QUIC & TLS

BoF overview

Background

QUIC crypto allows QUIC to start in fewer round trips

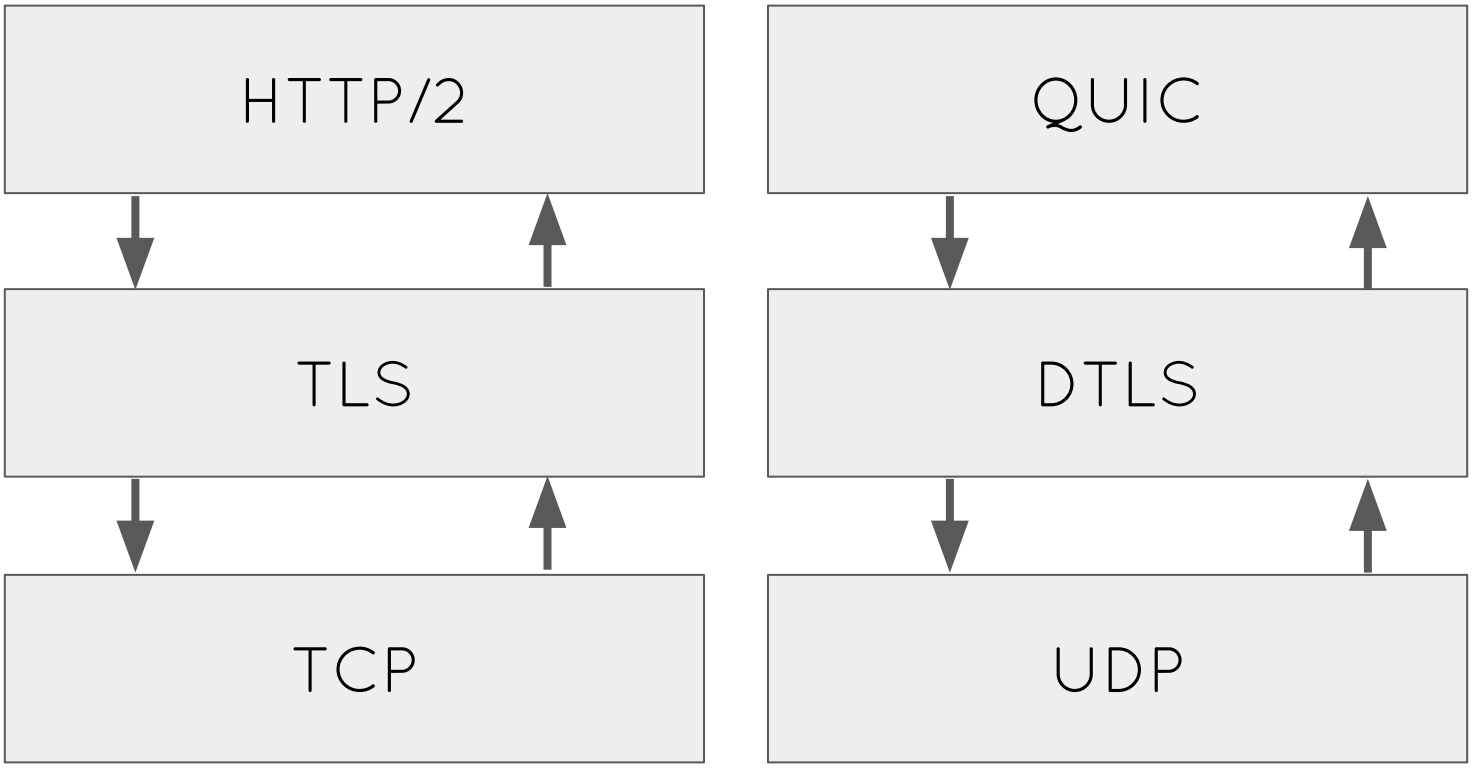
- 1 round trip on first contact (*)
- 0 round trip on return (**)

Saving round trips is a **huge** win

TLS 1.3 provides the same performance properties

- And several improvements over QUIC crypto

Security Modularization Classic

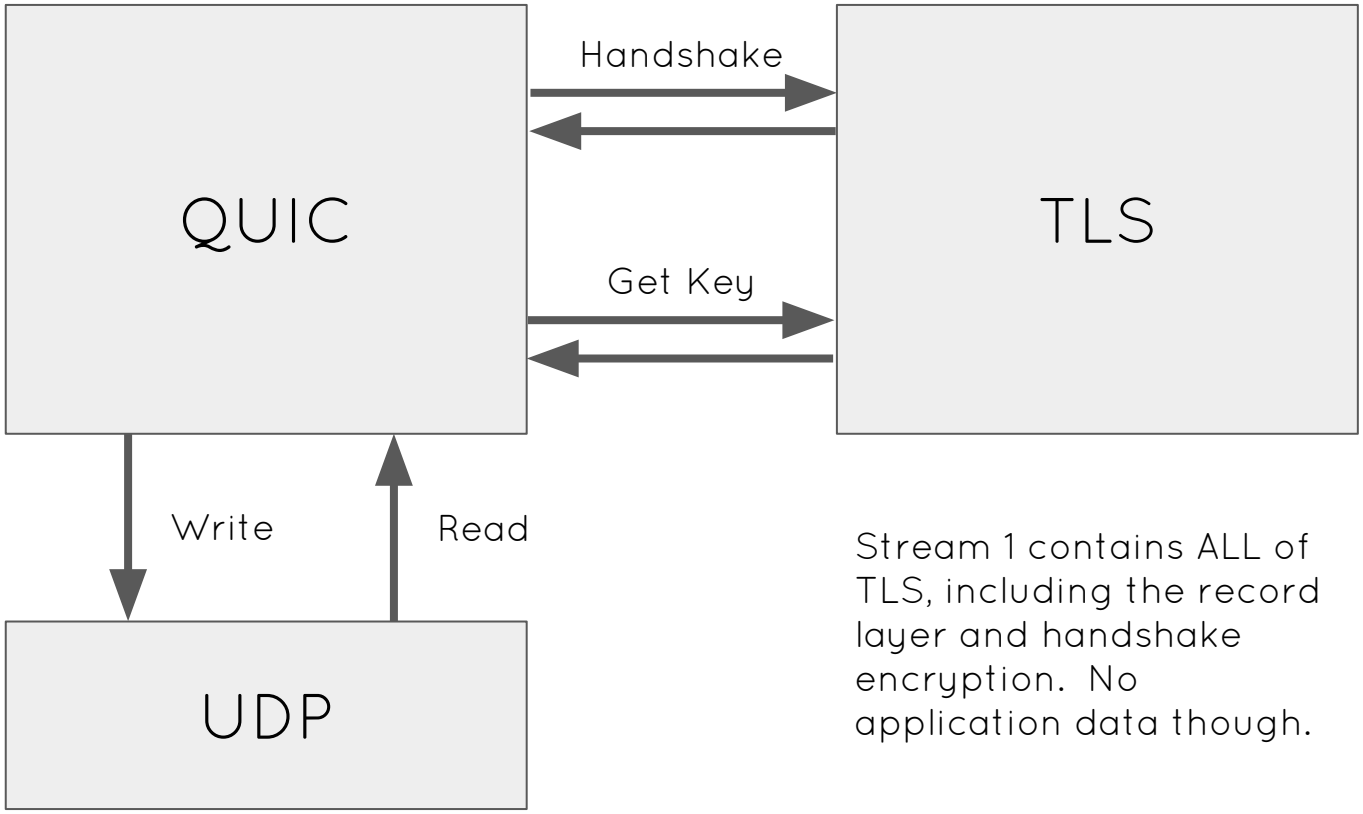


Problems

QUIC over DTLS might work, but

- DTLS loss recovery is primitive
- QUIC can't see loss/delay/etc... during handshake
- DTLS record format is a tad wasteful (*)

TLS as a Service



Stream 1 contains ALL of TLS, including the record layer and handshake encryption. No application data though.

Benefits

QUIC can provide ordering and reliability for TLS

- Use QUIC Stream 1 for TLS

QUIC can use its own record protection

- This is similar to the DTLS record structure

Use DTLS cookie or session ticket for DoS mitigation

- Completely transparent to clients

Complications

Generic exporters might be risky for use with 0-RTT

- Solution: a special key export

QUIC version negotiation isn't integrity protected

- Solution: bind to ALPN and validate
- Use an extension as needed for other parameters

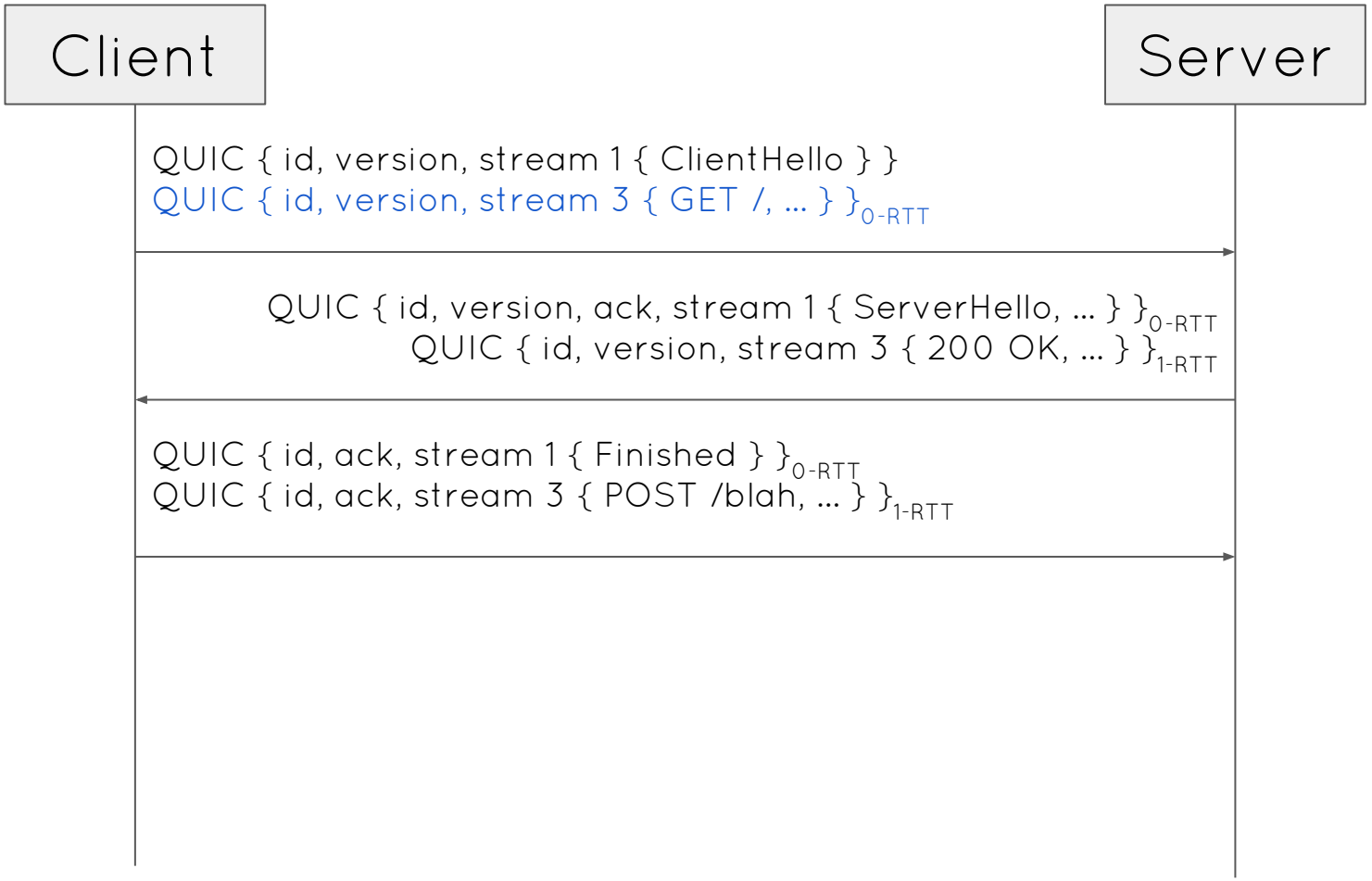
Flow control for TLS handshake

- Solution: make window big enough and don't worry

Transitions between keys aren't always easy

- No great solution here

Handshake


Client
Server
QUIC { id, version, stream 1 { ClientHello } }
QUIC { id, version, stream 3 { GET /, ... } }0-RTT
QUIC { id, version, ack, stream 1 { ServerHello, ... } }0-RTT
QUIC { id, version, stream 3 { 200 OK, ... } }1-RTT
QUIC { id, ack, stream 1 { Finished } }0-RTT
QUIC { id, ack, stream 3 { POST /blah, ... } }1-RTT