

QUIC

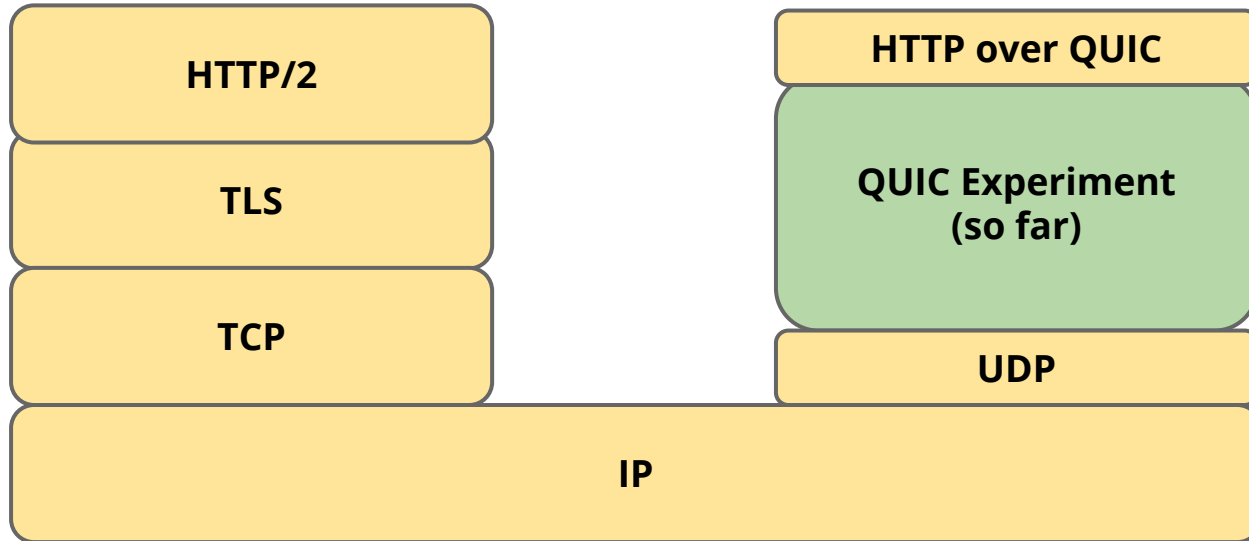
A New Internet Transport

Presenter: Jana Iyengar

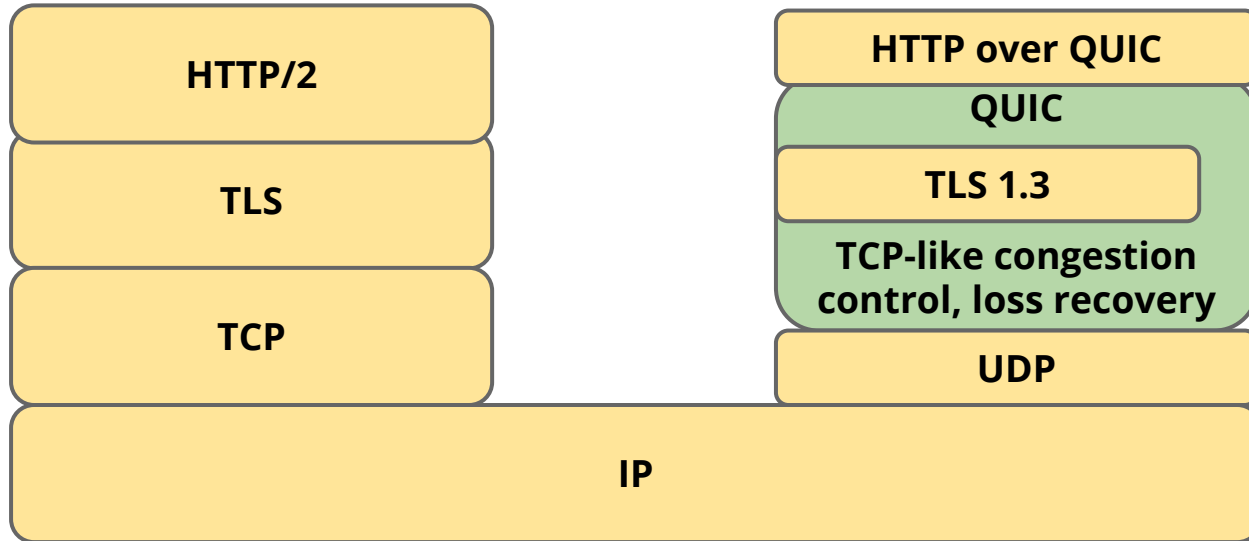
QUIC and the IETF

Nov 2013	Early design and experience (TSVAREA)
Mar 2015	QUIC handshake (SAAG)
Mar 2015 onwards	Replacing QUIC's handshake with TLS1.3
July 2015	BarBoF, experimental results
Nov 2015	Cubic bug in QUIC, TCP (TCPM)
July 2016	BoF NOW!

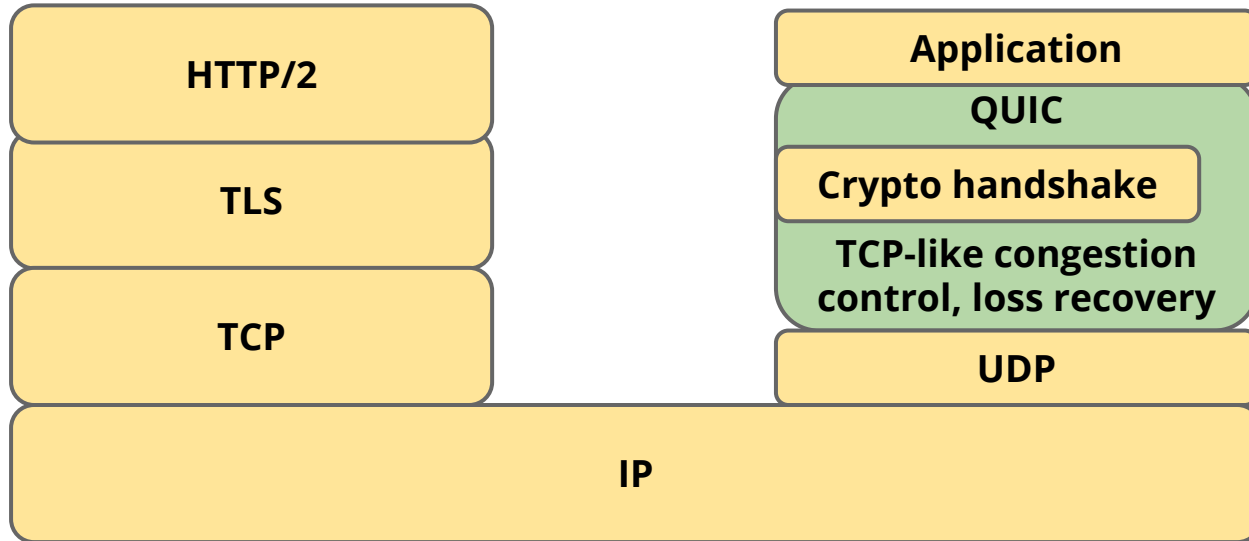
The QUIC Experiment



The IETF Proposal



Standardized QUIC



QUIC Design Aspirations

- Deployability and evolvability

QUIC Design Aspirations

- Deployability and evolvability
- Low latency connection establishment

QUIC Design Aspirations

- Deployability and evolvability
- Low latency connection establishment
- Multistreaming and per-stream flow control

QUIC Design Aspirations

- Deployability and evolvability
- Low latency connection establishment
- Multistreaming and per-stream flow control
- Better loss recovery and flexible congestion control

QUIC Design Aspirations

- Deployability and evolvability
- Low latency connection establishment
- Multistreaming and per-stream flow control
- Better loss recovery and flexible congestion control
- Resilience to NAT-rebinding

QUIC Design Aspirations

- Deployability and evolvability
- Low latency connection establishment
- Multistreaming and per-stream flow control
- Better loss recovery and flexible congestion control
- Resilience to NAT-rebinding
- Multipath for resilience and load sharing

Deployability and Evolvability

Uses UDP as the substrate

enables deployment through various middleboxes

userspace implementation enables rapid deployment

Deployability and Evolvability

Uses UDP as the substrate

enables deployment through various middleboxes

userspace implementation enables rapid deployment

Version negotiation

enables protocol wire format evolution

Deployability and Evolvability

Uses UDP as the substrate

enables deployment through various middleboxes

userspace implementation enables rapid deployment

Version negotiation

enables protocol wire format evolution

Fully authenticated and mostly encrypted headers

avoids network ossification

QUIC Streams

Multiplexed streams within a transport connection

multiple streams avoids HoL blocking

shared congestion control and loss recovery

two levels of flow control: stream and connection

Congestion Control & Loss Recovery

QUIC builds on decades of experience with TCP

Congestion Control & Loss Recovery

QUIC builds on decades of experience with TCP

Incorporates TCP best practices

TCP-like congestion control (NewReno, Cubic)

Congestion Control & Loss Recovery

QUIC builds on decades of experience with TCP

Incorporates TCP best practices

TCP-like congestion control (NewReno, Cubic)

FAK, TLP, F-RTO, Early Retransmit, ...

(also, time-based loss detection)

Congestion Control & Loss Recovery

QUIC builds on decades of experience with TCP

Incorporates TCP best practices

TCP-like congestion control (NewReno, Cubic)

FAACK, TLP, F-RTO, Early Retransmit, ...

(also, time-based loss detection)

Richer signaling than TCP

Richer Signaling Than TCP

Retransmitted packets consume new sequence number

no retransmission ambiguity

prevents loss of retransmission from causing RTO

Richer Signaling Than TCP

Retransmitted packets consume new sequence number

no retransmission ambiguity

prevents loss of retransmission from causing RTO

More verbose ACK

TCP supports up to 3 SACK ranges

QUIC supports up to 256 ACK ranges

explicit packet receive times

enables ACK decimation

QUIC Implementations

Chromium (open source)

<https://cs.chromium.org/chromium/src/net/quic/>

quic-go (open source implementation in Go)

<https://github.com/lucas-clemente/quic-go>

Christian Huitema's implementation

Debugging Tools: Wireshark

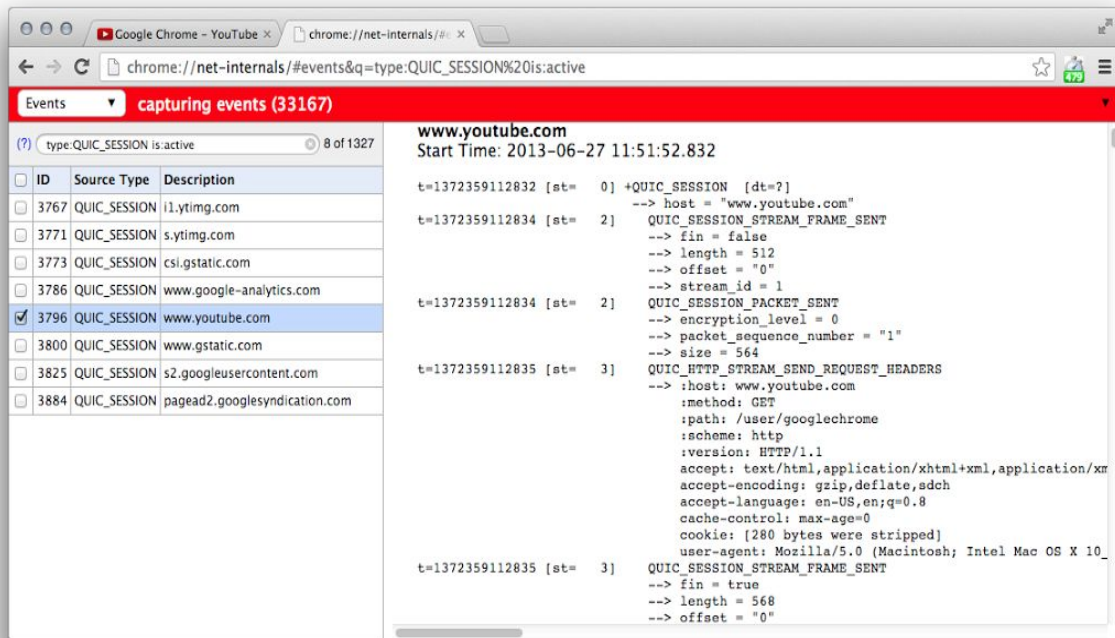
No.	Time	Source	Destination	Protoc	Length	Info
985	14.027869000	173.194.46.73	10.1.10.14	QUIC	1392	CID: 3182875774876983667, Seq: 1
986	14.028834000	10.1.10.14	173.194.46.73	QUIC	1392	CID: 3182875774876983667, Seq: 2
989	14.065914000	173.194.46.73	10.1.10.14	QUIC	1392	CID: 3182875774876983667, Seq: 2
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	2601:6:2c01:9300:69a8:92607:f8b0:4004:a::12	10.1.10.14	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	2601:6:2c01:9300:69a8	QUIC	1412	CID: 2735399198252988334, Seq: 1
1004	14.296658000	2601:6:2c01:9300:69a8:92607:f8b0:4004:a::12	10.1.10.14	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

Filter: Expression... Clear Apply Save

Frame 981: 1392 bytes on wire (11136 bits), 1392 bytes captured (11136 bits) on interface 0 (outbound)
Ethernet II, Src: Apple_bc:da:74 (78:31:c1:bc:da:74), Dst: Netgear_bf:79:04 (c4:04:15:bf:79:04)
Internet Protocol Version 4, Src: 10.1.10.14 (10.1.10.14), Dst: 173.194.46.73 (173.194.46.73)
User Datagram Protocol, Src Port: 51863 (51863), Dst Port: 80 (80)
QUIC (Quick UDP Internet Connections)
Public Flags: 0x0d
CID: 3182875774876983667
Version: 0024
Sequence: 1
Payload: 9f8da5bbb0e0724d965b22dc01a001000443484c4f130000...

Debugging Tools: Chrome

chrome://net-internals



The screenshot shows the Chrome DevTools 'net-internals' page. The top bar indicates 'capturing events (33167)'. A table on the left lists captured events, with the event for 'www.youtube.com' (ID 3796) selected. The right pane displays the details for this event, showing a QUIC session start and subsequent packet sends.

ID	Source Type	Description
3767	QUIC_SESSION	l1.ytimg.com
3771	QUIC_SESSION	s.ytimg.com
3773	QUIC_SESSION	csi.gstatic.com
3786	QUIC_SESSION	www.google-analytics.com
<input checked="" type="checkbox"/>	QUIC_SESSION	www.youtube.com
3800	QUIC_SESSION	www.gstatic.com
3825	QUIC_SESSION	s2.googleusercontent.com
3884	QUIC_SESSION	pagead2.googlesyndication.com

```
www.youtube.com
Start Time: 2013-06-27 11:51:52.832

t=1372359112832 [st= 0] +QUIC_SESSION [dt=?]
--> host = "www.youtube.com"
t=1372359112834 [st= 2] QUIC_SESSION_STREAM_FRAME_SENT
--> fin = false
--> length = 512
--> offset = "0"
--> stream_id = 1
t=1372359112834 [st= 2] QUIC_SESSION_PACKET_SENT
--> encryption_level = 0
--> packet_sequence_number = "1"
--> size = 564
t=1372359112835 [st= 3] QUIC_HTTP_STREAM_SEND_REQUEST_HEADERS
--> :host: www.youtube.com
:method: GET
:path: /user/googlechrome
:scheme: http
:version: HTTP/1.1
accept: text/html,application/xhtml+xml,application/xml;
accept-encoding: gzip,deflate,sdch
accept-language: en-US,en;q=0.8
cache-control: max-age=0
cookie: [280 bytes were stripped]
user-agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_
t=1372359112835 [st= 3] QUIC_SESSION_STREAM_FRAME_SENT
--> fin = true
--> length = 568
--> offset = "0"
```