inner observes loss on Outer as delay

Mitigation
Use delay as input to congestion control
Minor loss is hidden from Inner

*Inner TCP often needs to continue beyond tunnel*
Performance Concerns
Congestion window interactions and Bufferbloat

Concern
Bursts of packets delivered to other side of tunnel
Multiple layers of slow start

Mitigation
Avoid spurious retransmissions as much as possible
Performance Concerns
Head-of-Line Blocking

Concern
Major timeouts affect all Inner flows
Loss outside tunnel cannot be recovered while the tunnel is blocked

Mitigation
Increased timeouts on Inner flows can give Outer flow time to recover from losses
TCP Encapsulation for IKEv2

IKEv2 and ESP use UDP port 4500 generally

TCP Encapsulation sends those messages over a TCP stream on 4500 (but others ports can be configured)

TCP stream begins with a “Stream Prefix” of magic bytes to validate the protocol against previous non-standard uses of TCP 4500

Each datagram is framed with a 16-bit length field.

IKEv2 packets are distinguished from ESP by the first four bytes being all zeros (from UDP encapsulation)
Concerns with TCP Encapsulation

Packet loss induces large bursts, especially for a tunnel that may have inner TCP flows retransmitting.

Running TCP within TCP leads to window size issues, such as going through slow start both on outer and inner connections. Collaboration between outer and inner TCP would help.

Added head-of-line blocking between flows that were independent when using UDP.
Performance Tests

Setup:

Standard IKEv2 VPN server
Relay box in front of server, decapsulating TCP stream
Client modified to send IKEv2 and ESP packets over the TCP stream
Run multiple iperf TCP flows within the tunnel

Variables:

Encapsulation: ESP, ESP over UDP, ESP over TCP
Fixed random loss (0-3%) induced with Cerowrt router
Delay (0-500ms) induced with Cerowrt router
Performance Tests

Loss

Throughput (Mbits/sec) vs Induced Packet Loss

- No ESP Average
- ESP Average
- ESP Over UDP Average
- ESP Over TCP Average

TCP Encapsulation - TSVWG - T. Pauly, E. Kinnear, Apple - IETF 102
Performance Tests

Delay

Throughput (Mbits/sec)

Induced Delay (seconds)

- No ESP Average
- ESP Average
- ESP over UDP Average
- ESP over TCP Average
Conclusions

TCP encapsulation works, and is certainly preferable to no connectivity for UDP-based protocols.

Performance is tolerable, and degrades at roughly the same points as other tunnels (may be pathological cases, however).

Tuning the TCP connection used for encapsulation would likely improve its performance.