

# Understanding QUIC wire protocol



Anuradha Niroshan

Follow

Oct 1, 2017 · 4 min read

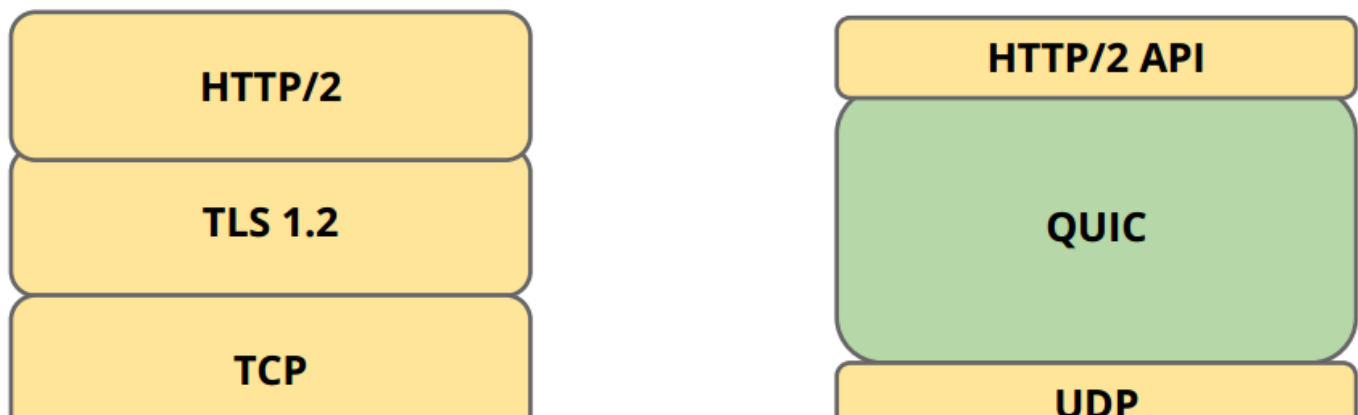
*Intention of this article is to give a simple explanation of how QUIC protocol works. In the first section I have introduced what is QUIC and the reasons for its emergence. Then I have discussed the wire protocol of QUIC with simple diagrams. Let's Start then.*

I encountered QUIC while I was searching for a project for GSOC 2017. Ok what's QUIC. It stands for Quick UDP internet connections.

QUIC is a new multiplexed and secure transport built atop UDP, designed from the ground up and optimized for HTTP/2 semantics. While built with HTTP/2 as the primary application protocol, QUIC builds on decades of transport and security experience, and implements mechanisms that make it attractive as a modern general-purpose transport. QUIC provides multiplexing and flow control equivalent to HTTP/2, security equivalent to TLS, and connection semantics, reliability, and congestion control equivalent to TCP.

## Where Does it fit in the stack ?

QUIC is a transport atop UDP. The exact position where QUIC fits can be shown by the following diagram. Theoretically it fills the gap between transport layer UDP protocol and application layer HTTP/2 protocol.





IP

QUIC — where it fits

## Why QUIC ?

Main goal of designing QUIC is to reduce web latency. This is achieved by combining the speed and possibilities of UDP with reliability of TCP.

To start a TCP connection 3 way handshake is performed. This adds a significant delay when creating a new connection. We need to negotiate TLS, to create a secure, encrypted, https connection, even more network packets have to be sent back and forth.

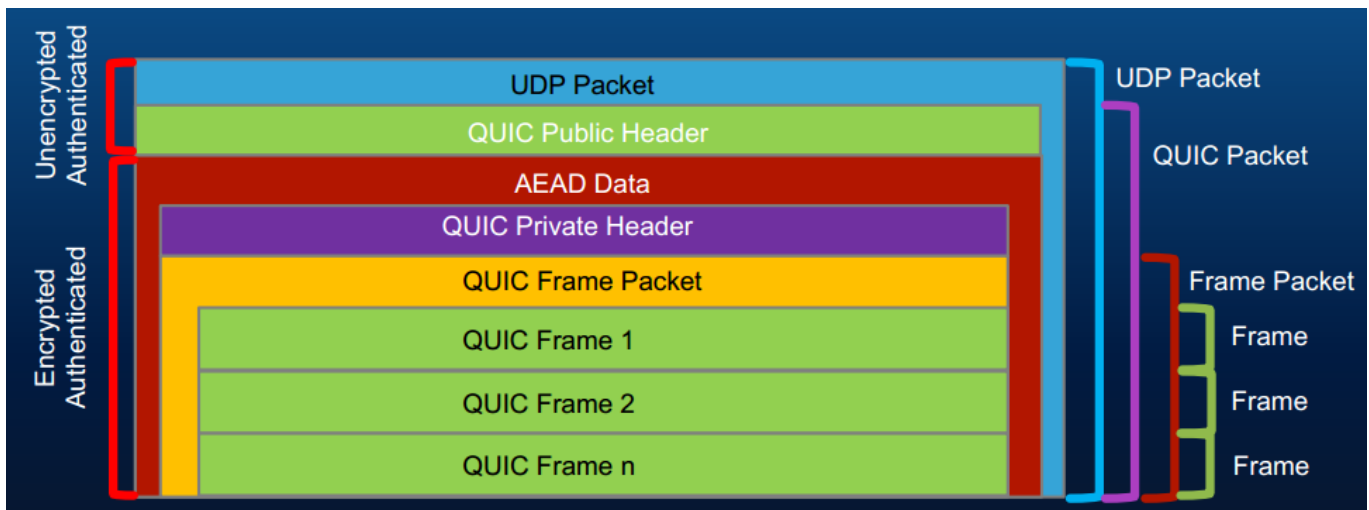
On the other hand UDP is connectionless. That means UDP doesn't establish connections as TCP does, so UDP does not perform this 3-way handshake and for this reason, it is referred to as an unreliable protocol. That doesn't mean UDP can't transfer data, it just doesn't negotiate how the connection will work, UDP just transmits and hopes for the best. The benefit is less time spent on the network to validate packets, the downside is that in order to be reliable, something has to be built *on top of* UDP to confirm packet delivery.

QUIC is very similar to TCP+TLS+HTTP2, but implemented on top of UDP. Key advantages of QUIC over TCP+TLS+HTTP2 include:

- Connection establishment latency
- Improved congestion control
- Multiplexing without head-of-line blocking
- Forward error correction
- Connection migration

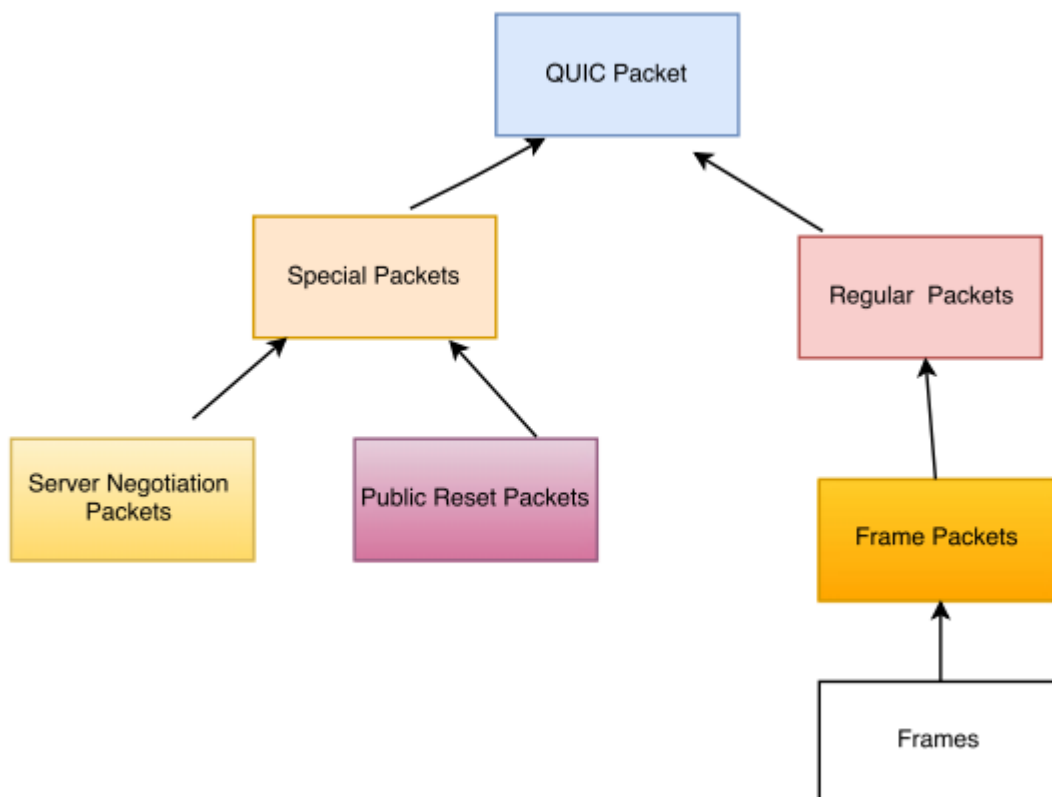
## Wire protocol

QUIC packet structure is shown by the following diagram.



Note : From QUIC version 'Q034' onwards private Header field is removed

All QUIC packets on the wire begin with public header sized between 1 to 51 bytes and it gives details about the rest of the packet. Decoding should be started from public header. QUIC packets can be categorized into special packets and Regular packets.



QUIC Packet Hirarchy

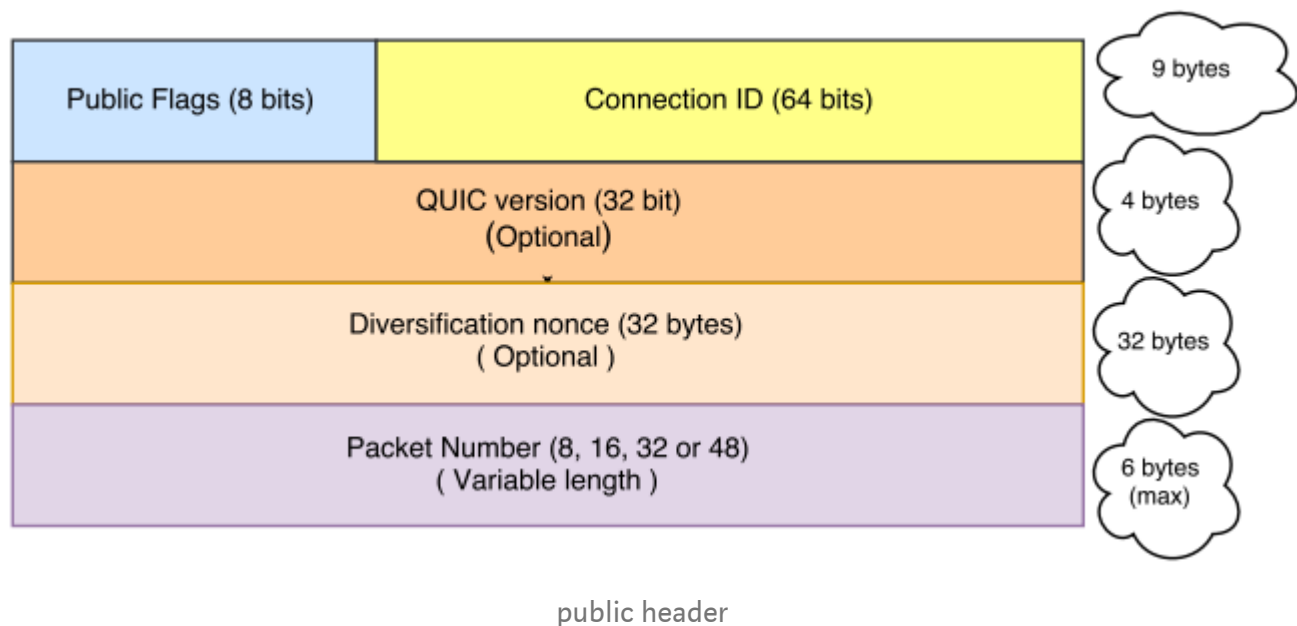
There are two types of special packets

1. Version Negotiation Packets — These packets are sent by servers to agree upon a QUIC version to use for communication.
2. Public Reset Packets — These packets are use to terminate communication

Frame Packets can be categorized as Regular Packets. Frame Packet consist of number of frames which has a specific type and payload.

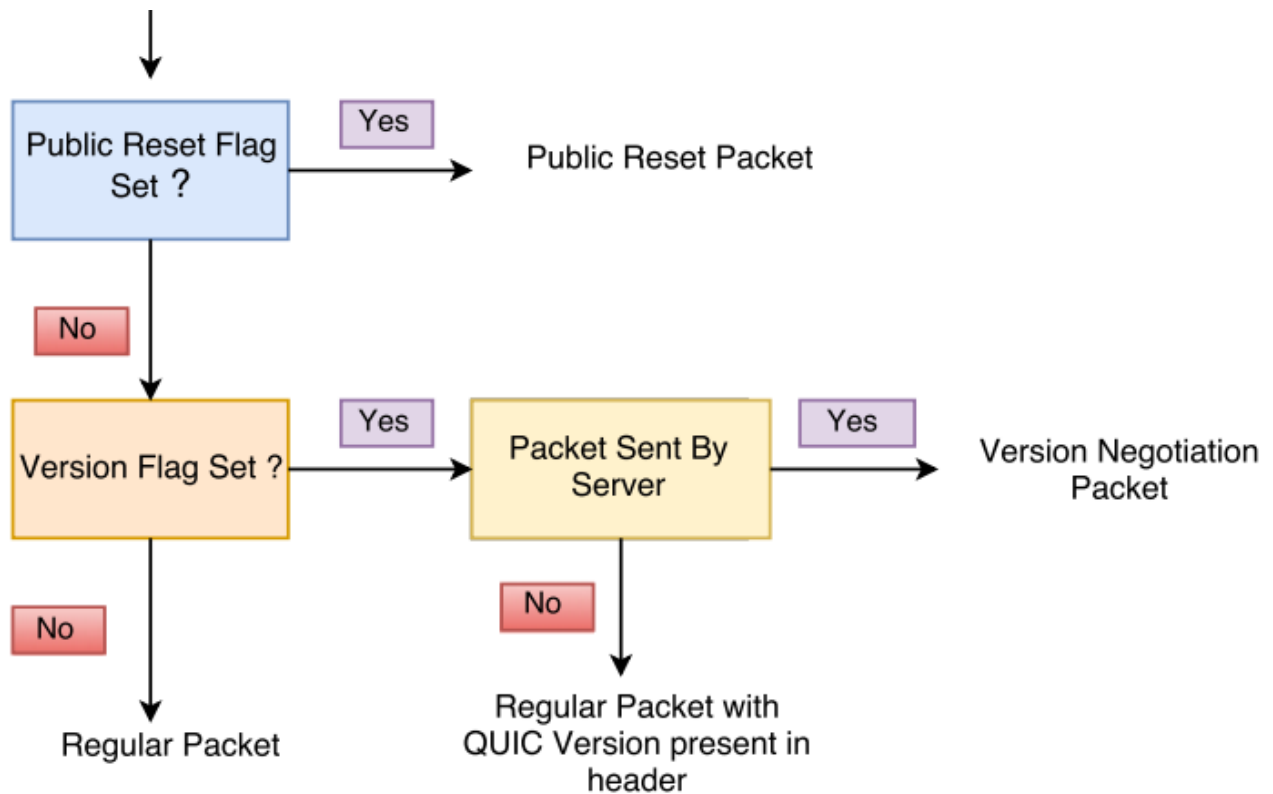
All integer values used in QUIC, including length, version, and type, are in little-endian byte order, and not in network byte order.

## Public header



Identify public flags (first byte of the Public header). It possess following details.

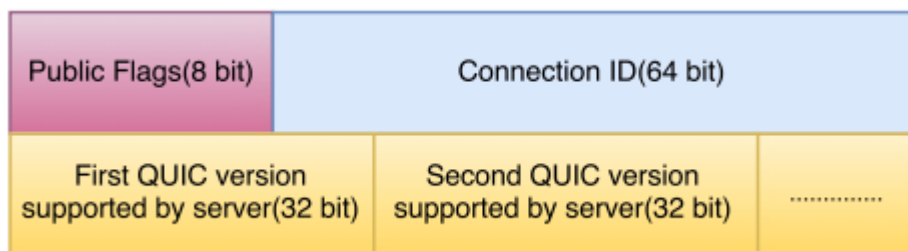
- Indicate presence of QUIC version (0x01)
- Indicate Reset packet (0x02)
- Indicate presence of diversification nonce(0x04)
- Indicate presence full 8 byte connectionId(0x08)
- Packet Number size(0x30)



Identify the type of the QUIC packet using flags

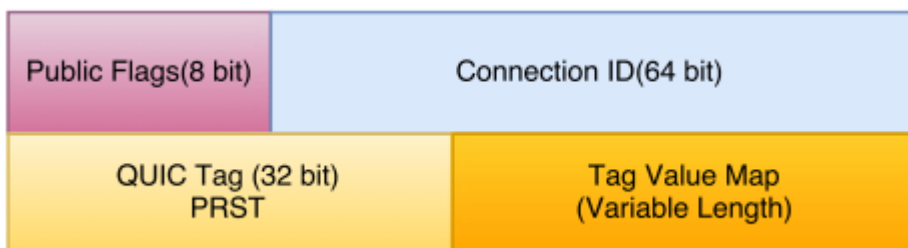
## Version negotiation packet

This packet is sent only by servers. It contains one or more QUIC versions supported by the server. It has the following format.



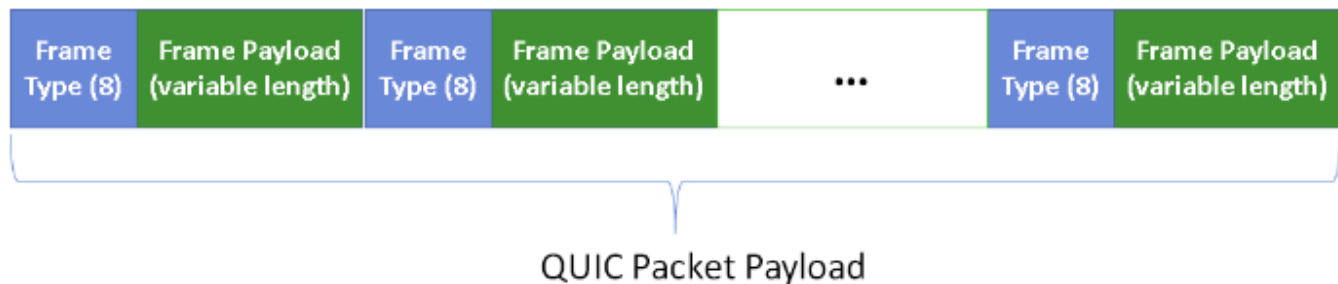
Version negotiation packet

## Public Reset Packet



public reset packet

## Frame Packet



Hope you got a good understanding of QUIC packet and what each bit represents. This is useful if you are implementing a codec for QUIC protocol. Having an understanding on how the protocol works is a must for that.

## References

[1]<https://www.chromium.org/quic>

[2]<https://ma.ttias.be/googles-quic-protocol-moving-web-tcp-udp/>

[3][http://www.inetdaemon.com/tutorials/internet/tcp/3-way\\_handshake.shtml](http://www.inetdaemon.com/tutorials/internet/tcp/3-way_handshake.shtml)

Quic   Http2   Protocol   Udp   Tcp Ip

[About](#)   [Help](#)   [Legal](#)