TSVWG
Internet-Draft
Intended status: Informational
Expires: January 29, 2021

A. Ferrieux, Ed.
I. Hamchaoui, Ed.
Orange Labs
I. Lubashev, Ed.
Akamai Technologies
D. Tikhonov, Ed.
LiteSpeed Technologies

July 28, 2020

# Packet Loss Signaling for Encrypted Protocols
**draft-ferrieuxhamchaoui-tsvwg-lossbits-03**

## Abstract

This document describes a protocol-independent method that employs two bits to allow endpoints to signal packet loss in a way that can be used by network devices to measure and locate the source of the loss. The signaling method applies to all protocols with a protocol-specific way to identify packet loss. The method is especially valuable when applied to protocols that encrypt transport header and do not allow an alternative method for loss detection.

## Status of This Memo

## Copyright Notice

Table of Contents

# 1. Introduction

## 1.1. Motivation for Passive On-Path Loss Observation

Packet loss is 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 (see [RFC8517]). These allow for quantitative estimation of packet loss by passive on-path observation. Additionally, the lossy segment (upstream or downstream from the observation point) can be quickly identified by moving the passive observer around.

With encrypted protocols, the equivalent transport headers are encrypted and passive packet loss observation is not possible, as described in [TRANSPORT-ENCRYPT].

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

## 1.2. 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 4.2)

- _downstream loss_ - loss between the observation point and the destination (Section 4.4)

- _observer loss_ - loss by the observer itself that does not cause downstream loss (Section 4.5)

The upstream and downstream loss together constitute _end-to-end loss_ (Section 4.1).

## 1.3. On-Path Loss Signaling

Following the recommendation in [RFC8558] of making path signals explicit, this document proposes adding two explicit loss bits to the clear portion of the protocol headers to restore network operators' ability to maintain high QoS. These bits can be added to an unencrypted portion of a header belonging to any protocol layer, e.g. IP (see [IP]) and IPv6 (see [IPv6]) headers or extensions, such as [IPv6AltMark], UDP surplus space (see [UDP-OPTIONS] and [UDP-SURPLUS]), reserved bits in a QUIC v1 header (see [QUIC-TRANSPORT]).

## 1.4. 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 as well as for monitoring the network by aggregating information from multiple flows and raising operator alarms if aggregate statistics indicate a potential problem.

## 2. 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].

## 3. Loss Bits

The draft introduces two bits that are to be present in packets capable of loss reporting. These are packets that include protocol headers with the loss bits. Only loss of packets capable of loss reporting is reported using loss bits.

Whenever this specification refers to packets, it is referring only to packets capable of loss reporting.

- Q: The "sQuare signal" bit is toggled every N outgoing packets as explained below in Section 3.1.

- L: The "Loss event" bit is set to 0 or 1 according to the Unreported Loss counter, as explained below in Section 3.2.

Each endpoint maintains appropriate counters independently and separately for each separately identifiable flow (each subflow for multipath connections).

### 3.1. Setting the sQuare Bit on Outgoing Packets

The sQuare Value is initialized to the Initial Q Value (0) 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]. The sQuare Bit can also be called an Alernate Marking bit.

Observation points can estimate the upstream losses by counting the number of packets during a half period of the square signal, as described in Section 4.

#### 3.1.1. 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 flows may not yield a useful upstream loss measurement if N is too large (see Section 4.2).

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 6).

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 tried in large-scale field tests and yielded good results. Alternatively, the sender MAY also choose a random N for each flow, increasing the chances of using a Q run length that gives the best signal for some flows.

The sender MUST keep the value of N constant for a given flow.

### 3.2. Setting the Loss Event Bit on Outgoing Packets

The Unreported Loss counter is initialized to 0, and 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 whatever loss detection machinery the protocol employs. If the protocol 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 due to the rescission.

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

For protocols, such as TCP ([TCP]), that allow network devices to change data segmentation, it is possible that only a part of the packet is lost. In these cases, the sender MUST increment Unreported Loss counter by the fraction of the packet data lost (so Unreported Loss counter may become negative when a packet with L=1 is sent after a partial packet has been lost).

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 4.

## 4. Using the Loss Bits for Passive Loss Measurement

### 4.1. End-To-End Loss

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

### 4.2. 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 upstream 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 flows, since they use independent counters.

## 4.3. 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 flow 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 flow, it should accumulate loss measurements over time periods of at least 4 times the typical RTT for the observed flows.

If the calculated upstream loss rate exceeds the end-to-end loss rate calculated in Section 4.1, then either the Q Period is too short for the amount of packet reordering or there is observer loss, described in Section 4.5.  If this happens, the observer SHOULD adjust the calculated upstream loss rate to match end-to-end loss rate.

## 4.4. 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)".

## 4.5. 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 {{upstreamloss}}). 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 4.1).

The need to adjust the upstream loss rate down to match end-to-end loss rate as described in Section 4.3 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 4.2. Alternatively, a high apparent upstream loss rate could be an indication of significant reordering, possibly due to packets belonging to a single flow being multiplexed over several upstream paths with different latency characteristics.

## 5. ECN-Echo Event Bit

While the primary focus of the draft is on exposing packet loss, modern networks can report congestion before they are forced to drop packets, as described in [ECN]. When transport protocols keep ECN-Echo feedback under encryption, this signal cannot be observed by the network operators. When tasked with diagnosing network performance problems, knowledge of a congestion downstream of an observation point can be instrumental.

If downstream congestion information is desired, this information can be signaled with an additional bit.

- E: The "ECN-Echo Event" bit is set to 0 or 1 according to the Unreported ECN Echo counter, as explained below in Section 5.1.

### 5.1. Setting the ECN-Echo Event Bit on Outgoing Packets

The Unreported ECN-Echo counter operates identicaly to Unreported Loss counter (Section 3.2), except it counts packets delivered by the network with CE markings, according to the ECN-Echo feedback from the receiver.

This ECN-Echo signaling is similar to ECN signaling in [ConEx]. ECN-Echo mechanism in QUIC provides the number of packets received with CE marks. For protocols like TCP, the method described in [ConEx-TCP] can be employed. As stated in [ConEx-TCP], such feedback can be further improved using a method described in [ACCURATE].

### 5.2. Using E Bit for Passive ECN-Reported Congestion Measurement

A network observer can count packets with CE codepoint and determine the upstream CE-marking rate directly.

Observation points can also estimate ECN-reported end-to-end congestion by counting packets in this direction with a E bit equal to 1.

The upstream CE-marking rate and end-to-end ECN-reported congestion can provide information about downstream CE-marking rate. Presence of E bits along with L bits, however, can somewhat confound precise estimates of upstream and downstream CE-markings in case the flow contains packets that are not ECN-capable.

## 6. Protocol Ossification Considerations

Accurate loss information is not critical to the operation of any protocol, though its presence for a sufficient number of flows is important for the operation of networks.

The loss bits are amenable to "greasing" described in [RFC8701], if the protocol designers are not ready to dedicate (and ossify) bits used for loss reporting to this function. The greasing could be accomplished similarly to the Latency Spin bit greasing in [QUIC-TRANSPORT]. Namely, implementations could decide that a fraction of flows should not encode loss information in the loss bits and, instead, the bits would be set to arbitrary values. The observers would need to be ready to ignore flows with loss information more resembling noise than the expected signal.

## 7. Security Considerations

Passive loss observation has been a part of the network operations for a long time, so exposing loss information to the network does not add new security concerns for protocols that are currently observable.

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

### 7.1. Optimistic ACK Attack

A defense against an Optimistic ACK Attack, decribed in [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.

A protocol that is more susceptible to an Optimistic ACK Attack with the loss signal provided by Q bit and uses a loss-based congestion controller, SHOULD shorten the current Q Run by the number of skipped packets numbers. For example, skipping a single packet number will invert the sQuare signal one outgoing packet sooner.

## 8. Privacy Considerations

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

New protocols commonly have specific privacy goals, and loss reporting must ensure that loss information does not compromise those privacy goals. For example, [QUIC-TRANSPORT] allows changing Connection IDs in the middle of a connection to reduce the likelihood of a passive observer linking old and new subflows to the same device. A QUIC implementation would need to reset all counters when it changes the destination (IP address or UDP port) or the Connection ID used for outgoing packets. It would also need to avoid incrementing Unreported Loss counter for loss of packets sent to a different destination or with a different Connection ID.

## 9. IANA Considerations

This document makes no request of IANA.

## 10. Change Log

### 10.1. Since version 02

- Minor improvement and clarifications

### 10.2. Since version 01

- Clarified Q Period selection

- Added an optional E (ECN-Echo Event) bit

- Clarified L bit calculation for protocols that allow partial data loss due to a change in segmentation (such as TCP)

### 10.3. Since version 00

- Addressed review comments

- Improved guidelines for privacy protections for QIUC

## 11. Acknowledgments

The sQuare bit was originally suggested by Kazuho Oku in early proposals for loss measurement and is an instance of the "alternate marking" as defined in [RFC8321].

## 12. References

### 12.1. Normative References

[ConEx] 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/info/rfc7713>.

[ConEx-TCP] Kuehlewind, M., Ed. and R. Scheffenegger, "TCP Modifications for Congestion Exposure (ConEx)", RFC 7786, DOI 10.17487/RFC7786, May 2016, <https://www.rfc-editor.org/info/rfc7786>.

[ECN] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <https://www.rfc-editor.org/info/rfc3168>.

[IP] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <https://www.rfc-editor.org/info/rfc791>.

[IPv6] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[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>.

[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/info/rfc8321>.

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

[TCP] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <https://www.rfc-editor.org/info/rfc793>.

## 12.2. Informative References

[ACCURATE]
Briscoe, B., Kuehlewind, M., and R. Scheffenegger, "More Accurate ECN Feedback in TCP", draft-ietf-tcpm-accurate-ecn-11 (work in progress), March 2020.

[IPv6AltMark]
Fioccola, G., Zhou, T., Cociglio, M., Qin, F., and R. Pang, "IPv6 Application of the Alternate Marking Method", draft-ietf-6man-ipv6-alt-mark-01 (work in progress), June 2020.

[QUIC-TRANSPORT]
Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", draft-ietf-quic-transport-29 (work in progress), June 2020.

[RFC8517] Dolson, D., Ed., Snellman, J., Boucadair, M., Ed., and C. Jacquenet, "An Inventory of Transport-Centric Functions Provided by Middleboxes: An Operator Perspective", RFC 8517, DOI 10.17487/RFC8517, February 2019, <https://www.rfc-editor.org/info/rfc8517>.

[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/info/rfc8701>.

[TRANSPORT-ENCRYPT]
Fairhurst, G. and C. Perkins, "Considerations around Transport Header Confidentiality, Network Operations, and the Evolution of Internet Transport Protocols", draft-ietf-tsvwg-transport-encrypt-16 (work in progress), July 2020.

[UDP-OPTIONS]
Touch, J., "Transport Options for UDP", draft-ietf-tsvwg-udp-options-08 (work in progress), September 2019.

[UDP-SURPLUS]
Herbert, T., "UDP Surplus Header", draft-herbert-udp-space-hdr-01 (work in progress), July 2019.

## Authors' Addresses

Alexandre Ferrieux (editor)
Orange Labs

EMail: alexandre.ferrieux@orange.com

Isabelle Hamchaoui (editor)
Orange Labs

EMail: isabelle.hamchaoui@orange.com

Igor Lubashev (editor)
Akamai Technologies

EMail: ilubashe@akamai.com

Dmitri Tikhonov (editor)
LiteSpeed Technologies

EMail: dtikhonov@litespeedtech.com