Trials and tribulations of migrating to IETF QUIC

Ian Swett @ maprg, IETF 105

Where is Google/Chrome now?

gQUIC **v46** is default enabled, v39 and v43 are still supported on the server

- Invariants-3 compatible with transport-19 packet types
- Future version will be invariants-4 compatible

V46 only supports 8 byte CIDs client -> server and 0 byte server -> client

Where is Google/Chrome now?

gQUIC **v46** is default enabled, v39 and **v43** are still supported on the server

- Invariants-3 compatible with transport-19 packet types
- Future version will be invariants-4 compatible

V46 only supports 8 byte CIDs client -> server and 0 byte server -> client

This talk is about v43 -> v46 (invariants-03)

Public Reset ->
Connection Close + Stateless Reset

Why is changing public reset hard?

Issue: LOTs of spots sent a public reset, each one had to be fixed

Why does it matter?: Handshake timeouts and idle timeouts are MUCH longer than sending a close/reset, so connections were stuck until they timed out if no packet or the wrong packet was sent.

When to send what?

gQUIC:

- If no state, always send a “public reset”

IETF QUIC:

- If short header and no state, send a Stateless Reset
- If long header and no state and it’s Initial, try to create a connection
- If long header and no state and it's Handshake, send an Initial close
  - If the version is not supported, send VN

QUIC identification

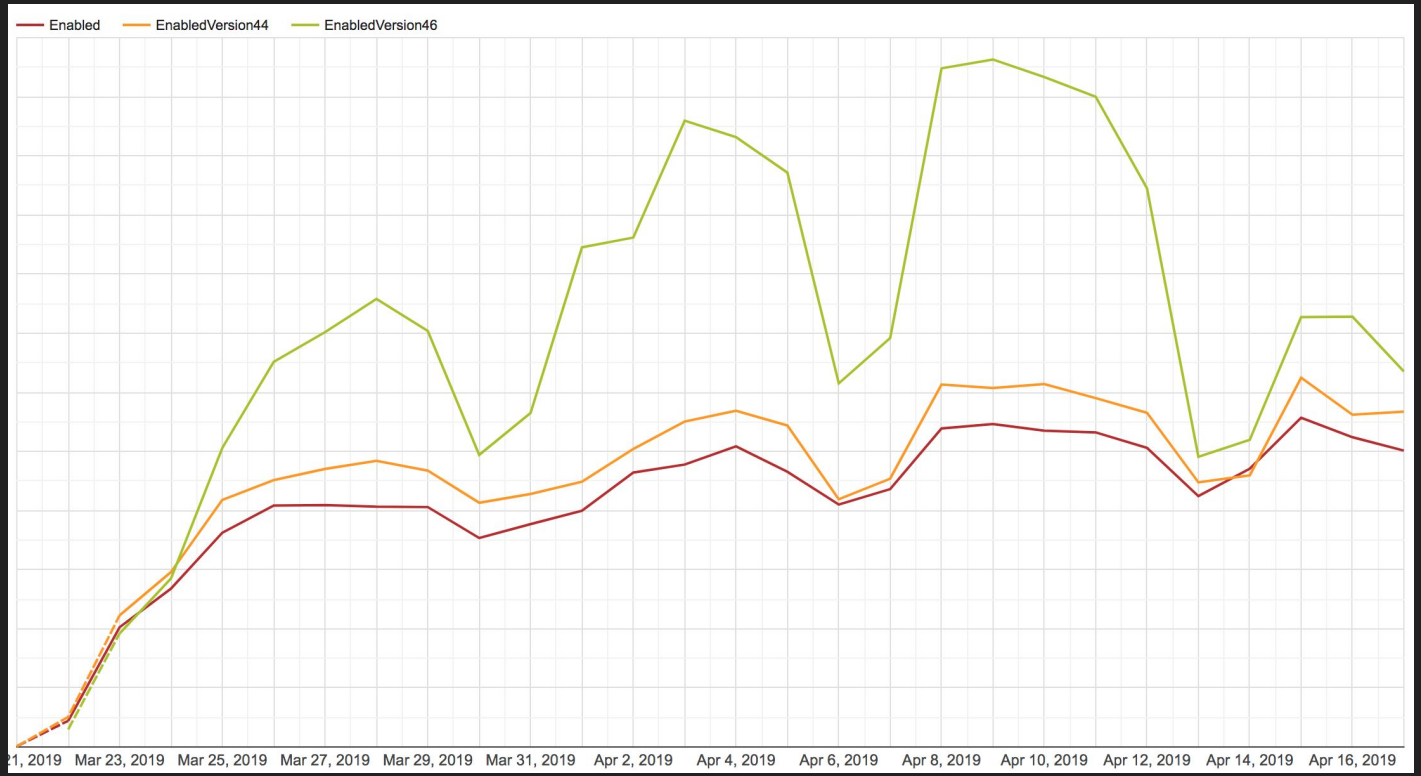
v46: Large increase in post-handshake blackholing

v46 blackholing almost 2x of v43

v46: Large increase in post-handshake blackholing

Suddenly improved April ~13th!

- Not server
- Not Chrome
- ?



What is TOO_MANY_RTOs?

On the 5th RTO, close the connection

Enabled by default on Chrome Desktop

Definitely a heuristic, but it’s better than nothing

- A great proxy for sudden blackholing
- But when were the connections being blackholed?

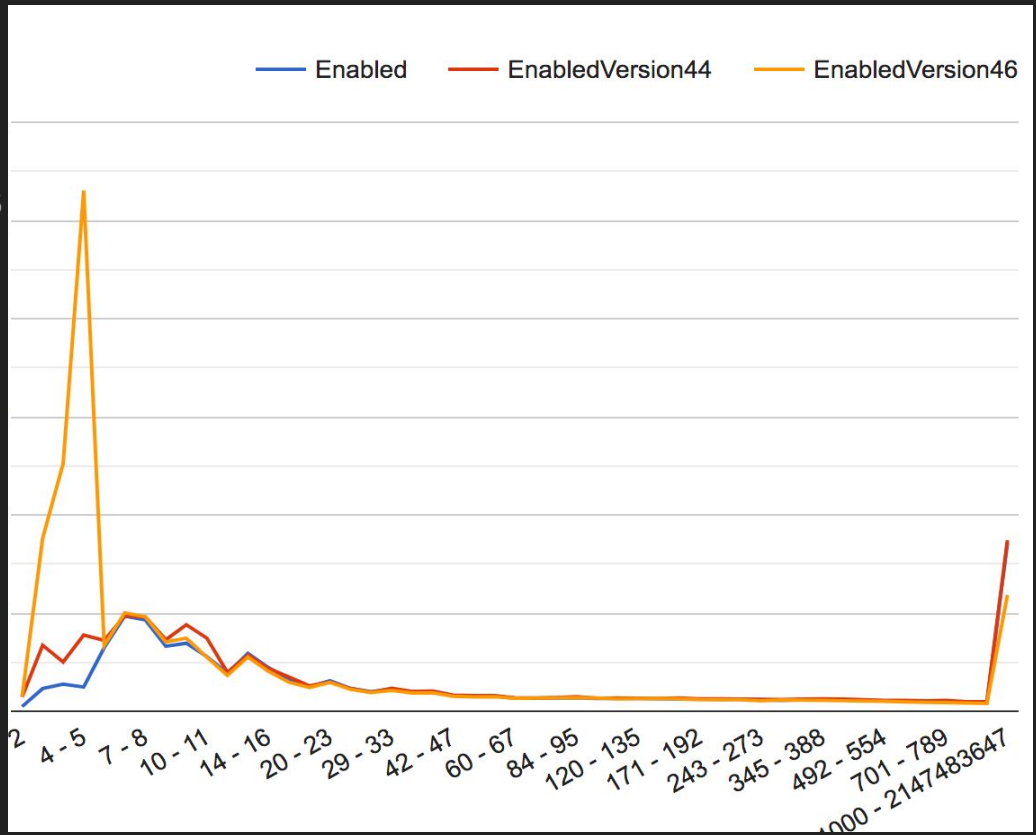
Note: gQUIC 5 RTOs ~= 7 PTOs because our 5 RTOs are after 2 TLPs are sent

When in a connection?

HUGE spike of TOO_MANY_RTOS at 2, 3 and 4 packets

Almost identical from 7 packets

# of packets received before TOO_MANY_RTOS



Turned out to be middlebox QUIC identification

Suddenly improved when a vendor updated their QUIC identification

Most users updated weekly, but some updated less frequently(ie: quarterly)

This caused a multi-week issue which was eventually diagnosed

How to block QUIC*

If QUIC is going to be blocked, ensure all packets in at least one direction are completely blocked.

Anything else is likely **very** user visible

*Or likely any other connection based transport

Antivirus QUIC blocking

Suddenly, QUIC usage among Windows users dropped measurably!

Eventually traced it to a single AV company

- At the time, v46 was not blocked
- v46 was default enabled, and then v46 was blocked :(

Slight change in SNI location

People have started inspecting gQUIC SNI in some locations

Most haven’t told us, so breakage is a real risk as gQUIC -> IETF QUIC

Realistically, there are 2+ more versions before final IETF QUIC

What’s Next?

CRYPTO frames, Invariants-5, TLS 1.3, header protection, etc

gQUIC is now closer to IETF QUIC, with many changes to go

- Some are visible to passive observers, so something will break...