QUIC
Internet-Draft
Intended status: Standards Track
Expires: 1 September 2025

A. Ferrieux, Ed.
Orange Labs
I. Lubashev, Ed.
Akamai Technologies
G. Fioccola, Ed.
Huawei Technologies
M. Ihlar, Ed.
Ericsson
F. Bulgarella
Telecom Italia - TIM
M. Cociglio

I. Hamchaoui
Orange Labs
M. Nilo
Telecom Italia - TIM
28 February 2025

Application of Explicit Measurement Techniques for QUIC Troubleshooting
draft-mdt-quic-explicit-measurements-02

Abstract

This document defines a protocol that can be used by QUIC endpoints to signal packet loss in a way that can be used by network devices to measure and locate the source of the loss.

Discussion of this work is encouraged to happen on the QUIC IETF mailing list quic@ietf.org (mailto:quic@ietf.org) or on the GitHub repository which contains the draft: https://github.com/igorlord/draft-mdt-quic-explicit-measurements (https://github.com/igorlord/draft-mdt-quic-explicit-measurements).

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 1 September 2025.

Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction

Packet loss is a hard and pervasive problem of day-to-day network operation. Proactively detecting, measuring, and locating it is crucial to maintaining high QoS and timely resolution of crippling end-to-end throughput issues. To this effect, in a TCP-dominated world, network operators have been heavily relying on information present in the clear in TCP headers: sequence and acknowledgment numbers, and SACKs when enabled. These allow for quantitative estimation of packet loss by passive on-path observation.

With QUIC, the equivalent transport headers are encrypted, and passive packet loss observation is not possible, as described in [RFC9065].

Measuring TCP loss between similar endpoints cannot be relied upon to evaluate QUIC loss. QUIC could be routed by the network differently and the fraction of Internet traffic delivered using QUIC is increasing every year. It is imperative to measure packet loss experienced by QUIC users directly.

The Alternate-Marking method [AltMark] defines a consolidated method to perform packet loss, delay, and jitter measurements on live traffic. However, as noted in [EXPLICIT-MEASUREMENTS], applying [AltMark] to end-to-end transport-layer connections is not easy because packet identification and marking by network nodes is prevented when QUIC encrypted transport-layer header is being used.

This document defines the Explicit Flow Measurement Protocol (EFMP) which is used by QUIC endpoints to enable packet loss measurements using Explicit Host-to-Network Flow Measurement Techniques defined in [EXPLICIT-MEASUREMENTS].

Measurement bits are sent in dedicated EFMP packets that are coalesced with other QUIC packets in UDP datagrams.

1.1. Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. On-Path RTT Observation

[QUIC-TRANSPORT] already introduces an explicit per-flow transport-layer signal for hybrid measurement of RTT. This signal consists of a Spin bit that toggles once per RTT.

3. On-Path Loss Observation

There are three sources of loss that network operators need to observe to guarantee high QoS:

* _upstream loss_ - loss between the sender and the observation point (Section 5.2)

* _downstream loss_ - loss between the observation point and the destination (Section 5.4)

* _observer loss_ - loss by the observer itself that does not cause downstream loss (Section 5.5)

The upstream and downstream loss together constitute _end-to-end loss_ (Section 5.1).

3.1. On-Path Loss Signaling Protocol

[EXPLICIT-MEASUREMENTS] introduces several techniques for using explicit loss bits in the clear portion of transport protocol headers to signal packet loss to on-path network devices. The explicit loss bits used in this document are the "sQuare signal" bit (Q) and the "Loss event" bit (L) (see Section 4.1 and Section 4.2). This approach follows the recommendations of [RFC8558] that recommends explicit path signals.

This document defines the Explicit Flow Measurement Protocol (EFMP) that takes inspiration from [TRAIN] that uses QUIC Long Header packets that are prepended to QUIC v1 or v2 packets as carriers of path signals.

While the exploitation of only Q can help in measuring the _upstream loss_ and only L can help in measuring the _end-to-end loss_, both are required to detect and measure the other types of losses (_downstream loss_ and _observer loss_).

3.2. Recommended Use of the Signals

The loss signal is not designed for use in automated control of the network in environments where loss bits are set by untrusted hosts. Instead, the signal is to be used for troubleshooting individual flows and for monitoring the network by aggregating information from multiple flows and raising operator alarms if aggregate statistics indicate a potential problem.

4. Loss Bits

The draft introduces two bits that are to be present in EFMP packets.

* Q: The "sQuare signal" bit is toggled every N outgoing packets, as explained below in Section 4.1.

* L: The "Loss event" bit is set to 0 or 1 according to the Unreported Loss counter, as explained below in Section 4.2.

Each endpoint maintains appropriate counters independently and separately for each connection 4-tuple and Destination Connection ID. Whenever this specification refers to connections, it is referring to packets sharing the same 4-tuple and Destination Connection ID. A "QUIC connection", however, refers to connections in the traditional QUIC sense.

4.1. Setting the sQuare Signal Bit on Outgoing Packets

The sQuare bit (Q bit) takes its name from the square wave generated by its signal. This method is based on the Alternate-Marking method [AltMark]. The sQuare Value is initialized to the Initial Q Value (0 or 1) and is reflected in the Q bit of every outgoing packet. The sQuare value is inverted after sending every N packets (a Q run). Hence, Q Period is 2*N. The Q bit represents "packet color" as defined by [RFC8321].

Observation points can estimate upstream losses by counting the number of packets during one period of the square signal, as described in Section 5.

4.1.1. Q Run Length Selection

The sender is expected to choose N (Q run length) based on the expected amount of loss and reordering on the path. The choice of N strikes a compromise -- the observation could become too unreliable in case of packet reordering and/or severe loss if N is too small, while short connections may not yield a useful upstream loss measurement if N is too large (see Section 5.2).

The value of N MUST be at least 64 and be a power of 2. This requirement allows an Observer to infer the Q run length by observing one period of the square signal. It also allows the Observer to identify flows that set the loss bits to arbitrary values (see Section 7).

If the sender does not have sufficient information to make an informed decision about Q run length, the sender SHOULD use N=64, since this value has been extensively tested in large-scale field tests and yielded good results. Alternatively, the sender MAY also choose a random N for each connection, increasing the chances of using a Q run length that gives the best signal for some connections.

The sender MUST keep the value of N constant for a given connection. The sender can change the value of N during a QUIC connection by switching to a new Destination Connection ID, if one is available.

4.2. Setting the Loss Event Bit on Outgoing Packets

The Loss Event bit uses the Unreported Loss counter maintained by the QUIC protocol. The Unreported Loss counter is initialized to 0, and the L bit of every outgoing packet indicates whether the Unreported Loss counter is positive (L=1 if the counter is positive, and L=0 otherwise). The value of the Unreported Loss counter is decremented every time a packet with L=1 is sent.

The value of the Unreported Loss counter is incremented for every packet that the protocol declares lost, using QUIC’s existing loss detection machinery. If the implementation is able to rescind the loss determination later, a positive Unreported Loss counter MAY be decremented due to the rescission, but it SHOULD NOT become negative.

This loss signaling is similar to loss signaling in [RFC7713], except the Loss Event bit is reporting the exact number of lost packets, whereas the Echo Loss bit in [RFC7713] is reporting an approximate number of lost bytes.

Observation points can estimate the end-to-end loss, as determined by the upstream endpoint, by counting packets in this direction with the L bit equal to 1, as described in Section 5.

5. Using Loss Bits for Passive Loss Measurement

5.1. End-To-End Loss

The Loss Event bit allows an observer to calculate the end-to-end loss rate by counting packets with the L bit value of 0 and 1 for a given connection. The end-to-end loss rate is the fraction of packets with L=1.

The assumption here is that upstream loss affects packets with L=0 and L=1 equally. If some loss is caused by tail-drop in a network device, this may be a simplification. If the sender congestion controller reduces the packet send rate after loss, there may be a sufficient delay before sending packets with L=1 that they have a greater chance of arriving at the observer.

5.2. Upstream Loss

Blocks of N (Q run length) consecutive packets are sent with the same value of the Q bit, followed by another block of N packets with an inverted value of the Q bit. Hence, knowing the value of N, an on-path observer can estimate the amount of loss after observing at least N packets. The upstream loss rate (u) is one minus the average number of packets in a block of packets with the same Q value (p) divided by N (u=1-avg(p)/N).

The observer needs to be able to tolerate packet reordering that can blur the edges of the square signal.

The observer needs to differentiate packets as belonging to different connections, since they use independent counters.

5.3. Correlating End-to-End and Upstream Loss

Upstream loss is calculated by observing packets that did not suffer the upstream loss. End-to-end loss, however, is calculated by observing subsequent packets after the sender’s protocol detected the loss. Hence, end-to-end loss is generally observed with a delay of between 1 RTT (loss declared due to multiple duplicate acknowledgments) and 1 RTO (loss declared due to a timeout) relative to the upstream loss.

The connection RTT can sometimes be estimated by timing protocol handshake messages. This RTT estimate can be greatly improved by observing a dedicated protocol mechanism for conveying RTT information, such as the latency Spin bit of [QUIC-TRANSPORT].

Whenever the observer needs to perform a computation that uses both upstream and end-to-end loss rate measurements, it SHOULD use upstream loss rate leading the end-to-end loss rate by approximately

1 RTT. If the observer is unable to estimate RTT of the connection, it should accumulate loss measurements over time periods of at least 4 times the typical RTT for the observed connections.

If the calculated upstream loss rate exceeds the end-to-end loss rate calculated in Section 5.1, then either the Q run length is too short for the amount of packet reordering or there is observer loss, described in Section 5.5. If this happens, the observer SHOULD adjust the calculated upstream loss rate to match end-to-end loss rate.

5.4. Downstream Loss

Because downstream loss affects only those packets that did not suffer upstream loss, the end-to-end loss rate (e) relates to the upstream loss rate (u) and downstream loss rate (d) as (1-u)(1-d)=1-e. Hence, d=(e-u)/(1-u).

5.5. Observer Loss

A typical deployment of a passive observation system includes a network tap device that mirrors network packets of interest to a device that performs analysis and measurement on the mirrored packets. The observer loss is the loss that occurs on the mirror path.

Observer loss affects upstream loss rate measurement, since it causes the observer to account for fewer packets in a block of identical Q bit values (see Section 5.2). The end-to-end loss rate measurement, however, is unaffected by the observer loss, since it is a measurement of the fraction of packets with the set L bit value, and the observer loss would affect all packets equally (see Section 5.1).

The need to adjust the upstream loss rate down to match end-to-end loss rate as described in Section 5.3 is a strong indication of the observer loss, whose magnitude is between the amount of such adjustment and the entirety of the upstream loss measured in Section 5.2. Alternatively, a high apparent upstream loss rate could be an indication of significant reordering, possibly due to packets belonging to a single connection being multiplexed over several upstream paths with different latency characteristics.

6. Implementation

6.1. EFMP Packet

An EFMP packet is a QUIC long header packet that follows the QUIC invariants; see Section 5.1 of [INVARIANTS].

Figure 1 shows the format of the EFMP packet using the conventions from Section 4 of [INVARIANTS].

```
EFMP Packet {
  Header Form (1) = 1,
  Reserved (1),
  Q Bit (1),
  L Bit (1),
  Spin Bit (1),
  Reserved (3),
  Version (32) = 0xTBD,
  Destination Connection ID Length (8),
  Destination Connection ID (0..2040),
  Source Connection ID Length (8),
  Source Connection ID (0..2040),
}
```

Figure 1: EFMP Packet Format

The most significant bit (0x80) of the packet indicates that this is a QUIC long header packet. The next bit (0x40) is reserved and can be set according to [QUIC-BIT].

The six least significant bits of the first octet of an EFMP packet forms the EFMP payload:

sQuare Signal Bit (Q): The first bit of the EFMP payload (0x20) is is the sQuare signal bit, set as described in Section 4.1.

Loss Event Bit (L): The second bit (0x10) is the Loss event bit, set as described in Section 4.2.

Latency Spin Bit (S): The third bit (0x8) is the latency spin bit. This bit is set to the value of the spin bit in the QUIC Short Header packet that follows directly after the EFMP packet in the same UDP datagram.

The three least significant bits (0x7) are reserved for future use.

An EFMP packet includes a Destination Connection ID field that is set to the same value as other packets in the same datagram; see Section 12.2 of [QUIC-TRANSPORT].

The Source Connection ID field is set to match the Source Connection ID field of any packet that follows. If the next packet in the datagram has a short header (Section 5.2 of [INVARIANTS]), the Source Connection ID field is empty.

EFMP packets are always coalesced with other QUIC packets and SHOULD be included as the first packet in a UDP datagram.

6.2. Transport Parameter

A QUIC endpoint indicates that it is willing to receive EFMP packets by including the transport parameter:

efmp_supported (0xTBD): efmp_supported transport parameter is an integer value, encoded as a variable-length integer, that can be set to 0 or 1, indicating the level of EFMP support. The value of 0 indicates that the endpoint is able to receive EFMP packets but will not be sending any, while the value of 1 indicates that the endpoint is also willing to send EFMP packets.

A client MUST NOT use remembered value of efmp_supported for 0-RTT connections.

Except for the cases outlined in Section 7, it is RECOMMENDED for the server to consistently include the efmp_supported parameter. This enables clients to utilize loss bits at their discretion.

6.3. EFMP Packet Processing

An EFMP packet is identified by the header form bit (0x80) of the first byte of a UDP datagram payload and the 32-bit version field with the value (0xTBD) that directly follows the first octet. Since the EFMP payload is part of the first octet, an observer does not need to process a packet beyond the version field.

7. Ossification Considerations

Accurate loss reporting is not critical for the operation of the QUIC protocol, though its presence in a sufficient number of connections is important for the operation of networks.

The use of EFMP is amenable to "greasing" described in [RFC8701] and MUST be greased. The greasing should be accomplished similarly to the latency Spin bit greasing in [QUIC-TRANSPORT]. Namely, implementations MUST NOT include efmp_supported transport parameter for a random selection of at least one in every 16 QUIC connections.

It is possible to observe packet reordering near the edge of the square signal. A middle box might observe the signal and try to fix packet reordering that it can identify, though only a small fraction of reordering can be fixed using this method. The Latency Spin bit signal edge can be used for the same purpose.

8. Security Considerations

The measurements described in this document do not involve new packets injected into the network causing potential harm to the network itself and to data traffic. The measurements could be harmed by a malicious endpoint misreporting losses or an attacker injecting artificial traffic. In the environments where such attacks are possible and cannot be identified by on-path observers, loss signal should not be used for automated control of the network.

In the absence of packet loss, the Q bit signal does not provide any information that cannot be observed by simply counting packets transiting a network path. The L bit signal discloses internal state of the protocol's loss detection machinery, but this state can often be gleaned by timing packets and observing congestion controller response. Hence, loss bits do not provide a viable new mechanism to attack QUIC data integrity and secrecy.

8.1. Optimistic ACK Attack

A defense against an Optimistic ACK Attack [QUIC-TRANSPORT] involves a sender randomly skipping packet numbers to detect a receiver acknowledging packet numbers that have never been received. The Q bit signal may inform the attacker which packet numbers were skipped on purpose and which had been actually lost (and are, therefore, safe for the attacker to acknowledge). To use the Q bit for this purpose, the attacker must first receive at least an entire Q run of packets, which renders the attack ineffective against a delay-sensitive congestion controller.

For QUIC v1 connections, if the attacker can make its peer transmit data using a single large stream, examining offsets in STREAM frames can reveal whether packet number skips are deliberate. In that case, the Q bit signal provides no new information (but it does save the attacker the need to remove packet protection). However, an endpoint that communicates using [DATAGRAM] and uses a loss-based congestion controller MAY shorten the current Q run by the number of skipped packets. For example, skipping a single packet number will invert the sQuare signal one outgoing packet sooner.

9. Privacy Considerations

To minimize unintentional exposure of information, loss bits provide an explicit loss signal -- a preferred way to share information per [RFC8558].

[QUIC-TRANSPORT] allows changing connection IDs in the middle of a QUIC connection to reduce the likelihood of a passive observer linking old and new subflows to the same device. Hence, a QUIC implementation would need to reset all counters when it changes connection ID used for outgoing packets. It would also need to avoid incrementing Unreported Loss counter for loss of packets sent with a different connection ID.

Accurate loss information allows identification and correlation of network conditions upstream and downstream of the observer. This could be a powerful tool to identify connections that attempt to hide their origin networks, if the adversary is able to affect network conditions in those origin networks. Similar information can be obtained by packet timing and inferring congestion controller response to network events, but loss information provides a clearer signal.

Implementations MUST allow administrators of clients and servers to disable loss reporting either globally or per QUIC connection. Additionally, as described in Section 7, loss reporting MUST be disabled for a certain fraction of all QUIC connections.

10. IANA Considerations

This document registers a new value in the QUIC Transport Parameter Registry:

Value: 0xTBD (if this document is approved)

Parameter Name: efmp_supported

Specification: Indicates that the endpoint supports the explicit flow measurement protocol. An endpoint that advertises this transport parameter can EFMP packets. An endpoint that advertises this transport parameter with value 1 can also send EFMP packets.

11. Change Log

TBD

12. Acknowledgments

The following people directly contributed key ideas that shaped this draft: Kazuho Oku, Christian Huitema.

13. References

13.1. Normative References

[AltMark] Fioccola, G., Ed., Cociglio, M., Mirsky, G., Mizrahi, T., and T. Zhou, "Alternate-Marking Method", RFC 9341, DOI 10.17487/RFC9341, December 2022, <https://www.rfc-editor.org/rfc/rfc9341>.

[EXPLICIT-MEASUREMENTS] Cociglio, M., Ferrieux, A., Fioccola, G., Lubashev, I., Bulgarella, F., Nilo, M., Hamchaoui, I., and R. Sisto, "Explicit Host-to-Network Flow Measurements Techniques", RFC 9506, DOI 10.17487/RFC9506, October 2023, <https://www.rfc-editor.org/rfc/rfc9506>.

[INVARIANTS] Thomson, M., "Version-Independent Properties of QUIC", RFC 8999, DOI 10.17487/RFC8999, May 2021, <https://www.rfc-editor.org/rfc/rfc8999>.

[QUIC-BIT] Thomson, M., "Greasing the QUIC Bit", RFC 9287, DOI 10.17487/RFC9287, August 2022, <https://www.rfc-editor.org/rfc/rfc9287>.

[QUIC-TRANSPORT] 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/rfc/rfc9000>.

[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/rfc/rfc2119>.

[RFC8558] Hardie, T., Ed., "Transport Protocol Path Signals", RFC 8558, DOI 10.17487/RFC8558, April 2019, <https://www.rfc-editor.org/rfc/rfc8558>.

[RFC8701] Benjamin, D., "Applying Generate Random Extensions And Sustain Extensibility (GREASE) to TLS Extensibility", RFC 8701, DOI 10.17487/RFC8701, January 2020, <https://www.rfc-editor.org/rfc/rfc8701>.

[RFC9065] Fairhurst, G. and C. Perkins, "Considerations around Transport Header Confidentiality, Network Operations, and the Evolution of Internet Transport Protocols", RFC 9065, DOI 10.17487/RFC9065, July 2021, <https://www.rfc-editor.org/rfc/rfc9065>.

13.2. Informative References

[DATAGRAM] Pauly, T., Kinnear, E., and D. Schinazi, "An Unreliable Datagram Extension to QUIC", RFC 9221, DOI 10.17487/RFC9221, March 2022, <https://www.rfc-editor.org/rfc/rfc9221>.

[RFC7713] Mathis, M. and B. Briscoe, "Congestion Exposure (ConEx) Concepts, Abstract Mechanism, and Requirements", RFC 7713, DOI 10.17487/RFC7713, December 2015, <https://www.rfc-editor.org/rfc/rfc7713>.

[RFC8321] Fioccola, G., Ed., Capello, A., Cociglio, M., Castaldelli, L., Chen, M., Zheng, L., Mirsky, G., and T. Mizrahi, "Alternate-Marking Method for Passive and Hybrid Performance Monitoring", RFC 8321, DOI 10.17487/RFC8321, January 2018, <https://www.rfc-editor.org/rfc/rfc8321>.

[TRAIN] Thomson, M., Huitema, C., and K. Oku, "Transparent Rate Adaptation Indications for Networks (TRAIN) Protocol", Work in Progress, Internet-Draft, draft-thomson-scone-train-protocol-00, 14 October 2024, <https://datatracker.ietf.org/doc/html/draft-thomson-scone-train-protocol-00>.

Authors' Addresses

Alexandre Ferrieux (editor)
Orange Labs
Email: alexandre.ferrieux@orange.com

Igor Lubashev (editor)
Akamai Technologies
Email: ilubashe@akamai.com

Giuseppe Fioccola (editor)
Huawei Technologies
Email: giuseppe.fioccola@huawei.com

Marcus Ihlar (editor)
Ericsson
Email: marcus.ihlar@ericsson.com

Fabio Bulgarella
Telecom Italia - TIM
Via Reiss Romoli, 274
10148 Torino
Italy
Email: fabio.bulgarella@guest.telecomitalia.it

Mauro Cociglio
Italy
Email: mauro.cociglio@outlook.com

Isabelle Hamchaoui
Orange Labs
Email: isabelle.hamchaoui@orange.com

Massimo Nilo
Telecom Italia - TIM
Via Reiss Romoli, 274
10148 Torino
Italy
Email: massimo.nilo@telecomitalia.it