QUIC

Google

Next generation multiplexed transport over UDP

lan Swett ianswett@google.com

Google

QUIC Quick UDP Internet Connections

- Multiplexed transport over UDP
- Reduce latency
- Rapid experimentation
- Open source development in Chromium

What does it replace?





Encryption

Comparable to TLS

Perfect forward secrecy, with more efficient handshake

IP spoofing protection Signed proof of address

more crypto details...

Google

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

Google

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

Google

NANOG on the road: QUIC

Deployment



Today: About half of Google to Chrome

Google

Debugging Tools: Chrome

chrome://net-internals

- Shows active QUIC sessions
- Captures every event
- Diagnoses handshake failures, packet loss, etc
- Important for filing Chromium <u>bugs</u>

0	0 0 0 ∫ Google Chrome - YouTube × Chrome://net-internals/#∈ ×												
← → C 🗋 chrome://net-internals/#events&q=type:QUIC_SESSION%20is:active													
Events Capturing events (33167)													
(?)	type	QUIC_SESSION is	active 8 of 1327	www.youtube.com Start Time: 2013-06-2	1:51:52.832	0							
	ID	Source Type	Description	t=1372359112832 [st=	0]	+QUIC_SESSION [dt=?]	1						
	3767	QUIC_SESSION	i1.ytimg.com	+=1372350112834 [e+=	21	> host = "www.youtube.com"							
	3771	QUIC_SESSION	s.ytimg.com	C-13/2333112034 [BC	- 1	> fin = false							
	3773	QUIC_SESSION	csi.gstatic.com			> length = 512 > offset = "0"							
	3786	QUIC_SESSION	www.google-analytics.com	+-1272250112024 (at-	21	> stream_id = 1							
J	3796	QUIC_SESSION	www.youtube.com	C=1372359112834 [SC=	2]	> encryption_level = 0							
	3800	QUIC_SESSION	www.gstatic.com			> packet_sequence_number = "1" > size = 564							
	3825	QUIC_SESSION	s2.googleusercontent.com	t=1372359112835 [st=	QUIC_HTTP_STREAM_SEND_REQUEST_HEADERS								
	3884	QUIC_SESSION	pagead2.googlesyndication.com			:method: GET							
						:path: /user/googlechrome :scheme: http							
						:version: HTTP/1.1							
						<pre>accept: text/html,application/xhtml+xml,application/xm accept_encoding: gzip.deflate.adch</pre>							
						accept-language: en-US,en;q=0.8							
						cache-control: max-age=0							
						user-agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_							
				t=1372359112835 [st=	3]	QUIC_SESSION_STREAM_FRAME_SENT							
						> length = 568							
						> offset = "0"	4						

Google

Debugging Tools: Wireshark

Parses

- Protocol: QUIC
- CID: Connection ID
- Seq: Sequence number
- Version: ie: Q024
- Public flags: 1 byte
- Payload: Encrypted

Filter			 Expression 	ession C	lear Apply	Save		
No.	Time	Source	Destination	Protoc 🕶	Length Info			
985	14.027869000	173.194.46.73	10.1.10.14	QUIC	1392 CID:	3182875774876983667,	Seq:	1
986	14.028834000	10.1.10.14	173.194.46.73	QUIC	1392 CID:	3182875774876983667,	Seq:	2
989	14.065914000	173.194.46.73	10.1.10.14	QUIC	1392 CID:	3182875774876983667,	Seq:	2
990	14.066812000	10.1.10.14	173.194.46.73	QUIC	79 CID:	3182875774876983667,	Seq:	3
991	14.194009000	10.1.10.14	173.194.46.73	QUIC	1392 CID:	3182875774876983667,	Seq:	4
992	14.194164000	10.1.10.14	173.194.46.73	QUIC	350 CID:	3182875774876983667,	Seq:	5
993	14.231536000	173.194.46.73	10.1.10.14	QUIC	85 CID:	3182875774876983667,	Seq:	3
994	14.258228000	173.194.46.73	10.1.10.14	QUIC	353 CID:	3182875774876983667,	Seq:	4
995	14.268285000	2601:6:2c01:9300:69a8:9	2607:f8b0:4004:a::12	QUIC	1412 CID:	2735399198252988334,	Seq:	1
997	14.270807000	10.1.10.14	216.58.216.238	QUIC	1392 CID:	2060901289831796684,	Seq:	1
998	14.273189000	10.1.10.14	173.194.46.76	QUIC	1392 CID:	16164325528471686122,	Seq:	1
999	14.277601000	10.1.10.14	173.194.46.73	QUIC	1392 CID:	9176532438181928584,	Seq:	1
1000	14.278560000	10.1.10.14	173.194.46.73	QUIC	1392 CID:	9176532438181928584,	Seq:	2
1001	14.278618000	10.1.10.14	173.194.46.73	QUIC	515 CID:	9176532438181928584,	Seq:	3
1002	14.284072000	10.1.10.14	173.194.46.73	QUIC	82 CID:	3182875774876983667,	Seq:	6
1003	14.295209000	2607:f8b0:4004:a::12	2601:6:2c01:9300:69a	8 QUIC	1412 CID:	2735399198252988334,	Seq:	1
1004	14.296658000	2601:6:2c01:9300:69a8:9	2607:f8b0:4004:a::12	QUIC	99 CID:	2735399198252988334,	Seq:	2
1005	14.309132000	216.58.216.238	10.1.10.14	QUIC	1392 CID:	2060901289831796684,	Seq:	1
1006	14.312428000	173.194.46.76	10.1.10.14	QUIC	1392 CID:	16164325528471686122,	Seq:	1
•)+
D Fra D Eth D Int	me 981: 1392 b ernet II, Src: ernet Protocol	ytes on wire (11136 bit Apple_bc:da:74 (78:31: Version 4, Src: 10.1.1	s), 1392 bytes captur cl:bc:da:74), Dst: Ne 0.14 (10.1.10.14), Ds	ed (11136) tgear_bf:79 st: 173.194	bits) on inte 9:04 (c4:04:1 .46.73 (173.1	rface 0 (outbound) 5:bf:79:04) 94.46.73)		
> Use	r Datagram Pro	tocol, Src Port: 51863	(51863), Dst Port: 80	(80)				
7 QUI	C (Quick UDP I	nternet Connections)						
ÞР	ublic Flags: 0	0x0d						
C	ID: 3182875774	1876983667						
v	ersion: Q024							
S	equence: 1							
Р	avload: 9f8da	bbb0e0724d965b22dc01a00	1000443484c4f130000.					

Google

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

Google

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

Google

Google

Source: QUIC in Chromium

Further reading: <u>Chromium Site</u> <u>Design doc and rationale</u>, <u>Wire spec</u>, <u>Crypto doc</u>

Mailing list: proto-quic@chromium.org

lan Swett ianswett@google.com