Internet Area Working Group
Internet-Draft
Updates: 3819 (if approved)
Intended status: Informational
Expires: 25 October 2026

G. White
CableLabs
I. Johansson
Ericsson
D. Das
Intel
C. Box
BT Group

23 April 2026

Proposal for Updates to Guidance on Packet Reordering
draft-white-intarea-reordering-03

Abstract

Several link technology standards mandate that equipment guarantee in-order delivery of layer 2 frames, apparently due to a belief that this is required by higher layer protocols. To meet this requirement they implement a "resequencing" operation to restore the original packet order. This can introduce delays that result in net degradation of performance. Modern TCP and QUIC implementations support features that significantly improve their tolerance to out-of-order delivery. This draft is intended to provide new information for layer 2 technology standards regarding the need to assure in-order delivery to support IETF 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 25 October 2026.

Copyright Notice

Table of Contents

1. Introduction

Several link technology standards mandate that equipment guarantee in-order delivery of layer 2 frames, apparently due to a belief that this is required by higher layer protocols. In addition, certain link types can introduce out-of-order arrivals at the end of the layer 2 link, which the receiving equipment is required to rectify by delaying higher sequenced frames until all lower sequenced frames can be delivered or are deemed lost. The delaying of higher sequenced frames is generally done without any knowledge of the higher layer protocols in use, let alone any knowledge of higher layer protocol contexts (e.g. TCP connections) in the case that the layer 2 link is carrying a multiplex of such contexts. It could, for example, be the case that all of the higher sequenced frames being delayed are carrying packets for different layer 4 contexts than a single lower-sequenced frame that triggered the delay. The result is that this "resequencing" operation can introduce delays that result in degradation of performance rather than improving it. Moreover, modern, performant TCP and QUIC implementations support features that

significantly improve their tolerance to out-of-order delivery.

Section 15 of [RFC3819] provides advice to Internet subnetwork designers regarding packet reordering. The advice seems reasonable: "try to avoid packet reordering whenever possible, but not if doing so compromises efficiency, impairs reliability, or increases average packet delay." However, along with this guidance come a couple of warnings that appear to have driven subnetwork designers towards assuring in-order delivery even if efficiency is impaired or average packet delay is increased. These warnings relate to the potential performance cost for TCP (as it was then defined) that arises due to out-of-order arrivals, and to the fact that out-of-order delivery would cause problems for "every header compression scheme currently standardized for the Internet."

The text of [RFC3819] does mention that research on improving TCP behavior in the face of packet reordering was underway. In the intervening 20+ years, that research has resulted in the definition of RACK [RFC8985] which significantly reduces TCP’s sensitivity to out-of-order arrivals. In addition, Section 6.1 of [RFC9002] discusses the usefulness of implementing similar mechanisms in QUIC, and provides the protocol tools to do so. The result is that having the link-layer delay packets to provide them in-order to the transport layer was helpful multiple decades ago when TCP implementations had more naive algorithms and were very resource-constrained. Given how modern implementations work, this "resequencing" at the link-layer is almost always harmful. Modern TCP and QUIC stacks can handle reordering well, thanks to both protocol features and implementations having more memory to work with. For these implementations, the delay induced by L2 resequencing will generally cause more harm than good. Furthermore, one of the biggest motivators for QUIC was to break head-of-line blocking and allow out-of-order delivery to the application layer. At this point we have large bodies of data showing that this improves real-world performance. Moreover, older implementations of TCP that don’t implement RACK don’t fail in the presence of some out-of-order packets. They may see reduced performance, but they do have the ability to correctly handle the receipt of packets out of order.

At the time of publication of [RFC3819], the "ROHC" header compression framework defined in [RFC3095] stated an operating assumption that the network would provide in-order delivery. This assumption is discussed in detail in [RFC4224], which provides guidance on implementing ROHC over channels that can reorder header-compressed packets, and explains different ways of implementing the profiles found in [RFC3095] over reordering channels. Subsequent to the publication of [RFC3819] the issue of reordering intolerance by ROHC was addressed by the development of ROHCv2 [RFC5225], which lists tolerance to reordering as one of its significant improvements over ROHC.

There may be other protocols or protocol implementations that are sensitive to reordering to some degree. For instance, Section 4.1 of [RFC2983] discusses IPSec [RFC2401] and L2TP [RFC2661] as being sensitive to packet reordering.

[RFC4303] (and the obsoleted [RFC2406]) specifies out of order handling for IPsec encryption (ESP). A sliding window scheme is used with a minimum size of 32 and a default of 64 packets. A conforming implementation will handle a fair amount of out of sequence delivery, how much is up to implementation choice and can possibly be set as a parameter. The impression is that IPsec, based on email on ipsec IETF reflector by Paul Koning 2026-03-26 (https://mailarchive.ietf.org/arch/msg/ipsec/nPntT13UfrbEF5Hl36T9xAFoYP4/), is not sensitive to reordering.

The current situation seems to be that several layer 2 technologies implement resequencing functions that increase cost and complexity of equipment, and may in many cases degrade network performance rather than improving it. Any benefits of resequencing may be primarily limited to older protocol implementations e.g. maintaining the performance of older TCP implementations that are no longer widely used and may not be particularly performant by modern standards.

2. Terminology

In this document, we adopt the following terminology.

Reordering:
: The process where the packet/frame sequence is made out-of-order from the originally transmitted order.

Resequencing:
: The process where an out-of-order packet/frame sequence is returned to the originally transmitted order.

3. Reordering in L2 Links

In current L2 link technologies, frame reordering comes from two primary sources: multi-link aggregation and layer-2 retransmission.

3.1. Multi-Link Aggregation

Some link technologies support the aggregation (within L2) of multiple links between a pair of nodes for the purpose of increasing the total capacity of the link. In many cases the individual links within the aggregation have different capacities and/or latencies from one another. When this is the case, the frame reception order differs from the frame transmission order.

3.2. Layer 2 Retransmissions

Some link technologies (particularly wireless), are subject to noise and interference that would result in an unacceptably high frame error rate if it were not corrected. These links commonly implement a method of acknowledgement and retransmission of lost frames, referred to as Automatic Repeat Request (ARQ). When a frame loss affects one or more frames within a transmission, and the frame(s) at the end are unaffected, the retransmitted frames will arrive out of sequence from the original ordering.

4. Resequencing in L2 Standards

This section discusses functions and requirements for resequencing in exising layer 2 standards.

4.1. LTE/5G

The layers involved in data transmission, from top to bottom, are PDCP, RLC, MAC and PHY. Packet reordering can occur on both the MAC and the RLC layer in LTE and 5G systems. Packet resequencing occurs on the PDCP layer.

On the MAC layer: An RLC packet data unit (RLC-PDU) is transmitted in one of up to 16 Hybrid ARQ processes on the MAC layer. This avoids the stop and wait for one transmission to succed before a new transmission can begin. Transmission on the MAC layer can however have a high block error rate (BLER), 10% or more is common especially when the traffic pattern and radio channel varies. Each retransmission adds a few milliseconds extra delay.

This results in packets delivered out of order to the RLC layer on the receiver side because some transmissions on the MAC layer succeed in one attempt while others may need several attempts, up to a configured limit.

HARQ failure is triggered when the max number of retransmissions is reached.

On the RLC Layer: The RLC layer is configured in two typical ways.

* RLC Unacknowledged mode (RLC-UM): Is typically used for voice services a.k.a VoLTE (Voice over LTE) or VoNR (Voice over NR). Data blocks are not retransmitted in this mode.

* RLC Acknowledged mode (RLC-AM): Is typically used for normal internet traffic, a.k.a Mobile Broadband (MBB). Retransmissions on this layer are less common than retransmissions on the MAC layer and they occur when HARQ failure is triggered on the MAC layer, or in the case that the MAC layer erroneously decodes a NACK as an ACK. Retransmission on the RLC layer causes 10s of milliseconds on extra delay, depending on how timers are configured.

Besides the causes for packet reordering listed above, packet reordering can occur with Dual Connectivity where 2 or more radio frequency bands are combined on the RLC layer, this because there can be a time skew of a few milliseconds over transport links between the PDCP layer and the RLC layer(s).

The RLC layer does resequencing on the RLC layer in LTE systems. On 5G systems however, resequencing is left to the PDCP layer. Resequencing on the PDCP layer is however optional according to the 5G standard but is mostly enabled out of concern for sensitivity to packet reordering on the transport and application layer. In addition radio bearers with robust header compression (RoHC) should enable resequencing.

4.2. Wi-Fi

The IEEE 802.11 specification (Wi-Fi) currently supports only in-order delivery. The 802.11 protocols inherit traffic delivery requirements from the 802.1Q specification (see section 6.5.3 in 802.1Q), which only permits only in-order delivery.

To utilize the medium efficiently, the 802.11 MAC aggregates data frames of a given traffic identifier (TID) and transmits them as a block. However, due to link errors, not all frames in the block may be received. A re-order buffer at the receiver holds up delivery of

data frames if a frame ahead of it is missing. A separate reorder buffer is maintained for each of the 8 TIDs. A later retransmission may deliver the missing frame at which point the missing frame and subsequent frames are delivered.

The process of in-order delivery is also taken advantage of as part of the security protocol. Each data frame is sequentially numbered with a packet number. To mitigate against a replay attack, the receiving MAC simply checks if the packet number of a received data frames is higher than the highest received one for that TID.

Allowing out-of-order delivery within the 802.11 specification for a given TID would require some adjustments to the 802.11 protocol. Some of the adjustments include the following:

* The reorder buffer would need to release packets without waiting for older packets to be received.

* The replay detection mechanism would need to be updated to not drop an older missing frame when it is successfully retransmitted.

4.3. DOCSIS

The Data-Over-Cable Service Interface Specifications provide broadband access over hybrid fiber-coaxial networks. The DOCSIS protocol does not support layer 2 retransmissions, but since the introduction of "channel bonding" functionality (a form of multi-link aggregation) in DOCSIS 3.0 in 2006, reordering can occur due to differences in capacity or latency between the bonded channels. All equipment built to that and later versions of DOCSIS (including DOCSIS 3.1 and DOCSIS 4.0) is required to support specific resequencing features to guarantee in-order delivery.

In the downstream direction (i.e. from the network to end-user) individual IP packets are encapsulated in layer 2 frames that generally include a 20-bit Downstream Service ID (DSID), a 1-bit Sequence Change Count, and a 16-bit Packet Sequence Number. The specifications include detailed, mandatory, requirements on the handling of these fields, including a requirement that cable modems support at least 16 simultaneous resequencing contexts, each of which having sufficient memory to hold packets for up to 13 ms (18 ms in older equipment) at the maximum forwarding rate of the device, mechanisms for rapid loss detection, and significant management and configuration mechanisms to support the feature.

In the upstream direction (i.e. from the end-user to the network) individual IP packets are encapsulated in layer 2 frames, then concatenated together and fragmented into "segments" for

transmission. The segment boundaries are determined by the size of the transmission opportunities (grants) and generally do not relate to the internal frame or packet boundaries. So, in general a segment could begin with the tail of one L2 frame, then have zero or more full L2 frames, then the head of a final frame. Each segment has a segment header which contains a 13-bit sequence number. The CMTS is required to reassemble the concatenated frame sequence, and forward the individual frames in order.

5. Proposed Updates to RFC3819 Guidance on Packet Reordering

This draft proposes to replace the contents of Section 15 of [RFC3819] with the following text.

The Internet architecture does not guarantee that packets will arrive in the same order in which they were originally transmitted; transport protocols must take this into account.

An earlier definition of TCP [RFC2581][RFC3517][RFC5681][RFC6675] defined the "fast retransmit" algorithm, which recommended that a TCP receiver send an immediate duplicate ACK when an out-of-order segment arrives, and it recommended that the TCP sender use the arrival of 3 duplicate ACKs as an indication that a segment has been lost. This loss determination triggers a retransmission and a reduction of the congestion window. In the presence of packet reordering, this duplicate-ACK-based loss detection mechanism can incorrectly deem a packet lost, and thus trigger a spurious retransmission and an unnecessary congestion window reduction, both of which can negatively impact the performance of the connection. Thus, one outcome of the implementation and deployment of the fast retransmit algorithm is that TCPs became significantly more sensitive to network reordering.

A more recent update to TCP [RFC8985] addresses this sensitivity by performing loss detection via time-based inferences derived from the ACK feedback, rather than using duplicate ACKs. In addition, Section 6.1 of [RFC9002] discusses the usefulness of implementing similar mechanisms in QUIC, and provides the protocol tools to do so. Implementations of these time-based loss detection methods have now been widely deployed.

Nonetheless, there do exist implementations of network protocols that do not work well in the presence of a significant amount of packet reordering.

This suggests that subnetwork implementers should try to avoid packet reordering when possible, but not if doing so compromises efficiency, impairs reliability, or increases average packet delay.

6. IANA Considerations

This memo includes no request to IANA.

7. Security Considerations

This document should not affect the security of the Internet.

8. References

8.1. Informative References

[RFC2401] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, DOI 10.17487/RFC2401, November 1998, <https://www.rfc-editor.org/info/rfc2401>.

[RFC2406] Kent, S. and R. Atkinson, "IP Encapsulating Security Payload (ESP)", RFC 2406, DOI 10.17487/RFC2406, November 1998, <https://www.rfc-editor.org/info/rfc2406>.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <https://www.rfc-editor.org/info/rfc4303>.

[RFC2661] Townsley, W., Valencia, A., Rubens, A., Pall, G., Zorn, G., and B. Palter, "Layer Two Tunneling Protocol "L2TP"", RFC 2661, DOI 10.17487/RFC2661, August 1999, <https://www.rfc-editor.org/info/rfc2661>.

[RFC2983] Black, D., "Differentiated Services and Tunnels", RFC 2983, DOI 10.17487/RFC2983, October 2000, <https://www.rfc-editor.org/info/rfc2983>.

[RFC3095] Bormann, C., Burmeister, C., Degermark, M., Fukushima, H., Hannu, H., Jonsson, L., Hakenberg, R., Koren, T., Le, K., Liu, Z., Martensson, A., Miyazaki, A., Svanbro, K., Wiebke, T., Yoshimura, T., and H. Zheng, "RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed", RFC 3095, DOI 10.17487/RFC3095, July 2001, <https://www.rfc-editor.org/info/rfc3095>.

[RFC3819] Karn, P., Ed., Bormann, C., Fairhurst, G., Grossman, D., Ludwig, R., Mahdavi, J., Montenegro, G., Touch, J., and L. Wood, "Advice for Internet Subnetwork Designers", BCP 89, RFC 3819, DOI 10.17487/RFC3819, July 2004, <https://www.rfc-editor.org/info/rfc3819>.

[RFC4224] Pelletier, G., Jonsson, L., and K. Sandlund, "RObust Header Compression (ROHC): ROHC over Channels That Can Reorder Packets", RFC 4224, DOI 10.17487/RFC4224, January 2006, <https://www.rfc-editor.org/info/rfc4224>.

[RFC5225] Pelletier, G. and K. Sandlund, "RObust Header Compression Version 2 (ROHCv2): Profiles for RTP, UDP, IP, ESP and UDP-Lite", RFC 5225, DOI 10.17487/RFC5225, April 2008, <https://www.rfc-editor.org/info/rfc5225>.

[RFC8985] Cheng, Y., Cardwell, N., Dukkipati, N., and P. Jha, "The RACK-TLP Loss Detection Algorithm for TCP", RFC 8985, DOI 10.17487/RFC8985, February 2021, <https://www.rfc-editor.org/info/rfc8985>.

[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>.

[RFC2581] Allman, M., Paxson, V., and W. Stevens, "TCP Congestion Control", RFC 2581, DOI 10.17487/RFC2581, April 1999, <https://www.rfc-editor.org/info/rfc2581>.

[RFC3517] Blanton, E., Allman, M., Fall, K., and L. Wang, "A Conservative Selective Acknowledgment (SACK)-based Loss Recovery Algorithm for TCP", RFC 3517, DOI 10.17487/RFC3517, April 2003, <https://www.rfc-editor.org/info/rfc3517>.

[RFC5681] Allman, M., Paxson, V., and E. Blanton, "TCP Congestion Control", RFC 5681, DOI 10.17487/RFC5681, September 2009, <https://www.rfc-editor.org/info/rfc5681>.

[RFC6675] Blanton, E., Allman, M., Wang, L., Jarvinen, I., Kojo, M., and Y. Nishida, "A Conservative Loss Recovery Algorithm Based on Selective Acknowledgment (SACK) for TCP", RFC 6675, DOI 10.17487/RFC6675, August 2012, <https://www.rfc-editor.org/info/rfc6675>.

Acknowledgements

Contributors

Thanks to all of the contributors.

Authors' Addresses

Greg White
CableLabs
United States of America
Email: g.white@cablelabs.com

Ingemar Johansson
Ericsson
Sweden
Email: ingemar.s.johansson@ericsson.com

Dibakar Das
Intel
United States of America
Email: dibakar.das@intel.com

Chris Box
BT Group
United Kingdom
Email: chris.box@bt.com