IPPM Working Group
Internet-Draft
Intended status: Experimental
Expires: September 1, 2019

M. Cociglio
Telecom Italia
G. Fioccola
Huawei Technologies
F. Bulgarella
R. Sisto
Politecnico di Torino

February 28, 2019

New Spin bit enabled measurements with one or two more bits
draft-cfb-ippm-spinbit-new-measurements-00

Abstract

This document introduces additional measurements by using the same spin bit signal as defined in [I-D.trammell-ippm-spin]. The spin bit signal alone is not enough to evaluate correctly in every network condition the RTT of a flow. In order to solve this problem, it is theorized the possibility of introducing an additional validation signal called delay bit, similar to what is done done by the Valid Edge Counter (VEC), but using just one bit instead of two. An alternative with two bits is also introduced with a so called loss bit.

Requirements Language

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 RFC 2119 [RFC2119].

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 September 1, 2019.

Table of Contents

1. Introduction

[I-D.trammell-ippm-spin] defines an explicit per-flow transport-layer signal for hybrid measurement of end-to-end RTT. This signal consists of three bits: a spin bit, which oscillates once per end-to-end RTT, and a two-bit Valid Edge Counter (VEC), which compensates for loss and reordering of the spin bit to increase fidelity of the signal in less than ideal network conditions.

In this document it is introduced the delay bit, that is a single bit signal that can be used together with the spin bit by passive observers to measure the RTT of a network flow, avoiding the spin bit ambiguities that arise as soon as network conditions deteriorate. Unlike the spin bit, which is actually set in every packet transmitted on the network, the delay bit is set only once per round trip.

This document defines a hybrid measurement RFC 7799 [RFC7799] path signal to be embedded into a transport layer protocol, explicitly intended for exposing end-to-end RTT to measurement devices on path.

The document introduces a mechanism applicable to any transport-layer protocol, then explains how to bind the signal to a variety of IETF transport protocols, and in particular to QUIC and TCP.

The application of the Spin bit to QUIC is described in [I-D.ietf-quic-spin-exp] which adds the spin bit only (without the VEC) to QUIC for experimentation purposes.

Note that both the spin bit and the delay bit are inspired by RFC 8321 [RFC8321]. This is also mentioned in [I-D.trammell-quic-spin].

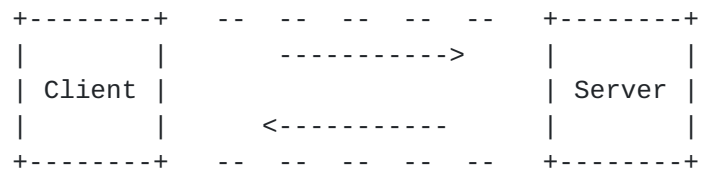
2. Spin bit and Delay bit mechanism

The main idea is to have a single packet, with a second marked bit (the delay bit), that bounces between client and server during the entire connection life. This single packet is called Delay Sample.

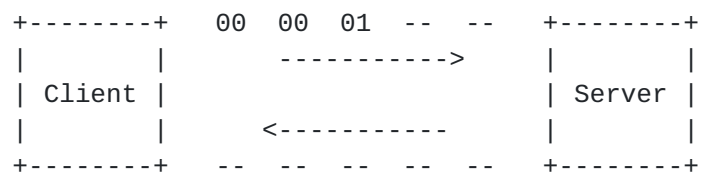
A simple observer placed in an intermediate point, tracking the delay sample and the relative timestamp in every spin bit period, can measure the end-to-end round trip delay of the connection. In the same way as seen with the spin bit and the VEC, it is possible to carry out other types of measurements. The next paragraphs give an overview of the observer capabilities.

In order to describe the delay sample working mechanism in detail, we have to distinguish two different phases which take part in the delay bit lifetime: initialization and reflection. The initialization is the generation of the delay sample, while the reflection realizes the bounce behavior of this single packet between the two endpoints.

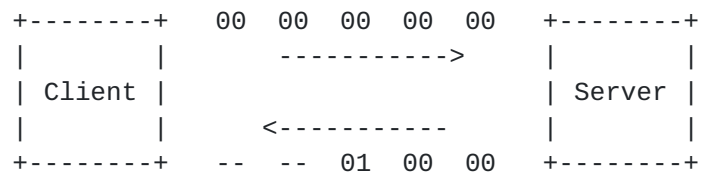
The next figure describes the Delay bit mechanism: the first bit is the spin bit and the second one is the delay bit.



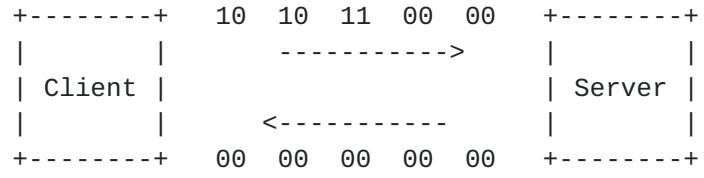
(a) No traffic at beginning.



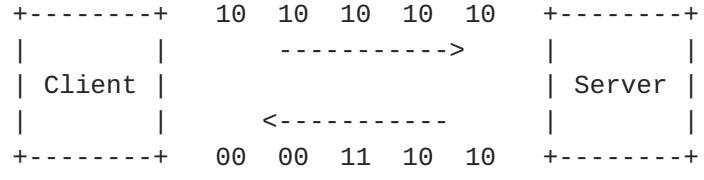
(b) The Client starts sending data and sets the first packet as Delay Sample.



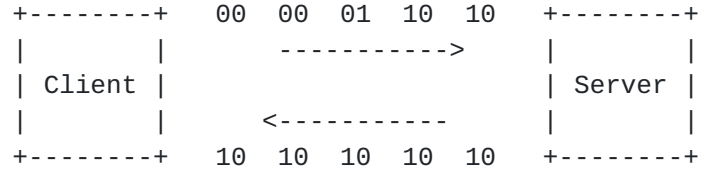
(c) The Server starts sending data and reflects the Delay Sample.



(d) The Client inverts the spin bit and reflects the Delay Sample.



(e) The Server reflects the Delay Sample.



(f) The client reverts the spin
bit and reflects the Delay Sample.

Figure 1: Spin bit and Delay bit

2.1. Delay Sample generation

During this first phase, endpoints play different roles. First of all a single delay sample must be bouncing per round trip period (and so per spin bit period). According to that statement and in order to simplify the general algorithm, the delay sample generation is in charge of just one of the two endpoints:

- the Client, when connection starts and spin bit is set to 0, initializes the delay bit of the first packet to 1, so it becomes the delay sample for that marking period. Only this packet is marked with the delay bit set to 1 for this round trip period; the other ones will carry only the spin bit;

- the server never initializes the delay bit to 1; its only task is to reflect the incoming delay bit into the next outgoing packet only if certain conditions occur.

Theoretically, in absence of network impairments, the delay sample should bounce between client and server continuously, for the entire duration of the connection. Actually, that is highly unlikely mainly for two different reasons:

1) the packet carrying the delay bit might be lost during its journey on the network which is unreliable by definition;

2) one of the two endpoints could stop or delay sending data because the application is limiting the amount of traffic transmitted;

To deal with these problems, the algorithm provides a procedure to regenerate the delay sample and to inform a possible observer that a problem has occurred, and then the measurement has to be restarted.

2.1.1. The recovery process

In order to relieve the server from tasks that go beyond the mere reflection of the sample, even in this case the recovery process belongs to the client. A fundamental assumption is that a delay sample is strictly related to its spin bit period. Considering this rule, the client verifies that every spin bit period ends with its delay sample. If that does not happen and a marking period

terminates without a delay sample, the client waits a further empty period; then, in the following period, it reinitializes the mechanism by setting the delay bit of the first outgoing packet to 1, making it the new delay sample. The empty period is needed to inform the intermediate points that there was an issue and a new delay measurement session is starting.

2.2. Delay Sample reflection

The reflection is the process that enables the bouncing of the delay sample between client and server. The behavior of the two endpoints is slightly different. With the exception of the client that, as previously exposed, generates a new delay sample, by default the delay bit is set to 0.

Server side reflection: when a packet with the delay bit set to 1 arrives, the server marks the first packet in the opposite direction as the delay sample, if it has the same spin bit value. While if it has the opposite spin bit value this sample is considered lost.

Client side reflection: when a packet with delay bit set to 1 arrives, the client marks the first packet in the opposite direction as the delay sample, if it has the opposite spin bit value. While if it has the same spin bit value this sample is considered lost.

In both cases, if the outgoing marked packet is transmitted with a delay greater than a predetermined threshold after the reception of the incoming delay sample (1ms by default), reflection is aborted and this sample is considered lost.

It is noteworthy that differently from what happens with the VEC for which the reflection always concerns the edge of the period, in this case reflection takes place for the packet that is carrying the delay bit regardless of its position within the period. For this reason it is necessary to introduce that condition of validation in order to identify and discard those samples that, due to reordering, might move to a contiguous period. Furthermore, by introducing a threshold for the retransmission delay of the sample, it is possible to eliminate all those measurements which, due to lack of traffic on the endpoints, would be overestimated and not true. Thus, the maximum estimation error, without considering any other delays due to flow control, would amount to twice the threshold (e.g. 2ms) per measurement, in the worst case.

3. Using the Spin bit and Delay bit for Hybrid RTT Measurement

Unlike what happens with the spin bit for which it is necessary to validate or at least heuristically evaluate the goodness of an edge, the delay sample can be used by an intermediate observer as a simple demarcator between a period and the following one eliminating the ambiguities on the calculation of the RTT found with the analysis of the spin-bit only. The measurement types, that can be done from the observation of the delay sample, are exactly the same achievable with the spin bit only (with or without the VEC).

3.1. End-to-end RTT measurement

The delay sample generation process ensures that only one packet marked with the delay bit set to 1 runs back and forth on the wire between two endpoints per round trip time. Therefore, in order to determine the end-to-end RTT measurement of a QUIC flow, an on-path passive observer can simply compute the time difference between two delay samples observed in a single direction. Note that a measurement, to be valid, must take into account the difference in time between the timestamps of two consecutive delay samples belonging to adjacent spin-bit periods. For this reason, an observer, in addition to intercepting and analyzing the packets containing the delay bit set to 1, must maintain awareness of each spin period in such a way as to be able to assign each delay sample to its period and, at the same time, identifying those periods that do not contain it.

3.2. Half-RTT measurement

An on-path passive observer that is sniffing traffic in both directions -- from client to server and from server to client -- can also use the delay sample to measure "upstream" and "downstream" RTT components. Also known as the half-RTT measurement, it represents the components of the end-to-end RTT concerning the paths between the client and the observer (upstream), and the observer and the server (downstream). It does this by measuring the delay between a delay sample observed in the downstream direction and the one observed in the upstream direction, and vice versa. Also in this case, it should verify that the two delay samples belong to two adjacent periods, for the upstream component, or to the same period for the downstream component.

3.3. Intra-domain RTT measurement

Taking advantage of the half-RTT measurements it is also possible to calculate the intra-domain RTT which is the portion of the entire RTT used by a QUIC flow to traverse the network of a provider (or part of

it). To achieve this result two observers, able to watch traffic in both directions, must be employed simultaneously at ingress and egress of the network to be measured. At this point, to determine the delay between the two observers, it is enough to subtract the two computed upstream (or downstream) RTT components.

4. Observer's algorithm and Waiting Interval

Given below is a formal summary of the functioning of the observer every time a delay sample is detected. A packet containing the delay bit set to 1:

- if it has the same spin bit value of the current period and no delay sample was detected in the previous period, then it can be used as a left edge (i.e., to start measuring an RTT sample), but not as a right edge (i.e., to complete and RTT measurement since the last edge). If the observation point is symmetric (i.e., it can see both upstream and downstream packets in the flow) and in the current period a delay sample was detected in the opposite direction (i.e., in the upstream direction), the packet can also be used to compute the downstream RTT component.

- if it has the same spin bit value of the current period and a delay sample was detected in the previous period, then it can be used at the same time as a left or right edge, and to compute RTT component in both directions.

Like stated previously, every time an empty period is detected, the observer must restart the measurement process and consider the next delay sample that will come as the beginning of a new measure, then as a left edge. As a result, being able to assign the delay sample to the corresponding spin period becomes a crucial factor for the proper functioning of the entire algorithm.

Considering that the division into periods is realized by exploiting the spin bit square wave, it is easy to understand that the presence of spurious spin edges -- caused by packet reordering -- would inevitably lead the observer to overestimate the amount of periods actually present in the transmission. This results in a greater number of empty periods detected and the consequent decrease of the actual RTT samples achievable. Therefore, in order to maximize the performance of the whole algorithm, the observer must implement a mechanism to filter out spurious spin edges.

To face this problem the waiting interval has to be introduced. Basically, every time a spin bit edge is detected, the observer sets a time interval during which it rejects every potential spurious edges observed on the wire. While, at the end of the interval it

starts again to accept changes in the spin bit value. This guarantees a proper protection against the spurious edges in relation to the size of the interval itself. For instance, an interval of 5ms is able to filter out edges that have been reordered by a maximum of 5ms. Clearly, the mechanism does its job for intervals smaller than the RTT of the observed connection (if RTT is smaller than the waiting interval the observer can't measure the RTT).

5. Adding a Loss bit to Delay bit and Spin bit

It is possible to introduce a mechanism to evaluate also the packet loss together with the delay measurement. In particular, the Client can select and mark a train of packets for this purpose, by using a loss bit, additionally to the spin bit and delay bit.

These packets bounce between Client and Server to complete two rounds and an Observer counts the marked packets during the two rounds and compares the counters to find Round Trip(RT) losses.

The problem to be solved is to choose the right number of packets to mark to avoid marked packets congestion on the slowest traffic direction. But the solution is simple, because it is enough to choose the number of packets that transit on the slowest direction during an RTT.

5.1. Round Trip Packet Loss measurement

The Client generates a train of marked packets (Packet Loss Samples) by using the additional bit called Loss bit. The marked packets are generated at the slowest direction rate (only when a packet arrives the Client marks an outgoing packet). The Server reflects these packets accordingly and, as a consequence, it could insert some not-marked packets. Then the client reflects the marked packets and the server reflects the marked packets again. The Client generates a new train of marked packets and so on.

The Packet Loss calculation can be made after the comparison of counters taken by the on-path passive observer. Indeed the Observer in the middle (upstream or downstream) sees the packet train twice and so it calculates the Observer Round Trip Packet Loss that, statistically, will be equal to the end-to-end Round Trip Packet Loss. So this measurement can be simply referred as Round Trip Packet Loss (RTPL).

In addition, this methodology allows Half-RTPL measurement and Intra-domain RTPL measurement, in the same way as described in the previous Sections for RTT measurement.

The method allows the packet loss calculation for a portion of the traffic but it is useful to perform RT Packet Loss measurement that gives useful information coupled with RTT.

6. Protocols

6.1. QUIC

The binding of this signal to QUIC is partially described in [I-D.ietf-quic-spin-exp], which adds the spin bit only to QUIC.

From an implementation point of view, the delay bit is placed in the partially unencrypted (but authenticated) QUIC header, alongside the spin bit, occupying one of the two bits left reserved for future experiments. As things stand, according to [I-D.ietf-quic-transport], the proposed scheme of the first header's byte would be 01SDRKPP.

6.2. TCP

The signal can be added to TCP by defining bit 4 of bytes 13-14 of the TCP header to carry the spin bit, and eventually bits 5 and 6 to carry additional information, like the delay bit and the loss bit.

7. Security Considerations

The privacy considerations for the hybrid RTT measurement signal are essentially the same as those for passive RTT measurement in general.

8. Acknowledgements

tbc

9. IANA Considerations

tbc

10. References

10.1. Normative References

[I-D.ietf-quic-spin-exp]
Trammell, B. and M. Kuehlewind, "The QUIC Latency Spin Bit", draft-ietf-quic-spin-exp-01 (work in progress), October 2018.

[I-D.ietf-quic-transport]
Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", draft-ietf-quic-transport-18 (work in progress), January 2019.

[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>.

[RFC7799] Morton, A., "Active and Passive Metrics and Methods (with Hybrid Types In-Between)", RFC 7799, DOI 10.17487/RFC7799, May 2016, <https://www.rfc-editor.org/info/rfc7799>.

[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>.

10.2. Informative References

[I-D.trammell-ippm-spin]
Trammell, B., "An Explicit Transport-Layer Signal for Hybrid RTT Measurement", draft-trammell-ippm-spin-00 (work in progress), January 2019.

[I-D.trammell-quic-spin]
Trammell, B., Vaere, P., Even, R., Fioccola, G., Fossati, T., Ihlar, M., Morton, A., and S. Emile, "Adding Explicit Passive Measurability of Two-Way Latency to the QUIC Transport Protocol", draft-trammell-quic-spin-03 (work in progress), May 2018.

Authors' Addresses

Mauro Cociglio
Telecom Italia
Via Reiss Romoli, 274
Torino 10148
Italy

Email: mauro.cociglio@telecomitalia.it

Giuseppe Fioccola
Huawei Technologies
Riesstrasse, 25
Munich 80992
Germany

Email: giuseppe.fioccola@huawei.com

Fabio Bulgarella
Politecnico di Torino

Email: fabio.bulgarella@guest.telecomitalia.it

Riccardo Sisto
Politecnico di Torino
Corso Duca degli Abruzzi, 24
Torino 10129
Italy

Email: riccardo.sisto@polito.it