Use IETF Packet Header Wire Format in Google QUIC

Status: Final
Authors: fayang@chromium.org
Last Updated: 2019-06-17

Overview

In versions 43 and below, Google QUIC packets on the wire begin with a public header:

```
+-+-+-+-+-+-+-+-+
|0|0|P|P|C|D|R|V|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                     [Connection ID (64)]                      +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

As part of making QUIC IETF compatible, the QUIC packet header is changing to one of two formats, starting in Google QUIC version 46:

1) Long Header (used for packets that are sent prior to the completion of version negotiation and establishment of 1-RTT keys):

```
+-+-+-+-+-+-+-+-+
|1|1|T|T|R|R|P|P|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Version (32)                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|DCIL(4)|SCIL(4)|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Destination Connection ID (0/64)              ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Source Connection ID (0/64)                 ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Packet Number (8/16/24/32)              ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

2) Short Header (used after the version and 1-RTT keys are negotiated):

```
+-+-+-+-+-+-+-+-+
|0|1|R|R|R|R|P|P|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Destination Connection ID (0..144)           ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Packet Number (8/16/24/32)              ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

And the following packet types are defined for packets with long header: Initial, Retry, Handshake and 0-RTT Protected (details can be found in QUIC IETF draft).

This document describes how to support the new header formats and also discusses how to support both Google and IETF header formats during the transition.

Future versions of Google QUIC will have full variable length asymmetric connection ID support.

Assign Long Header Packet Types

Before connection encryption level switches to forward secure, long packet type is purely determined by the packet's encryption level (Retry type is not used yet):

Encryption level	Packet Type
ENCRYPTION_INITIAL	INITIAL
ENCRYPTION_HANDSHAKE	HANDSHAKE
ENCRYPTION_ZERO_RTT	ZERO_RTT_PROTECTED

Variable Length Connection ID

A Google QUIC connection is identified by an 8-byte connection ID, and server to client load balancing is not supported. Such that in IETF header format, both source and destination connection ID are the same:

Packet Header Type	From	To	Connection ID
Long header	Client	Server	8-byte Destination Connection ID 0-byte Source Connection ID
Long header	Server	Client	0-byte Destination Connection ID 8-byte Source Connection ID
Short header	Client	Server	8-byte Destination Connection ID
Short header	Server	Client	0-byte Destination Connection ID

Received packets with unexpected length of connection IDs are dropped by an endpoint.

When Receiving Packets

Client side is trivial, as it starts with one version. When a server receives a packet, use both the most significant bit and the demultiplexing bit (0x40) of the first byte to determine whether this is a Google QUIC or IETF QUIC packet. If the significant bit is set, this packet is considered as an IETF long header packet. Else if demultiplexing bit is set, this packet is considered as an IETF short header packet. Else the packet is considered as a Google QUIC packet.

Version Negotiation and Stateless Reset

A server needs to send the right version negotiation packet depends on either the client speaks IETF QUIC or Google QUIC.

In Google QUIC, public reset can be used to terminate a connection both pre and post handshake. Stateless reset uses in IETF QUIC can only be used post handshake because the stateless reset token is provided when handshake succeeds, such that a connection close packet is used to terminate a connection pre-handshake.

Deployment

This IETF packet header format is implemented and deployed as QUIC version 46.