

Core Protocol Spec

draft-hamilton-quick-transport-protocol-01

Draft Outline

- Overview of QUIC features
- Packet Types and Formats
- Life of a Connection
 - Describes Version Negotiation and Crypto handshake requirements
- Frame Types and Formats
- Packetization and Reliability
- Streams: QUIC's Data Structuring Abstraction
- Flow Control
- Error Codes
- Security Considerations

Current State

Draft documents QUIC version Q034, which is tethered to the current Google deployment.

- "Diversification Nonce" is for QUIC crypto; not useful with TLS 1.3
- Timestamps are in a home-grown floating-point format
- Numbers are little-endian

If adopted, based on wg discussion and consensus, the draft will move away from Q034 design.

Discussions on mailing list so far

- Packet number vs. Sequence number
 - Use "Packet number", since TCP seq num has different semantics.

Discussions on mailing list so far

- Packet number vs. Sequence number
 - Use "Packet number", since TCP seq num has different semantics.
- Endian-ness
 - Q034 uses little-endian numbers, since Chrome runs only on little-endian machines.
 - Also, there are about 21 big-endian machines left in the world.
 - However, IETF uses big-endian for everything.
 - QUIC will interact with HTTP/2 and with TLS, both of which use big-endian.

Discussions on mailing list: Versioning

- QUIC uses 32-bit version.
 - Q034 is represented by the ASCII values of the four characters
That is, "Q034" is 0x 51 30 33 36

Discussions on mailing list: Versioning

- QUIC uses 32-bit version.
 - Q034 is represented by the ASCII values of the four characters
That is, "Q034" is 0x 51 30 33 36
- Martin's strawman proposal:
 - 0xffff0000-0xffffffff for experimentation
 - 0xffff5100 and up as corresponding to versions of the draft
 - (Also, ALPN label tracks QUIC version referred to by HTTP mapping doc)

Discussions on mailing list: Versioning

- Jana's mod to Martin's strawman:
 - Reserve 0x00000001 to 0x0000ffff (16K versions) for standards
 - 0x00000001 - 0x0000fff (4096 versions) to QUIC standards
 - 0x00001000 - 0x0000ffff (~60K versions) for draft versions
 - 0x00010000 upwards (~4 billion versions) for experimentation
 - Conveniently, "Q034" falls in this space

Discussions on mailing list: Versioning

- Jana's mod to Martin's strawman:
 - Reserve 0x00000001 to 0x0000ffff (16K versions) for standards
 - 0x00000001 - 0x0000fff (4096 versions) to QUIC standards
 - 0x00001000 - 0x0000ffff (~60K versions) for draft versions
 - 0x00010000 upwards (~4 billion versions) for experimentation
 - Conveniently, "Q034" falls in this space
- Suggestion: Wiki page for experimental versions
 - Is self-registration adequate?

Other topics for review and discussion

- Transport params format and exhaustive listing
 - List and discuss semantics of connection options
- Clean up and group error codes (editorial)
- Remove diversification nonce (editorial)

Other topics for review and discussion

- Transport params format and exhaustive listing
 - List and discuss semantics of connection options
- Clean up and group error codes (editorial)
- Remove diversification nonce (editorial)
- Reconsider ACK timestamp format
 - Currently a home-grown floating-point wire format
- Connection ID is currently client-chosen
- Addressing: Connection ID and relationship to 5-tuple
 - Relevance to QUIC mobility

Other topics for review and discussion

- Transport params format and exhaustive listing
 - List and discuss semantics of connection options
- Clean up and group error codes (editorial)
- Remove diversification nonce (editorial)
- Reconsider ACK timestamp format
 - Currently a home-grown floating-point wire format
- Connection ID is currently client-chosen
- Addressing: Connection ID and relationship to 5-tuple
 - Relevance to QUIC mobility
- Packet size / Path MTU recommendation
- Functional API, ala RFC 793 (?)