Network Working Group A. Pelov
Internet-Draft IMT Atlantique
Intended status: Informational C. Gomez
Expires: 2 October 2025Universitat Politecnica de Catalunya/Fundacio i2CAT
31 March 2025

SCHC Reliability Fragmentation
draft-pelov-schc-rel-fragmentation-rule-format-03

Abstract

This document specifies a fragmentation mode used for reliability within the SCHC framework. Building on the fragmentation mechanisms defined in RFC8724, this rule format is tailored to ensure the reliable delivery of small messages that do not trigger conventional fragmentation. A key enhancement is the inclusion of a size field, indicating the total byte-length of the message, and modifications to the state machine to support a persistent session with wrap-around windows. Two operational modes are defined:

* RelNoAck: A mode derived from SCHC No-Ack fragmentation, where fragments are transmitted without expecting per-fragment acknowledgments. Losses are tolerated within a configured threshold.

* RelAckOnErr: A mode derived from SCHC Ack-On-Error fragmentation, where the receiver actively monitors for missing fragments and initiates recovery through explicit negative acknowledgments.

These modes offer operators the flexibility to balance recovery overhead against latency and reliability requirements in constrained network environments.

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 2 October 2025.

Table of Contents

1. Introduction

RFC8724 specifies the SCHC framework for compressing and fragmenting IPv6/UDP packets for LPWANs. While its fragmentation mechanism efficiently segments large messages, small messages that do not meet the fragmentation threshold remain vulnerable to loss due to the absence of recovery procedures. This document introduces a new rule format for reliable fragmentation that adapts SCHC Fragmentation for small-message reliability. It does so by adding a size field to every fragment and by modifying the state machine to maintain a persistent session with cyclic windowing, thereby enabling recovery even when individual fragments are lost.

2. Reliability Fragmentation Overview

The fundamental enhancement in Reliability Fragmentation is the extension of SCHC Fragmentation to small messages. This is achieved by:

* Adding a Size Field: Each fragment carries an additional field indicating the total size (in bytes) of the SCHC-compressed message. This allows the receiver to understand the complete expected payload even when messages are not naturally segmented.

* Persistent Session with Wrap-Around Windows: Unlike conventional fragmentation where the session terminates upon complete message reassembly, the Reliability Fragmentation session remains open indefinitely. The window indices wrap around, enabling recovery only within the bounds of the maintained window memory.

* Two Operational Modes: Two distinctly different modes are defined to address diverse network conditions:

  - RelNoAck: Optimized for environments where low latency is prioritized and occasional losses are acceptable.

  - RelAckOnErr: Designed for scenarios requiring strict reliability by actively recovering lost fragments.

3. Detailed Description of Reliability Modes

3.1. RelNoAck Mode

RelNoAck mode is derived from the SCHC No-Ack fragmentation mechanism. Its main characteristics are:

* Non-blocking Transmission: Fragments MAY BE sent without waiting for acknowledgments for each fragment.

* Size Field Utilization: The inclusion of a size field enables the receiver to immediately ascertain the complete message length, even if fragments arrive out of sequence.

* Loss Tolerance: In the absence of acknowledgment-driven recovery, a configurable threshold of tolerated loss is defined. If fragment loss remains within this threshold, the upper layers may accept a partial reassembly.

* Simplified State Machine: The state machine does not trigger explicit recovery procedures upon detecting a missing fragment. Instead, fragments are forwarded as they arrive, and any gaps are either ignored or handled by upper-layer protocols if the loss is deemed acceptable.

This mode is ideal for networks where retransmission overhead is undesirable and where some loss does not critically affect application performance. Another important use-case is for networks with challenged bidirectionality, such as satellite connectivity.

3.2. RelAckOnErr Mode

RelAckOnErr mode builds on the SCHC Ack-On-Error fragmentation mechanism. Its operation involves:

* Error Detection: The receiver monitors the sequence of fragments using the embedded RuleID and fragmentation parameters.

* Explicit Recovery Trigger: If a fragment is identified as missing within the active window, the receiver generates a negative acknowledgment or error report to the sender.

* Retransmission Mechanism: Upon receiving the error notification, the sender retransmits the missing fragment(s) to ensure complete message reconstruction.

* Enhanced Reliability: This mode provides robust recovery, ensuring that even if one or more fragments are lost, the complete message is eventually delivered without error.

* Dynamic Window Management: The state machine continuously manages the cyclic window, tracking received and missing fragments, and initiating recovery procedures only within the window’s scope.

RelAckOnErr is best suited for applications where data integrity is critical and where the overhead of retransmissions is justified by the need for complete and error-free delivery.

4. Packet Format for Reliability Fragmentation

The packet format for Reliability Fragmentation extends the SCHC Fragmented Header format specified in RFC8724 with an additional size field. The structure is as follows:

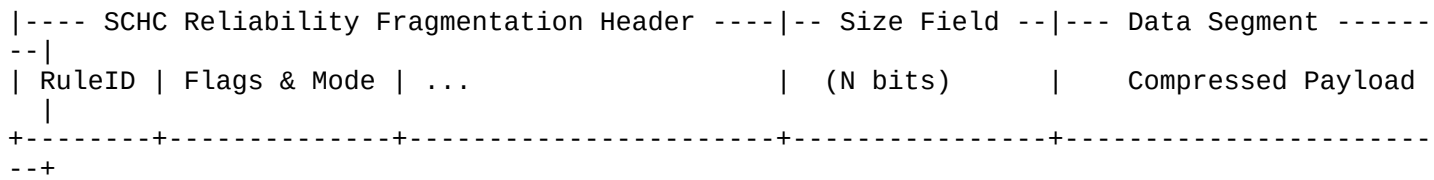

Figure 1: SCHC Reliability Fragmentation Packet Format

* Reliability RuleID: Identifies that the packet adheres to the Reliability Fragmentation format.

* Flags & Mode: Indicate the operational mode (RelNoAck or RelAckOnErr) and other control parameters.

* Size Field: An N-bit field, defined in the SCHC Context, representing the total length of the compressed message expressed in L2-words.

* Data Segment: The payload portion. This could be the result of SCHC Compression or SCHC Aggregation.

5. State Machine Modifications

The state machine for Reliability Fragmentation is adapted from the SCHC Fragmentation state machine in RFC8724 with the following key modifications:

1. Persistent Session:
   The session remains open after the complete transmission of a message, allowing the state machine to support continuous monitoring and recovery within a cyclic window.

2. Window Wrap-Around:
   Sequence numbers or window indices wrap around. Recovery procedures are constrained to the current window maintained in memory, ensuring resource constraints are respected.

3. Mode-Specific Behavior:

   * In RelNoAck mode, no major changes are necessary.

   * In RelAckOnErr mode, the receiver-side state machine actively monitors for missing fragments. Upon detecting a gap bigger than the tolerated threshold, it triggers a recovery procedure that requests retransmission of the missing fragment(s).

6. Operational Considerations

The design of Reliable Fragmentation is intended to seamlessly integrate with existing SCHC operations while providing enhanced reliability for small messages. Key considerations include:

* Immediate Upper-Layer Delivery:
  In both modes, received fragments are promptly forwarded to the upper layers.

* Policy Flexibility:
  Network operators can select between the SEND_OUT_OF_ORDER behavior (suitable for RelNoAck) and SEND_IN_ORDER behavior (necessary for RelAckOnErr) based on application latency and recovery requirements.

* Dynamic MTU Adaptation:
  Changes in the L2 Maximum Transmission Unit require dynamic adjustment of the window size. The system ensures that all fragments adhere to the current MTU, preserving data integrity.

7. Flow Diagram

The diagram below illustrates the flow of data through the Reliability Fragmentation process:

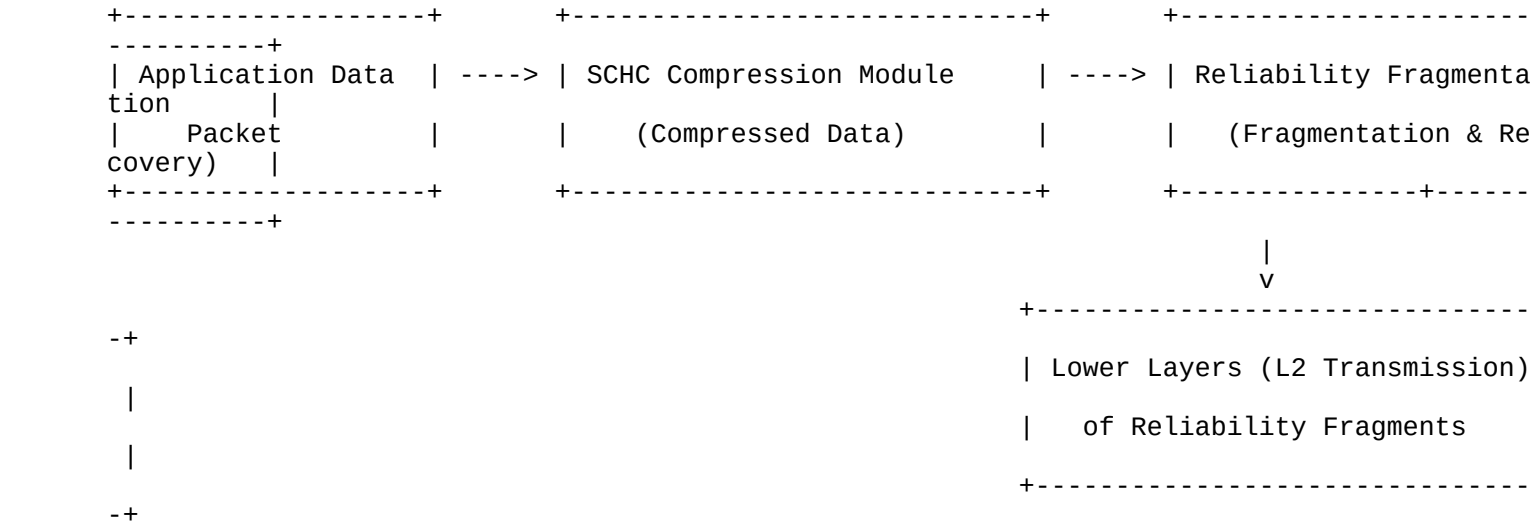

Figure 2: Data Flow for SCHC Reliability Fragmentation

8. Security Considerations

The modifications introduced by Reliability Fragmentation, such as the persistent session and additional size field, do not fundamentally alter the SCHC security model defined in RFC8724. Implementations must ensure that integrity and authenticity checks cover all fragments and that recovery procedures do not create new vulnerabilities.

9. IANA Considerations

No IANA Considerations.

10. Examples and Use Cases

10.1. Example 1: RelNoAck for Sensor Networks

In a sensor network where data is periodically transmitted, the RelNoAck mode is employed. Sensors compress their data using SCHC Compression, and the resulting packets are processed by the Reliable Fragmentation module. Fragments are sent continuously without waiting for acknowledgments. Each individual message is sent to the upper layers as soon as it is received. Occasional losses are tolerated within a predefined threshold, making this mode suitable for non-critical monitoring applications.

10.1.1. Example 2: RelAckOnErr for Critical Data Delivery

In applications where data integrity is needed, such as in industrial monitoring, the RelAckOnErr mode is utilized. Each fragment contains a separate message. as in RelNoAck mode, each message is transfered to the upper layers as soon as it is received. If a missing fragment is detected, Ack-On-Err mechanisms are used to recover for the missing fragments.

11. Normative References

[RFC8724] Minaburo, A., Toutain, L., Gomez, C., Barthel, D., and JC. Zuniga, "SCHC: Generic Framework for Static Context Header Compression and Fragmentation", RFC 8724, DOI 10.17487/RFC8724, April 2020, <https://www.rfc-editor.org/info/rfc8724>.

Authors' Addresses

Alexander Pelov
IMT Atlantique
2bis rue de la Chataigneraie
35536 Cesson-S迪vign迪
France
Email: alexander.pelov@imt-atlantique.fr

Carles Gomez
Universitat Politecnica de Catalunya/Fundacio i2CAT
C/Esteve Terradas, 7
08860 Castelldefels
Spain
Phone: +34-93-413-7206
Email: carlesgo@entel.upc.edu