MQTT Over QUIC

William Yang

2022-02-17 Thurs

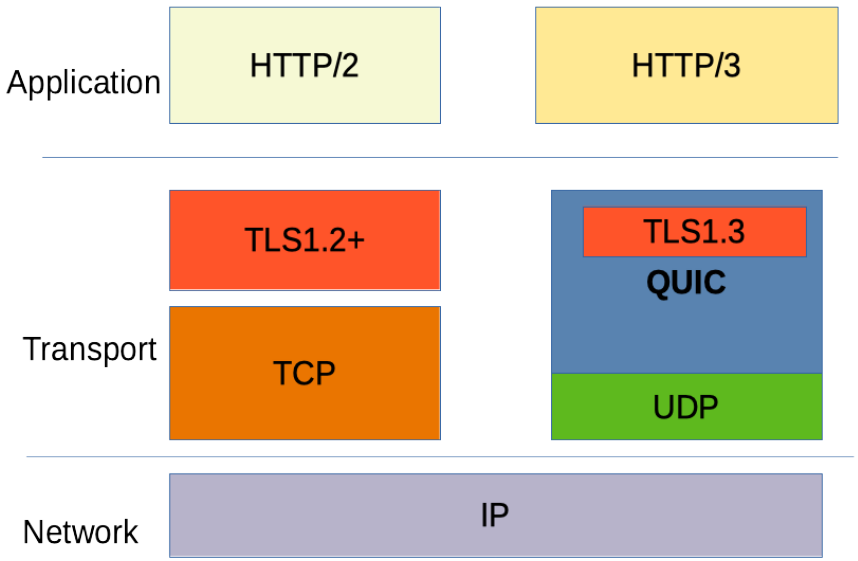
Outline

About Us

Me: William Yang, Software Developer from EMQ
EMQX: An Open-Source, Distributed MQTT Broker for IoT
`https://github.com/emqx/emqx`

IETF HTTP/3

In May 2021, the IETF (Internet Engineering Task Force) decided to call HTTP mapping over QUIC "HTTP/3" and made it a worldwide standard.

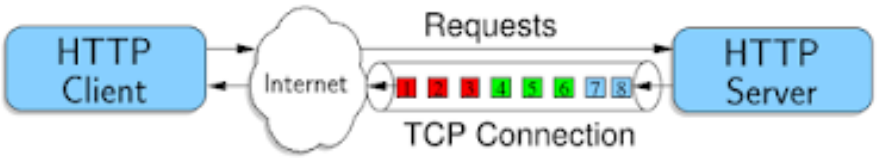


QUIC

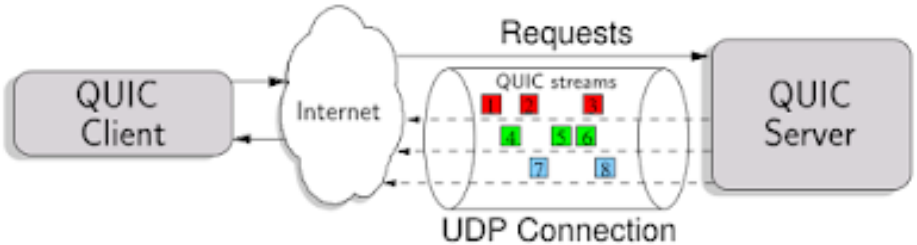
- Secure Transport Protocol, Over UDP/TLS1.3
- Connection-Oriented, reliable delivery
- Reduce overhead during connection setup (1-RTT, or even 0-RTT)
- Multiplexing in one connection with 'Streams'
- Congestion Control at the application level
- Multipath, path migration
- QUIC stack can be implemented in the user-space(application-space)

TCP/SSL vs QUIC

	TCP/TLS1.2	QUIC
Connection Setup	3-4 RTT	0-1 RTT
Head-of-line Blocking	Yes	NO
Congestion control	Kernel	Userspace
Flow control	Connection	Connection&Stream
Network Address Migration	Reconnect only	Supported



a) HTTP



b) QUIC

Could MQTT run over QUIC?

In MQTT 5.0: "The protocol runs over TCP/IP, or over other network protocols that provide ordered, lossless, bidirectional connections"

QUIC:

- [x] Connection-Oriented
- [x] Ordered (In the same stream)
- [x] Bidirectional

PoC in EMQX 5.0

Since early 2021, EMQX has a PoC of running MQTT over QUIC for both client and server side.

- Transport replacement
- One bidirectional stream per connection
- Performance test

New opportunity | Connection Multiplexing

1. MQTT Topic per Stream
   For independent topics, each topic could have independent stream within same connection to eliminate the effects from other topics such as long blocking or flowcontrol at the receiving side.
2. Streams for different QoS, optimized flow control
   example: In flow control, QoS 0 traffic should give way to high QoS traffic.
3. Separate controlling messages into different streams.
   MQTT Control messages could be sent in unidirectional or bidirectional streams for asynchronous processing.
   example: Client could send UNSUBSCRIBE messages though 'controlling stream' asynchronously to ask server side stop sending the data which is no longer interested.

New opportunity | Address Migration

1. MQTT Client address migrations
   - lossy network
   - when at the edge of two networks
     Probe network path of both
   - NAT rebinding
     For publishing only clients
2. MQTT Server address migrations (Server redirection)
   Support stateful server side switchover in the cluster without massive reconnects from clients.
3. OPs friendly, loadbalancer switchover without massive reconnects.

Scope of the work

We have a good reference: 'HTTP/3' (draft)

1. Basic
   One bidirectional stream in one connection. Treat 'Connection + Stream' as 'Connection' in TCP/SSL Close connection when stream is closed.
2. Advanced
   - 2.1 ALPN service discovering (TCP/SSL <-> QUIC switching)
   - 2.2 Define some trasport profile for how multiple streams is utilized Stream per QoS Stream per Topic
   - 2.3 Define the scenario of recoverable connection After sending a DISCONNECT message, the connection should be unrecoverable.

What's Next?

- Invite TC feedback
- Request the TC approve initial development of a work product likely to become a Committee Node Draft.
- TC review initial offering and decide next step e.g., stop or if a new SC is required to progress.