

QUIC

Deployment Experience @Google

Presenter: Ian Swett

Deployment timeline: June, 2013



Chrome Canary



Google

Deployment timeline: April, 2014



Chrome Stable Desktop



Google

Deployment timeline: 2015



Deployment timeline: 2016



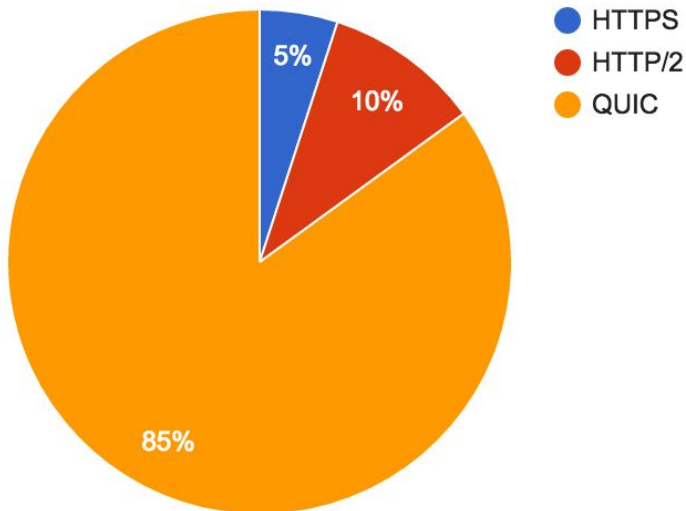
Deployment at Google

QUIC used for every major Google Site on Desktop and Android Chrome

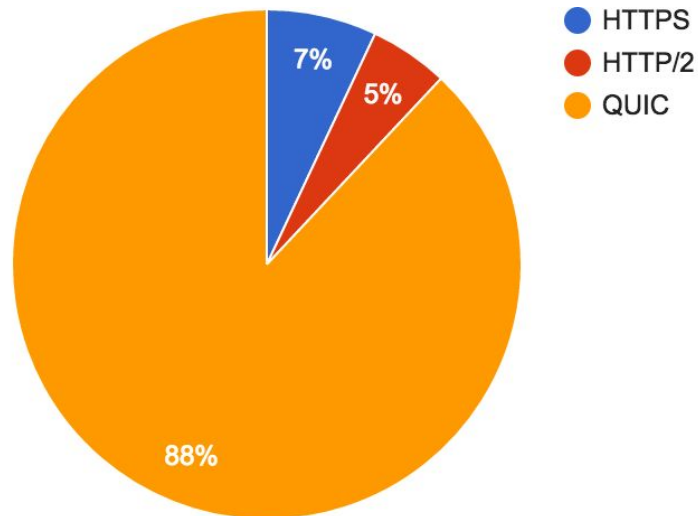
- Disabled for domains requiring PCI compliance.

Many Google Android Apps

Chrome Requests



Chrome Bytes Received



Fallback to HTTP/2

What if UDP is blocked?

- Chrome seamlessly falls back to HTTP/TCP

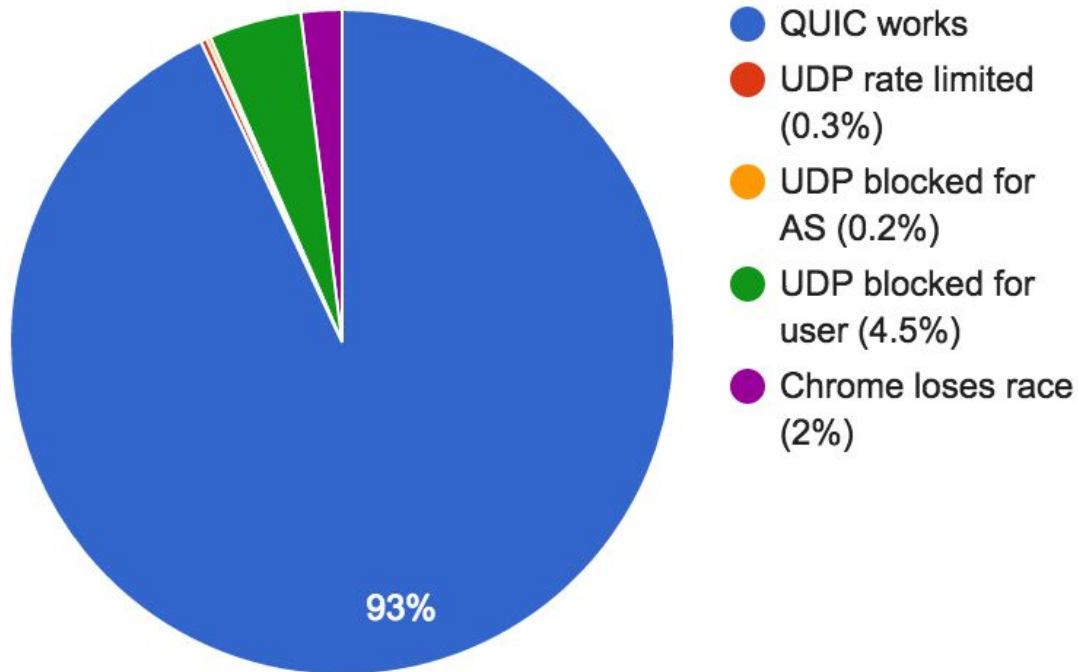
What if the path MTU is too small?

- QUIC handshake fails, Chrome falls back to TCP

What if a client doesn't want to use QUIC?

- Chrome flag / administrative policy to disable QUIC

QUIC: Does it work?



Since Initial Launch, UDP rate limiting has decreased by 2/3rds

Measuring and Monitoring

Controlled Experiments

Client Side

Latency, Bandwidth, Quality of Experience, Connection Close Errors

Server Side

Latency, Bandwidth, ORTT Rate, Connection Close Errors, Handshake Failures, Transport Metrics

Fine Grained Analysis

By ASN, Server, OS, Version

Performance on Google properties

Faster page loading times

- 5% faster on average
- 1 second faster for web search at 99th-percentile

Improved YouTube Quality of Experience

- 30% fewer rebuffers (video pauses)

Where are the gains from?

0-RTT

- Over 50% of the latency improvement (at median and 95th-percentile)

Improved loss recovery

- Fewer timeout based retransmissions improve tail latency and YouTube video rebuffer rates

Other, smaller benefits

- e.g. head of line blocking, more efficient framing, consistent clients, RWIN issues.

QUIC

Source: [QUIC in Chromium](#)

Page: www.chromium.org/quic

IETF Mailing List: quic@ietf.org