QUIC

Next generation multiplexed transport over UDP

Ian Swett
ianswett@google.com
QUIC
Quick UDP Internet Connections

- Multiplexed transport over UDP
- Reduce latency
- Rapid experimentation
- Open source development in Chromium
What does it replace?

- HTTP/2
- TLS 1.2
- TCP
- IP

- HTTP/2
- QUIC
- UDP
Encryption

Comparable to TLS
Perfect forward secrecy, with more efficient handshake

IP spoofing protection
Signed proof of address

more crypto details...
Congestion Control

Comparable to Linux TCP
TCP Cubic, FACK, TLP, F-RTO, Early Retransmit...

Better Loss Detection
Retransmission uses a new sequence number

More flexibility going forward
Improved congestion feedback, control over acking
Why is Google deploying QUIC?

Faster page loading times
- 3% faster on average
- 1s faster for web search at 99%

Improved YouTube Quality of Experience
- 30% fewer rebuffers (video pauses)

More improvements to come
- Bandwidth resumption, Forward Error Correction, etc

Recent Blog Post
Deployment

Today: About half of Google to Chrome
Debugging Tools: Chrome

- chrome://net-internals
  - Shows active QUIC sessions
  - Captures every event
  - Diagnoses handshake failures, packet loss, etc
  - Important for filing Chromium bugs
Debugging Tools: Wireshark

- Protocol: QUIC
- CID: Connection ID
- Seq: Sequence number
- Version: ie: Q024
- Public flags: 1 byte
- Payload: Encrypted
What does QUIC need to work well?

UDP port 443
- QUIC uses UDP port 443, since it provides equivalent functionality to HTTP/2.

Same QoS handling for UDP as TCP
- No rate limiting
- QoS differences are very difficult to detect client side

Reasonable Stateful FW/NAT timeouts
- QUIC currently uses a conservative timeout of 30s
- If a NAT binding expires, the user’s connection suddenly fails
Common Questions?

What if UDP is blocked?
- Chrome races TCP and QUIC and TCP will win.
- Chrome won’t try QUIC for that domain for 5 minutes.
- Exponentially back off from there.

What if the path MTU is too small?
- QUIC’s handshake packets are a fixed size (1392 bytes) and always the largest packet in a connection.
- If the MTU is too small, the handshake will fail.

What if the QoS is different?
- QoS differences are not detected Chrome side.
- We’re working on detecting extreme cases of rate limiting.
QUIC

Source: QUIC in Chromium

Further reading: Chromium Site Design doc and rationale, Wire spec, Crypto doc

Mailing list: proto-quic@chromium.org

Ian Swett
ianswett@google.com