Export of QUIC Information in IP Flow Information Export (IPFIX)

draft-lin-opsawg-ipfix-quic-header-00

Changwang Lin (New H3C)
Yisong Liu (China Mobile)
Yao Liu (ZTE)

IETF-122, March 2025

Background and Motivation

- **QUIC is a UDP-based transport protocol and is widely used in various network services.**
- **How to collect and export QUIC information in forwarding traffic is not yet defined.**
- **This document proposes exporting the QUIC information by extending IPFIX.**

```
Long Header Packet {
  Header Form (1) = 1,
  Fixed Bit (1) = 1,
  Long Packet Type (2),
  Type-Specific Bits (4),
  Version (32),
  Destination Connection ID Length (8),
  Destination Connection ID (0..160),
  Source Connection ID Length (8),
  Source Connection ID (0..160),
  Type-Specific Payload (..),
}
```

```
Short Header Packet
1-RTT Packet {
  Header Form (1) = 0,
  Fixed Bit (1) = 1,
  Spin Bit (1),
  Reserved Bits (2),
  Key Phase (1),
  Packet Number Length (2),
  Destination Connection ID (0..160),
  Packet Number (8..32),
  Packet Payload (8..),
}
```

```
Packet Payload {
  Frame (8..) ...,
}

Frame {
  Frame Type (i),
  Type-Dependent Fields (..),
}

STREAM Frame {
  Type (i) = 0x08..0x0f,
  Stream ID (i),
  [Offset (i)],
  [Length (i)],
  Stream Data (..),
}
```

QUIC message format

Main Content (1)

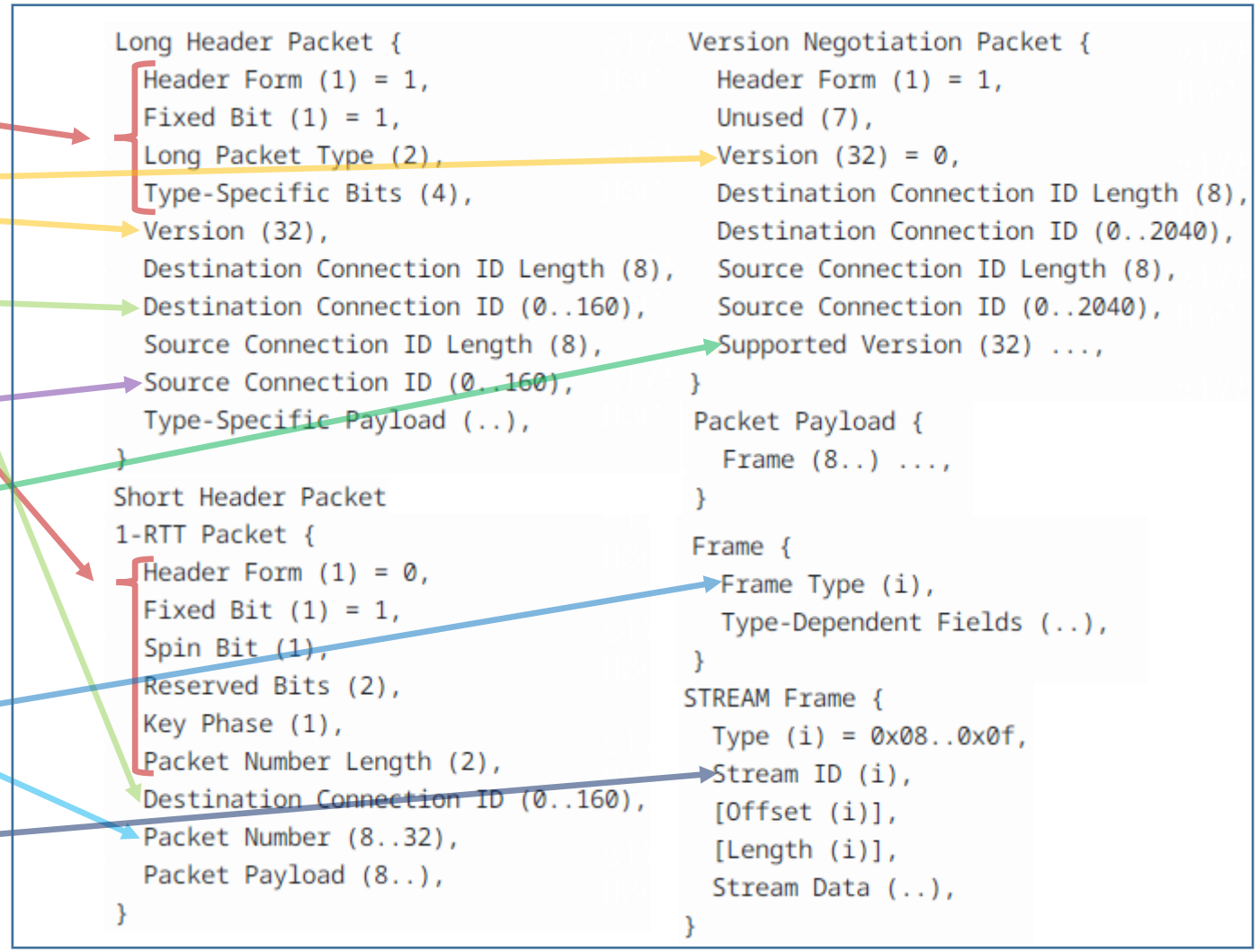
- **Specifies a set of new IPFIX Information Elements for QUIC**
- **Currently, 8 Information Elements have been defined.**

Main Content (2)

8 Information Elements are:

- **quicHeaderFlag**
- **quicVersion**
- **quicDestinationConnectionID**
- **quicSourceConnectionID**
- **quicSupportedVersion**
- **quicPacketNumber**
- **quicFrameType**
- **quicStreamID**

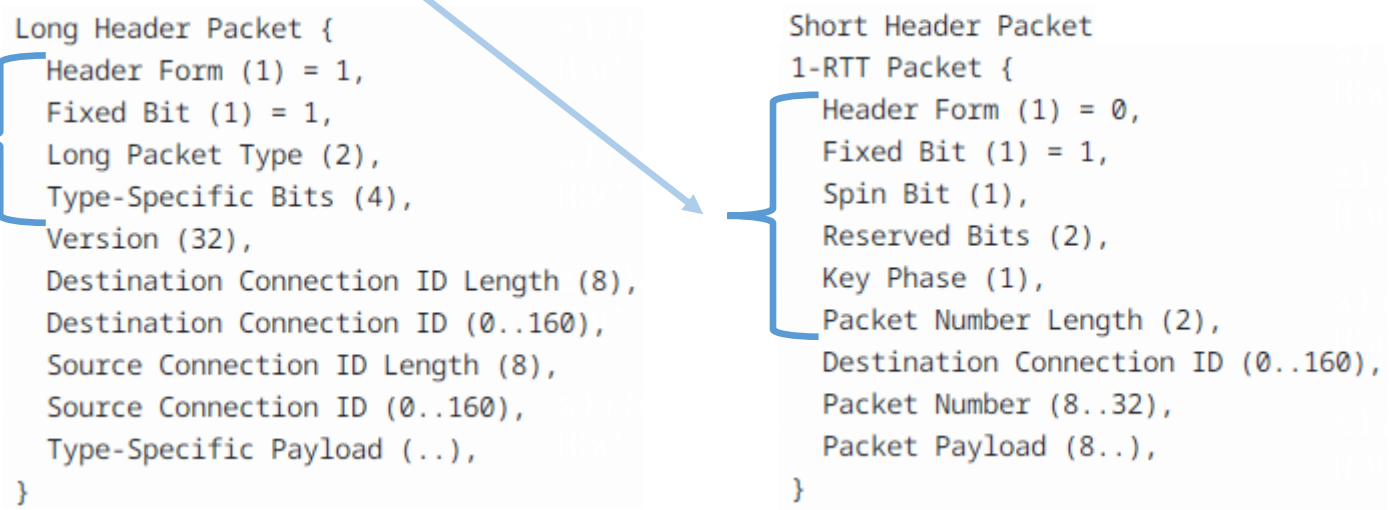
QUIC message format



Detailed Information Elements (1)

- **quicHeaderFlag**

  8-bit flag defined in the QUIC Header, as the first byte of QUIC Packet, which primarily indicates the Packet Type and the Length of followed fields.

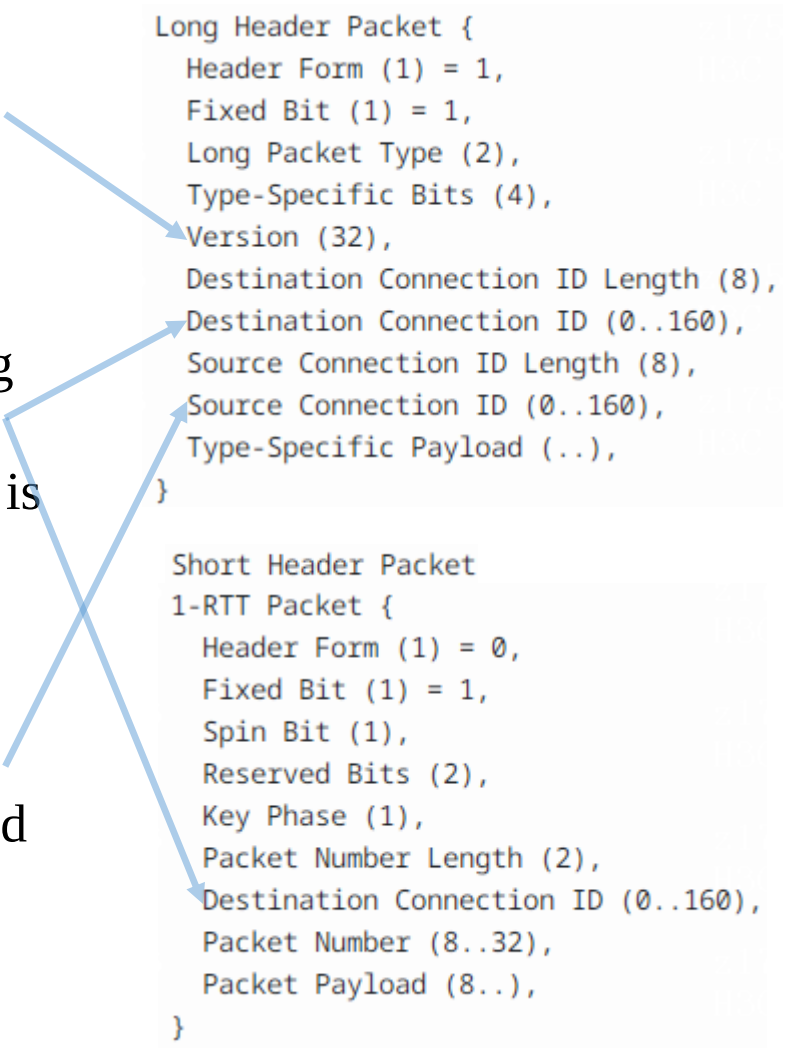


Detailed Information Elements (2)

- **quicVersion**
  32-bit QUIC Version that is in use and determines how the rest of the protocol packet fields are interpreted.

- **quicDestinationConnectionID**
  Defined as a series of octets in IPFIX. Included in the Long Header or Short Header of QUIC Packet. The Destination Connection ID is chosen by the recipient of the packet and is used to provide consistent routing.

- **quicSourceConnectionID**
  Defined as a series of octets in IPFIX. Located in the Long Header of QUIC Packet. The Source Connection ID is used to set the Destination Connection ID used by the peer.



Detailed Information Elements (3)

- **quicSupportedVersion**

  32-bit Supported Version included by Version Negotiation packet with the Long Header. Indicates the supported QUIC version of server.

```
Version Negotiation Packet {
  Header Form (1) = 1,
  Unused (7),
  Version (32) = 0,
  Destination Connection ID Length (8),
  Destination Connection ID (0..2040),
  Source Connection ID Length (8),
  Source Connection ID (0..2040),
  Supported Version (32) ...,
}
```

- **quicPacketNumber**

  32-bit unsigned integer. Appears in some packet types such as Initial packet, 0-RTT packet and Handshake packet. The underlying packet number increases with each packet sent in a given packet number space.

```
Short Header Packet
1-RTT Packet {
  Header Form (1) = 0,
  Fixed Bit (1) = 1,
  Spin Bit (1),
  Reserved Bits (2),
  Key Phase (1),
  Packet Number Length (2),
  Destination Connection ID (0..160),
  Packet Number (8..32),
  Packet Payload (8..),
}
```

Detailed Information Elements (4)

- **quicFrameType**

  Frame Type that indicates the type of Frame contained in the Payload of QUIC Packet. The Frame Type value uses a variable-length integer encoding which means that integers are encoded on 1, 2, 4, or 8 bytes and can encode 6-, 14-, 30-, or 62-bit values, respectively. Here, defined as 64-bit unsigned integer.

```
Packet Payload {
  Frame (8..) ...,
}
```

```
Frame {
  Frame Type (i),
  Type-Dependent Fields (..),
}
```

- **quicStreamID**

  The Stream ID included in the Frame related to Stream such as RESET_STREAM frame, STOP_SENDING frame, STREAM frame and MAX_STREAM_DATA frame. According to the Frame type, Indicates which stream the Frame is located or applied. A 62-bit integer.
  Here, defined as 64-bit unsigned integer.

```
STREAM Frame {
  Type (i) = 0x08..0x0f,
  Stream ID (i),
  [Offset (i)],
  [Length (i)],
  Stream Data (..),
}
```

Next Steps

- **Request more review to improve the draft.**
- **Any comments welcomed.**

Thank You!