Next-generation networking: QUIC and HTTP 3

Dirkjan Ochtman
June 2019

About me

 **@djc**  **@djco**

- https://dirkjan.ochtman.nl/
- Member of ING's Distributed Ledger Technology ("blockchain") team since December 2018

Started Rust 2 years ago.

Why QUIC & HTTP/3

Uber: Employing QUIC Protocol to Optimize App Performance

We witnessed a **reduction of 10-30 percent in tail-end latencies** for HTTPS traffic at scale in our rider and driver apps. In addition to improving the performance of our apps in low connectivity networks, QUIC gives us end-to-end control over the flow of packets in the user space.

https://eng.uber.com/employing-quic-protocol/

Latency

Google: "the loading time has increased accordingly from 0.4 to 0.9 seconds. (..) decreased traffic and ad revenues by 20%"

Amazon: "every 100 ms increase in load time of Amazon.com decreased sales by one percent"

Microsoft: "tests (..) showed that when search results pages were slowed by one second queries per user declined by 1.0% and ad clicks per user declined by 1.5%. After slowing the search results page by two seconds queries per user declined by 2.5% and ad clicks per user declined by 4.4%."

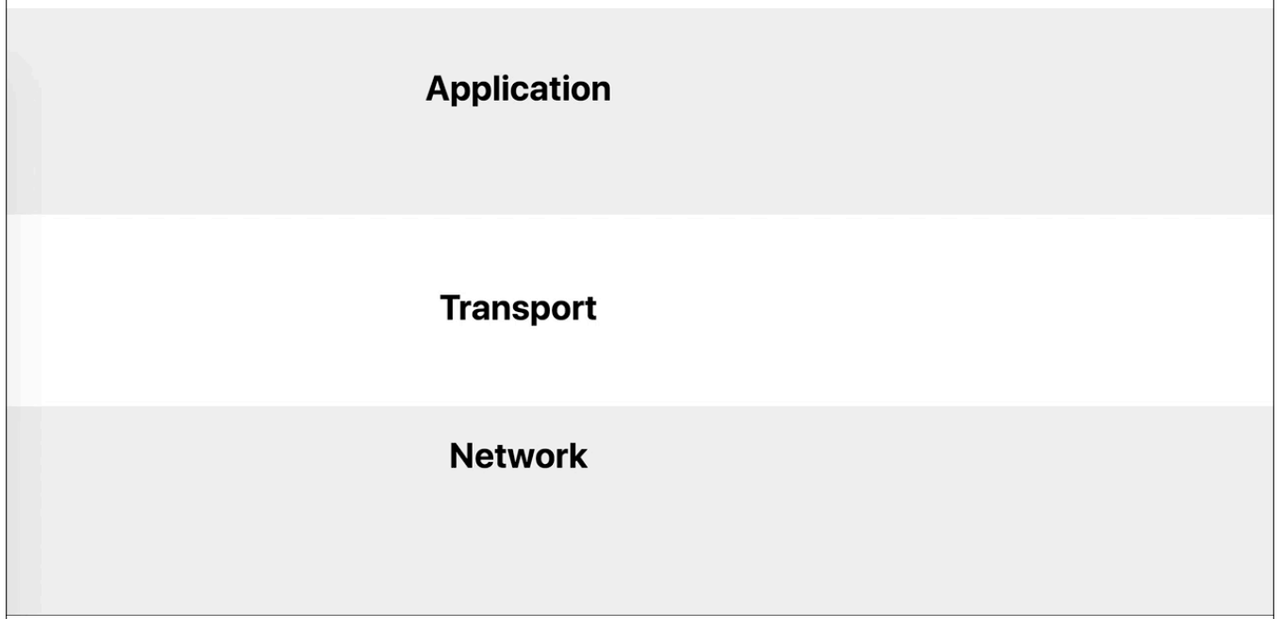
Table of contents

Context & history

- Re-envision TCP after ~40 years of use
- QUIC: started at Google in 2012, announced publicly in 2013; IETF working group established in 2016
- Expect QUIC v1 in Q3 2019
- gQUIC is slowly converging with iQUIC

RFC 793 (TCP) published in 1981.

Head-of-Line blocking



QUIC goals

- Minimize latency
- Provide multiplexing
- Changes limited to path endpoints
- Support connection migration
- Provide always-secure transport
- Expect rapid distributed development and testing

Latency: connection establishment and overall transport for applications, starting with HTTP/2.

QUIC documents

- Invariants: things that cannot change in future versions
- Transport: core protocol logic
- Recovery: loss detection and congestion control
- TLS: how to secure the transport payloads
- QPACK: header compression (static & dynamic)
- HTTP: mapping of HTTP (2) semantics to QUIC

4 kinds of streams; HTTP uses stream 2 and 3 for control traffic

Recovery

- Differences with TCP
  - Monotonically increasing packet numbers
  - No reneging (reduces memory pressure on sender)
  - More ACK ranges (less reliant on timeouts)
  - Explicit correction for delayed acknowledgements

TCP conflates transmission sequence number at the sender with delivery sequence number at the receiver, which results in retransmissions of the same data carrying the same sequence number. QUIC separates the two: QUIC uses a packet number for transmissions, and any data that is to be delivered to the receiving application(s) is sent in one or more streams, with delivery order determined by stream offsets encoded within STREAM frames.

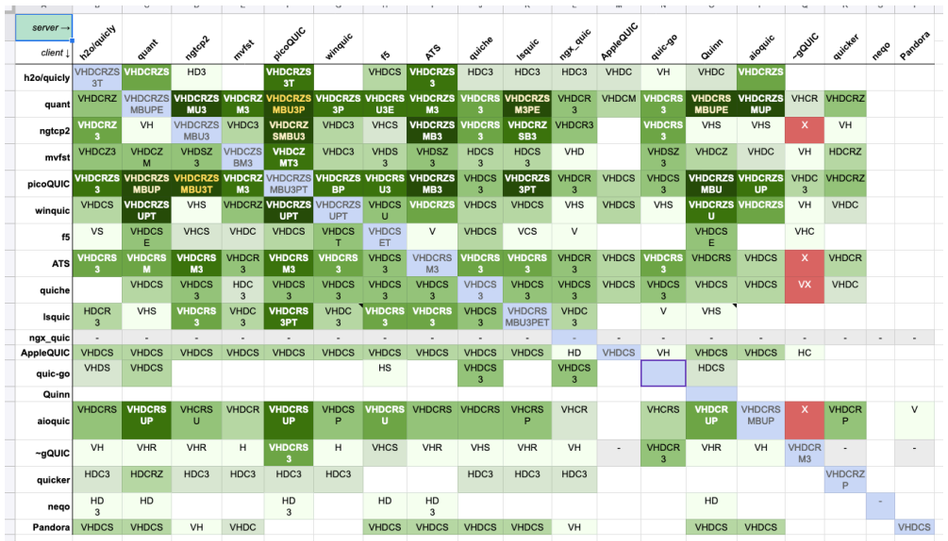
TLS 1.3

- RFC 8446 published in August 2018
- Requires Perfect Forward Secrecy
- Removes many insecure algorithms and features
- Eliminates one round trip for (most) cold sessions
- Warm sessions can send data in first packet
  - But watch for replay attacks (idempotent only?)

QUIC TLS

- TLS 1.3 handshake messages with a custom extension
  - Extension for announcing transport parameters
- After handshake, encrypt payloads with negotiated secret
  - Application data is not wrapped in TLS records

Interoperability matrix



server → / client ↓	h2o/quicly	quant	ngtcp2	mvfst	picoQUIC	winquic	f5	ATS	quiche	lsquic	ngx_quic	AppleQUIC	quic-go	Quinn	aioquic	~gQUIC	quicker	neqo	Pandora
h2o/quicly	VHDCRZS 3T	VHDCRZS	HD3		VHDCRZS 3T		VHDCS	VHDCRZS 3	HDC3	HDC3	HDC3	VHDC	VH	VHDC	VHDCRZS				
quant	VHDCRZ	VHDCRZS MBUPE	VHDCRZS MU3	VHDCRZ M3	VHDCRZS MBU3P	VHDCRZS 3P	VHDCRS U3E	VHDCRZS M3	VHDCRS 3	VHDCRZS M3PE	VHDCR 3	VHDCM	VHDCRS 3	VHDCRS MBUPE	VHDCRZS MUP	VHCR	VHDCRZ		
ngtcp2	VHDCRZ 3	VH	VHDCRZS MBU3	VHDC3	VHDCRZ SMBU3	VHDC3	VHCS	VHDCRZS MB3	VHDCRS 3	VHDCRZ SB3	VHDCR3		VHDCRS 3	VHS	VHS	X	VH		
mvfst	VHDCZ3	VHDCZ M	VHDSZ 3	VHDCZS BM3	VHDCZ MT3	VHDC3	VHDS 3	VHDSZ 3	HDCS 3	HDCS 3	VHD		VHDSZ 3	VHDCZ	VHDC	VH	HDCRZ		
picoQUIC	VHDCRZS 3	VHDCRZS MBUP	VHDCRZS MBU3T	VHDCRZ M3	VHDCRZS MBU3PT	VHDCRZS BP	VHDCRS U3	VHDCRZS MB3	VHDCS 3	VHDCRZS 3PT	VHDCR 3	VHDCS	VHDCS 3	VHDCRZS MBU	VHDCRZS UP	VHDC 3	VHDCRZ		
winquic	VHDCS	VHDCRZS UPT	VHS	VHDCRZ	VHDCRZS UPT	VHDCRZS UPT	VHDCS U	VHDCRZS	VHDCS	VHDCS	VHS	VHDCS	VHS	VHDCRZS U	VHDCRZS	VH	VHDC		
f5	VS	VHDCS E	VHCS	VHDC	VHDCS	VHDCS T	VHDCS ET	V	VHDCS	VCS	V			VHDCS E		VHC			
ATS	VHDCRS 3	VHDCRS M	VHDCRS M3	VHDCR 3	VHDCRS M3	VHDCRS 3	VHDCS 3	VHDCRS M3	VHDCRS 3	VHDCRS 3	VHDCR 3	VHDCS	VHDCRS 3	VHDCRS	VHDCS	X	VHDCR		
quiche		VHDCS	VHDCS 3	HDC 3	VHDCS 3	VHDCS 3	VHDCS 3	VHDCS 3	VHDCS 3	VHDCS 3	VHDCS 3	VHDCS	VHDCS 3	VHDCS	VHDCS	VX	VHDC		
lsquic	HDCR 3	VHS	VHDCRS 3	VHDC 3	VHDCRS 3PT	VHDC 3	VHDCRS 3	VHDCRS 3	VHDCS 3	VHDCRS MBU3PET	VHDC 3		V	VHS					
ngx_quic	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
AppleQUIC	VHDCS	VHDCS	VHDCS	VHDCS	VHDCS	VHDCS	VHDCS	VHDCS	VHDCS	VHDCS	HD	VHDCS	VH	VHDCS	VHDCS	HC			
quic-go	VHDS	VHDCS					HS		VHDCS 3		VHDCS 3			HDCS					
Quinn																			
aioquic	VHDCRS	VHDCRS UP	VHCRS U	VHDCR	VHDCRS UP	VHDCS P	VHDCRS U	VHDCRS	VHDCRS	VHCRS P	VHCR		VHCRS	VHDCR UP	VHDCRS MBUP	X	VHDCR P		V
~gQUIC	VH	VHR	VHR	H	VHDCRS 3	H	VHCS	VHR	VHS	VHR	VH	-	VHDCR 3	VHR	VH	VHDCR M3	-		-
quicker	HDC3	HDCRZ	HDC3	HDC3	HDC3	HDC3			HDC3	HDC3	HDC3						VHDCRZ P		
neqo	HD 3	HD			HD 3		HD	HD 3						HD				-	
Pandora	VHDCS	VHDCS	VH	VHDC			VHDCS	VHDCS	VHDCS	VHDCS	VH			VHDCS	VHDCS				VHDCS

Version Negotiation — Handshake — Stream Data — Connection Close — Resumption — 0-RTT — Stateless Retry — Migration — HTTP 3

mvfst / WinQuic / F5 / quiche / AppleQUIC / Quinn / aioquic / gQUIC

Mapping application protocols

- Mapping HTTP is part of the QUIC WG charter
- Other developments:
  - DNS over QUIC
  - NETCONF over QUIC
  - Unreliable datagram extension
  - P2P communication support

Table of contents

HTTP 1

- RFC 2616 was published in June 1999
- *The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems.*

HTTP 1-bis

- New set of RFCs published June 2014:
  - 7230: Message Syntax and Routing
  - 7231: Semantics and Content
  - 7232: Conditional Requests
  - 7233: Range Requests
  - 7234: Caching
  - 7235: Authentication

"HTTP is one of the most successful and widely-used protocols on the Internet today. However, its specification has several editorial issues. Additionally, after years of implementation and extension, several ambiguities have become evident, impairing interoperability and the ability to easily implement and use HTTP."

SPDY

- Announced by Google in 2009
  - True request pipelining
  - Message framing mechanism
  - Mandatory compression
  - Priority scheduling
  - Bi-directional communication
- First draft of HTTP 2 in 2012

HTTP 2

RFC 7450 published in May 2015:

*this specification describes an **optimized expression** of the semantics of the Hypertext Transfer Protocol (HTTP), referred to as HTTP version 2 (HTTP/2). HTTP/2 enables a **more efficient use of network resources** and a **reduced perception of latency** by introducing header field compression and **allowing multiple concurrent exchanges** on the same connection.*

HTTP/3 vs HTTP/2

- TCP's lack of visibility into parallel requests at the HTTP level limited the possible performance gains
- QUIC moves multiplexing and flow control into transport
- HTTP 3 simplifies HTTP 2 by relying on QUIC
  - Replace HPACK with QPACK to prevent HOL blocking
  - Different framing because some H2 concerns are already covered by the QUIC protocol layer

HTTP semantics are used for a broad range of services on the Internet. These semantics have commonly been used with two different TCP mappings, HTTP/1.1 and HTTP/2. HTTP/2 introduced a framing and multiplexing layer to improve latency without modifying the transport layer. However, TCP's lack of visibility into parallel requests in both mappings limited the possible performance gains.

The QUIC transport protocol incorporates stream multiplexing and per-stream flow control, similar to that provided by the HTTP/2 framing layer. By providing reliability at the stream level and congestion control across the entire connection, it has the capability to improve the performance of HTTP compared to a TCP mapping. QUIC also incorporates TLS 1.3 at the transport layer, offering comparable security to running TLS over TCP, but with improved connection setup latency (unless TCP Fast Open [RFC7413]} is used).

How to start

- Server advertises the availability of an equivalent HTTP/3 endpoint via the Alt-Svc response header or the HTTP/2 ALTSVC frame, using the appropriate ALPN token:
  - `Alt-Svc: h3=":50781"`
- On receipt, a client MAY attempt to establish a QUIC connection to the indicated address to start a HTTP session.
- Connectivity problems can result in QUIC connection establishment failure, in which case the client SHOULD continue using the existing connection.
- Servers MAY serve HTTP/3 on any UDP port

ALPN = Application-Layer Protocol Negotiation

Stream types

- Client-started bi-directional streams are request streams
- Server-started bi-directional streams are not allowed
- Unidirectional streams start with an explicit type:
  - Control stream
  - Push stream
  - Some types reserved for future use