TCP Encapsulation

Experience and experiments with IKE over TCP

Tommy Pauly
Transport Area Open Meeting
IETF 101, March 2018, London

Why TCP Encapsulation?

NATs and Firewalls notoriously treat UDP traffic badly

- ESP uses 20 second keepalives with NATs
- As of 2015, error rates of 3-8% seen with UDP
- Mainly blocked on captive networks or enterprise networks

TCP gets through networks with higher success rates (especially if the traffic looks like port 443 traffic)

TCP Encapsulation for IKEv2

RFC 8229

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 an inner TCP would help.

Added **head-of-line blocking** between flows that were independent when using UDP

Performance Tests

Setup:

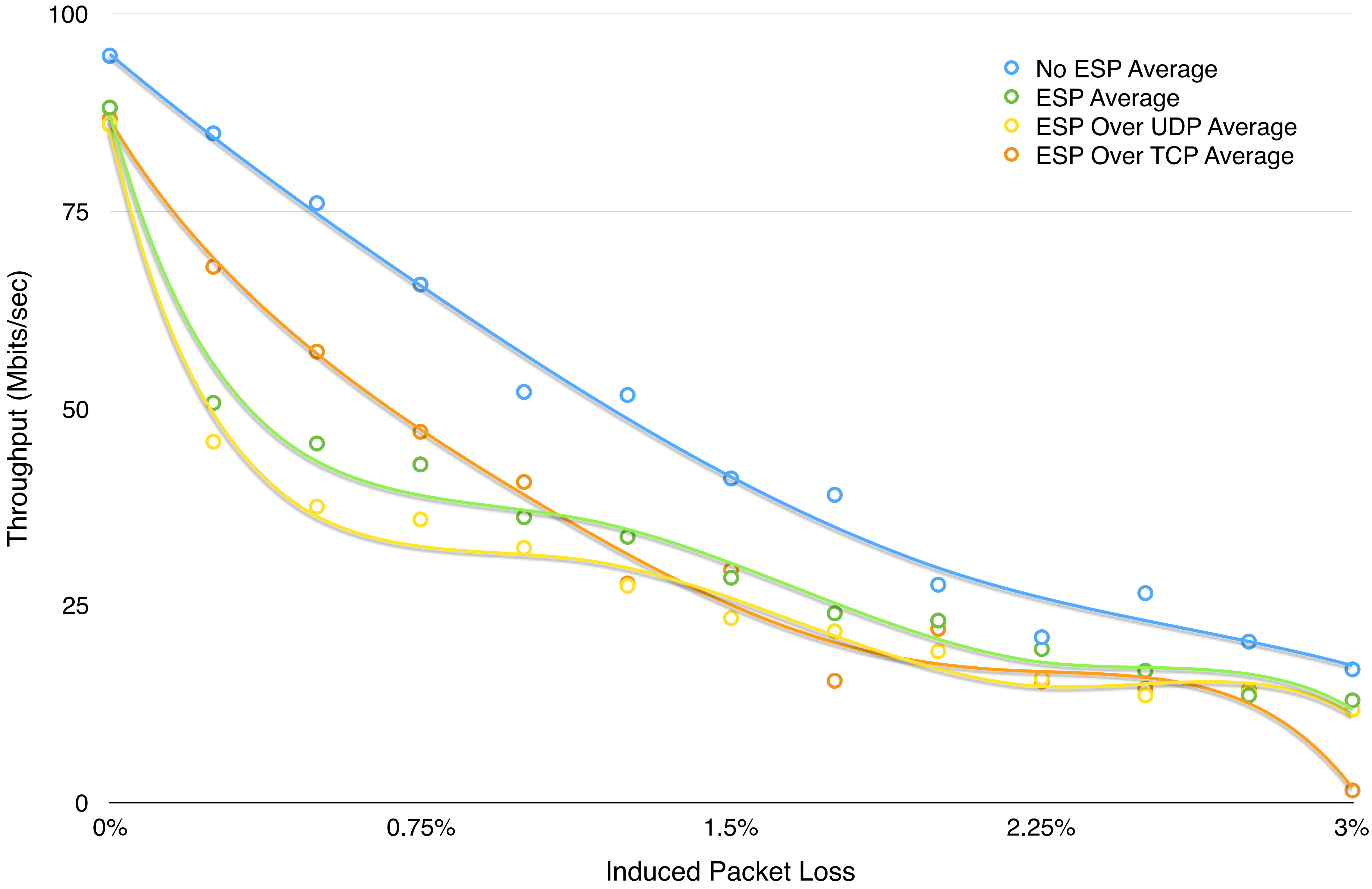
- Standard IKEv2 VPN server
- Relay box in front of server, decapsulating TCP stream
- Client modified to send IKEv2 and ESP packets over the TCP stream
- Run multiple iperf TCP flows within the tunnel

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

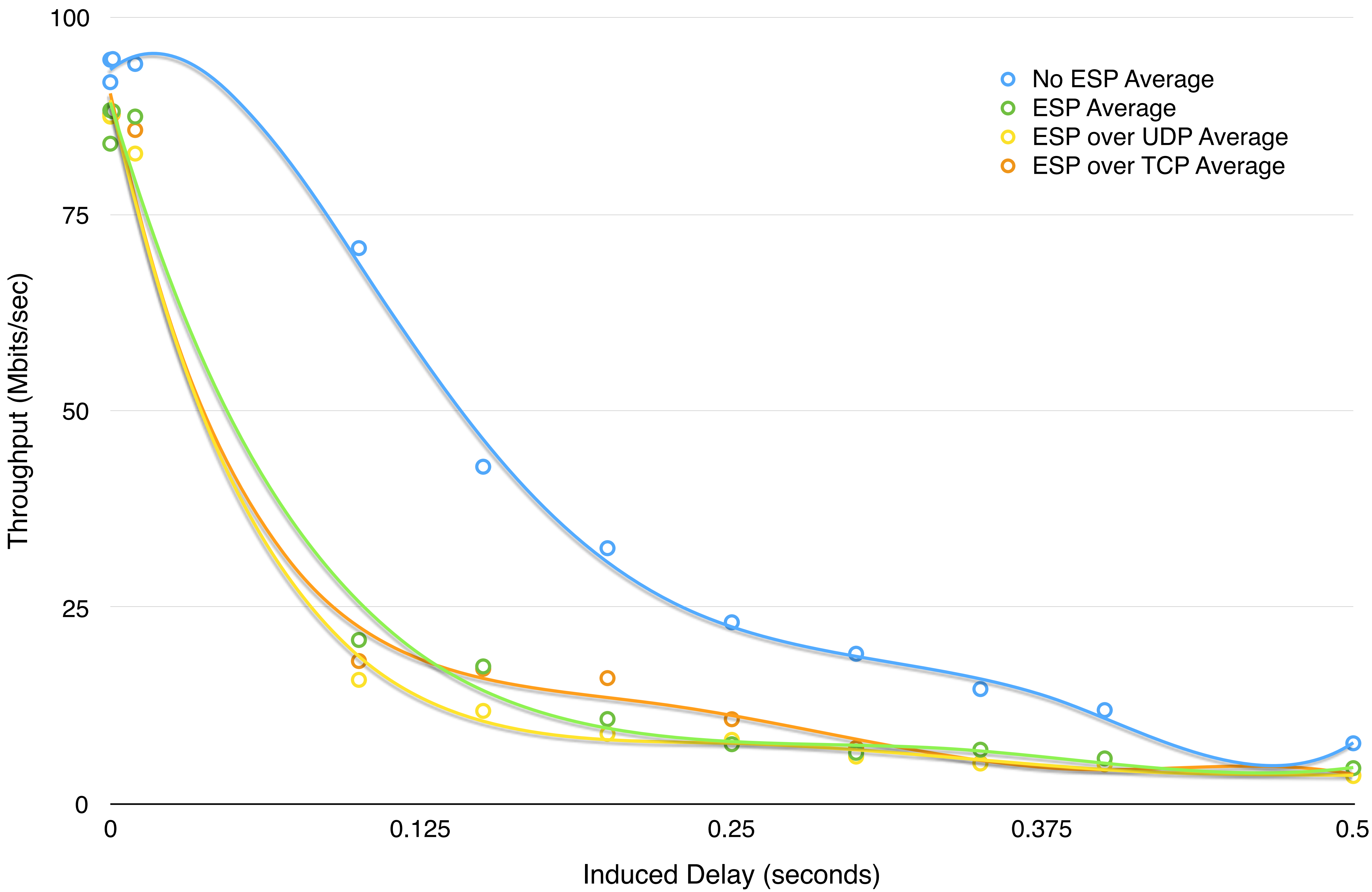
Performance Tests

Loss



Performance Tests

Delay



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