

Performance Enhancement via QUIC Proxy

Contact Persons

Zahed Sarker (zaheduzzaman.sarker@ericsson.com),

Marcus Ihlar (marcus.ihlar@ericsson.com),

Mirja Kühlewind (mirja.kuehlewind@ericsson.com)

IETF 106, Singapore – Nov 20, 2019 – EToSat side meeting

Motivation and Design Goals

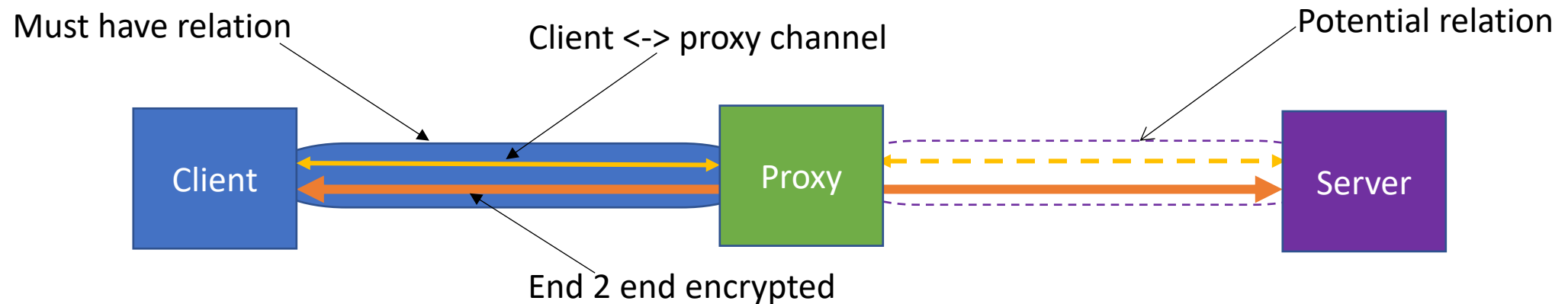


- Motivation: Make network support functions collaborative and explicit
 - Use QUIC a substrate/tunnelling protocol to proxies/VPN server/load balancers...
 - Client and/or server selects proxy and function
- Design Goals
 - Don't interfere with end-to-end data security
 - Explicitly addressed L4 collaborative proxy where proxy operate with consent from the client (and optionally server)
 - Proxy is discoverable or pre-configured
 - Explicit protected communication channel between client and the proxy

Collaborative Performance Enhancement



- Client explicitly opens connection to proxy and requests forwarding
 - uses MASQUE
- QUIC proxy provides secure forwarding and performance enhancement services
 - For example - topology hiding, congestion control support (mobile/satellite), access policy enforcement
- QUIC proxy may optionally also open a tunnel to server (if supported by server)
- Multiple security context
 - End-to-end security is intact while collaborating with proxy
- Inner and outer transport layer interaction
 - For example - Congestion control selection





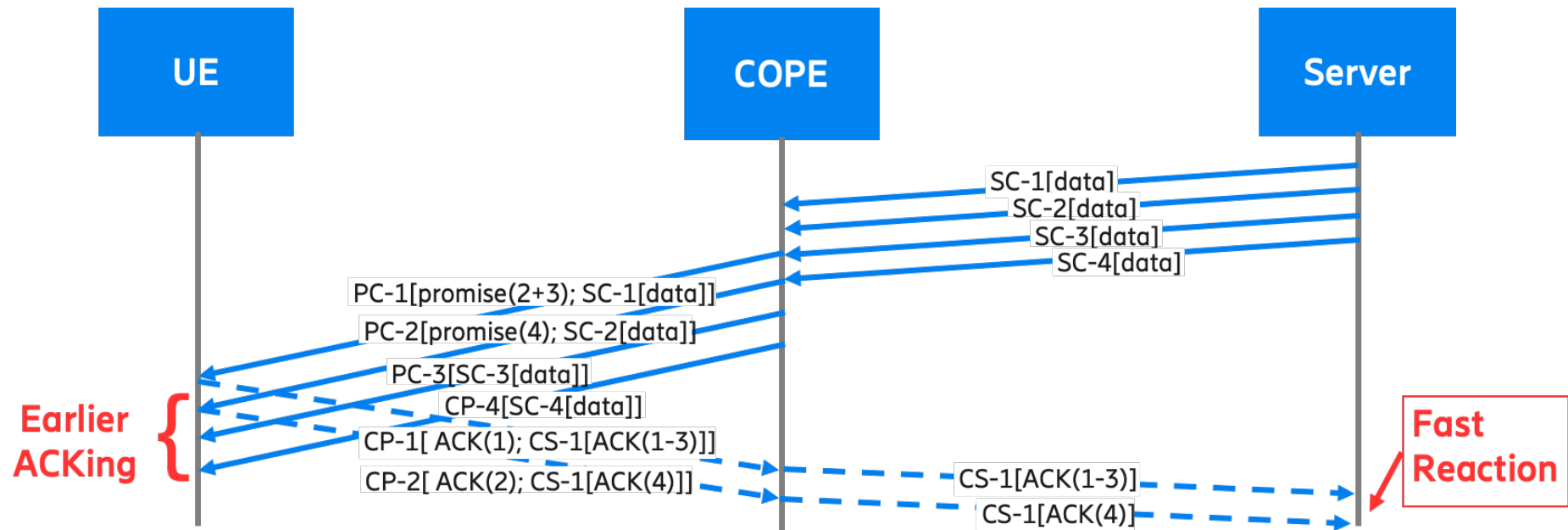
Two approaches explored

- Proxy “promises” delivery of data to the client ensuring server progress
 - Local recovery is used to conceal local network impairments
- Turning off end-to-end inner congestion control
 - to use domain specific congestion control on legs of outer connection



The Promise

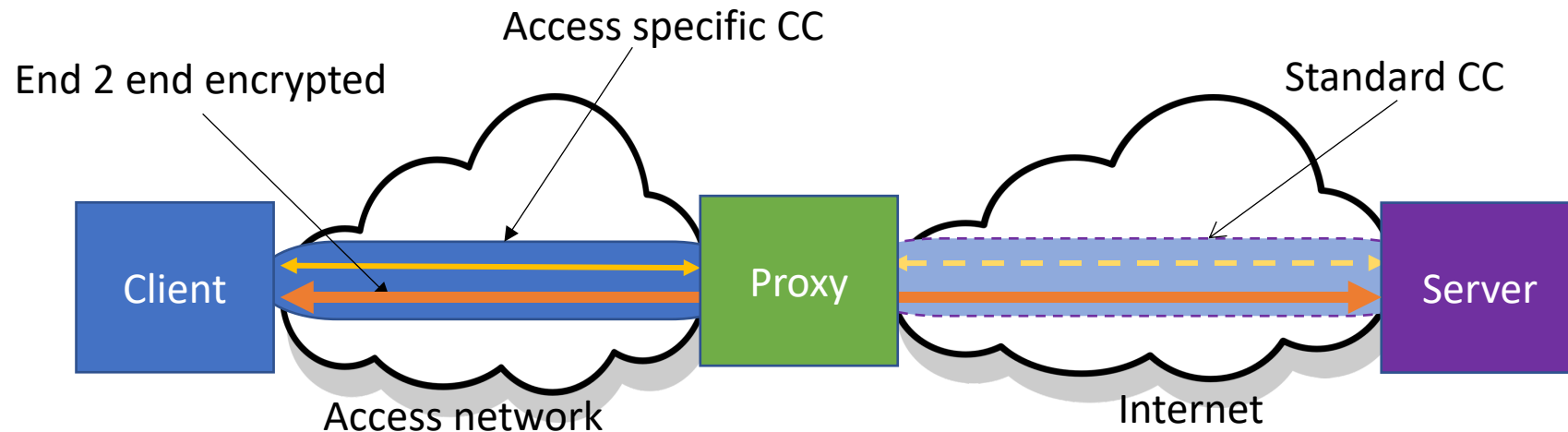
- Proxy receives packets from the server that it is not able to forward immediately (due to channel impairments) and buffers locally
- Proxy “promises” delivery of data to the client ensuring continuous server progress to be able to quickly “fill the link” when resources become available



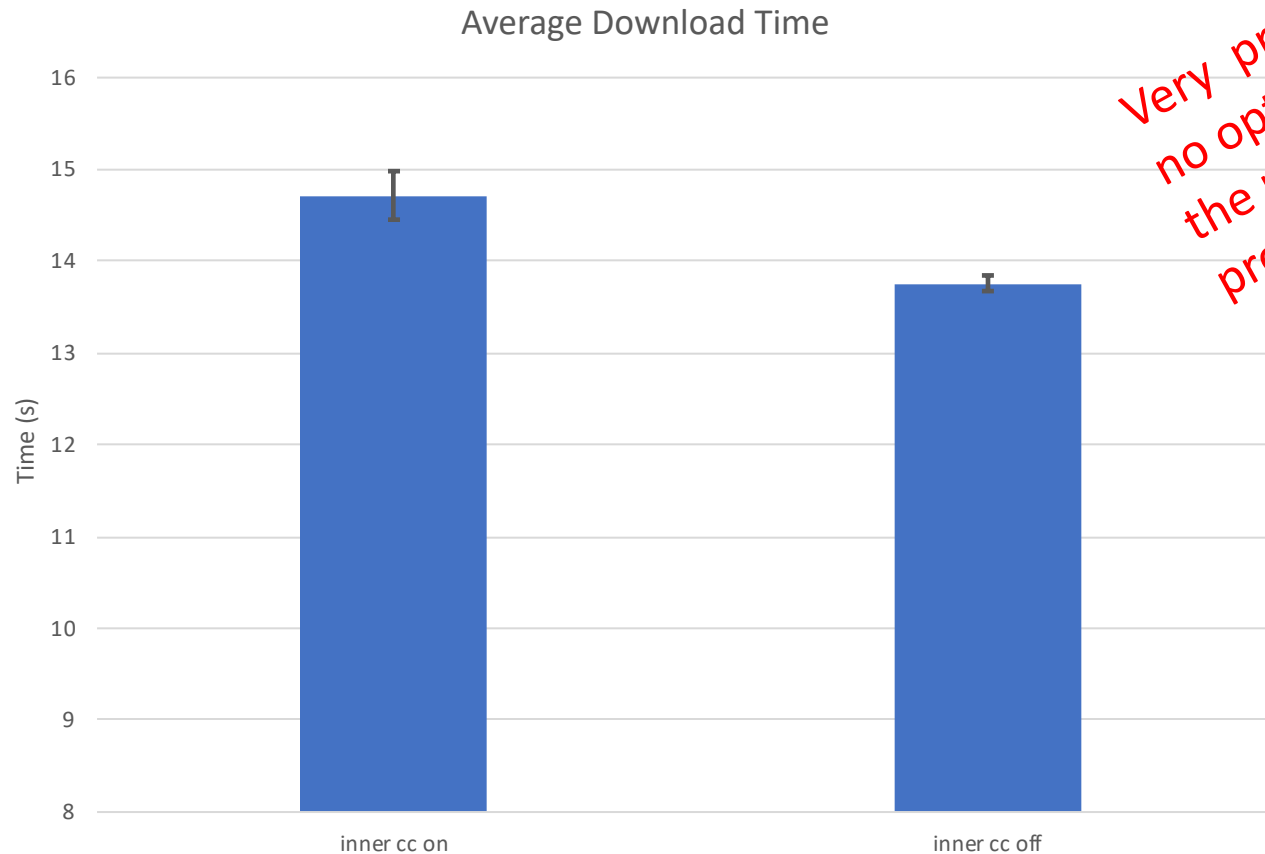


Domain expertise

- Proxy has explicit outer connection to both client and server
- Proxy uses domain specific congestion control
- Endpoints can disable e2e inner congestion control



Download time improvement with local loss recovery.



Very preliminary example of results with no optimization in place; this is just to show the potential of such concept and a working prototype

- Download of 1MB files over large delay path and lossy last hop.
- Simulated Network Setup
 - **Server <-> Proxy**
 - RTT 500 ms
 - BW 50 mbps
 - Random loss 0%
 - **Proxy <-> Client**
 - RTT 25 ms
 - BW 10 mbps
 - Random loss 1%

Client, server and proxy is prototyped based on Facebook's MVFST QUIC implementation