Proposal for Updates to Guidance on Packet Reordering

draft-white-intarea-reordering

INTAREA – IETF 122

March 17, 2025

Greg White, Ingemar Johansson, Dibakar Das

Background

- L2 technologies can have link characteristics that result in packet *reordering*, e.g.:
  - Link Aggregation
    - When aggregated links have different latency or capacity
  - L2 retransmissions
    - Common in wireless links
- Several L2 standards introduce requirements for *resequencing* to restore the frame sequence by delaying early-arriving frames
  - Is this what IP protocols want?

Existing IETF guidance

- RFC3819 §15
  - The Internet architecture does not guarantee that packets will arrive in the same order in which they were originally transmitted; transport protocols like TCP must take this into account.
  - [L2 links] should try to avoid packet reordering whenever possible, but not if doing so compromises efficiency, impairs reliability, or increases average packet delay.

  - But…
    - … reordering does come at a cost with TCP as it is currently defined … the TCP sender needlessly reduces the congestion window and performance suffers.
    - Note that every header compression scheme currently standardized for the Internet requires in-order packet delivery on the link between compressor and decompressor.

Goals

- Provide up-to-date information for L2 technology SDOs on the impact that packet reordering & resequencing have on IP protocols.
- Enable SDOs to make informed decisions for future standards.

Potential benefits of resequencing

- TCP/QUIC
  - Older non-RACK TCP implementations can interpret out-of-order packets as loss, triggering retransmissions and CWND reductions that could affect throughput
  - There may be some QUIC implementations that don’t (yet?) support RACK
- Header Compression (ROHC)
  - ROHC (RFC3095 ca. 2001) profiles can be sensitive to reordering
    - RFC4224 discusses the reordering sensitivity of ROHC schemes.
  - ROHCv2 profiles (RFC5225 ca. 2008) provide improved robustness to out-of-order packets
- IPsec replay protection
  - reordering within a stream causes later packets of the same stream to the dropped as replay attacks
  - interleaving of different reordered streams has no new impact over interleaving of order-preserving streams
- IP fragmentation reassembly
  - reordering within a stream set may or may not have any impact over interleaving of order preserving streams
  - interleaving of different reordered streams consumes more resources, as each pending stream requires a max-receive buffer

Potential downsides to resequencing

- Cost and complexity
  - Requires certain protocol features (sequence numbering, etc.)
  - Requires resequencing buffers & logic

- End-to-end performance degradation
  - Introduces delay on packets that arrive early
  - Delayed packets might not be a protocol that benefits from resequencing
  - Delayed packets might not be associated with the same L3/L4 context as the late arriving packet

Current state?

- Modern, performant TCP and QUIC implementations (i.e. those that implement RACK) are harmed by resequencing
  - Those that don’t support RACK aren’t particularly performant and/or performance sensitive.
- Is ROHCv1 still used? Is it used in an end-to-end manner, i.e. not tied to a specific link technology?
- How robust are IPsec implementations?
- How robust are IP fragmentation reassembly implementations?

L2 Case Studies

- LTE/5G
  - Reordering due to L2 retransmissions (HARQ) & link aggregation (Dual Connectivity), can reach 10s of ms.
  - Resequencing happens at both MAC and RLC/PDCP layers
  - PDCP layer resequencing (5G) can be disabled (but rarely is)
- Wi-Fi
  - Reordering due to L2 retransmissions
  - AP & STA are required to ensure in-order delivery within a TID (priority level)
  - Receiver also uses sequence numbers for replay attack protection
- DOCSIS
  - Reordering due to link aggregation (channel bonding), can reach 13 ms
  - Downstream: CM is required to support 16 resequencing contexts, resequencing can be disabled (but rarely is)
  - Upstream: L2 frames are concatenated and then fragmented into segments. Segmentation reassembly requires r
  - Lost frames/segments can cause resequencing delays