{QUIC,Hello} from OpenSSL

Alexandr Nedvedicky
sashan@openssl.org
https://github.com/sashan

History of QUIC in OpenSSL

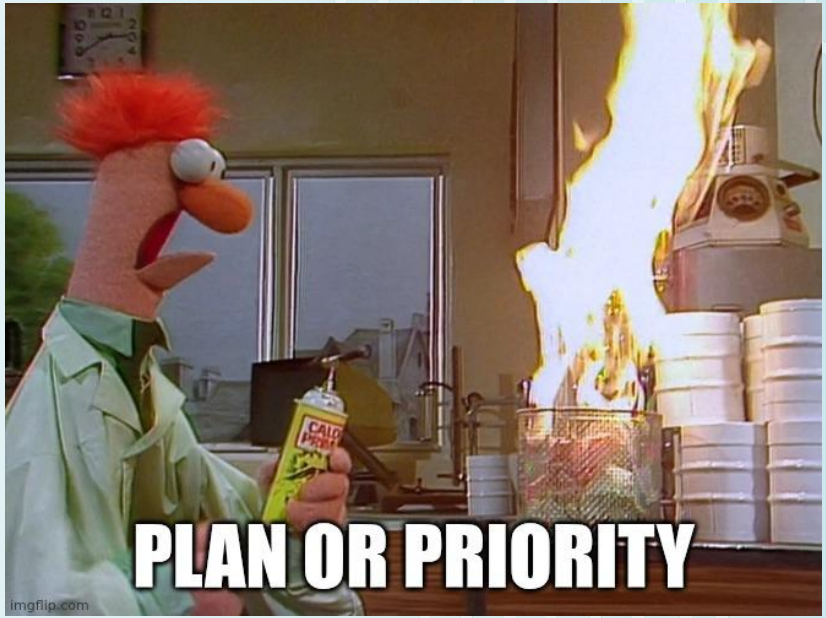
- 3.2, 2023, client side
- 3.3, 2024, SSL_poll() arrives
  - spring 24, Hugo Landau leaves OpenSSL
- 3.5, 2025, server side + handshake layer API for 3rd party QUIC stacks (BoringSSL-like)

?TODO?

- 0-RTT, connection migration, ?no-copy I/O API? (offer alternative to SSL_poll())
- Make QUIC stack more robust
- Improve testing story

What next

- Improve testing, I wish QUIC in scapy
  - The current practice is to derive malicious QUIC stack from existing implementation
  - The goal is to trigger various corner cases such as holes in recv window, deliberately delaying ACK… to determine DOS conditions and verify mitigations are effective
- Subscription I/O to unleash zero-copy

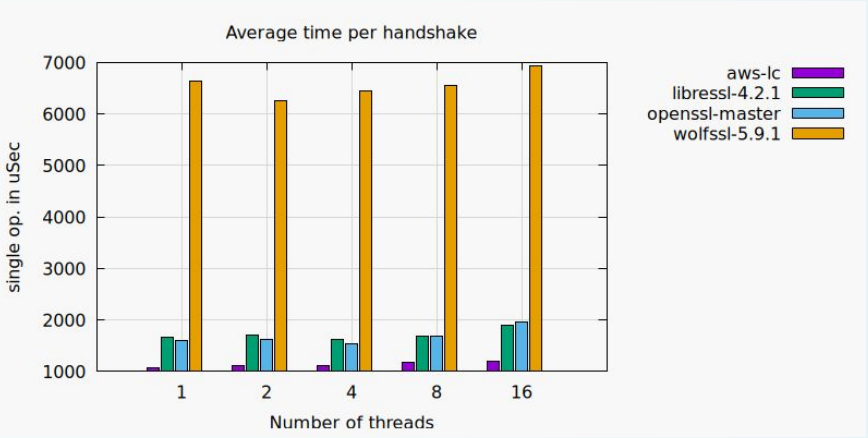


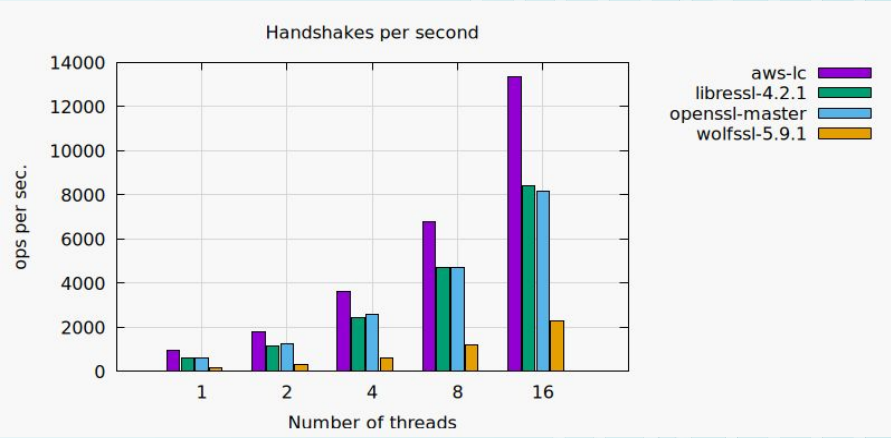
Can OpenSSL/QUIC get curl ride again?

- Jan 2024 #0535f6e by icing adds OpenSSL QUIC stack to curl
- Jan 2026 #6aaac9d by badger removes OpenSSL QUIC stack from curl
  - Looks like friction comes from interfacing OpenSSL I/O (SSL_read()/SSL_write()) with libhttp3

What else is on now

- Still sorting out mem/perf Issues
  - #31143 (dropping method store refcount)
  - #29117 (making ASN1 strings opaque)
  - #31184 (caching parsed certs)
  - Side channel mitigation (const. Time math)
- regular benchmarks (openssl vs. openssl)
  - The cross library benchmark is still work in progress (see next slide)





Charts from handshake tools from OpenSSL project. openssl-bench supports openssl/boring/rust I have not looked at it yet to add support for other SSL libraries.


MEEK
imgflip.com