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

draft-lin-opsawg-ipfix-quic-header-04

Presenter: Changwang Lin
Co-author: Changwang Lin (New H3C)
Yisong Liu (China Mobile)
Yao Liu (ZTE)
Xueting Li (China Telecom)

IETF-126, Vienna, July 2026

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 (..),
}
```

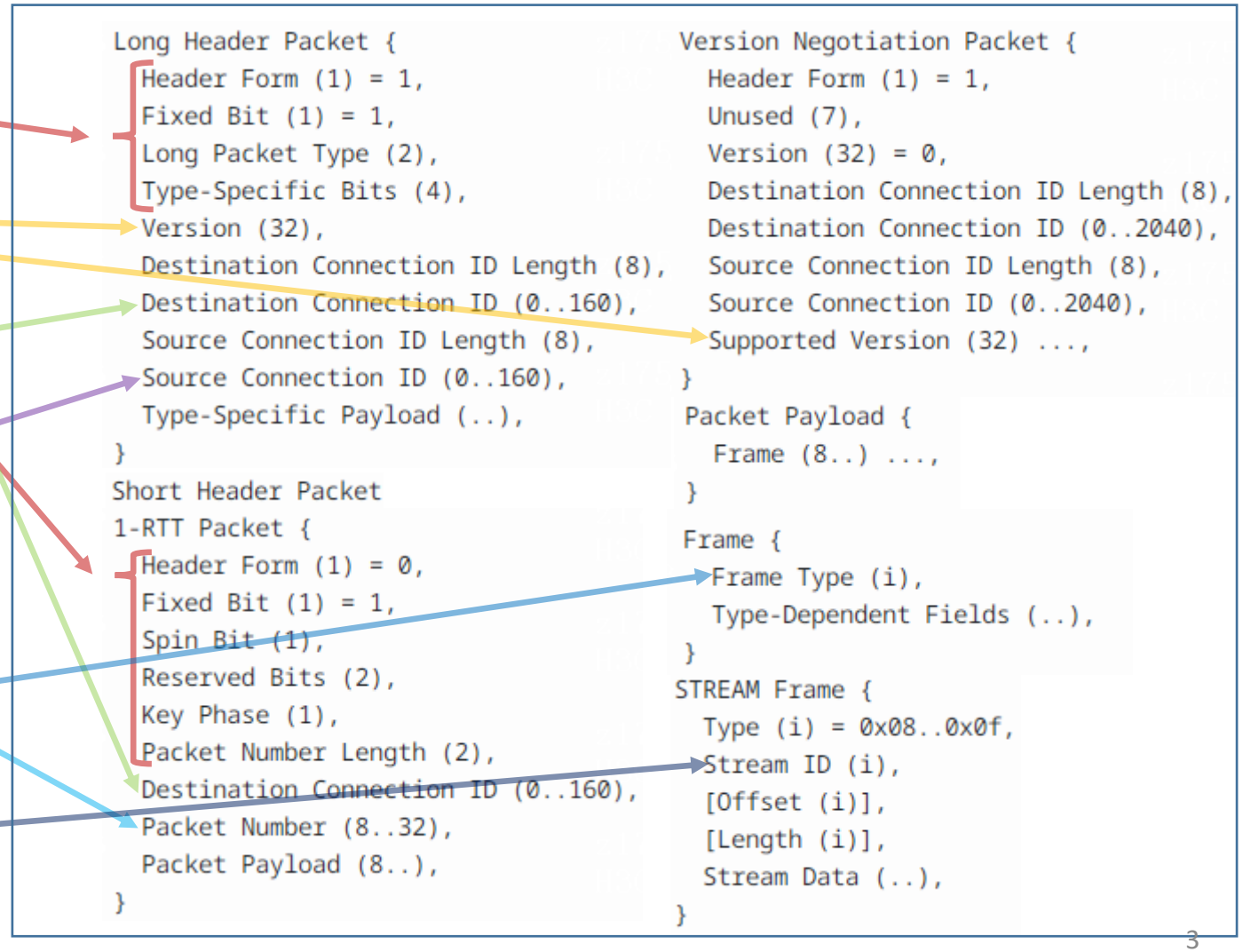
QUIC message format

Main Content

Defined 7 Information Elements are:

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

QUIC message format



Updates

- **Presented at IETF-122 & 124 and comments at the meeting and on the mailing list are appreciated from :**

  Med, Benoit, Chongfeng

- **Update this draft from v03 to v04 based on the comments：**
  https://mailarchive.ietf.org/arch/msg/opsawg/2KUkJSc-j1MTfontadHoO51AgrQ/

  - **Updates 1:** (Terminology) Adds explicit clarification that the term "**flow**" in this document aligns with the **IPFIX** definition, not the QUIC definition.

  - **Updates 2**: (Use Cases) Enhances the description of packet discard monitoring, aligning it with the broader **standardized discard reporting framework** ([draft-ietf-opsawg-discardmodel], [draft-evans-opsawg-ipfix-discard-class-ie]).

  - **Updates 3:** (Use Cases) Refines the language describing how QUIC information (Header, Frame, Stream ID) is associated with IPFIX Flow Records for both Long and Short Header packets, using more precise terminology.

  - **Updates 4:** Adds a **new co-author :** Xueting Li.

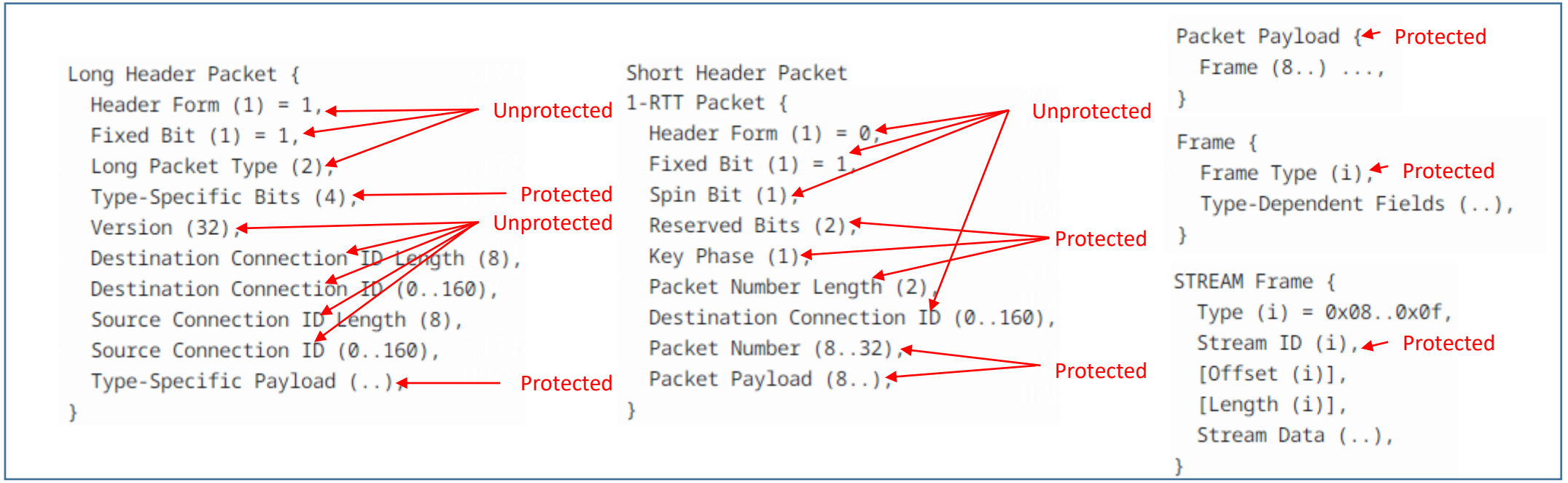
Next Steps

- **Any questions or comments are welcomed.**
- **Ask for WG adoption.**

Thank You!

Background and Motivation

- QUIC Packet includes Long Header Packet and Short Header Packet.
  - Long Header Packets are used during QUIC connection establishment.
  - Short Header Packets are used after a QUIC connection is established.
- **QUIC packets have different levels of cryptographic protection based on the type of packet.**
- **The protection details for QUIC packet parameters are listed below.**



QUIC packet protection parameters

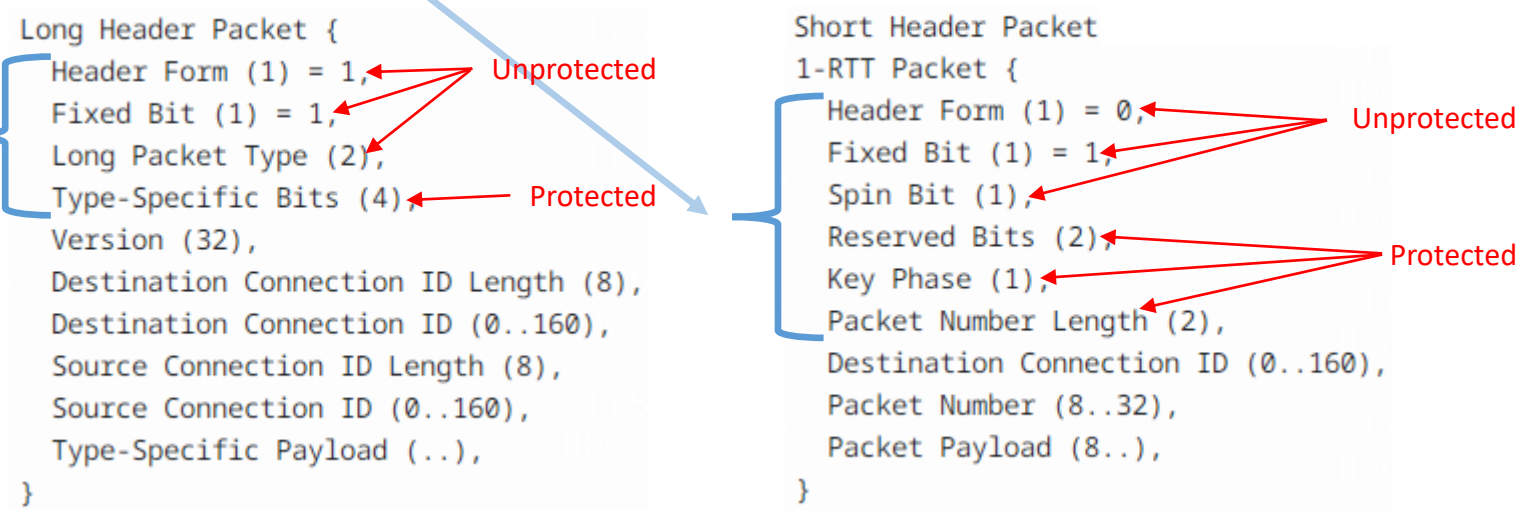
Main Content (1)

- **Specifies a set of new IPFIX Information Elements for QUIC**
- **7 Information Elements (IEs) are defined.**
- **The protected values of some IEs can only be obtained on a device that can decrypt QUIC packets, such as endpoint devices of QUIC connection.**

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.



For quicHeaderFlag, the last 4 bits for Long Header Packet and last 5 bits for Short Header Packet are protected, and the remaining bits are not protected.

Detailed Information Elements (2)

- **quicVersion** (Not Protected)

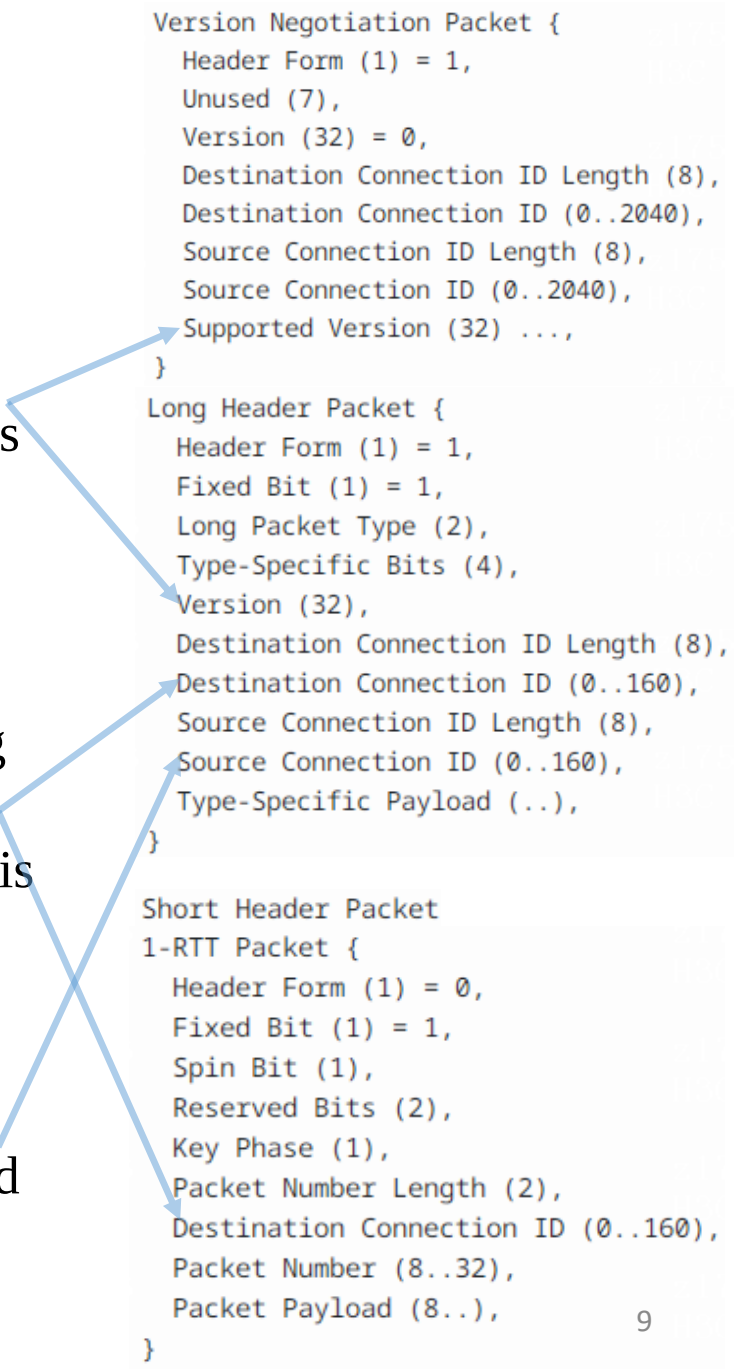
  32-bit QUIC Version that is in use and determines how the rest of the protocol packet fields are interpreted, or indicates the supported QUIC version of server for Version Negotiation packet.

- **quicDestinationConnectionID** (Not Protected)

  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** (Not Protected)

  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)

- **quicPacketNumber (Protected)**

  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.

- **quicFrameType (Protected)**

  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.

- **quicStreamID (Protected)**

  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. It is a 62-bit integer.
  Here, defined as 64-bit unsigned integer.

```
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 (..),
}
```

Use Cases

- **For Long Header Packet**, the following parameters is used to represent a QUIC flow:
  - Five-tuple (protocol, source and destination IP address, source and destination port).
  - Source Connection ID.
  - Destination Connection ID.

  Example:

  When the Long Packet Type is observed in a Flow during a QUIC connection, Flow information includes:

  Five-tuple + Source Connection ID + Destination Connection ID + **Header Flag**

- **For Short Header Packet**, the following parameters is used to represent a QUIC flow:
  - Five-tuple (protocol, source and destination IP address, source and destination port).
  - Destination Connection ID.

  Example:

  When the Frame Type is observed in a Flow, Flow information includes:

  Five-tuple + Destination Connection ID + **Frame Type**

  When the Stream ID is observed in a Flow, Flow information includes:

  Five-tuple + Destination Connection ID + **Stream ID**

```
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 (..),
}
```

Operational Considerations

- The **length of the Destination Connection ID** (or the value itself) must be **preconfigured or known in advance** on these intermediate devices for the tracking to work effectively.
- **The protected packet fields must be decrypted before they can be obtained.**
- quicHeaderFlag, quicVersion, quicDestinationConnectionID, and quicSourceConnectionID **can be obtained in all on-path devices.**
- quicPacketNumber, quicFrameType, and quicStreamID **can be obtained in endpoint devices or on-path devices which are capable of decrypting QUIC packets.**

