

IP Performance Measurement WG  
Internet-Draft  
Intended status: Experimental  
Expires: July 13, 2019

B. Trammell, Ed.  
ETH Zurich  
January 09, 2019

**An Explicit Transport-Layer Signal for Hybrid RTT Measurement**  
**draft-trammell-ippm-spin-00**

**Abstract**

This document 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. It describes the algorithm for generating the signal, approaches for observing it to passively measure end-to-end latency, and proposes methods for adding it to a variety of IETF transport protocols.

**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 July 13, 2019.

**Copyright Notice**

Copyright (c) 2019 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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

<a href="#">1.</a>	<a href="#">Introduction . . . . .</a>	<a href="#">2</a>
<a href="#">2.</a>	<a href="#">Signal Generation Mechanism . . . . .</a>	<a href="#">3</a>
<a href="#">2.1.</a>	<a href="#">Illustrating the Mechanism . . . . .</a>	<a href="#">4</a>
<a href="#">3.</a>	<a href="#">Using the Signal for Hybrid RTT Measurement . . . . .</a>	<a href="#">6</a>
<a href="#">4.</a>	<a href="#">Using Only the Spin Bit . . . . .</a>	<a href="#">8</a>
<a href="#">5.</a>	<a href="#">Binding the Hybrid RTT Measurement Signal to Transport Protocols . . . . .</a>	<a href="#">8</a>
<a href="#">5.1.</a>	<a href="#">QUIC . . . . .</a>	<a href="#">8</a>
<a href="#">5.2.</a>	<a href="#">TCP . . . . .</a>	<a href="#">8</a>
<a href="#">6.</a>	<a href="#">Discussion . . . . .</a>	<a href="#">9</a>
<a href="#">6.1.</a>	<a href="#">Why the transport layer? . . . . .</a>	<a href="#">9</a>
<a href="#">7.</a>	<a href="#">Privacy and Security Considerations . . . . .</a>	<a href="#">10</a>
<a href="#">8.</a>	<a href="#">IANA Considerations . . . . .</a>	<a href="#">11</a>
<a href="#">9.</a>	<a href="#">Contributors . . . . .</a>	<a href="#">12</a>
<a href="#">10.</a>	<a href="#">Acknowledgments . . . . .</a>	<a href="#">12</a>
<a href="#">11.</a>	<a href="#">Informative References . . . . .</a>	<a href="#">12</a>
	<a href="#">Author's Address . . . . .</a>	<a href="#">14</a>

## [1. Introduction](#)

Latency is a key metric to understanding network operation and performance, and passive measurability of round trip times (RTT) is a useful and important, if generally unintentional, feature of many transport protocols. Passive measurement allows inspection of latency on productive traffic, avoiding problems with different treatment of productive and measurement traffic, and enables opportunistic measurement of latency without active measurement overhead.

However, since these features are largely accidental, methods for passive latency measurement are transport-dependent, and different heuristics for deriving metrics from these accidental signals may lead to non-comparable values. For example, methods applicable can be exclusively based on the TCP timestamp option [[RFC7373](#)] (see [[CACM-TCP](#)]), leverage both timestamps and matching sequence and acknowledgment numbers (see [[TMA-QOF](#)]), or rely on ACK-clocking in flows transmitting at a stable rate (see [[CARRA-RTT](#)]). In addition, they rely on features that may change or have undesirable side effects. For example, [[CARRA-RTT](#)] makes implicit assumptions about congestion control and pacing that may not hold for all senders, and timestamp-based methods require the TCP timestamp option to operate

Trammell

Expires July 13, 2019

[Page 2]

effectively, which adds 10 bytes of overhead to every packet and provides a relatively large amount of information for sender fingerprinting [[ZANDER-TS](#)].

This document defines a hybrid measurement [[RFC7799](#)] path signal [[PATH-SIGNALS](#)] to be embedded into a transport layer protocol, explicitly intended for exposing end-to-end RTT to measurement devices on path, following the principles elaborated in [[IPIM](#)]. 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. An evaluation of the spin bit and VEC mechanism in a variety of simulated and Internet testbed environments is given in [[IMC-SPIN](#)].

The document starts with a mechanism applicable to any transport-layer protocol, then explains how to bind the signal to a variety of IETF transport protocols, and describes a measurement methodology for deriving RTT samples from the signal.

## **[2.](#) Signal Generation Mechanism**

The hybrid RTT measurement signal consists of two parts:

- o a spin bit, which oscillates once per end-to-end RTT. Measuring the time between observed edges in the spin bit therefore provides an RTT sample.
- o a valid edge counter (VEC), which serves to reject edges in the spin bit signal which are invalid, due to loss, reordering, or delay.

This signal is encoded as three bits in the transport-layer header, or as a transport-layer option or extension, and the mechanism for generating these bits consists of a receive-side procedure for updating signal state, and a send-side procedure for encoding the signal on a packet.

On receiving a packet on a given connection, the receiver:

1. checks to see whether the packet is the latest packet for the connection. If not, it does not update any signal state. The method for determining whether a packet is the latest packet in a connection is necessarily transport-dependent.
2. takes the spin bit from the incoming packet and updates its NEXT\_SPIN value for the connection. If the receiver was the responder (server) of the connection, it sets NEXT\_SPIN to the to



the spin bit on the incoming packet. If the receiver was the initiator (client) of the connection, it sets NEXT\_SPIN to the inverse of the spin bit on the incoming packet.

3. updates its NEXT\_VEC value for the connection as follows. If the incoming spin bit value is the same as LAST\_SPIN (see below), it sets NEXT\_VEC to 0. Otherwise, if the incoming VEC value is 3, it sets NEXT\_VEC to 3. Otherwise, it sets NEXT\_VEC to the incoming VEC value plus 1.
4. stores the value of the incoming spin bit as LAST\_RECV\_SPIN for subsequent VEC generation.
5. stores the time at which this packet was received as RECV\_TIME.

On sending a packet on a given connection, the sender:

1. sets the outgoing spin bit to NEXT\_SPIN
2. sets the outgoing VEC to 0 if NEXT\_SPIN is the same as LAST\_SENT\_SPIN. Otherwise, it compares the current system time to RECV\_TIME. If more than a configurable delay has passed since RECV\_TIME, it sets the outgoing VEC to  $\max(\text{NEXT\_VEC}, 1)$ . Otherwise, it sets the outgoing VEC to NEXT\_VEC

This mechanism causes the spin bit to oscillate once per round trip time, and the VEC to count up to 3 and hold on each edge on the spin bit signal, in the absence of lost or reordered edges. Delays in sending an edge due to quiescence cause the VEC to reset to 1. Observation points can therefore estimate the end-to-end latency by observing these edges, as described in [Section 3](#). See [Section 2.1](#), below, for an illustration of this mechanism in action.

### **[2.1. Illustrating the Mechanism](#)**

To illustrate the operation of this signal, we consider a simplified model of a single bidirectional path between client and server as a queue with slots for five packets, and assume that both client and server sent packets at a constant rate. If each packet moves one slot in the queue per clock tick, note that this network has a RTT of 10 ticks. In the figures below, the signal is shown as two characters. The first denotes the value of the spin bit ( $\wedge = 1$ ,  $\vee = 0$ ), the second the value of the VEC (0-3). - means no packet in flight.

Initially, no packets are in flight, so there is no signal, as shown in Figure 1.



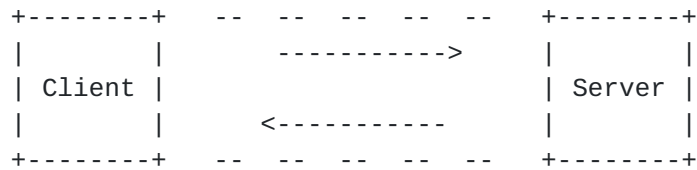


Figure 1: Initial state, no packets between client and server

The client begins sending packets with the spin bit and VEC set to zero, as shown in Figure 2.

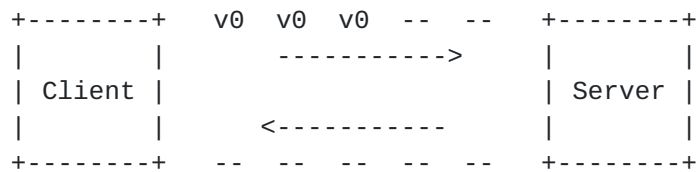


Figure 2: Client begins sending packets

The first packet arrives at the server five ticks later. It reflects the spin bit, and increments the VEC on its first packet, as shown in Figure 3.

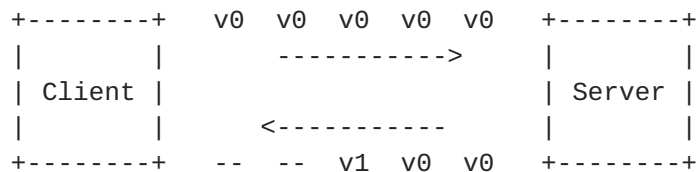


Figure 3: Server reflects first packet, sets edge

When the client receives this edge, again five ticks later, it inverts the spin bit and increments the VEC, as shown in Figure 4. In this way, the spin signal begins to oscillate around the path, with one edge in flight at any given time.

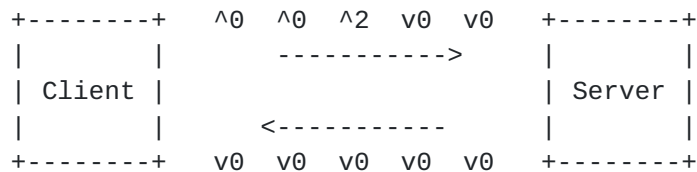


Figure 4: Client inverts spin bit, increments edge

And in turn, when this edge reaches the server, the VEC increments again, reaching its stable value of 3, as shown in Figure 5.





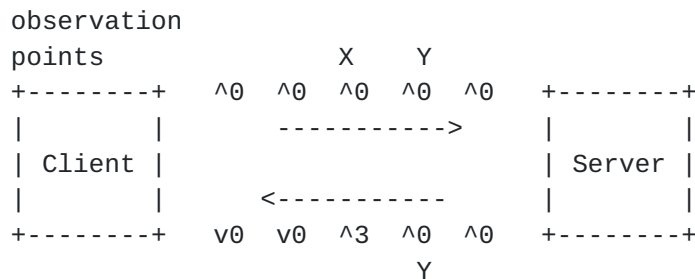


Figure 5: Server reflects edge, increments VEC

Here we can also see how measurement works. An observer watching the signal at single observation point X in Figure 5 will see an edge every 10 ticks, i.e. once per RTT. An observer watching the signal at a symmetric observation point Y in Figure 5 will see a server-client edge 4 ticks after the client-server edge, and a client-server edge 6 ticks after the server-client edge, allowing it to compute the components of RTT between itself and the client and between itself and the server.

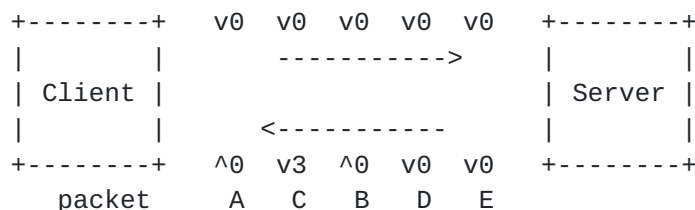


Figure 6: How the VEC detects reordering

Figure 6 shows how this mechanism works in the presence of reordering. Here, we assume the transport provides some form of packet sequencing (such as QUIC [QUIC-TRANSPORT] packet numbers or TCP [RFC0793] sequence numbers). Packet C carries the spin edge, and packet B is reordered on the way to the client. In this case, the client will begin sending spin 1 after the arrival of packet C, and ignore the spin bit flip to 1 on packet B, since  $B < C$ ; i.e. it does not increment the highest packet number seen. An on-path observer can also reject the spurious edges carried by packets B and D, even without knowledge of the transport protocol's sequence numbering (or, as is the case with QUIC, when the transport protocol's sequence numbering is encrypted), since the VEC is 0 on these packets.

### 3. Using the Signal for Hybrid RTT Measurement

When at least one sender is sending packets at full rate (i.e., is neither application nor flow-control limited), and the other sender is sending at least one packet per RTT (e.g. as is the case with the TCP acknowledgment-only packets on), the spin bit oscillates once per



RTT, and the VEC counts up to 3 and holds on the edges in the spin bit (the first packet carrying a new spin bit value in each direction). An on-path observer can observe the time difference between these edges in the spin bit signal in a single direction to measure one sample of end-to-end RTT. Note that this measurement, as with transport-specific passive RTT measurement, includes any transport protocol delay (e.g., delayed sending of acknowledgements) and/or application layer delay (e.g., waiting for a request to complete). These RTT samples can be used

The VEC can be used by observers to determine whether an edge in the spin bit signal is valid or not, as follows:

- o A packet containing an apparent edge in the spin signal with a VEC of 0 is not a valid edge, but may have been caused by reordering or loss, or was marked as delayed by the sender. It should therefore be ignored.
- o A packet containing an apparent edge in the spin signal with a VEC of 1 can be used as a left edge (i.e., to start measuring an RTT sample), but not as a right edge (i.e., to take an RTT sample since the last edge).
- o A packet containing an apparent edge in the spin signal with a VEC of 2 can be used as a left edge, but not as a right edge. If the observation point is symmetric (i.e., it can see both upstream and downstream packets in the flow), the packet can also be used to take a component RTT sample on the segment of the path between the observation point and the direction in which the previous VEC 1 edge was seen.
- o A packet containing an apparent edge in the spin signal with a VEC of 3 can be used as a left edge or right edge, and can be used to compute component RTT in either direction.

Taking only valid samples ensures that the RTT estimate provided is accurate. However, in some situations, it may result in a low sample rate. Since the VEC resets to one when a sender determines that its edge is delayed, bursty traffic on one side of the connection will cause the VEC not to count up to 3 very often. Likewise, a connection on which many edges are lost (because the connection itself is very lossy) will cause many samples to be rejected as well. Observers may choose to use heuristics in addition to VEC analysis to increase the sample rate in challenging network or traffic environments.



#### **4. Using Only the Spin Bit**

Note that, in the absence of loss and reordering, the single spin bit on its own suffices to provide one accurate RTT sample per RTT to on-path observers. Instead of using two additional bits for the VEC to reject bad samples caused by less than ideal network conditions, protocol designers can instead opt to add only the spin bit to the protocol, and shift the burden of correcting the RTT sample stream to observers, in keeping with the third principle elaborated in [IPIM]: the cost of deriving measurements from measurable protocols should be shifted from the participants to the measurement consumers where possible. Indeed, this is the approach followed by QUIC when adding the spin signal to the protocol (see [Section 5.1](#)).

#### **5. Binding the Hybrid RTT Measurement Signal to Transport Protocols**

The following subsections define how to bind the spin bit to various IETF transport protocols. As of this writing, bindings are specified for QUIC and TCP.

##### **5.1. QUIC**

This signal was originally specified for the QUIC transport protocol [[QUIC-TRANSPORT](#)], as the encrypted design of that protocol makes passive RTT measurement impossible. The binding of this signal to QUIC is partially described in [[QUIC-SPIN-EXP](#)], which adds the spin bit only (without the VEC) to QUIC for experimentation purposes.

The "latest packet" determination for QUIC is made using the QUIC packet number: only packets which have a packet number greater than the highest packet number seen are considered when generating the signal.

Note that, when used with QUIC, the signal only appears on short header packets; long header packets are ignored for the purposes of generating the signal. Since either the client or the server may start sending short header packets first, both sides initialize their NEXT\_SPIN value to 0.

##### **5.2. 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 bits 5 and 6 to carry the VEC, as shown in Figure 7.



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
								R		C		E		U	
					S		VEC		s		W		C		R
								v		R		E		G	
+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+

Figure 7: Definition of bytes 13 and 14 of TCP header with spin bit S and VEC

The "latest packet" determination for TCP is made using the TCP sequence and acknowledgment numbers: only packets which have a sequence number greater than highest sequence number seen, accounting for wraparound, or which have a sequence number equal to the last sequence number seen and an acknowledgment number higher than the highest acknowledgment number seen, accounting for wraparound, are considered when generating the signal.

Since use of the reserved bits may cause connectivity issues in situations where overzealous interpretation by devices on path of "must be zero" for the reserved bits in byte 13 of the TCP header [RFC0793], the addition of the signal to TCP includes a simple fallback mechanism. The client sets NEXT\_SPIN to 1 and NEXT\_VEC to 0 on its initial SYN. If this SYN is lost, the client disables generation of the signal for the life of the connection.

A cursory initial evaluation presented in [IMC-SPIN] suggests that the deployability of a latency spin signal in the reserved bits of TCP is on the order of equivalent to the deployability of a latency spin signal carried in a newly-defined experimental TCP option [RFC6694].

## 6. Discussion

This signal is the result of work carried out in various BoFs and working groups in the IETF. This section attempts to answer questions that have been posed in those contexts about approaches such as that outlined in this document.

Additional discussion of privacy and security relevant questions is given in [Section 7](#).

### 6.1. Why the transport layer?

As this path signal is (by definition) designed for consumption by devices on the path, and the transport layer is designed for end-to-end operation, an obvious question presents itself: isn't this a





layer violation? The answer is both "not really" and "it doesn't matter".

The signal defined in this document is designed to measure per-connection, end-to-end RTT. The per-connection nature of the signal leverages the assumption that all packets of a given connection (n-tuple flow, including transport layer ports) will be routed over the same path over a given time interval (on the scale of multiple RTTs) to ensure observability at all points along the path. As it is necessarily a per-connection signal, it is best carried at the transport layer. In addition, the need to reject retransmitted or duplicated packets in the generated signal implies the need for sequence or packet numbering, which is also inherently per-connection, and therefore a transport-layer function.

In any case, adding this signal to network layer protocols is unlikely to prove deployable. IPv6 hop-by-hop and destination options [[RFC8200](#)] do not work on a significant minority of measured network paths [[RFC7872](#)], and IPv4 [[RFC0791](#)] options are even less usable.

## 7. Privacy and Security Considerations

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

A question was raised during the discussion of this signal within the QUIC working group and the QUIC RTT Design Team: does passive RTT measurement pose a privacy risk? The short answer is no [[PAM-RTT-PRIVACY](#)]. Normal variations in Internet RTT are great enough that RTT measurements are not useful for geolocation of an endpoint beyond the resolution and error available with even low-quality, freely-available IP address geolocation. In the event that the true endpoint address is not known (for example, in the case of anonymity networks), latency information could be used for deanonymization. However, in this case, the signal will not carry end-to-end RTT, rather exit-to-public-end RTT, as these networks typically terminate transport-layer connections.

RTT information may be used to infer the occupancy of queues along a path; indeed, this is part of its utility for performance measurement and diagnostics. When a link on a given path has excessive buffering (on the order of hundreds of milliseconds or more), such that the difference in delay between an empty queue and a full queue dwarfs normal variance and RTT along the path, RTT variance during the lifetime of a flow can be used to infer the presence of traffic on the bottleneck link. In practice, however, this is not a concern for hybrid measurement of congestion-controlled traffic, since any

Trammell

Expires July 13, 2019

[Page 10]

observer in a situation to observe RTT passively need not infer the presence of the traffic, as it can observe it directly.

In addition, since RTT information contains application as well as network delay, patterns in RTT variance from minimum, and therefore application delay, can be used to infer or fingerprint application-layer behavior. However, as with the case above, this is not a concern with passive measurement, since the packet size and interarrival time sequence, which is also directly observable, carries more information than RTT variance sequence.

We therefore conclude that the high-resolution, per-flow exposure of RTT for passive measurement as provided by this signal poses negligible marginal risk to privacy.

Since the hybrid RTT measurement signal is disconnected from transport mechanics, an endpoint implementing the signal that has a model of the actual network RTT and a target RTT to expose can "lie" about its spin bit edges, by anticipating or delaying observed edges, even without coordination with and the collusion of the other endpoint. When passive measurement is used for purposes where one endpoint might gain a material advantage by representing a false RTT, e.g. SLA verification or enforcement of telecommunications regulations, this situation raises a question about the trustworthiness of the RTT measurements produced from this signals

This issue must be appreciated by users of information produced from sampling the hybrid RTT measurement signal. In the case of TCP, mitigation is trivial as existing passive measurement methods can be used to verify the operation of the signal. The case of QUIC is harder, as in the general case it is impossible to verify explicit path signals with two complicit endpoints connected via an encrypted channel (see [[WIRE-IMAGE](#)]). However, here there are also verification methods possible. A lying server could be contacted by an honest client under the control of a verifying party, and the client's RTT estimate compared with the spin-bit exposed estimate. A server/client pair that collaborate to lie may be subject to dynamic analysis along paths with known RTTs. We consider the ease of verification of lying in situations where this would be prohibited by regulation or contract, combined with the consequences of violation of said regulation or contract, to be a sufficient incentive in the general case not to do it.

## **8. IANA Considerations**

This document has no current actions for IANA.



Should consensus emerge that deployment of the spin bit in TCP is worth pursuing, a companion document submitted to the TCP Maintenance and Minor Extensions (TCPM) Working Group would need to request the following assignments in the IANA TCP Header Flags registry for the purposes of carrying the Spin Bit and Valid Edge Counter on TCP packets:

- o Bit 4: Hybrid RTT Measurement Spin Bit, as defined in this document
- o Bit 5: Hybrid RTT Measurement Valid Edge Counter, high-order bit, as defined in this document
- o Bit 6: Hybrid RTT Measurement Valid Edge Counter, low-order bit, as defined in this document

## **9. Contributors**

This work is based in part on [[QUIC-SPIN](#)], of which it is a generalization. In addition to the editor(s) and author(s) of this document, [[QUIC-SPIN](#)] was the work of Piet De Vaere, Roni Even, Giuseppe Fioccola, Thomas Fossati, Marcus Ihlar, Al Morton, and Emile Stephan.

## **10. Acknowledgments**

Many thanks to Christian Huitema, who originally proposed the spin bit as pull request 609 on [[QUIC-TRANSPORT](#)]. Thanks to Tobias Buehler for feedback on the draft, and for Alexandre Ferrieux for input on the Valid Edge Counter. Special thanks to the QUIC RTT Design Team for discussions leading especially to the privacy and security considerations section.

This work is partially supported by the European Commission under Horizon 2020 grant agreement no. 688421 Measurement and Architecture for a Middleboxed Internet (MAMI), and by the Swiss State Secretariat for Education, Research, and Innovation under contract no. 15.0268. This support does not imply endorsement.

## **11. Informative References**

[CACM-TCP]

Strowes, S., "Passively Measuring TCP Round-Trip Times (in Communications of the ACM)", October 2013.



## [CARRA-RTT]

Carra, D., Avrachenkov, K., Alouf, S., Blanc, A., Nain, P., and G. Post, "Passive Online RTT Estimation for Flow-Aware Routers Using One-Way Traffic (NETWORKING 2010, LNCS 6091, pp. 109-121)", 2010.

## [IMC-SPIN]

De Vaere, P., Buehler, T., Kuehlewind, M., and B. Trammell, "Three Bits Suffice - Explicit Support for Passive Measurement of Internet Latency in QUIC and TCP (ACM IMC 2018)", November 2018.

## [IPIM]

Allman, M., Beverly, R., and B. Trammell, "Principles for Measurability in Protocol Design (ACM CCR 47(2), pp. 2-12)", April 2017.

## [PAM-RTT-PRIVACY]

Trammell, B. and M. Kuehlewind, "Revisiting the Privacy Implications of Two-Way Internet Latency Data (PAM 2018, LNCS 10771, pp. 73-84)", March 2018.

## [PATH-SIGNALS]

Hardie, T., "Transport Protocol Path Signals", [draft-iab-path-signals-03](#) (work in progress), January 2019.

## [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.

## [QUIC-SPIN-EXP]

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

## [QUIC-TRANSPORT]

Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", [draft-ietf-quic-transport-17](#) (work in progress), December 2018.

## [RFC0791]

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





- [RFC0793] Postel, J., "Transmission Control Protocol", STD 7, [RFC 793](#), DOI 10.17487/RFC0793, September 1981, <<https://www.rfc-editor.org/info/rfc793>>.
- [RFC6694] Moonesamy, S., Ed., "The "about" URI Scheme", [RFC 6694](#), DOI 10.17487/RFC6694, August 2012, <<https://www.rfc-editor.org/info/rfc6694>>.
- [RFC7373] Trammell, B., "Textual Representation of IP Flow Information Export (IPFIX) Abstract Data Types", [RFC 7373](#), DOI 10.17487/RFC7373, September 2014, <<https://www.rfc-editor.org/info/rfc7373>>.
- [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>>.
- [RFC7872] Gont, F., Linkova, J., Chown, T., and W. Liu, "Observations on the Dropping of Packets with IPv6 Extension Headers in the Real World", [RFC 7872](#), DOI 10.17487/RFC7872, June 2016, <<https://www.rfc-editor.org/info/rfc7872>>.
- [RFC8200] 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>>.
- [TMA-QOF] Trammell, B., Gugelmann, D., and N. Brownlee, "Inline Data Integrity Signals for Passive Measurement (in Proc. TMA 2014)", April 2014.
- [WIRE-IMAGE] Trammell, B. and M. Kuehlewind, "The Wire Image of a Network Protocol", [draft-trammell-wire-image-04](#) (work in progress), April 2018.
- [ZANDER-TS] Zander, S. and S. Murdoch, "An Improved Clock-skew Measurement Technique for Revealing Hidden Services (USENIX Security 2008, pp. 211-225)", 2008.

## Author's Address

Brian Trammell (editor)  
ETH Zurich

Email: [ietf@trammell.ch](mailto:ietf@trammell.ch)

