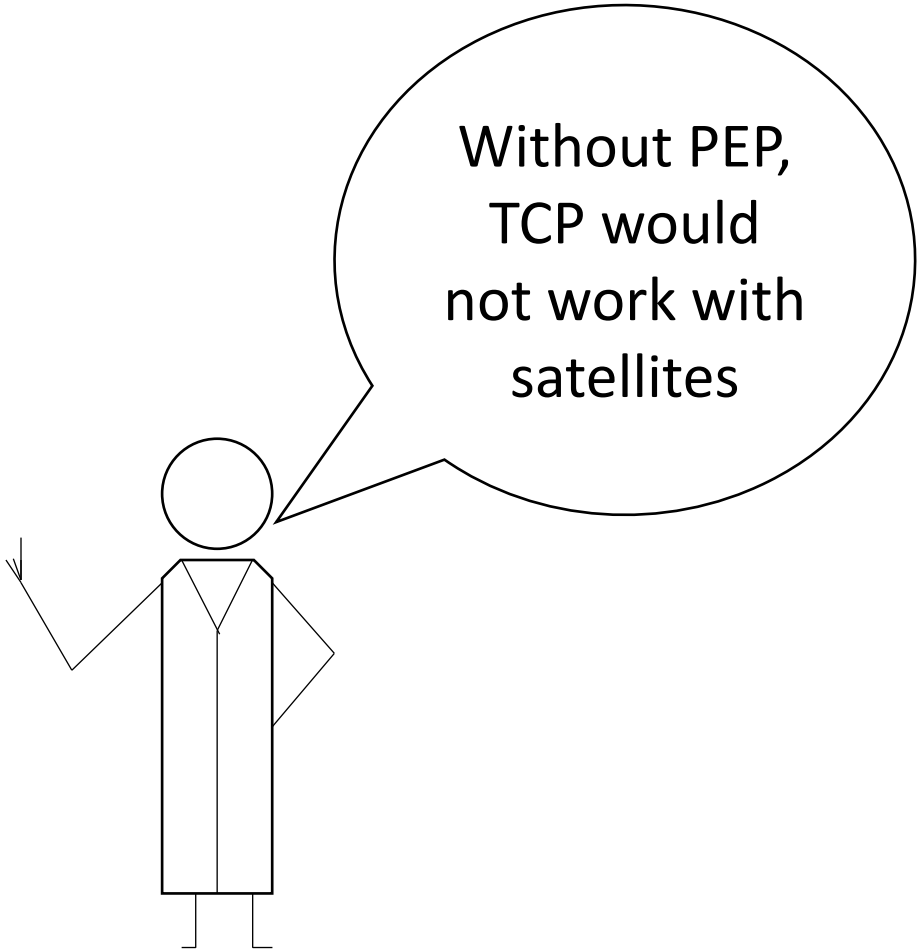
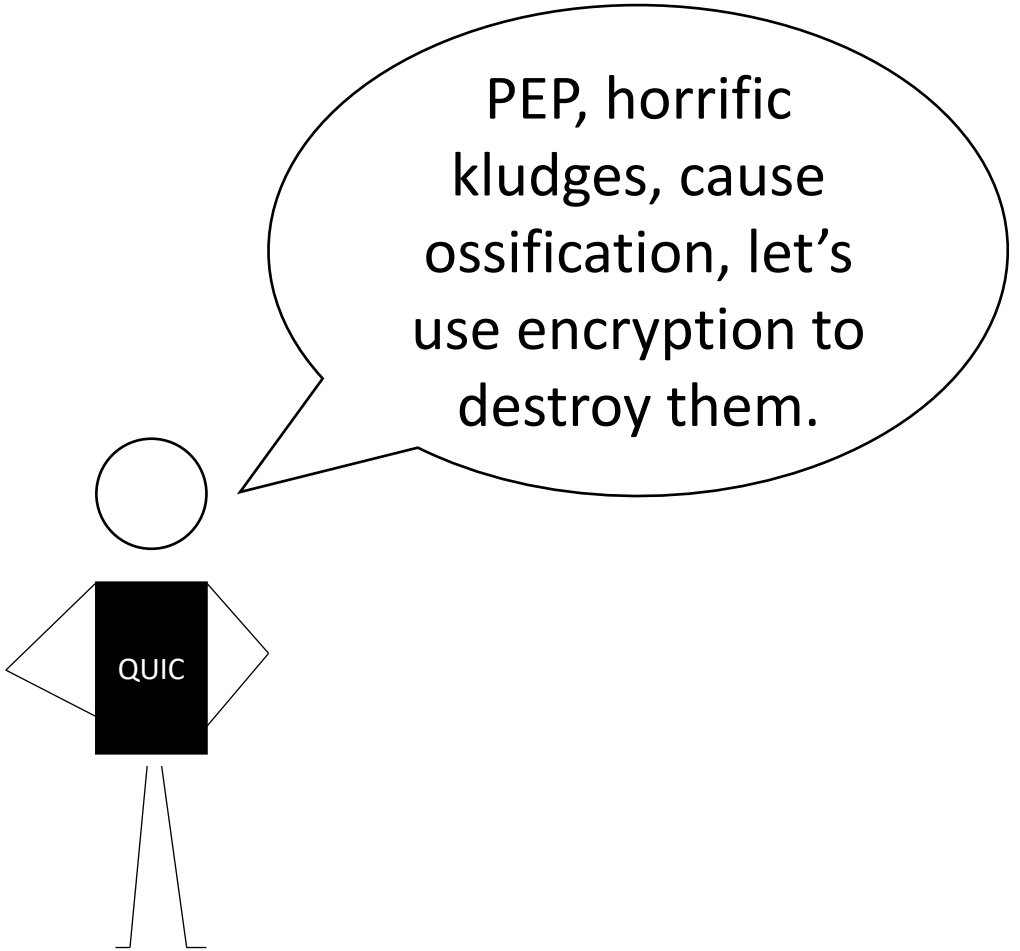
Making QUIC work with satellites

Christian Huitema

January 19, 2022


Without PEP, TCP would not work with satellites

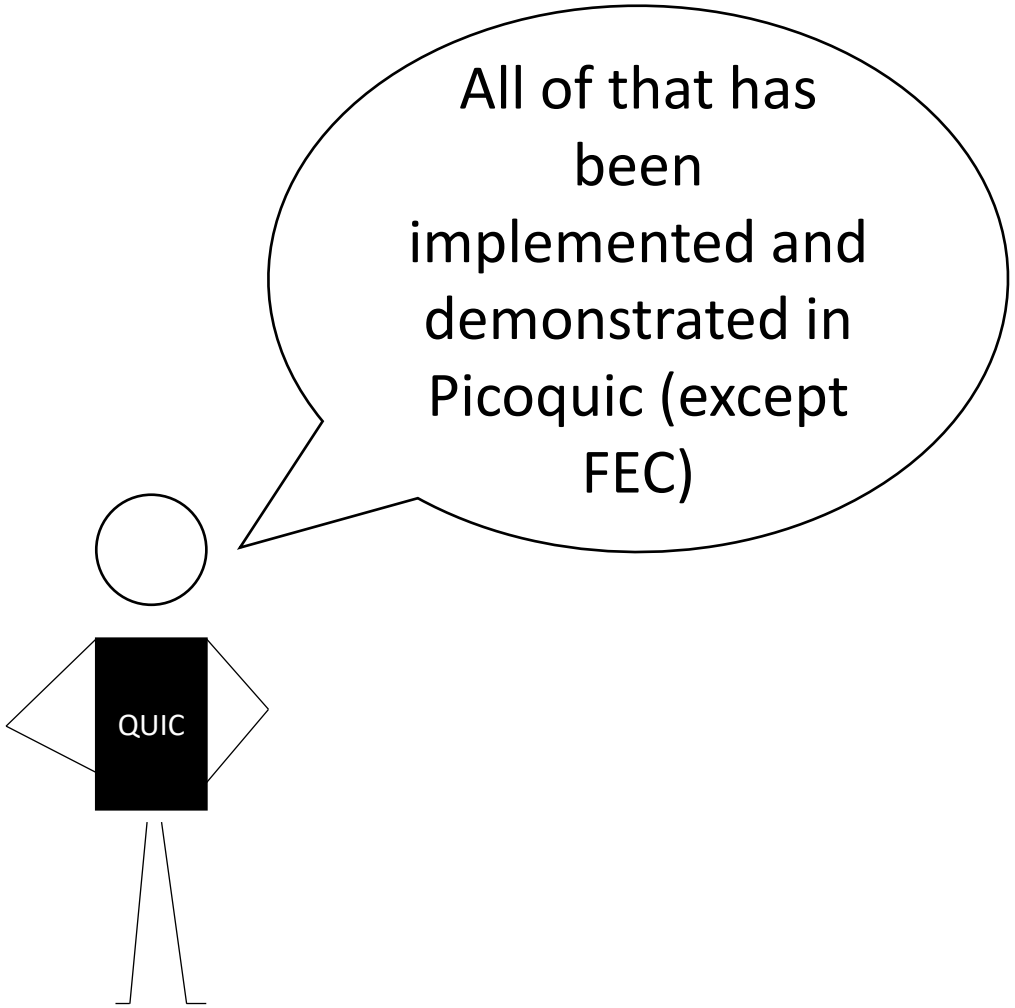

PEP, horrific kludges, cause ossification, let’s use encryption to destroy them.
QUIC

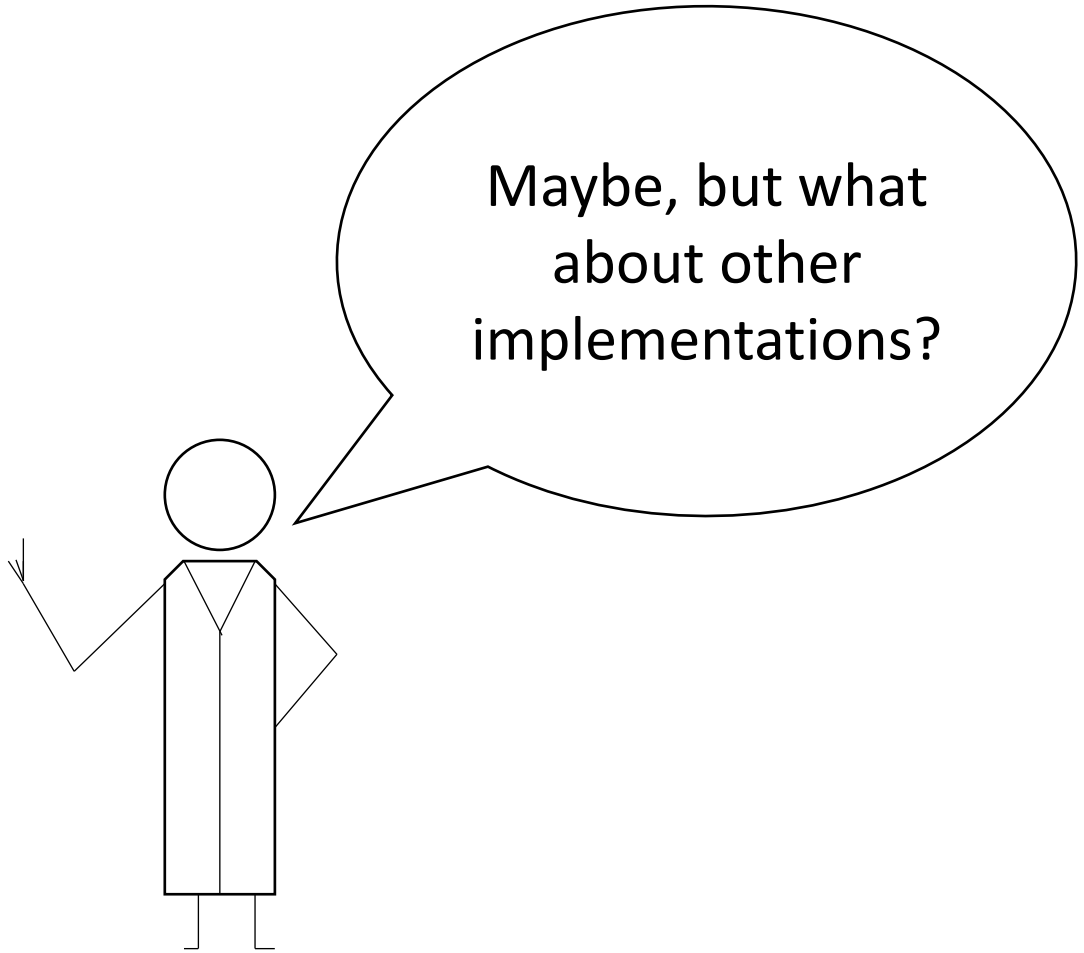
What did break in the past

- TCP slow start
  - log2(BDP/starting Window)*RTT
  - Not just start up, also recovery from trains of packet losses
- Loss recovery
  - Too many losses drive CWIN down
  - Too many losses to carry in SACK lists
  - Losses on Wi-Fi link slow to correct on satellite link
- ACK Congestion
  - Common case of 100Mps downlink, 1 Mps uplink
  - One ACK for 2 packets causes uplink congestion

Challenges can be solved end-to-end

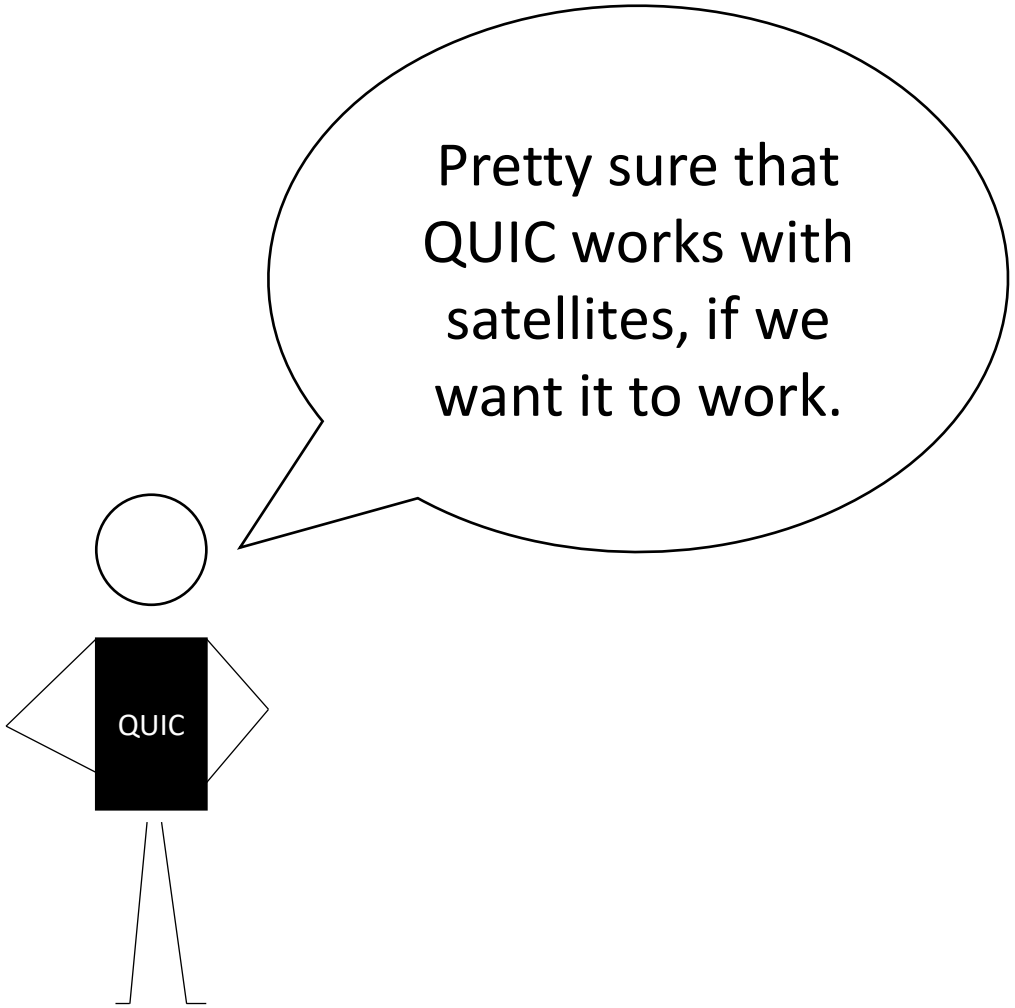
- Connection startup
  - Remember previous connections, EBDP options
  - Use bandwidth evaluation during startup, variants of packet train, ramp faster
- Loss Recovery
  - Select CC algorithms that are less sensitive to packet loss – BBR
  - QUIC ACK frames can carry large number of ACK-ranges
  - Possibly, develop FEC for QUIC
- ACK Congestion
  - QUIC delayed ACK extension
  - Typically one ACK for 64KB of packets


All of that has been implemented and demonstrated in Picoquic (except FEC)
QUIC


Maybe, but what about other implementations?

The real issue is testing!

- If it is not tested, it probably does not work
  - Added satellite scenarios to Picoquic test suite, all tests pass.
- Scenario are mostly tested if “main line”
  - Wireless was for a long time “exotic”, is now mainline
  - High speed transmissions are mainline in QUIC
  - Long delay links, EBMP, FEC, not so much
- For “exotic” scenarios, specific software required
  - Delayed ACK Extension is mainline in clients
  - Server-side improvements may be sufficient
  - Servers can implement EBMP and BBR unilaterally.


Pretty sure that QUIC works with satellites, if we want it to work.
QUIC