OPSAWG
Internet-Draft
Intended status: Standards Track
Expires: 30 August 2026

Y. Liu, Ed.
China Mobile
C. Lin, Ed.
New H3C Technologies
T. Graf, Ed.
Swisscom

26 February 2026

IPFIX Protocol over QUIC
draft-llg-opsawg-ipfix-over-quic-02

Abstract

The IP Flow Information Export (IPFIX) Protocol provides a means for transmitting Traffic Flow information over the network. IPFIX Data and Template Records can be carried over a number of transport protocols from an IPFIX Exporting Process to an IPFIX Collecting Process. The supported transport protocols are SCTP, UDP and TCP. QUIC could provide useful, reliable and secure semantics for IPFIX Protocol in particular as a single connection could carry multiple traffic flows over streams, enabling much better efficiency and performance for Exporter and Collector. This document describes how to use IPFIX Protocol over the QUIC transport protocol, named IPFIXoQUIC.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 30 August 2026.

Copyright Notice

Table of Contents

1. Introduction

In order to transmit Traffic Flow information from an Exporting Process to a Collecting Process, a common representation of flow data and a standard means of communicating them are required. [RFC7011] specifies a protocol called IP Flow Information Export (IPFIX) Protocol, and describes how the IPFIX Data and Template Records of IPFIX Protocol are carried over a few of transport protocols from an Exporter to a Collector.

The IPFIX Protocol Specification has been designed to be transport protocol independent, and allows a mapping to define how it could be implemented over any specific transport protocol. In other words, the Exporter can export to multiple Collectors using independent transport protocols. At present, some transport protocols are defined to carry IPFIX Protocol: SCTP transport [RFC9260], UDP transport [RFC9293], TCP transport [RFC768].

As the network continues to expand, the data traffic it carries is increasing, placing a significant load challenge on the IPFIX transport protocol. To prevent potential security threats (such as DoS attacks), it is highly desirable that it must ensure the confidentiality and integrity of IPFIX data transferred from an Exporting Process to a Collecting Process. Authentication mechanism is also desirable to prevent the collection of data from an unauthorized Exporting Process or the export of data to an unauthorized Collecting Process.

QUIC [RFC9000] conforms to the above requirements, therefore is also an appropriate transport protocol for IPFIX Protocol. QUIC is UDP-based multiplexed and secure transport protocol that provides connection-oriented and stateful interaction between a client and server. It can provide low latency and encrypted transport with resilient connections.

QUIC uses multiple simultaneous streams to carry data in one direction. Each stream is a separate unidirectional or bidirectional channel consisting of an ordered stream of bytes. In Addition, each stream has its own flow control, which limit bytes sent on a stream, together with flow control of the connection. Moreover, QUIC does not have the TCP shortcomings such as head of line blocking.

Compared to the transport protocols already supported by IPFIX [RFC7011], QUIC offers the following advantages:

* QUIC provides reliability and congestion control similar to SCTP and TCP. It is more universally used and supported across the global Internet environment because QUIC is a UDP-based protocol.

* QUIC has built-in TLS encryption (typically TLS 1.3), offering end-to-end security to ensure data confidentiality and integrity. In contrast, encryption for SCTP, UDP, and TCP requires additional protocols such as DTLS, increasing configuration complexity.

* QUIC supports multi-streaming similar to SCTP, allowing multiple data streams to be multiplexed within a single connection. Each stream transmits data independently, avoiding the head-of-line blocking issue found in TCP.

* QUIC has a faster connection establishment speed than TCP and SCTP. QUIC requires 1~2 handshakes, TCP requires 3 handshakes, and SCTP requires 4 handshakes.

* QUIC supports MTU discovery similar to SCTP and TCP, which can dynamically optimize packet fragmentation, reduce data loss caused by fragmentation failure. This gives QUIC strong adaptability to different MTU conditions.

* QUIC's connection migration feature can make it easy to maintain sessions seamlessly even when IP addresses change (e.g., during Wi-Fi to cellular handover), making it well-suited for IPFIX collection in mobile networks. SCTP connection migration is more complicated, and all backup addresses must be exchanged in advance.

* QUIC based on UDP is a pure user-mode implementation that does not require operating system kernel support. QUIC is flexible to deploy and easy to expand. QUIC has strong penetration and is less likely to be intercepted by enterprise firewalls/NAT than TCP and SCTP.

Therefore, QUIC is a proper secure and reliable transport protocol for the message transmission mechanism of IPFIX Protocol. This document specifies how to use QUIC as the transport protocol for IPFIX Protocol.

2. Terminology and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

In this document, the terms "client" and "server" are used to refer to the two ends of the QUIC connection. The client actively initiates the QUIC connection. The terms "exporter" and "collector" are used to refer to the two ends of the IPFIX Protocol session. An exporter establishes and keeps open a connection to one or more collectors, generally, an "exporter" is a "client" meanwhile a "collector" is a "server".

* Client: The endpoint that initiates a QUIC connection, the Exporter or Exporting Process.

* Server: The endpoint that accepts a QUIC connection, the Collector or Collecting Process.

3. Connection Management

3.1. Connection Establishment

QUIC connection establishment is described in [RFC9000]. During establishing connection, IPFIXoQUIC support is indicated by selecting the Application-Layer Protocol Negotiation (ALPN) [RFC7301] token as listed in the IANA Section 7 in the TLS handshake.

The Exporter MUST also act as the client meanwhile the Collector must also act as the server.

The Exporter should be the initiator of the QUIC connection to the Collector meanwhile the Collector acts as a connection acceptor. The Exporter MAY support more than one active connection to different Collector. The Exporter MAY also support more than one active connection to the same Collector to avoid head-of-line blocking across Observation Domains.

The Exporter MAY record an alarm if the underlying QUIC connection establishment time out; this timeout should be configurable on the Exporter.

If the Collecting Process does not acknowledge an attempt by the Exporting Process to establish a connection, QUIC will automatically retry connection establishment using exponential backoff. The Exporting Process MAY open a backup QUIC connection to a Collecting Process in advance, if it supports Collecting Process failover.

3.1.1. Early Data

The QUIC protocol uses TLS 1.3 messages to secure the transport. And TLS 1.3 supports Early Data (also known as 0-RTT data) [RFC9001]. Early Data is a mechanism defined in TLS 1.3 [I-D.ietf-tls-rfc8446bis] that enables a client to transmit flow records during the initial handshake when resuming a previous session. Note that TLS 1.3 can be used without Early Data as per Appendix F.5 of [I-D.ietf-tls-rfc8446bis]. This functionality is only available when the client and server share a Pre-Shared Key (PSK), obtained either externally or through a prior handshake.

As detailed in Section 2.3 of [I-D.ietf-tls-rfc8446bis], Early Data provides weaker security than standard TLS, lacking forward secrecy and replay attack protection while requiring pre-established PSK credentials. For IPFIXoQUIC, these security limitations present significant concerns:

* Flow records frequently contain sensitive operational data that could be exploited if replayed.

* Duplicate flow reporting could corrupt traffic analysis and monitoring systems.

* The stateful nature of flow collection amplifies vulnerability to replay-based manipulation.

* The marginal latency improvements offered by 0-RTT provide insufficient justification for accepting these risks in monitoring applications.

In accordance with Appendix F.5 of [I-D.ietf-tls-rfc8446bis], which mandates explicit profiling for Early Data usage, this document specifies that IPFIXoQUIC implementations MUST NOT utilize Early Data (0-RTT). Clients MUST NOT include early_data extensions in ClientHello messages, and servers MUST reject such extensions if presented. Implementations MUST configure their TLS 1.3 stacks to disable 0-RTT functionality.

3.2. Connection Termination

3.2.1. QUIC Connection Termination Process

The typical QUIC connection termination process is described in [RFC9000].

3.2.2. IPFIXoQUIC Considerations for Connection Termination

When an IPFIX session is implemented based on a QUIC connection, the idle timeout should be disabled or the QUIC max_idle_timeout should be set appropriately in order to keep the QUIC connection persistent even if the IPFIX session is idle.

When an Exporting Process is shut down, it SHOULD shut down the QUIC connection.

When a Collecting Process no longer wants to receive IPFIX Messages, it SHOULD close its end of the connection. The Collecting Process SHOULD continue to read IPFIX Messages until the Exporting Process has closed its end.

When detecting abnormal termination of a QUIC connection established with the exporting process, the collecting process MUST retain its ability to accept new incoming QUIC connections.

When an Exporting Process is detecting the abnormal interruption of the QUIC connection to the Collecting Process, it SHOULD try to re-establish the connection. Connection timeouts and retry schedules SHOULD be configurable. The default configuration is that an Exporting Process MUST NOT attempt to establish a connection more frequently than once per minute.

4. Stream Mapping and Usage

QUIC [RFC9000] uses multiple simultaneous streams to carry data in one direction. QUIC Streams provide a lightweight, ordered byte-stream abstraction to an application. Streams can be unidirectional or bidirectional meanwhile streams can be initiated by either the client or the server. Unidirectional streams carry data in one direction: from the initiator of the stream to its peer. Bidirectional streams allow for data to be sent in both directions.

QUIC uses Stream ID to identify the stream. The least significant bit (0x1) of the stream ID identifies the initiator of the stream (client with the bit set to 0). The second least significant bit (0x2) of the stream ID distinguishes between bidirectional streams (with the bit set to 0) and unidirectional streams [RFC9000].

4.1. Unidirectional Stream from Exporter to Collector

According to the architecture for the export of measured IP Flow information defined in [RFC5470], IPFIX Data Records and Templates are carried via a number of transport protocols from IPFIX Exporting Process to IPFIX Collecting Process. Therefore, the IPFIX connection is unidirectional from Exporter to Collector.

Based on the above description, The IPFIX messages are initiated by the Exporter and no reply is needed from the Collector. So the IPFIX messages MAY be mapped into one unidirectional stream whose stream type is 0x3 according to section 2.1 of [RFC9000].

4.2. Multiple Stream Usage

For achieving much better Transmission efficiency and performance for Exporter and Collector, An Exporting Process MAY create more than one QUIC stream per connection. Each of these streams may be used for the transmission of IPFIX Messages containing Data Sets, Template Sets, and/or Options Template Sets.

IPFIX messages can be differentiated by various characteristics, such as five-tuple information (Source IP address, source port, destination IP address, destination port, and transport layer protocol), only transport layer protocol (UDP and TCP), ranges of source or destination IP addresses, and method per Template, and carried over different QUIC streams accordingly.

For five-tuple information, each five-tuple's IPFIX messages are transmitted through a QUIC stream. If there are many five-tuples, a large number of QUIC streams must be created, so this approach should be used with caution.

For only transport layer protocol, IPFIX messages of each protocol are transmitted through a QUIC stream.

For ranges of source or destination addresses, IPFIX messages with specified source or destination IP address segment are transmitted through a QUIC stream. The size of the IP address range can be determined by the configuration.

For the method for exporting a Template Record and its associated Data Sets in a single SCTP stream (as specified in [RFC6526]), both each Template and its corresponding Data Records can also be exported within a single QUIC stream.

Moreover, if the specified feature's IPFIX message is binding to a QUIC stream ID, meaning that the IPFIX message with fixed features is only transmitted through a QUIC stream of fixed ID, the collector will only receive the specified feature's IPFIX message from the designated stream ID. This may simplify the implementation of the collector.

According to one of the features, the number of streams created may exceed the limit of a QUIC connection, in which case a new QUIC connection needs to be created.

5. Endpoint Authentication

IPFIXoQUIC uses QUIC which uses TLS version 1.3 or greater. Therefore, the TLS handshake process can be used for IPFIXoQUIC endpoint authentication. A third-party authentication mechanism can also be applied for IPFIXoQUIC endpoint authentication, such as a TLS client certificate.

6. Operational Considerations

6.1. Configuration Parameters

The decision to use IPFIXoQUIC instead of the SCTP-based/TCP-based/UDP-based mechanism in [RFC7011] is an operational decision, and an implementation MUST provide a configuration mechanism to enable IPFIXoQUIC as optional transmission protocol on the IPFIX session.

A configuration mechanism should be implemented to enable selection between single-stream or multi-stream QUIC transmission for IPFIX messages. For multi-stream QUIC transmission, the characteristics of IPFIX messages exported in a single stream should be determined by configuration.

Some connectivity problems (such as blocking UDP) could result in a failure to establish a QUIC connection. When this happens, the Exporter SHOULD attempt to establish an IPFIX session via other transport protocols.

6.2. Congestion Avoidance

QUIC controls the rate at which data can be sent from the Exporting Process to the Collecting Process, using a mechanism that considers both network congestion and the receiver’s capabilities. Therefore, an IPFIX Exporting Process may not be able to send IPFIX Messages at the rate the Metering Process generates them, either due to network congestion or because the Collecting Process cannot process IPFIX Messages quickly enough. As long as congestion is transient, the Exporting Process can buffer IPFIX Messages for transmission. However, buffering is inherently limited due to both resource constraints and timeliness requirements, so persistent and/or severe congestion may lead to a situation where the Exporting Process is blocked.

When an Exporting Process has Data Records to send but the transmission buffer is full and wants to avoid blocking, it can choose to drop some Data Records. Dropped Data Records MUST be accounted for, so that lost record counts can later be reported as described in Section 4.3 of [RFC7011].

6.3. Reliability

QUIC ensures reliable delivery of data from the Exporting Process to the Collecting Process.

6.4. MTU

QUIC uses path MTU discovery to determine the maximum packet size suitable for transmission of IPFIX Messages, aiming to ensure that packets can be delivered to the destination by all nodes in the network path without fragmentation.

6.5. Failover

If the Collecting Process does not acknowledge an attempt by the Exporting Process to establish a connection, QUIC will automatically retry connection establishment using exponential back off. The Exporter MAY log an alarm if the underlying QUIC connection establishment times out; this timeout should be configurable on the Exporter.

The Exporting Process MAY open a backup QUIC connection to a Collecting Process in advance, if it supports Collecting Process failover.

QUIC identifies connections using connection IDs instead of relying on the IP address and port at the network layer. This allows QUIC to continue data transmission without reconstructing the connection when the network path changes.

7. IANA Considerations

This document creates a new registration for the identification of IPFIXoQUIC in the "Application Layer Protocol Negotiation (ALPN) Protocol IDs" registry established in [RFC7301].

The "IPFIXoQ" string identifies IPFIXoQUIC:

* Protocol: IPFIXoQUIC

* Identification Sequence: 0x49 0x50 0x46 0x49 0x58 0x6f 0x51 ("IPFIXoQ")

* Specification: This document

In addition, it is requested for IANA to reserve a UDP port TBD for ’IPFIX over QUIC’.

8. Security Considerations

This document replaces the transport protocol layer of IPFIX from other transport protocols to QUIC. The basic protocol specification of IPFIX is not modified, and therefore the new security risks are not introduced to the basic IPFIX protocol. The security considerations of [RFC7011] are applicable to this document.

IPFIXoQUIC enhances transport-layer security for IPFIX session according to [RFC9000]. This document does not require to support third-party authentication (e.g., backend Authentication) due to the fact that TLS does not specify this way of authentication. If third-party authentication is needed, TLS client certificates are recommended to be used here.

9. References

9.1. Normative References

[RFC768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <https://www.rfc-editor.org/info/rfc768>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC6526] Claise, B., Aitken, P., Johnson, A., and G. Muenz, "IP Flow Information Export (IPFIX) Per Stream Control Transmission Protocol (SCTP) Stream", RFC 6526, DOI 10.17487/RFC6526, March 2012, <https://www.rfc-editor.org/info/rfc6526>.

[RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, DOI 10.17487/RFC7011, September 2013, <https://www.rfc-editor.org/info/rfc7011>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <https://www.rfc-editor.org/info/rfc9000>.

[RFC9001] Thomson, M., Ed. and S. Turner, Ed., "Using TLS to Secure QUIC", RFC 9001, DOI 10.17487/RFC9001, May 2021, <https://www.rfc-editor.org/info/rfc9001>.

[RFC9260] Stewart, R., Tuexen, M., and K. Nielsen, "Stream Control Transmission Protocol", RFC 9260, DOI 10.17487/RFC9260, June 2022, <https://www.rfc-editor.org/info/rfc9260>.

[RFC9293] Eddy, W., Ed., "Transmission Control Protocol (TCP)", STD 7, RFC 9293, DOI 10.17487/RFC9293, August 2022, <https://www.rfc-editor.org/info/rfc9293>.

[I-D.ietf-tls-rfc8446bis] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", Work in Progress, Internet-Draft, draft-ietf-tls-rfc8446bis-14, 13 September 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-tls-rfc8446bis-14>.

9.2. Informative References

[RFC5470] Sadasivan, G., Brownlee, N., Claise, B., and J. Quittek, "Architecture for IP Flow Information Export", RFC 5470, DOI 10.17487/RFC5470, March 2009, <https://www.rfc-editor.org/info/rfc5470>.

[RFC7301] Friedl, S., Popov, A., Langley, A., and E. Stephan, "Transport Layer Security (TLS) Application-Layer Protocol Negotiation Extension", RFC 7301, DOI 10.17487/RFC7301, July 2014, <https://www.rfc-editor.org/info/rfc7301>.

Authors' Addresses

Yisong Liu (editor)
China Mobile
China
Email: liuyisong@chinamobile.com

Changwang Lin (editor)
New H3C Technologies
Beijing
China
Email: linchangwang.04414@h3c.com

Thomas Graf (editor)
Swisscom
Binzring 17
CH-8045 Zurich
Switzerland
Email: thomas.graf@swisscom.com