L, Q, R, and T - Which Spin Bit Cousin Is Here to Stay?

Ike Kunze, Klaus Wehrle, Jan Rüth
{kunze,wehrle,rueth}@comsys.rwth-aachen.de
RWTH Aachen University

ABSTRACT

Network operators utilize traffic monitoring to locate and fix faults or performance bottlenecks. This often relies on intrinsic protocol semantics, e.g., sequence numbers, that many protocols share implicitly through their packet headers. The arrival of (almost) fully encrypted transport protocols, such as QUIC, significantly complicates this monitoring as header data is no longer visible to passive observers. Recognizing this challenge, QUIC offers explicit measurement semantics by exposing the spin bit to measure a flow's RTT. Ongoing efforts in the IETF IPPM working group argue to expose further information and enable the passive quantification of packet loss. This work implements and evaluates four currently proposed measurement techniques (*L*-, *Q*-, *R*-, and *T*-bit). We find that all techniques generally provide accurate loss estimations, but that longer algorithmic intervals for *Q* and *R*, yet foremost for *T*, complicate detecting very small loss rates or loss on short connections. Deployment combinations of *Q* & *R* as well as *Q* & *L*, thus, have the best potential for accurately grasping the loss in networks.

CCS CONCEPTS

• **Networks** → **Network monitoring**.

ACM Reference Format:
Ike Kunze, Klaus Wehrle, Jan Rüth. 2021. L, Q, R, and T - Which Spin Bit Cousin Is Here to Stay?. In *Applied Networking Research Workshop (ANRW '21), July 24–30, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3472305.3472319

1 INTRODUCTION

The growing complexity of networks makes monitoring their performance increasingly challenging. Yet, accurate information on the current network status is vital so that operators can act on errors in a timely manner or quickly identify misbehaving flows. The types of available monitoring solutions significantly depend on the kind of observed network. Datacenters, e.g., allow for detailed monitoring [14] as operators can leverage both, passive measurements of existing production traffic and carefully injected active measurement traffic to obtain a complete picture of the network status. In contrast, telco operators are mostly limited to passive monitoring as comprehensive active measurements are either infeasible or would not provide representative information [6].

These passive measurements often rely on specific semantics of the deployed transport protocols. For example, packet loss can be estimated using TCP's sequence numbers and acknowledgments [2]. The ongoing shift to (almost) fully encrypted transports [27], such as QUIC [16], challenges these approaches as intrinsic protocol semantics are no longer visible to passive observers and thus lost for measurements.

Instead, new ways for enabling passive measurements are required. Allman et al. [1] argue that measurability should be explicitly included into the protocols. Adhering to this line of thought, QUIC incorporates a dedicated measurement bit, the spin bit, which is visible to passive observers and enables round-trip time (RTT) measurements [7].

The spin bit's concept is similar to *alternate marking* techniques, e.g., defined in RFC 8321 [11], whose idea is to modulate signals onto production traffic by carefully controlling the values of designated measurement bits which an observer can then measure. However, these concepts are deployed entirely in the network without cooperation or knowledge of the end-hosts. Currently, the IETF IPPM working group [15] discusses ways to bring these principles into end-host protocols. There are four distinct proposals for such *Explicit Flow Measurements (EFMs)*, called the L-, Q-, R-, and T-Bits [5] that enable packet loss measurements. Yet, there is little well-documented knowledge on their effectiveness in measuring packet loss and next to no academic research.

In this paper, we thus evaluate the four proposals using Mininet [22] and find that they mostly yield reasonable loss estimates. Yet, longer algorithmic phases make some approaches miss very seldom loss or loss on short connections. Specifically, this work contributes the following:

- We implement the end-host logic of the four measurement mechanisms in aioquic [21] (source code available at [19]).
- We devise a Mininet-based study to observe the four mechanisms in different network settings (setup available at [20]).

ANRW '21, July 24–30, 2021, Virtual Event, USA

- We find that all mechanisms provide decent results subject to random loss; however, Q-, R-, and T-Bit are challenged in times of burst loss or short transmissions.

Structure. Sec. 2 introduces EFMs, discusses related work, and presents the four mechanisms under study. We then discuss our Mininet-based testbed in Sec. 3 before evaluating the mechanisms in three scenarios. First, we inspect their behavior subject to random loss (Sec. 4.1), and, second, to burst loss (Sec. 4.2). Lastly, we evaluate the impact of different flow lengths in Sec. 4.3. Finally, Sec. 5 concludes the paper.

2 EXPLICIT FLOW MEASUREMENTS

Although having been an integral part of the early Internet [18], network measurements have mostly been designed independently from protocols and, as such, have usually relied on externally visible protocol semantics. Prominent examples are measurements of the RTT that base on the TCP handshake semantic [17] or on the timestamp option [28, 29]. However, as argued for and discussed by Allman et al. [1], explicit measurement capabilities integrated into protocols are desirable for a number of reasons, e.g., for providing a solid framework for making inferences on the network state.

2.1 Related Work

In this context, principles described in RFC 8321 [11] have sparked a new push towards embedding measurements into protocols. While the original *alternate marking* is intended for measurements within the network, i.e., without the participation of end-hosts, its principle of modeling measurable signaling information onto a communication channel has been picked up for the application to end-to-end flow measurements, also called *Explicit Flow Measurements (EFMs)*. QUIC even includes one instance following this scheme for enabling RTT measurements: the spin bit [16]. In essence, the server of a connection always *reflects* the received spin bit, while the client *flips* the spin bit after one RTT, thus creating an oscillating pattern. On-path observers can collect the end-to-end RTT by measuring the time between changes of the spin bit value as the half period of the square wave signal produced by the spin bit equals the RTT.

De Vaere et al. [7] present and evaluate the spin bit. In addition to the version incorporated in QUIC, they also describe a more sophisticated variant, the *Valid Edge Counter (VEC)*, which requires two additional bits and allows for filtering out invalid spin bit edges, e.g., due to reordering. Using a Mininet-based evaluation and traffic passing through real networks, the authors show the general applicability of the spin bit. Bulgarella et al. [4] propose a simplified version called the delay bit, which only requires one additional bit. In Mininet experiments, they show that their variant achieves similar performance as the VEC. Meanwhile, the delay bit has also been proposed as a standalone technique [5].

In this work, we focus on EFM-based mechanisms to measure packet loss [5]: the L-, Q-, R-, and T-Bits, which we will explain in the following sections. There is little work overall and especially no published academic work on the loss mechanisms. At IETF, Ferrieux et al. [10] provide an initial analysis of the Q- and L-Bits. They present measurements between a CDN and an ISP and compare results obtained in different countries. Additionally, they remark that flows have to have a certain length for the Q-Bit to be applicable.

2.2 EFM-based Loss Measurements

Cociglio et al. [5] currently propose four mechanisms for enabling packet loss measurements. While the L-Bit builds upon sender-side loss-detection provided by the transport protocol, the other three approaches rely on the periodic transmission of marked packets. The Q-Bit creates a constant square wave signal by flipping a bit after sending a fixed number of packets. The R-Bit builds upon this by itself marking packets for received Q bits. The T-Bit creates an initial train of packets that is then reflected several times between the server and the client. In the following, we briefly outline the approaches, how they enable packet loss measurements, as well as end-host and observer logics; please refer to [5] for an in-depth explanation. Sec. 2.3 then jointly discusses the techniques and analyzes their differences, especially regarding their loss observation capabilities.

2.2.1 Loss Event Bit (L-Bit). The L-Bit relies on loss-detection by the transport protocol and essentially reports the number of packets that have been declared lost to the network. For this, whenever the sender detects a loss, she increases a $counter_L$ by one. To convey the number of lost packets to an on-path observer, she sets the L-Bit (in the observable packet header) in the next packet and decreases the $counter_L$ by one and does so as long as it is above zero. Thus, for each packet lost, the sender marks exactly one packet.

End-Host Logic. The sender only has to check the current value of $counter_L$ and set the L-Bit accordingly. However, it requires a loss-detection mechanism on the transport layer.

Observer Logic. An on-path observer can simply count the number of observed L-Bit markings to determine the number of lost packets and might additionally count the number of overall transmitted packets to determine a loss rate.

2.2.2 Round Trip Loss Bit (T-Bit). The T-Bit is a multi-phase algorithm whose phases synchronize to the spin-bit periods. In a first step, spanning at least two spin-bit periods, the client transmits a train of packets with a set T-Bit (*generation train*). The server then reflects as many packets with a set T-Bit as it has received, i.e., lost packets are not reflected.

After a pause phase that spans at least one spin-bit period, the reflection phase begins. In this phase, the client first transmits as many packets with a set T-Bit to the server as it has received from the server in the previous reflection, i.e., again marked packets may have been lost. Finally, the server reflects all packets with a set T-Bit that it has received. At the end of this process, there is again a pause phase.

End-Host Logic. The main task for hosts implementing the T-Bit is to correctly assign received T-Bit packets to the different algorithm phases and then set the lengths of outgoing phases accordingly. The burden of deciding when to start which phase entirely lies on the client. The main challenge is that the underlying spin-bit periods, and with them the envisioned state transitions, are known to be susceptible to reordering [7], potentially causing faulty state transitions.

Observer Logic. Observers can determine lost packets between the generation and reflection phase by comparing the number of marked packets in both phases. For this, they have to detect and distinguish the different algorithm phases. The pause phases simplify the detection, while generation and reflection can be distinguished by comparing observations of two subsequent phases and choosing the one with a greater value as the generation phase.

2.2.3 Square Bit (Q-Bit). A Q-Bit sender creates a square wave signal by first sending N packets (a Q-Block) with an unset Q-Bit and subsequently N packets with a set Q-Bit.

End-Host Logic. The sender only has to alternate between the two signal levels after a fixed number of packets. Our implementation uses a Q-Block length of $N = 64$.

Observer Logic. If the value of N is known, on-path observers can simply compare the number of marked/unmarked packets during a Q-Block with the expected number N to determine the number of lost packets. If N is unknown, the observers can deduce N from the number of received packets as Cociglio et al. [5] propose choosing N as a power of 2. In both cases, the observers have to correctly assign the counted packets to Q-Blocks even when reordering occurs near to the flanks of the square waves. As a solution, Cociglio et al. [5] propose to add fixed thresholds to the phase change detection. Our implementation uses a threshold of 8 so that a new signal level is only assumed after 8 values of the new phase have been detected, while all intermediate values are still assigned to the previous phase.

2.2.4 Reflection Square Bit (R-Bit). The R-Bit is co-designed with and builds upon the Q-Bit, essentially using the same mechanism. Initially, the R-Bit is unset on outgoing packets. After the first Q-Block has been received, the R-Bit is toggled and its value is set on as many packets as have been received during the Q-Block. It then flips again and similarly denotes the number of packets in the subsequently received Q-Block and so forth. Since Q and R periods may overlap in time, e.g., when the packet count in each direction is highly asymmetric, the number of packets that need R-marking is adjusted when one or more new Q-Blocks end before the current R-Block is completed by setting it to the average number of packets per Q-Block since the last R-Block started. As such, the R-Bit conveys the average number of packets received per Q-Block, thus enabling the observation of a statistical loss rate.

End-Host Logic. The sender has to incorporate observer logic for the Q-Bit. More specifically, she has to to be aware of incoming Q-Bit phases and detect Q-Block boundaries even in light of reordering to correctly count the packets belonging to each phase. Additionally, she has to keep track of the average and properly react to changes such as the average dropping below the current R-marking count.

Observer Logic. In addition to the consideration regarding reordering for the Q-Bit, there is one significant challenge for an R-Bit observer: she cannot know the number of initially transmitted R-Bit packets. Thus, she can only compare the counted R-Bit packets to N, i.e., the initial Q-Block length.

This directly brings up the question of what conclusions can be drawn from the measurements of the different mechanisms. In the following considerations, we first discuss the varying path resolutions of the different mechanisms before concluding our theoretical findings in a general discussion.

2.3 Discussion

While all of the four techniques enable passive packet loss measurements, they differ in the complexity of the end-host and observer logic, and in their path resolution, i.e., which conclusions on the state of the network can be drawn from the measurements when seeing what portion of the traffic.

2.3.1 Path Resolution. The accuracy and applicability of the techniques depends on where an observer monitors traffic and in which direction. The following rationale is along the lines presented by Cociglio et al. [5]. Fig. 1 visualizes a network consisting of one client, one server, and an observer in between and highlights which parts of the network are covered by each technique. We first focus on a downstream observer that monitors traffic from the server to the client.

Q-Bit. The Q-Bit allows for measuring packet loss on the link *Downstream 1* as the observer can simply compare the counted number of Q-Bits to the Q-Block length N.

L-Bit. The L-Bit captures the overall loss of *Downstream 1* + *Downstream 2* (*downstream loss*). However, loss on *Downstream 1* may skew the observations as L-Bit markings can be lost prior to the observation point.

R-Bit. The R-Bit resolution is even broader as a uni-directional observer can only determine the *three-quarters loss* [5], i.e., the packet loss between the client and the downstream-side observer (*Downstream 1*, *Upstream 1*, and *Upstream 2*).

T-Bit. Finally, the T-Bit yields loss estimates for the observer-to-observer path, i.e., the entire communication path.

Note that the same considerations apply for an observer that only monitors upstream traffic. Further observations are possible when combining different approaches (the draft [5], e.g., suggests Q+R-Bit and Q+L-Bit) or when considering a bi-directional observer. Those are out of scope of this work.

2.3.2 General Discussion. Based on the presentation in this section, we can already make a few statements regarding the applicability of the different variants. Multi-stage algorithms, such as the R-Bit (as it depends on Q) and the T-Bit, require more complex state handling and are thus more difficult to implement. The Q-Bit and L-Bit are straightforward to implement, both on the end-hosts and on the observers, although mechanisms to handle reordering are required for the Q-Bit.

The T-Bit alone allows for the most flexible determination of packet loss across all network segments, especially if the observer is bi-directional. According to the IETF IPPM mailing list, the combination of Q- and R-Bits, as well as the T-Bit variant are currently under testing at Telecom Italia [26]. The Q+L combination is deployed at Akamai and Orange [24] and also featured in the lsquic [23] implementation. This already indicates that the different combinations of the loss mechanisms seem to be deployable effectively.

After this general discussion, we next provide an experimental comparison of the different loss mechanisms for which we will first discuss our methodology.

3 METHODOLOGY

As discussed in the previous Sec. 2, the four mechanisms differ significantly in their fundamental working principles and in their path resolution which makes a comprehensive comparison challenging. We consequently focus on evaluating the core characteristics of the mechanisms and do not investigate possible algorithm combinations or the possibility to locate loss on specific network segments. In the following, we describe our evaluation setup in more depth.

Network Scenario. We perform our study in a Mininet-based testbed visualized in Fig. 1 which consists of two end-hosts (*Server*, *Client*) and one intermediate switch (*Observer*). Similar approaches have already been used for investigating the spin bit [7] and the delay bit [4]. We shape a steady base delay of 10 ms on all links using *tc netem*.

Network Impairments. To study the effects of different loss characteristics, we select link *Downstream 1* for our experiments as a downstream observer can detect loss occurring on this link using all four mechanisms. For inducing the impairments, we place a dedicated network arbiter on *Downstream 1* which we then configure using *tc netem*.

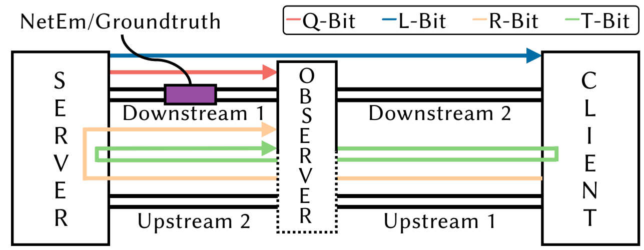

Figure 1: Our Mininet-based topology consists of two end-hosts (*Server, Client*) and one intermediate *Observer*. A netem-based arbiter induces network impairments on *Downstream 1*.

EFM Implementation. For our experiments, we utilize the aioquic [21] QUIC implementation. We extend it by implementing all four loss mechanisms and additionally extend the short header by an additional byte (without header protection) that we insert behind the original spin bit. This way, we can capture all mechanisms at the same time and compare their behavior subject to the same traffic. Our modified aioquic implementation is available at [19].

Measurements. We do not perform online observations. Instead, we capture incoming traffic on our *Observer* using *tcpdump* and then determine the behavior of the mechanisms offline. We additionally derive a groundtruth of the actual loss as we can only configure loss probabilities and the true loss can thus vary. For this, we collect queue statistics on our network arbiter using *eBPF* which we then correlate with general statistics on the traffic coming into the arbiter which we again capture using *tcpdump*.

In our analysis, we can derive the results obtained by the observer using our traffic captures and then compare them to groundtruth values based on our network arbiter.

4 EVALUATION

Our evaluation consists of three parts. First, we judge the general effectiveness of EFM-based loss measurements subject to idealized random loss behavior (Sec. 4.1). In this setting, we generate symmetric traffic between *Client* and *Server* that is not congestion-controlled similar to [4] and [7]. We then investigate the effect of burst losses while still keeping congestion control disabled (Sec. 4.2). Finally, we enable congestion control and switch to an asymmetric traffic scheme mirroring a typical download, i.e., *Client* requests files of different sizes from *Server* (Sec. 4.3). This enables investigating two aspects, 1) the performance of the EFM-based schemes subject to fluctuations in the sending rates, and 2), the impact of connection durations on the expressiveness of the measurements, i.e., whether they are equally effective for short-lived requests.

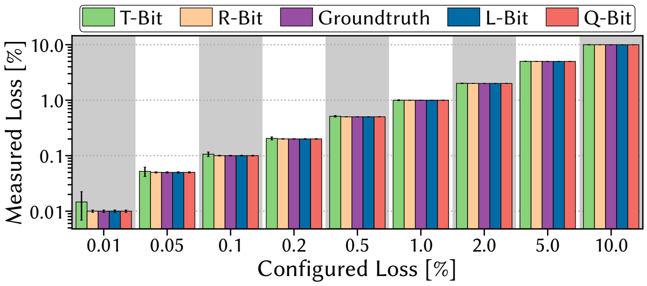

Figure 2: Mean loss rates with 99% CIs reported by the four mechanisms as well as the groundtruth for different random loss rates.

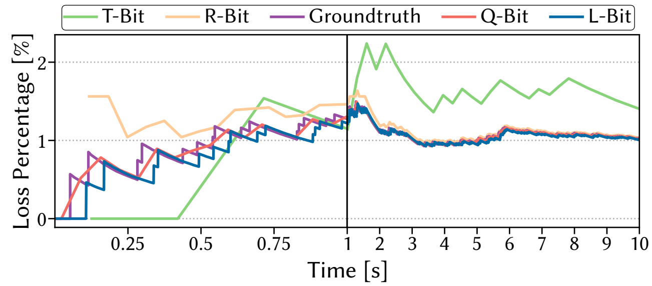

Figure 3: Cumulative loss rates of one selected setting with a configured mean random loss rate of 1% within the 1st second (left) and over 9 more seconds (right).

For each scenario, we perform 30 independent measurements and derive cumulative loss percentages as derived by the EFM schemes, i.e., loss rates summarizing a whole measurement run. If not stated otherwise, we report the mean over all runs together with 99% confidence intervals (CIs).

4.1 Random Loss

The first network impairment in our study is random loss to investigate the general effectiveness of the EFM-based loss mechanisms. Using *netem loss random*, we configure different random loss rates ranging from 0.01% to 10%. To obtain statistically meaningful results, even for the very low loss percentages, we transmit approximately one million packets in each experiment run. Following the methodology described by Biondini [3], this number is required to allow for an acceptable resolution of the groundtruth and should provide a relation of 0.1 between standard deviation and expected value of the resulting loss samples. Fig. 2 shows the estimated loss rates as well as the groundtruth on a logarithmic scale for the different mechanisms across the different scenarios.

As can be seen, all measurement techniques are able to derive relatively accurate loss rates that are very close to the groundtruth values. Only the T-Bit has problems correctly determining the low loss percentages as its lengthy pause periods, accounting for at least one third of each measurement cycle, make it easily miss lost packets.

A closer look. To get a better feeling for the actual behavior of the different mechanisms, we next closely inspect the accuracy across one example run. Fig. 3 illustrates the first second (left) and nine more seconds (right) of one handpicked measurement run in the 1% random loss setting. Note that all data points represent the total loss as provided by the different mechanisms at the time of loss detection.

A first result is that the mechanisms provide different numbers of observations due to their specific algorithmic periodicities. The Q-Bit, e.g., requires 64 transmitted packets before it can report a measurement. The R-Bit needs further 64 packets, as it first needs 64 in one direction that are then reflected. Consequently, the first measurement of the R-Bit comes after the first measurement of the Q-Bit. While the T-Bit does not have fixed-length periods, it relies on the spin bit for its phase transitions which generally prolongs the interval between measurements and causes it to provide the fewest measurements. We observe that the L-Bit best tracks the evolution of the groundtruth, but as it relies on end-host loss detection, it slightly lags behind due to the end-hosts having to detect the loss in the first place. Overall, we can see that the L-Bit and the Q-Bit are good approximations of the groundtruth while the R-Bit and especially the T-Bit require longer time to get close to the groundtruth.

While random loss enables deriving a baseline, loss often comes in bursts, e.g., when network buffers overflow, causing a series of consecutive packets to be dropped. Thus, we next investigate the impact of burst loss on the EFM mechanisms.

4.2 Burst Loss

We model burst loss using the simple Gilbert model [8, 12, 13] and choose a fixed overall loss percentage of 1% as this has shown adequate results in the random loss investigation. We then derive corresponding parameter settings using the methodology described by Nasralla et al. [25] to model different mean burst sizes. Note that the corresponding state transition probabilities within the simple Gilbert model become very small when keeping the average loss rate at 1%. We thus again transmit approximately one million packets in each experiment to achieve statistically meaningful results.

Fig. 4 depicts our results for four different average burst sizes. As can be seen, the L-Bit achieves a very high accuracy and is very close to the groundtruth for all investigated burst sizes. In contrast, the other approaches show decreasing accuracy for larger burst sizes. The main reason is that the loss detection techniques built into the transport protocols will eventually report on the lost packets in the case of the L-Bit. Thus, even though some L-marked packets will also get lost, the L-Bit will eventually report on all lost packets.

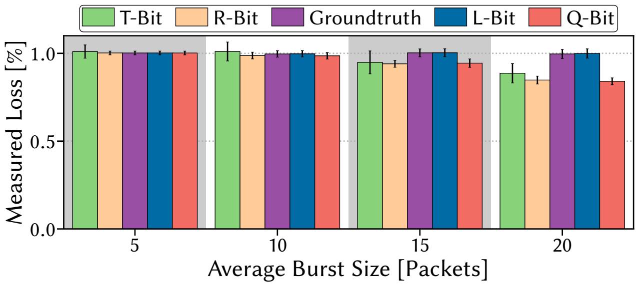

Figure 4: Mean loss rates with 99% CIs reported by the four mechanisms and the groundtruth for different configured burst loss rates at a fixed loss rate of 1%.

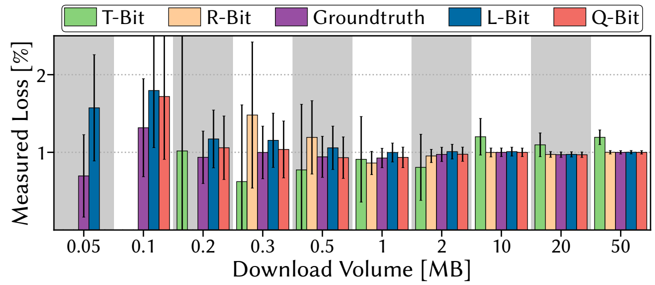

Figure 5: Mean loss rates with 99% CIs reported by the four mechanisms and the groundtruth for different download volumes at a fixed random loss rate of 1%.

In contrast, the other approaches rely entirely on signaling information and, furthermore, build upon periodic behavior. In the worst case, entire periods might be wiped out if the burst sizes become too large. For example, burst loss spanning an entire Q-Block length will remain unnoticed by the observer.[1] Consequently, the R-Bit provides similar results and, for both, increasing burst loss sizes cause an increasing difference between the real and the estimated loss rates. The T-Bit, on the other hand, shows a fluctuating behavior with larger confidence intervals. This is due to the fact that the observer needs to resynchronize to the T-Bit phases, thus yielding many invalid data points and thus reducing the overall number of measurements.

While the previous two settings are well-suited to investigate the general performance of the different EFM mechanisms, real flows governed by congestion control would not keep steady transmission rates when loss appears as a proxy for congestion. We thus next investigate the effect of enabling congestion control and of various flow lengths.

4.3 Flow Lengths

To study the usefulness of the EFM mechanisms on more realistic traffic, we switch to the significantly asymmetric scenario of a download. Additionally, we enable QUIC's congestion control (aioquic defaults to New Reno) to see whether this also plays a role. In contrast to the considerations in Sec. 4.1 and Sec. 4.2, we no longer ensure that there is enough traffic in each setting to achieve meaningful results, but instead explicitly aim to find out whether different flow lengths affect the accuracy of the loss mechanisms. We investigate the accuracy of the EFM mechanisms at 1% random loss with different download sizes, ranging from 50 kB, i.e., around 40 packets, to 50 MB, i.e., 40 000 packets, so covering typical website sizes, streaming video chunks, or larger assets.

[1]The burst resilience can be extended to two Q-Blocks (cf. [9]), but this mechanism is not yet part of the EFM draft and not included here.

For the small volume sizes, our results in Fig. 5 are very unstable, even the measured groundtruth varies a lot around the configured loss due to the small number of packets. Another observation, that follows from Fig. 3, is that the Q-, R- and T-Bits require a certain amount of transmitted data to actually provide measurements, namely, in our case, 100kB (Q-Bit), 200kB (T-Bit), and 300kB (R-Bit). In contrast, the L-Bit provides measurements right from the start.

Overall, we observe that the accuracy only reaches stable levels in the MB range. The phase-based algorithms require longer flows before producing reliable results, and, as such, they appear less suited to capture loss of short-sized interactions. Additionally, since T-Bit and R-Bit rely on receiver-generated packets, which, in this setting, are created through ACKs, the chosen ACK ratio further affects the applicability.

5 CONCLUSION

Explicit Flow Measurements provide a novel way of explicitly monitoring connection statistics by passive observers in the network. While measurements of the network delay are already incorporated in QUIC in the form of the spin bit, there is ongoing debate whether and how to enable packet loss measurements. In this work, we investigate four proposals, L-, Q-, R-, and T-Bit, currently discussed in the IETF IPPM working group. Using a Mininet testbed, we study the proposals across several network settings and find that they are generally capable of providing accurate measurements when subject to random loss. This changes in times of burst loss, as longer algorithmic phases of Q-, R-, and T-Bit make them easily miss loss if an entire phase is wiped out. On the other hand, the L-Bit provides an accuracy close to the groundtruth, but requires an accurate loss detection by the transport. Concluding, we find that the standalone L-Bit or combinations, such as Q+R-Bit or Q+L-Bit, look the most promising, at least when only considering measurement accuracy. While the required path resolution might also affect real deployment decisions, we leave corresponding investigations and considerations for future work.

REFERENCES

[1] Mark Allman, Robert Beverly, and Brian Trammell. 2017. Principles for Measurability in Protocol Design. *ACM SIGCOMM Computer Communication Review (CCR '17)* 47, 2 (2017). https://doi.org/10.1145/3089262.3089264

[2] Peter Benko and Andras Veres. 2002. A Passive Method for Estimating End-to-End TCP Packet Loss. In *IEEE Global Telecommunications Conference (GLOBECOM '02)*. https://doi.org/10.1109/GLOCOM.2002.1189102

[3] Gino Biondini. 2015. An Introduction to Rare Event Simulation and Importance Sampling. In *Handbook of Statistics*. Vol. 33. Elsevier, 29–68. https://doi.org/10.1016/B978-0-444-63492-4.00002-2

[4] Fabio Bulgarella, Mauro Cociglio, Giuseppe Fioccola, Guido Marchetto, and Riccardo Sisto. 2019. Performance Measurements of QUIC Communications. In *ACM Applied Networking Research Workshop (ANRW '19)*. https://doi.org/10.1145/3340301.3341127

[5] M. Cociglio, A. Ferrieux, G. Fiocolla, I. Lubashev, F. Bulgarella, I. Hamchaoui, M. Nilo, R. Sisto, and D. Tikhonov. 2021. *Explicit Flow Measurements Techniques*. Internet-Draft. IETF. https://datatracker.ietf.org/doc/draft-mdt-ippm-explicit-flow-measurements/ Work in Progress.

[6] Mauro Cociglio, Giuseppe Fioccola, Guido Marchetto, Amedeo Sapio, and Riccardo Sisto. 2019. Multipoint Passive Monitoring in Packet Networks. *IEEE/ACM Transactions on Networking (TON '19)* 27, 6 (2019). https://doi.org/10.1109/TNET.2019.2950157

[7] Piet De Vaere, Tobias Bühler, Mirja Kühlewind, and Brian Trammell. 2018. Three Bits Suffice: Explicit Support for Passive Measurement of Internet Latency in QUIC and TCP. In *ACM Internet Measurement Conference (IMC '18)*. https://doi.org/10.1145/3278532.3278535

[8] E. O. Elliott. 1963. Estimates of Error Rates for Codes on Burst-Noise Channels. *Bell System Technical Journal* 42, 5 (1963), 1977–1997. https://doi.org/10.1002/j.1538-7305.1963.tb00955.x

[9] Ericsson. 2021. Spindump on GitHub. https://github.com/EricssonResearch/spindump

[10] Alexandre Ferrieux, Isabelle Hamchaoui, and Igor Lubashev. 2019. Packet Loss Signaling for Encrypted Protocols. https://datatracker.ietf.org/meeting/105/materials/slides-105-maprg-packet-loss-signaling-for-encrypted-protocols-01.pdf.

[11] G. Fioccola, A. Capello, M. Cociglio, L. Castaldelli, M. Chen, L. Zheng, G. Mirsky, and T. Mizrahi. 2018. *Alternate-Marking Method for Passive and Hybrid Performance Monitoring*. RFC 8321. IETF. http://tools.ietf.org/rfc/rfc8321.txt

[12] E. N. Gilbert. 1960. Capacity of a Burst-Noise Channel. *Bell System Technical Journal* 39, 5 (1960), 1253–1265. https://doi.org/10.1002/j.1538-7305.1960.tb03959.x

[13] Gerhard Haßlinger and Oliver Hohlfeld. 2008. The Gilbert-Elliott Model for Packet Loss in Real Time Services on the Internet. In *VDE GI/ITG Conference-Measurement, Modelling and Evalutation of Computer and Communication Systems (MMB '08)*. https://www.vde-verlag.de/proceedings-en/563090019.html

[14] Qun Huang, Haifeng Sun, Patrick PC Lee, Wei Bai, Feng Zhu, and Yungang Bao. 2020. OmniMon: Re-architecting Network Telemetry with Resource Efficiency and Full Accuracy. In *ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM '20)*. https://doi.org/10.1145/3387514.3405877

[15] IP Performance Measurement Working Group. 2021. IP Performance Measurement (ippm). https://datatracker.ietf.org/wg/ippm/about/.

[16] Jana Iyengar and Martin Thomson. 2021. *QUIC: A UDP-Based Multiplexed and Secure Transport*. RFC 9000. Internet Engineering Task Force. http://tools.ietf.org/rfc/rfc9000.txt

[17] Hao Jiang and Constantinos Dovrolis. 2002. Passive Estimation of TCP Round-Trip Times. *ACM SIGCOMM Computer Communication Review (CCR '02)* 32, 3 (2002). https://doi.org/10.1145/571697.571725

[18] Leonard Kleinrock. 2010. An Early History of the Internet. *IEEE Communications Magazine* 48, 8 (2010). https://doi.org/10.1109/MCOM.2010.5534584

[19] I. Kunze et al. 2021. aioquic on GitHub. https://github.com/COMSYS/aioquic

[20] I. Kunze et al. 2021. EFM evaluation framework on GitHub. https://github.com/COMSYS/efm-evaluation-anrw

[21] Lainé, Jeremy. 2021. aioquic on GitHub. https://github.com/aiortc/aioquic

[22] Bob Lantz, Brandon Heller, and Nick McKeown. 2010. A Network in a Laptop: Rapid Prototyping for Software-Defined Networks. In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks (Hotnets-IX)*. https://doi.org/10.1145/1868447.1868466

[23] LiteSpeed Technologies Inc. 2021. lsquic on GitHub. https://github.com/litespeedtech/lsquic

[24] Igor Lubashev. 2021. Re: [ippm] Comparing Alternate Marking and Explicit Flow Measurements (Spin bit, ...). IETF IPPM Mailing List. https://mailarchive.ietf.org/arch/msg/ippm/v1je6osxrRS1zBCruPqEBnDyw8Y/

[25] Moustafa M. Nasralla, Chaminda T.E.R. Hewage, and Maria G. Martini. 2014. Subjective and Objective Evaluation and Packet Loss Modeling for 3D Video Transmission over LTE Networks. In *IEEE International Conference on Telecommunications and Multimedia (TEMU '14)*. https://doi.org/10.1109/TEMU.2014.6917770

[26] Massimo Nilo. 2021. [ippm] Explicit Flow Measurements implementations. IETF IPPM Mailing List. https://mailarchive.ietf.org/arch/msg/ippm/3EpGbp78CtimKTrEa4oKnDsXEZk/

[27] Jan Rüth, Ingmar Poese, Christoph Dietzel, and Oliver Hohlfeld. 2018. A First Look at QUIC in the Wild. In *Springer Passive and Active Measurement (PAM '18)*. https://doi.org/10.1007/978-3-319-76481-8_19

[28] Stephen D. Strowes. 2013. Passively Measuring TCP Round-Trip Times. *Communications of the ACM (CACM '13)* 56, 10 (Oct. 2013), 57–64. https://doi.org/10.1145/2507771.2507781

[29] Bryan Veal, Kang Li, and David Lowenthal. 2005. New Methods for Passive Estimation of TCP Round-Trip Times. In *Spinger Passive and Active Measurement Conference (PAM '05)*. https://doi.org/10.1007/978-3-540-31966-5_10