QUIC
Internet-Draft
Intended status: Standards Track
Expires: 23 April 2026

J. Iyengar
Fastly
I. Swett
Google
M. K端hlewind
Ericsson
20 October 2025

QUIC Acknowledgment Frequency
draft-ietf-quic-ack-frequency-12

Abstract

This document specifies an extension to QUIC that enables an endpoint to request its peer change its behavior when sending or delaying acknowledgments.

Note to Readers

Discussion of this draft takes place on the QUIC working group mailing list (quic@ietf.org), which is archived at https://mailarchive.ietf.org/arch/search/?email_list=quic. Source code and issues list for this draft can be found at https://github.com/quicwg/ack-frequency.

Working Group information can be found at https://github.com/quicwg.

Status of This Memo

Table of Contents

1. Introduction

The QUIC transport protocol recommends sending an ACK frame after receiving at least two ack-eliciting packets; see Section 13.2 of [QUIC-TRANSPORT]. However, the data receiver determines how frequently to send acknowledgments in response to ack-eliciting packets, without any ability for the data sender to influence this behavior. This document specifies an extension to QUIC that enables an endpoint to request its peer change its behavior when sending or delaying acknowledgments.

This document defines a new transport parameter that indicates support of this extension and specifies two new frame types to request changes to the peer's acknowledgement behavior.

1.1. Terms and Definitions

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

In the rest of this document, "sender" refers to a QUIC data sender (and acknowledgment receiver). Similarly, "receiver" refers to a QUIC data receiver (and acknowledgment sender).

This document uses terms, definitions, and notational conventions described in Section 1.2 and Section 1.3 of [QUIC-TRANSPORT].

2. Motivation

A receiver acknowledges received packets, but can delay sending these acknowledgments. Delaying acknowledgments can impact a data sender's throughput, loss detection and congestion controller performance, as well as CPU utilization at both endpoints.

Reducing the frequency of acknowledgments can improve connection and endpoint performance in the following ways:

* Sending UDP datagrams is very CPU intensive on some platforms. A data receiver can decrease its CPU usage by reducing the number of acknowledgement-only packets sent. Experience shows that this reduction can be critical for high packet rate connections.

* Similarly, receiving UDP datagrams can also be CPU intensive. Reducing the acknowledgement frequency also reduces the data sender's CPU usage because it receives and processes fewer acknowledgment-only packets.

* For asymmetric link technologies, such as DOCSIS, LTE, and satellite, connection throughput in the forward path can become constrained when the reverse path is filled by acknowledgment packets [RFC3449]. When traversing such links, reducing the number of acknowledgments can achieve higher connection throughput, lower the impact on other flows or optimise the overall use of transmission resources [Cus22].

* The rate of acknowledgment packets can reduce link efficiency, including transmission opportunities or battery life, as well as transmission opportunities available to other flows sharing the same link.

However, as discussed in Section 8, a unilateral reduction in acknowledgement frequency can lead to undesirable consequences for congestion control and loss recovery. [QUIC-TRANSPORT] specifies a simple delayed acknowledgment mechanism (Section 13.2.1 of [QUIC-TRANSPORT]) without any ability for the data sender to influence this behavior. A data sender's constraints on the acknowledgment frequency need to be taken into account to maximize congestion controller and loss recovery performance. This extension provides a mechanism for a data sender to signal its preferences and constraints.

3. Negotiating Extension Use

After a data receiver advertises support for this extension, two new frames can be sent by the data sender to provide guidance about delaying and sending ACK frames. These frames are the ACK_FREQUENCY frame (see Section 4) and the IMMEDIATE_ACK frame (see Section 5).

Endpoints advertise their support for receiving ACK_FREQUENCY and IMMEDIATE_ACK frames as described in this document by sending the following transport parameter (Section 7.2 of [QUIC-TRANSPORT]):

min_ack_delay (0xff04de1b): A variable-length integer representing the minimum amount of time, in microseconds, that the endpoint sending this value is willing to delay an acknowledgment. This limit could be based on the data receiver's clock or timer granularity. min_ack_delay is used by the data sender to avoid requesting too small a value in the Requested Max Ack Delay field of the ACK_FREQUENCY frame.

An endpoint's min_ack_delay MUST NOT be greater than its max_ack_delay. Endpoints that support this extension MUST treat receipt of a min_ack_delay that is greater than the max_ack_delay as a connection error of type TRANSPORT_PARAMETER_ERROR. Note that while the endpoint's max_ack_delay transport parameter is in milliseconds (Section 18.2 of [QUIC-TRANSPORT]), min_ack_delay is specified in microseconds.

The min_ack_delay transport parameter is a unilateral indication of support for receiving ACK_FREQUENCY frames. If an endpoint sends the transport parameter, the peer is allowed to send ACK_FREQUENCY and IMMEDIATE_ACK frames independent of whether it also sends the min_ack_delay transport parameter or not.

Until an ACK_FREQUENCY or IMMEDIATE_ACK frame is received, sending the min_ack_delay transport parameter does not cause the endpoint to change its acknowledgment behavior.

Endpoints MUST NOT remember the value of the min_ack_delay transport parameter they received for use in a subsequent connection. Consequently, ACK_FREQUENCY and IMMEDIATE_ACK frames cannot be sent in 0-RTT packets, as per Section 7.4.1 of [QUIC-TRANSPORT].

This Transport Parameter is encoded as per Section 18 of [QUIC-TRANSPORT].

4. ACK_FREQUENCY Frame

Delaying acknowledgments as much as possible reduces work done by the endpoints as well as network load. A data sender's loss detection and congestion control mechanisms however need to be tolerant of this delay at the peer. A data sender signals the conditions under which it wants to receive ACK frames using an ACK_FREQUENCY frame, shown below:

```
ACK_FREQUENCY Frame {
  Type (i) = 0xaf,
  Sequence Number (i),
  Ack-Eliciting Threshold (i),
  Requested Max Ack Delay (i),
  Reordering Threshold (i),
}
```

Following the common frame format described in Section 12.4 of [QUIC-TRANSPORT], ACK_FREQUENCY frames have a type of 0xaf, and contain the following fields:

Sequence Number: A variable-length integer representing the sequence

number assigned to the ACK_FREQUENCY frame by the sender so receivers ignore obsolete frames. A sending endpoint MUST send monotonically increasing values in the Sequence Number field to allow obsolete ACK_FREQUENCY frames to be ignored when packets are processed out of order.

Ack-Eliciting Threshold: A variable-length integer representing the maximum number of ack-eliciting packets the recipient of this frame receives without sending an acknowledgment. A receiving endpoint SHOULD send at least one ACK frame after receiving more than this many ack-eliciting packets. A value of 0 results in a receiver immediately acknowledging every ack-eliciting packet. By default, an endpoint sends an ACK frame for every other ack-eliciting packet, as specified in Section 13.2.2 of [QUIC-TRANSPORT], which corresponds to a value of 1.

Requested Max Ack Delay: A variable-length integer representing the value to which the data sender requests the data receiver update its max_ack_delay (Section 18.2 of [QUIC-TRANSPORT]). The value of this field is in microseconds, unlike the max_ack_delay transport parameter, which is in milliseconds. On receipt of a valid value, the endpoint SHOULD update its max_ack_delay to the value provided by the peer. Note that values of 2^14 milliseconds or greater are invalid for max_ack_delay, as specified in Section 18.2 of [QUIC-TRANSPORT]. A value smaller than the min_ack_delay advertised by the peer is also invalid. Receipt of an invalid value MUST be treated as a connection error of type PROTOCOL_VIOLATION.

Reordering Threshold: A variable-length integer that indicates the maximum packet reordering before eliciting an immediate ACK, as specified in Section 6.2. If no ACK_FREQUENCY frames have been received, the data receiver immediately acknowledges any subsequent packets that are received out-of-order, as specified in Section 13.2 of [QUIC-TRANSPORT], corresponding to a default value of 1. A value of 0 indicates out-of-order packets do not elicit an immediate ACK.

ACK_FREQUENCY frames are ack-eliciting and congestion controlled. When an ACK_FREQUENCY frame is lost, the sender is encouraged to send another ACK_FREQUENCY frame, unless an ACK_FREQUENCY frame with a larger Sequence Number value has already been sent. However, it is not forbidden to retransmit the lost frame (see Section 13.3 of [QUIC-TRANSPORT]), because the receiver will ignore duplicate or out-of-order ACK_FREQUENCY frames based on the Sequence Number.

A receiving endpoint MUST ignore a received ACK_FREQUENCY frame unless the Sequence Number value in the frame is greater than the largest processed value.

5. IMMEDIATE_ACK Frame

A sender can use an ACK_FREQUENCY frame to reduce the number of acknowledgments sent by a receiver, but doing so increases the likelihood that time-sensitive feedback is delayed as well. For example, as described in Section 8.3, delaying acknowledgments can increase the time it takes for a sender to detect packet loss. Sending an IMMEDIATE_ACK frame can help mitigate this problem.

An IMMEDIATE_ACK frame can be useful in other situations as well. For example, if a sender wants an immediate RTT measurement or if a sender wants to establish receiver liveness as quickly as possible. PING frames (Section 19.2 of [QUIC-TRANSPORT]) are ack-eliciting, but if a PING frame is sent without an IMMEDIATE_ACK frame, the receiver might not immediately send an ACK based on its local ACK strategy.

By definition IMMEDIATE_ACK frames are ack-eliciting and they are also congestion controlled. An endpoint SHOULD send a packet containing an ACK frame immediately upon receiving an IMMEDIATE_ACK frame. An endpoint MAY delay sending an ACK frame despite receiving an IMMEDIATE_ACK frame. For example, an endpoint might do this if a large number of received packets contain an IMMEDIATE_ACK or if the endpoint is under heavy load.

```
IMMEDIATE_ACK Frame {
  Type (i) = 0x1f,
}
```

IMMEDIATE_ACK frames do not need to be retransmitted.

6. Sending Acknowledgments

Prior to receiving an ACK_FREQUENCY frame, endpoints send acknowledgments as specified in Section 13.2.1 of [QUIC-TRANSPORT].

After receiving an ACK_FREQUENCY frame and updating its max_ack_delay and Ack-Eliciting Threshold values (Section 4), the data receiver sends an acknowledgment when one of the following conditions are met since the last acknowledgement was sent:

* The number of received ack-eliciting packets is greater than the Ack-Eliciting Threshold.

* max_ack_delay amount of time has passed and at least one ack-eliciting packet has been received.

An acknowledgment can be sent earlier based on the value of the Reordering Threshold when a gap in packet numbers is detected, see Section 6.2.

Section 6.3 and Section 6.4 describe exceptions to this strategy.

All packets still have to be acknowledged at least once with this extension, as stated in Section 13.2.1 of [QUIC-TRANSPORT]. With large values for Ack-Eliciting Threshold or the Reordering Threshold, implementations might accumulate multiple new ACK ranges before sending an ACK. As such, implementations have to take more care to avoid truncating ACK ranges before they are sent at least once. As discussed in Section 13.2.4 of [QUIC-TRANSPORT] this does not guarantee that every acknowledgment is seen by the sender. Therefore, when ACK frames are sent less often, the effect of lost or re-ordered packets with ACK frames needs to be considered more carefully when trimming the ACK range.

6.1. Response to long idle periods

It is important to receive timely feedback after long idle periods, e.g. update stale RTT measurements. When no acknowledgment has been sent in over one smoothed round trip time (Section 5.3 of [QUIC-RECOVERY]), receivers are encouraged to send an acknowledgment soon after receiving an ack-eliciting packet. This is not an issue specific to this document, but the mechanisms specified herein could create excessive delays.

6.2. Response to Out-of-Order Packets

As specified in Section 13.2.1 of [QUIC-TRANSPORT], endpoints are expected to send an immediate acknowledgment upon receipt of a reordered ack-eliciting packet. After an ACK_FREQUENCY frame with a Reordering Threshold value other than 1 has been received, this extension delays immediate acknowledgements to reordered ack-eliciting packets and the gaps they can create.

If the most recent ACK_FREQUENCY frame received from the peer has a Reordering Threshold value of 0, the endpoint SHOULD NOT send an immediate acknowledgment in response to packets received out of order, and instead rely on the peer’s Ack-Eliciting Threshold and Requested Max Ack Delay for sending acknowledgments.

If the most recent ACK_FREQUENCY frame received from the peer has a Reordering Threshold value larger than 1, the endpoint tests the amount of reordering before deciding to send an acknowledgment. The specification uses the following definitions:

Largest Unacked: The largest packet number among all received ack-eliciting packets.

Largest Acked: The Largest Acknowledged value sent in an ACK frame.

Largest Reported Missing: The largest packet number that could be declared lost with the specified Reordering Threshold, which is Largest Acked - Reordering Threshold.

Unreported Missing: Packets with packet numbers between the Largest Unacked and Largest Reported Missing that have not yet been received.

An endpoint that receives an ACK_FREQUENCY frame with a non-zero Reordering Threshold value SHOULD send an immediate ACK whenever it receives an ack-eliciting, out-of order packet whose packet number is outside the reordering window of the peer, i.e. when * the difference between the smallest Unreported Missing packet and the Largest Unacked packet is greater than or equal to the Reordering Threshold value; or * the received packet number is less than or equal to Largest Acked - Reordering Threshold.

The first condition triggers an ACK as soon as the reordering threshold is reached and a packet can be declared lost at the sender. The second condition addresses packets that have been received later, out of order and are already spuriously declared lost by the sender. This enables the sender to detect spurious retransmissions and revert the congestion control status accordingly.

Sending such an additional ACK resets the max_ack_delay timer and the Ack-Eliciting Threshold counter, as any ACK would.

See Section 6.2.1 for examples explaining this behavior. See Section 9 for guidance on how to choose the reordering threshold value when sending ACK_FREQUENCY frames.

6.2.1. Examples

Note that the following examples assume that packets are processed one-by-one in the received order as indicated below.

When the reordering threshold is 1, any time a packet is received and there is a missing packet, an immediate acknowledgement is sent.

If the reordering theshold is 3 and acknowledgements are only sent due to reordering, the sequence in Table 1 would occur:

Received Packet	Largest Unacked	Largest Acked	Largest Reported Missing	Unreported Missing	Send Acknowledgement
0	0	-	-	-	No
1	1	-	-	-	No
3	3	-	-	2	No
4	4	-	-	2	No
5	5	-	-	2	Yes (5 - 2 >= 3)
8	8	5	2	6,7	No
9	9	5	2	6,7	Yes (9 - 6 >= 3)
10	10	9	6	7	Yes (10 - 7 >= 3)

Table 1: Acknowledgement behavior with a reordering threshold of 3

Note that in this example, the receipt of packet 9 triggers an ACK that reports both packets 6 and 7 as missing. However, the receipt of packet 10 needs to trigger another immediate ACK because the sender will be unable to declare packet 7 as lost (with a reordering threshold of 3) until it receives an ACK reporting the reception of packet 10.

If the reordering threshold is 5 and acknowledgements are only sent due to reordering, the sequence in Table 2 would occur:

Received Packet	Largest Unacked	Largest Acked	Largest Reported Missing	Unreported Missing	Send Acknowledgement
0	0	-	-	-	No
1	1	-	-	-	No
3	3	-	-	2	No
5	5	-	-	2,4	No
6	6	-	-	2,4	No
7	7	-	-	2,4	Yes (7 - 2 >= 5)
8	8	7	2	4	No
9	9	7	2	4	Yes (9 - 4 >= 5)

Table 2: Acknowledgement behavior with a reordering threshold of 5

6.3. Expediting Explicit Congestion Notification (ECN) Signals

If the Ack-Eliciting Threshold is larger than 1, an endpoint SHOULD send an immediate acknowledgement when a packet marked with the ECN Congestion Experienced (CE) [RFC3168] codepoint in the IP header is received and the previously received packet was not marked CE. From there on, if multiple consecutive CE-marked packets are received or only non-CE-marked packet received, the endpoint resumes sending acknowledgements based on the Ack-Eliciting Threshold or max_ack_delay. Therefore, CE-marking only triggers an immediate acknowledgement when there is a transition from non-CE-marked to CE-marked.

If the Ack-Eliciting Threshold is 0, every ack-eliciting packet is immediately acknowledged, whether it is CE marked or not. If the Ack-Eliciting Threshold is 1, the default behavior as specified in RFC9000 applies, which recommends to immediately acknowledge all packets marked with CE (see Section 13.2.1 of [QUIC-TRANSPORT]).

Acknowledging the first CE marked packet immediately maintains the peer’s response time to congestion events, while also reducing the ACK rate compared to Section 13.2.1 of [QUIC-TRANSPORT] during

extreme congestion or when peers are using DCTCP [RFC8257] or other congestion controllers (e.g. [I-D.ietf-tsvwg-aqm-dualq-coupled]) that mark more frequently than classic ECN [RFC3168].

6.4. Batch Processing of Packets

To avoid sending multiple acknowledgments in rapid succession, an endpoint can process all packets in a batch before determining whether to send an ACK frame in response, as stated in Section 13.2.2 of [QUIC-TRANSPORT].

7. Computation of Probe Timeout Period

After requesting an update to the data receivers's max_ack_delay, a data sender can use this new value in later computations of its Probe Timeout (PTO) period; see Section 5.2.1 of [QUIC-RECOVERY].

Until the packet carrying the ACK_FREQUENCY frame is acknowledged, the endpoint MUST use the greater of the current max_ack_delay and the value that is in flight when computing the PTO period. Doing so avoids spurious PTOs that can be caused by an update that decreases the peer's max_ack_delay.

While it is expected that endpoints will have only one ACK_FREQUENCY frame in flight at any given time, this extension does not prohibit having more than one in flight. When using max_ack_delay for PTO computations, endpoints MUST use the maximum of the current value and all those in flight.

When the number of in-flight ack-eliciting packets is larger than the ACK-Eliciting Threshold, an endpoint can expect that the peer will not need to wait for its max_ack_delay period before sending an acknowledgment. In such cases, the endpoint MAY exclude the peer's max_ack_delay from its PTO calculation. When Reordering Threshold is set to 0 and loss prevents the peer from receiving enough packets to trigger an immediate acknowledgment, the receiver will wait max_ack_delay, increasing the chances of a premature PTO. Therefore, if Reordering Threshold is set to 0, the PTO MUST be larger than the peer's max_ack_delay.

When sending PTO packets, one can include an IMMEDIATE_ACK frame to elicit an immediate acknowledgment. This avoids delaying acknowledgements of PTO packets by the acknowledgment delay, reducing tail latency and allowing the sender to exclude the peer's max_ack_delay from subsequent PTO calculations.

8. Determining Acknowledgment Frequency

This section provides some guidance on a sender's choice of acknowledgment frequency and discusses some additional considerations. Implementers can select an appropriate strategy to meet the needs of their applications and congestion controllers.

8.1. Congestion Control

A sender needs to be responsive to notifications of congestion, such as a packet loss or an ECN CE marking. Decreasing the acknowledgment frequency can delay a sender's response to network congestion or cause it to underutilize the available bandwidth.

To limit the consequences of reduced acknowledgment frequency, a sender can use the extension in this draft to request a receiver to send an acknowledgment at least once per round trip, when there are ack-eliciting packets in flight, in the following ways:

A data sender can set the Requested Max Ack Delay value to no more than the estimated round trip time. The sender can also improve feedback and robustness to variation in the path RTT by setting the Ack-Eliciting Threshold to a value no larger than number of maximum-sized packets that fit into the current congestion window. Alternatively, a sender can send an IMMEDIATE_ACK frame if no acknowledgement has been received for more than one round trip time. If the packet containing an IMMEDIATE_ACK is lost, detection of that loss will be delayed by the Reordering Threshold or Requested Max Ack Delay.

When setting the Requested Max Ack Delay as a function of the RTT, it is usually better to use the Smoothed RTT (smoothed_rtt) (Section 5.3 of [QUIC-RECOVERY]) or another estimate of the typical RTT, but not the minimum RTT (min_rtt) (Section 5.2 of [QUIC-RECOVERY]). This avoids eliciting an unnecessarily high number of acknowledgments when min_rtt is much smaller than smoothed_rtt.

Note that the congestion window and the RTT estimate change over the lifetime of a connection and therefore might require sending updates in an ACK_FREQUENCY frames to ensure optimal performance, though not every change should trigger an update. Usually, it is not necessary to send an ACK_FREQUENCY frame more than once per RTT and likely even less frequently. Ideally, an ACK_FREQUENCY frame is sent only when a relevant change in the congestion window or smoothed RTT is detected that impacts the local setting of the reordering threshold or locally-selected calculation of the either Ack-Eliciting Threshold or the Requested Max Ack Delay.

It is possible that the RTT is smaller than the receiver's timer granularity, as communicated via the min_ack_delay transport parameter, preventing the receiver from sending an acknowledgment every RTT in time unless packets are acknowledged immediately. In these cases, Reordering Threshold values other than 1 can delay loss detection more than an RTT.

8.1.1. Application-Limited Connections

A congestion controller that is limited by the congestion window relies upon receiving acknowledgments to send additional data into the network. An increase in acknowledgment delay increases the delay in sending data, which can reduce the achieved throughput. Congestion window growth can also depend upon receiving acknowledgments. This can be particularly significant in slow start (Section 7.3.1 of [QUIC-RECOVERY]), when delaying acknowledgments can delay the increase in congestion window and can create larger packet bursts.

If the sender is application-limited, acknowledgments can be delayed unnecessarily when entering idle periods. Therefore, if no further data is buffered to be sent, a sender can send an IMMEDIATE_ACK frame with the last data packet before an idle period to avoid waiting for the acknowledgment delay.

If there are no inflight packets, no acknowledgments will be received for at least a round trip when sending resumes. The max_ack_delay and Ack-Eliciting Threshold values used by the receiver can further delay acknowledgments. In this case, the sender can include an IMMEDIATE_ACK or ACK_FREQUENCY frame in the first Ack-Eliciting packet to avoid waiting for substantially more than a round trip for an acknowledgment.

8.2. Burst Mitigation

Receiving an acknowledgment can allow a sender to release new packets into the network. If a sender is designed to rely on the timing of peer acknowledgments ("ACK clock"), delaying acknowledgments can cause undesirable bursts of data into the network. In keeping with Section 7.7 of [QUIC-RECOVERY], a sender can either employ pacing or limit bursts to the initial congestion window.

8.3. Loss Detection and Timers

Acknowledgments are fundamental to reliability in QUIC. Consequently, delaying or reducing the frequency of acknowledgments can cause loss detection at the sender to be delayed.

A QUIC sender detects loss using packet thresholds on receiving an acknowledgment (Section 6.1.1 of [QUIC-RECOVERY]); delaying the acknowledgment therefore delays this method of detecting losses.

Reducing acknowledgment frequency reduces the number of RTT samples that a sender receives (Section 5 of [QUIC-RECOVERY]), making a sender's RTT estimate less responsive to changes in the path's RTT. As a result, any mechanisms that rely on an accurate RTT estimate, such as time-threshold-based loss detection (Section 6.1.2 of [QUIC-RECOVERY]) or the Probe Timeout (PTO) (Section 6.2 of [QUIC-RECOVERY]), will be less responsive to changes in the path's RTT, resulting in either delayed or unnecessary packet transmissions.

A sender might use timers to detect loss of PMTU probe packets (Section 14 of [QUIC-TRANSPORT]). A sender MAY bundle an IMMEDIATE_ACK frame with any PMTU probes to avoid triggering such timers.

8.4. Connection Migration

To avoid additional delays to connection migration confirmation when using this extension, a client can bundle an IMMEDIATE_ACK frame with the first non-probing frame (Section 9.2 of [QUIC-TRANSPORT]) it sends or it can send only an IMMEDIATE_ACK frame, which is a non-probing frame.

An endpoint's congestion controller and RTT estimator are reset upon confirmation of migration (Section 9.4 of [QUIC-TRANSPORT]); this changes the pattern of acknowledgments received after migration.

Therefore, an endpoint that has sent an ACK_FREQUENCY frame earlier in the connection ought to send a new ACK_FREQUENCY frame upon confirmation of connection migration with updated information, e.g. to consider the new RTT estimate.

9. Setting the Reordering Threshold Value

To ensure timely loss detection, a data sender can send a Reordering Threshold value that is the same as the loss detection packet threshold. If the reordering threshold is smaller than the packet threshold, an acknowledgement is unnecessarily sent before the packet can be declared lost. If the value is larger, it can cause unnecessary delays in loss detection (Section 6.1.1 of [QUIC-RECOVERY]).

In order to avoid unnecessary immediate acknowledgements, senders SHOULD implement adaptive packet threshold loss detection and communicate the increased Reordering Threshold value to the receiver.

10. Security Considerations

An improperly configured or malicious data sender could request a data receiver to acknowledge more frequently than its available resources permit. However, there are two limits that make such an attack largely inconsequential. First, the acknowledgment rate is bounded by the rate at which data is received. Second, ACK_FREQUENCY and IMMEDIATE_ACK frames can only request an increase in the acknowledgment rate, but cannot enforce it.

Section 21.9 of [QUIC-TRANSPORT] provides further guidance on peer denial of service attacks that could abuse control frames, including ACK frames as well as the newly herein specified ACK_FREQUENCY and IMMEDIATE_ACK frames, to cause disproportional processing costs without observable impact on the state of the connection. Especially, the IMMEDIATE_ACK frame does not only imply processing cost for receiving and processing the control frame itself but can also cause additional sending of packets. However, in general, with this extension, a sender cannot force a receiver to acknowledge more frequently than the receiver considers safe based on its resource constraints.

11. IANA Considerations

This document defines a new transport parameter to advertise support of the extension described in this document and two new frame types to registered by IANQ in the respective "QUIC Protocol" registries under https://www.iana.org/assignments/quic/quic.xhtml (https://www.iana.org/assignments/quic/quic.xhtml).

11.1. QUIC Transport Parameter

The following entry in Table 3 has been requested to be provisionally added to the "QUIC Transport Parameters" registry under the "QUIC Protocol" heading.

Value	Parameter Name.	Specification
0xff04de1b	min_ack_delay	Section 3

Table 3: Addition to QUIC Transport Parameters Entries

When this document is approved, IANA is requested to assign a permanent allocation of a codepoint in the 0-63 range to replace the provisional codepoint described above.

11.2. QUIC Frame Types

The following frame types have requested to be provisionally added to the "QUIC Frame Types" registry under the "QUIC Protocol" heading.

Value	Frame Name	Specification
0xaf	ACK_FREQUENCY	Section 4
0x1f	IMMEDIATE_ACK	Section 5

Table 4: Addition to QUIC Frame Types Entries

When this document is approved, IANA is requested to change the registration to a permanent allocation of these frame types with the values described above.

12. References

12.1. Normative References

[QUIC-TRANSPORT]
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/rfc/rfc9000>.

[QUIC-RECOVERY]
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/rfc/rfc9002>.

[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/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

12.2. Informative References

[Cus22] Custura, A., Jones, T., Secchi, R., and G. Fairhurst, "Reducing the acknowledgement frequency in IETF QUIC", DOI 10.1002/sat.1466, name IJSCN, October 2022, <https://doi.org/10.1002/sat.1466>.

[RFC3449] Balakrishnan, H., Padmanabhan, V., Fairhurst, G., and M. Sooriyabandara, "TCP Performance Implications of Network Path Asymmetry", BCP 69, RFC 3449, DOI 10.17487/RFC3449, December 2002, <https://www.rfc-editor.org/rfc/rfc3449>.

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

[RFC8257] Bensley, S., Thaler, D., Balasubramanian, P., Eggert, L., and G. Judd, "Data Center TCP (DCTCP): TCP Congestion Control for Data Centers", RFC 8257, DOI 10.17487/RFC8257, October 2017, <https://www.rfc-editor.org/rfc/rfc8257>.

[I-D.ietf-tsvwg-aqm-dualq-coupled] De Schepper, K., Briscoe, B., and G. White, "Dual-Queue Coupled Active Queue Management (AQM) for Low Latency, Low Loss, and Scalable Throughput (L4S)", Work in Progress, Internet-Draft, draft-ietf-tsvwg-aqm-dualq-coupled-25, 29 August 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-tsvwg-aqm-dualq-coupled-25>.

Acknowledgments

The following people directly contributed key ideas that shaped this draft: Bob Briscoe, Kazuho Oku, Marten Seemann.

Thanks for the reviews by Lucas Pardue, Martin Thomson, Magnus Westerlund, Kazuho Oku, Marten Seemann, Gorry Fairhurst, Ingemar Johansson, Christina Huitema, Michael Eriksson, Martin Duke, Nick Banks, Gaurav Singh, Mike Bishop, Neal Cardwell, Rui Paulo, Joseph Beshay, Alexis La Goutte, Vidhi Goel, Dmitri Tikhonov, Marco Munizaga and Matt Joras.

Authors’ Addresses

Jana Iyengar
Fastly
Email: jri.ietf@gmail.com

Ian Swett
Google
Email: ianswett@google.com

Mirja K端hlewind
Ericsson
Email: mirja.kuehlewind@ericsson.com