Network Working Group M. Bagnulo
Internet-Draft Universidad Carlos III de Madrid
Intended status: Experimental 5 February 2025
Expires: 9 August 2025

rLEDBAT for QUIC
draft-bagnulo-iccrg-rledbat-quic-00

Abstract

This document specifies rLEDBAT for QUIC, a set of mechanisms that enable the execution of a less-than-best-effort congestion control algorithm for QUIC at the receiver end. This draft explores adaptation strategies for integrating rLEDBAT with QUIC’s framework, aiming to maintain compatibility with QUIC.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

QUIC [RFC9000] is a transport protocol designed to provide fast connection establishment, multiplexed streams without head-of-line blocking, and built-in encryption using TLS 1.3. QUIC operates over UDP, allowing it to function efficiently even in environments with high latency or packet loss. With its user-space implementation and flexible congestion control mechanisms, QUIC has been widely adopted for web traffic, including HTTP/3.

Receiver-driven Low Extra Delay Background Transport (rLEDBAT)[I-D.irtf-iccrg-rledbat] is a congestion control mechanism designed for background data transfers in TCP. By leveraging delay-based congestion signals, rLEDBAT enables a receiver to control the sender’s transmission rate to minimize interference with foreground traffic. This approach ensures that background transfers utilize available bandwidth without degrading the performance of latency-sensitive applications. rLEDBAT achieves this by dynamically adjusting the receiver’s advertised window, effectively signaling the sender to slow down when network delays increase. rLEDBAT has been adopted for background traffic, including software updates, backups and prefetching.

The goal of this document is to define how rLEDBAT can be adapted for QUIC. Unlike TCP, QUIC does not expose the receiver’s advertised window in the same way, requiring a different approach for implementing receiver-driven congestion control. This draft explores possible mechanisms to extend rLEDBAT for QUIC . The proposed design aims to support background data transfers in QUIC without introducing excessive network latency or disrupting higher-priority traffic.

2. Designing rLEDBAT for QUIC

In order to enable a QUIC receiver to execute rLEDBAT, we need to devise the means for the QUIC receiver to feed information about the delay and packet loss to the LBE congestion control algorithm executed at the receiver. In addition, the LBE congestion control algorithm produces a congestion window value as an output which is to be used by the QUIC receiver to modulate the rate at which the QUIC sender is transmitting. We need to analyze how all this can be accomplished.

2.1. RTT Measurements

A QUIC receiver can measure the RTT passively or actively. Passive measurements of the RTT can be done by simply measuring the RTT of the exchange packets. In the QUIC protocol, all packets exchanged are acknowledged, so simply matching the packet sent by the receiver and the ACK frame generated by the sender allows the receiver to passively measure the RTT.

There is, however, a potential problem. RFC 9002 [RFC9002] explicitly states that an incoming ACK frame should be used to compute the RTT if and only if the following two conditions are met: (i) the largest acknowledged packet number is newly acknowledged, and (ii) at least one of the newly acknowledged packets was ack-eliciting.

Unfortunately, a common use case for rLEDBAT is the one where the receiver is downloading a large file from the sender, meaning that the receiver is not sending any data and, as such, is only generating non-ack-eliciting frames (i.e., it is only generating ACK frames from the incoming data). The reason for such a recommendation is that ACKs for non-ack-eliciting frames can be arbitrarily delayed, as the receiver of such frames is not required to issue the corresponding ACK frame within a certain time bound (which is the case for ack-eliciting frames).

This is similar to what happens in rLEDBAT for TCP, and the discussion presented in Section 3.2.1.1 of the rLEDBAT for TCP draft holds. That is, in the common use case for rLEDBAT, the sender is transmitting a large file, so, in general, it will have data to send, enabling an accurate measurement of the RTT most of the time. Similar considerations also apply, namely, the rLEDBAT receiver should apply a minimum filter when estimating the RTT in order to discard large samples caused by periods in which the sender has no data to send. Also, in cases where it is the flow control exerted by the receiver that is preventing the sender from sending more data, the sender should discard samples generated while reducing the sender’s window through flow control.

2.1.1. Reported ACK Delay

QUIC allows the sender of an ACK frame to report any additional delay that the issuer has incurred before returning the ACK. This can be used to correct the RTT computation by discounting the reported delay in the RTT estimation. While this is straightforward for ack-eliciting frames, it is less so for non-ack-eliciting frames.

The ACK for a non-ack-eliciting frame is included in the next ack-eliciting packet generated. It is unclear whether the value reported in the ACK delay field of such an ACK frame includes all the time since the packet was received and the ACK was issued. If so, then this can help to correct the RTT computation. If not, then it is uncertain what value of ACK delay these frames include, if any.

This then creates another question: if these frames include an unusable value for the ACK delay, should the receiver use the ACK delay value included in the ACK for the ack-eliciting frames? The problem is that making corrections for some of the samples and not for others will distort the RTT estimation.

2.1.2. Active RTT Measurement

A QUIC receiver can also perform active measurements to measure the RTT. The QUIC protocol includes PING frames that require the receiver to respond with an ACK frame. The receiver can periodically generate PING frames and compute the RTT using those. The receiver should then issue several (3?) PING frames every RTT to allow a robust measurement of the RTT, even if one or more of such PING frames are lost.

The benefit of this approach is that it provides accurate RTT measurements, irrespective of the traffic pattern generation of the sender. The downside is that it requires extra logic to generate these PING frames and requires extra link capacity and processing to handle them.

2.2. Packet Loss Detection

A QUIC rLEDBAT receiver can detect packet loss through retransmission of data. For that, the receiver inspects the received data frames and detects a retransmission as follows: suppose that X is the largest offset corresponding to data received so far (not necessarily all data corresponding to a smaller offset was received already), and Y is the largest packet number in the application data space for a packet received so far. A packet with packet number m and including a data frame with offset n is a retransmission if m > Y and n < X.

2.3. Controlling the Sender’s Rate

The QUIC receiver can control the sender’s rate using the MAX_DATA frame. The MAX_DATA frame carries the amount of data that can be sent in that connection, across all streams. A QUIC receiver with rLEDBAT enabled will convey the congestion window computed by the LEDBAT++ algorithm running in the receiver using the MAX_DATA frame.

Similarly to the case of TCP, a QUIC rLEDBAT receiver should refrain from shrinking the announced value. While it is not an error, it will not have any effect on the sender. So, again similarly to the case of TCP, the QUIC rLEDBAT receiver should only reduce the announced value in the amount of bytes acknowledged.

3. Security Considerations

TBD

4. IANA Considerations

No actions are required from IANA.

5. Acknowledgements

This work was supported by the EU through the StandICT project RXQ.

6. Informative References

[I-D.irtf-iccrg-rledbat]
Bagnulo, M., Garcia-Martinez, A., Montenegro, G., and P. Balasubramanian, "rLEDBAT: receiver-driven Low Extra Delay

Background Transport for TCP", Work in Progress, Internet-Draft, draft-irtf-iccrg-rledbat-10, 3 February 2025, <https://datatracker.ietf.org/doc/html/draft-irtf-iccrg-rledbat-10>.

[RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <https://www.rfc-editor.org/info/rfc9000>.

[RFC9002] Iyengar, J., Ed. and I. Swett, Ed., "QUIC Loss Detection and Congestion Control", RFC 9002, DOI 10.17487/RFC9002, May 2021, <https://www.rfc-editor.org/info/rfc9002>.

Author’s Address

Marcelo Bagnulo
Universidad Carlos III de Madrid
Email: marcelo@it.uc3m.es