SCHC Working Group
Internet-Draft
Intended status: Informational
Expires: 13 November 2025

S. Sirohi
L. Toutain
IMT Atlantique

12 May 2025

QUIC compression using SCHC
draft-sirohi-schc-quic-compression-00

Abstract

This document specifies a mechanism for applying Static Context Header Compression (SCHC) to QUIC (Quick UDP Internet Connections) packets. It leverages SCHC to compress both the QUIC packet header and the headers or type information of QUIC frames contained within the packet payload. This approach aims to significantly reduce QUIC overhead, optimizing bandwidth usage. This is particularly beneficial for valuable in bandwidth-sensitive scenarios like deep space communications.

This document defines the SCHC architecture.

Table of Contents

1. Introduction

QUIC [rfc9000] provides a secure, reliable, multiplexed transport protocol over UDP. While efficient, its packet headers and frame headers can still represent significant overhead relative to the payload, especially when carrying small amounts of application data. This overhead can be detrimental on bandwidth-sensitive links, such as those found in deep space communication.

SCHC [rfc8724] provides a powerful compression mechanism optimized for environments where endpoints share static contexts describing the expected traffic patterns, allowing for significant compression based on predictability, similar in principle to statistical compression approaches. This document proposes a method to apply SCHC compression in two stages within the QUIC packet processing pipeline: first to individual QUIC frames' headers/metadata, and second to the outer QUIC packet header itself.

The goal is to minimize the transmission size of QUIC packets over links where SCHC is deployed, potentially enabling more efficient QUIC usage in deep space communication scenarios and other environments where bandwidth is at a premium.

2. Terminology

* *SCHC:* Static Context Header Compression [rfc8724].

* *QUIC:* Transport protocol as defined in [rfc9000].

* *QUIC Frame:* Units of data carried within QUIC packets (e.g., STREAM, ACK, PING).

* *Compressed Frame (CF):* The result of applying SCHC compression rules to the header/type/metadata of a single QUIC Frame. The frame's payload (if any) remains uncompressed at this stage.

* *Inner SCHC Packet:* An aggregation of one or more Compressed Frames, prefixed by a SCHC Rule ID and containing length information. This forms the payload for QUIC encryption.

* *Outer QUIC Header:* The standard QUIC packet header (short or long).

* *Compressed QUIC Header:* The result of applying SCHC compression rules to the Outer QUIC Header.

* *RID:* SCHC Rule ID used for compression/decompression.

* *L:* Length field associated with a Compressed Frame within the Inner SCHC Packet.

3. Architecture Overview

The proposed mechanism modifies the QUIC packet construction process by introducing SCHC compression at two distinct stages. Fig. 1 illustrates the overall flow.

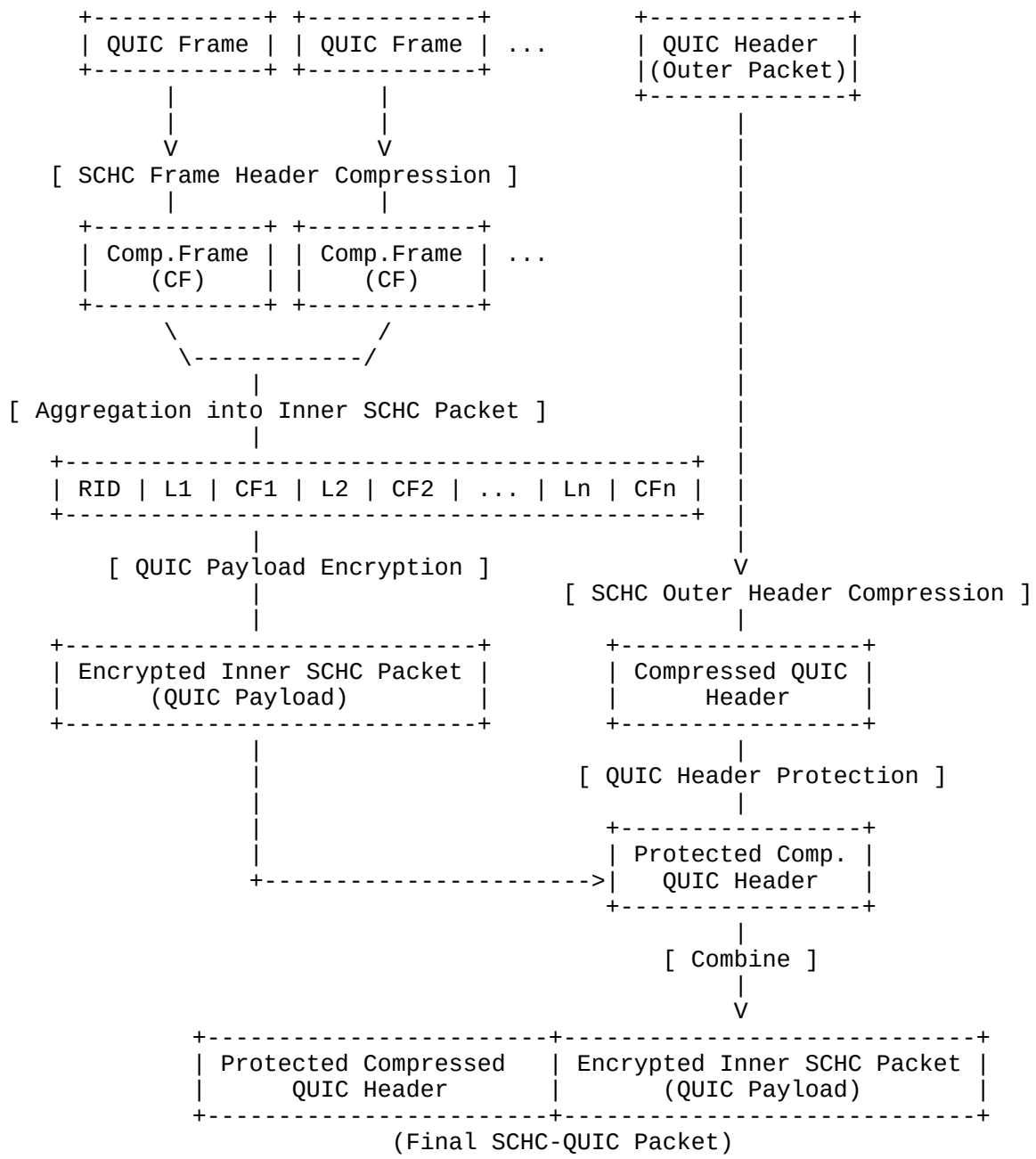

Figure 1: SCHC-QUIC Compression Flow

Figure 1: SCHC-QUIC Compression Architecture

The specific steps involved are detailed below:

3.1. QUIC Frame Header Compression

* Before QUIC frames are encrypted, their headers (and potentially other predictable metadata, depending on the SCHC rule) are processed using a set of predefined SCHC rules specific to QUIC frame types.

* Each QUIC Frame (e.g., STREAM frame header, ACK frame ranges, PING frame type) is compressed into a "Compressed Frame" (CF).

* The payload of frames like STREAM frames is NOT compressed by SCHC at this stage but remains associated with its compressed header/ type.

* Specific SCHC rules MUST be defined for various QUIC frame types and their fields.

3.2. Inner SCHC Packet Aggregation

* One or more Compressed Frames (CFs) destined for the same QUIC packet are aggregated.

* This aggregation is formed into an "Inner SCHC Packet". The exact structure and encoding for this aggregation require further specification. It needs to allow the decompressor to correctly parse the individual CFs and any associated uncompressed payload data.

* This might involve prefixing the aggregation with a SCHC Rule ID (RID) that identifies the set of SCHC rules used for the frame header compression (Section 3.1) and potentially defines the structure of the aggregation itself.

* It may also involve including explicit Length (L) fields or using other delimiting mechanisms defined by a specific SCHC "aggregation rule" to separate the CFs and their associated payloads within the Inner SCHC Packet.

3.3. QUIC Payload Encryption

* The _entire_ Inner SCHC Packet (RID, L/CF pairs) is treated as the plaintext payload for QUIC packet encryption.

* Standard QUIC packet protection [rfc9001] is applied to encrypt this payload.

3.4. Outer QUIC Header Compression

* The QUIC packet header (Long or Short Header, including fields like DCID, SCID, Packet Number, etc.) is compressed using a _separate_ set of SCHC rules designed for QUIC packet headers.

* This results in a "Compressed QUIC Header".

* The SCHC context and rules for outer header compression are distinct from those used for inner frame compression.

3.5. QUIC Header Protection

* Standard QUIC Header Protection [rfc9001] is applied to the _Compressed QUIC Header_.

* The specific mechanism needs to ensure that header protection can be correctly applied and removed on the compressed header format. This might involve defining SCHC rules such that the fields needed for the protection algorithm (e.g., Packet Number, payload sample) remain accessible, predictable, or are handled appropriately by the SCHC rule definition. This interaction requires careful definition (see Section 6).

The final packet sent over the link consists of the Protected (Compressed QUIC Header) followed by the Encrypted (Inner SCHC Packet).

4. Packet Formats (Informative)

This section provides a conceptual view based on the architecture. Precise field sizes and encodings are outside the scope of this initial draft.

4.1. Inner SCHC Packet

The structure of the Inner SCHC Packet needs to encapsulate one or more Compressed Frames (CFs) and their associated payloads (if any) in a way that can be parsed by the decompressor.

One potential approach, drawing inspiration from techniques like those used in [I-D.ietf-schc-icmpv6-compression], could involve:

* Using different SCHC Rule IDs (RIDs) for the Inner SCHC packet itself. Each RID might implicitly define the number and types of QUIC frames contained, or define a general structure.

* Employing SCHC Matching Operators (MOs) and Compression/ Decompression Actions (CDAs) to handle the variable fields within each frame’s compressed representation (CF).

* Using variable-length encoding mechanisms defined within SCHC or the specific rule to handle lengths of payloads or other variable elements.

A simplified conceptual diagram might look like:

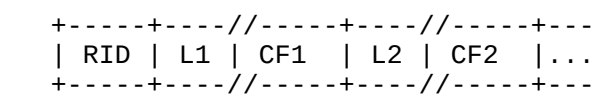

Figure 2: Agregation Frame

* *RID:* SCHC Rule ID for frame compression rules.

* *L:* Length of the following CF (potentially including uncompressed payload).

* *CF:* Compressed Frame (compressed header/type + uncompressed payload, if any).

4.2. Final SCHC-QUIC Packet Structure

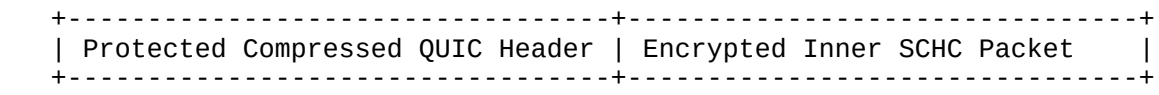

Figure 3: Full Message

* *Protected Compressed QUIC Header:* The result of applying SCHC compression (Section 3.4) and then QUIC Header Protection (Section 3.5) to the Outer QUIC Header.

* *Encrypted Inner SCHC Packet:* The result of encrypting the structure defined in Section 4.1 using QUIC Payload Encryption (Section 3.3).

5. SCHC Context Considerations

* This mechanism REQUIRES two distinct SCHC Contexts to be established between the SCHC compression/decompression endpoints:

  - *Frame Context:* Contains rules for compressing QUIC Frame headers/types. Identified by the RID in the Inner SCHC Packet.

- *Packet Header Context:* Contains rules for compressing the Outer QUIC Header. The method for identifying the applicable rule for this context (e.g., implicit, lower-layer signaling) needs further definition.

* The establishment, management, and synchronization of these SCHC contexts are outside the scope of this document but are essential prerequisites for operation (e.g., using mechanisms defined in [rfc8724] or other means).

6. Header Protection

QUIC packet headers contain sensitive information, like the Packet Number, which must be protected from pervasive observation while still allowing network devices to route packets based on other header fields (e.g., Connection IDs). QUIC employs a dedicated Header Protection mechanism, separate from the payload encryption, to achieve this. This section details the standard QUIC Header Protection mechanism defined in [rfc9001], explains how it interacts with the proposed SCHC compression of the outer QUIC header, and discusses potential implications.

6.1. Standard QUIC Header Protection Mechanism

QUIC Header Protection uses a key derived specifically for this purpose and applies a pseudo-random mask to obfuscate certain header fields. The process ensures that only endpoints possessing the correct header protection key can remove the mask and correctly interpret the protected fields.

The core of header protection is generating a 5-byte mask. This mask generation relies on:

* The Header Protection Key

* A Ciphertext Sample: A small sample (typically 16 bytes, matching the AEAD algorithm’s tag length) is taken from the encrypted payload of the QUIC packet. The sample starts at an offset calculated relative to the start of the Packet Number field in the header. Crucially, for sampling purposes, the Packet Number field is always assumed to be 4 bytes long (its maximum possible encoded length), regardless of its actual encoded length in the packet. This ensures the sampling offset is consistent even before the header is unprotected. Packets too short to provide a full sample must be discarded.

The header protection key and the ciphertext sample are fed into a specific cipher algorithm to produce the 5-byte mask.

The generated 5-byte mask is XORed with specific parts of the QUIC header:

* First Byte: The first byte of the mask is XORed with specific bits in the first byte of the header (the Flags byte). For Long Headers, this includes the 2 Reserved bits and the 2 Packet Number Length bits. For Short Headers, this includes the Reserved bits and the Packet Number Length bits (5 bits total). The most significant bits (Header Form, Fixed Bit, Long Packet Type / Spin Bit) are not masked.

* Packet Number Field: The remaining 4 bytes of the mask are XORed with the first 1 to 4 bytes of the Packet Number field, matching its encoded length

6.2. Header Protection with SCHC Outer Header Compression

In the SCHC-QUIC mechanism proposed, SCHC rules are applied to the Outer QUIC Header first. This compresses fields like Connection IDs, Packet Number, and flags based on the established SCHC Packet Header Context. The output is a "Compressed QUIC Header". This compressed header contains the residue of the compression the information that could not be elided by the SCHC rules. The Rule ID (RID) for the outer header compression might also form part of this output. QUIC Header Protection is then applied to the resulting "Compressed QUIC Header" as depicted in the Fig. 4.

This approach introduces specific requirements and interactions:

* *Protecting the Residue:* The Header Protection mask is applied to the compressed representation. The SCHC rules used for the outer header must be designed such that the sensitive information QUIC aims to protect (primarily the Packet Number and its length) are either present in the residue in a predictable location and format, allowing the standard HP masking to be applied effectively or are reconstructible from the residue after HP removal, ensuring the HP process itself can derive the necessary inputs (like the original Packet Number for nonce generation, even if the PN field itself is compressed in the residue).

* *Protecting the Outer Rule ID:* The SCHC outer header compression scheme requires transmitting the Rule ID (RID) to identify which set of compression rules are applied, this RID could potentially also be obfuscated by the Header Protection mask. Applying HP to the RID in this way contributes to making traffic analysis harder, as it obscures which specific compression rule was used for the outer header until HP is removed. Note: This applies only to the RID for the outer header compression. The RID for the inner frame compression is part of the Inner SCHC Packet, which becomes the encrypted payload and is protected by QUIC payload encryption.

* *Ciphertext Sampling:* The standard HP mechanism calculates the ciphertext sample offset based on the Packet Number field location in the unprotected header. When HP is applied to a compressed header, the SCHC rule and the HP implementation must agree on how this offset is determined before HP removal and header decompression. The mechanism must ensure that the location of the (potentially compressed) Packet Number field within the compressed header is defined, allowing the receiver to calculate the correct sample offset from the encrypted payload before removing HP. This might require preserving the relative position or provide a fixed offset for sampling.

By applying HP to the compressed header residue (and potentially the outer RID), we aim to obfuscate the remaining header information and the specific compression details, making it more difficult for observers to infer traffic patterns or the structure of the compressed data.

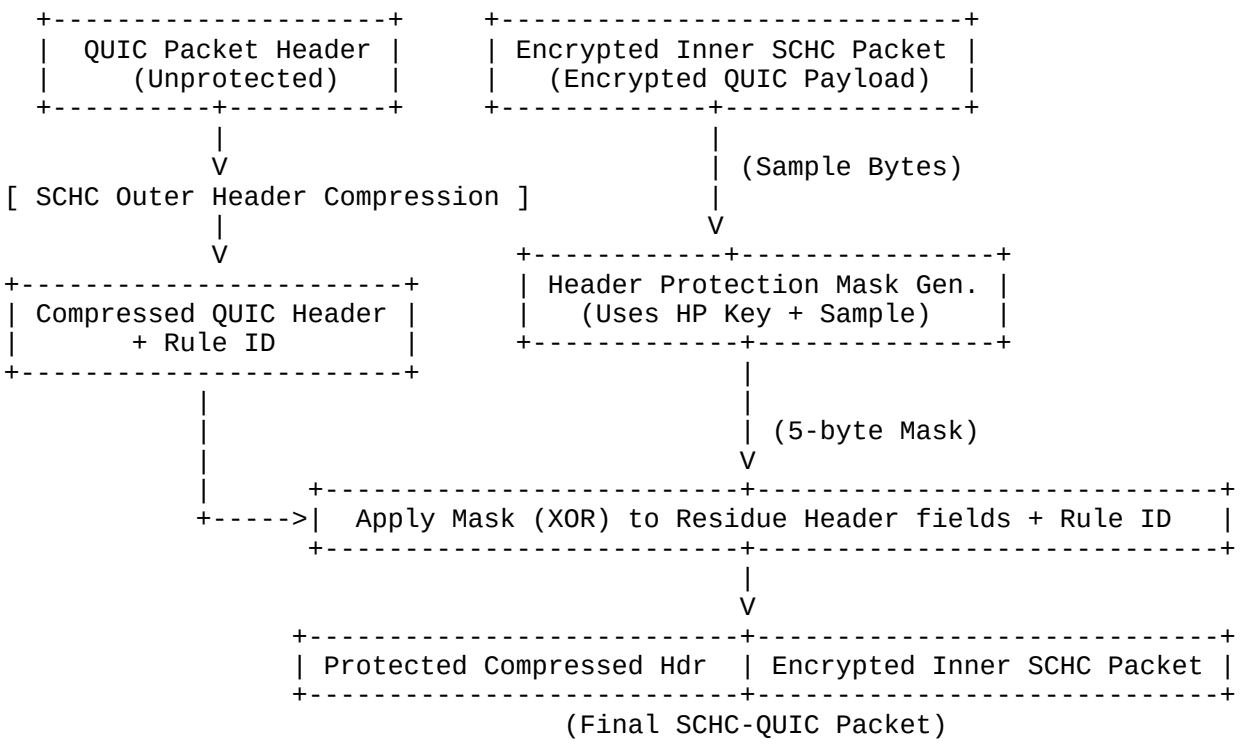

Figure 4: Header Protection with SCHC

7. Security Considerations

* SCHC itself does not provide encryption. This mechanism relies on QUIC's standard payload encryption and header protection.

* Payload encryption is applied _after_ frame aggregation, ensuring the confidentiality and integrity of the compressed frames and their payloads.

* As detailed in Section 6, Header protection is applied _after_ outer QUIC header compression. This interaction requires careful design choices as mentioned above.

* Compression techniques can potentially leak information through variations in output size. The security implications of this for QUIC metadata (frame types, packet numbers, etc.) MUST be analyzed.

* The security of the SCHC context establishment and management process is paramount. Compromised contexts could lead to incorrect decompression or information disclosure.

8. Acknowledgements

The authors would like to thank (in alphabetic order): Nicolas Kuhn

9. Normative References

[I-D.ietf-schc-icmpv6-compression]
Barthel, D. and L. Toutain, "Static Context Header Compression (SCHC) for the Internet Control Message Protocol (ICMPv6)", Work in Progress, Internet-Draft, draft-ietf-schc-icmpv6-compression-01, 25 November 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-schc-icmpv6-compression-01>.

[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/rfc/rfc8724>.

[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/rfc/rfc9000>.

[rfc9001] Thomson, M., Ed. and S. Turner, Ed., "Using TLS to Secure QUIC", RFC 9001, DOI 10.17487/RFC9001, May 2021, <https://www.rfc-editor.org/rfc/rfc9001>.

Authors' Addresses

Samar Sirohi
IMT Atlantique
rue de la Chataigneraie
35576 Cesson-Sevigne Cedex
France
Email: samar.sirohi@imt-atlantique.net

Laurent Toutain
IMT Atlantique
rue de la Chataigneraie
35576 Cesson-Sevigne Cedex
France

Email: laurent.toutain@imt-atlantique.fr