

Sliding Window Random Linear Code (RLC) Forward Erasure Correction (FEC) Schemes for QUIC

Vincent Roca, Ian Swett, Marie-Jose Montpetit

► **To cite this version:**

Vincent Roca, Ian Swett, Marie-Jose Montpetit. Sliding Window Random Linear Code (RLC) Forward Erasure Correction (FEC) Schemes for QUIC. Internet Research Task Force, Coding for Efficient Network Communications Research Group (NWCRG) .. 2018. <hal-01919876>

HAL Id: hal-01919876

<https://hal.inria.fr/hal-01919876>

Submitted on 12 Nov 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

nwcrg
Internet-Draft
Intended status: Informational
Expires: December 23, 2018

V. Roca
INRIA
I. Swett
Google
M-J. Montpetit
Triangle Video
June 21, 2018

Sliding Window Random Linear Code (RLC) Forward Erasure Correction (FEC)
Schemes for QUIC
[draft-roca-nwcrg-rlc-fec-scheme-for-quic-00](#)

Abstract

This document specifies Sliding Window Random Linear Code (RLC) Forward Erasure Correction (FEC) Schemes for the QUIC transport protocol, in order to recover from packet losses.

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 December 23, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in [Section 4.e](#) of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Definitions and Abbreviations	3
3. Procedures	3
3.1. Source Symbols Mapping	3
3.2. Pseudo-Random Number Generator (PRNG)	4
3.3. Coding Coefficients Generation Function	4
4. Sliding Window RLC FEC Scheme over $GF(2^{8})$ when Protecting a Single QUIC Stream	4
4.1. Formats and Codes	4
4.1.1. Configuration Information	4
4.1.2. REPAIR Frame Format	5
4.1.3. Additional Procedures	6
4.2. FEC Code Specification	6
4.2.1. Encoding Side	6
4.2.2. Decoding Side	6
5. Sliding Window RLC FEC Scheme over $GF(2^{8})$ when Protecting Several QUIC Streams	6
6. Security Considerations	7
7. IANA Considerations	7
8. Acknowledgments	7
9. References	7
9.1. Normative References	7
9.2. Informative References	8
Authors' Addresses	8

1. Introduction

QUIC [[QUIC-transport](#)] is a new transport that aims at improving network performance by enabling out of order delivery, partial reliability, and methods of recovery besides retransmission, while also improving security. This document specifies FEC schemes for Sliding Window Random Linear Code (RLC) [[RLC](#)] to recover from lost packets within a single QUIC stream or across several QUIC streams, compliant with the FEC coding framework for QUIC [[Coding4QUIC](#)].

The ability to add FEC coding in QUIC may be beneficial in several situations:

- o for a robust transmission of latency sensitive traffic, for instance real-time flows, since it enables to recover packet losses independently of the round trip time;

- o for the transmission of contents to a large set of QUIC reception endpoints, since the same repair frame may help recovering several different packet losses at different receivers;
- o for multipath communications, since repair traffic adds diversity.

2. Definitions and Abbreviations

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Terms and definitions that apply to coding are available in [nc-taxonomy]. More specifically, this document uses the following definitions:

Packet versus Symbol: a Packet is the unit of data that is exchanged over the network while a Symbol is the unit of data that is manipulated during the encoding and decoding operations

Source Symbol: a unit of data originating from the source that is used as input to encoding operations

Repair Symbol: a unit of data that is the result of a coding operation

This document uses the following abbreviations:

E: size of an encoding symbol (i.e., source or repair symbol), assumed fixed (in bytes)

3. Procedures

This section introduces the procedures that are used by these FEC Schemes.

3.1. Source Symbols Mapping

The present FEC Scheme follows the source symbols mapping specified in [Coding4QUIC]. Figure 1 illustrates this mapping.

```

< -E- > < -E- > < -E- > < -E- >
+-----+-----+-----+-----+
|< -- Frame 1 -- >< ----- Frame | source symbols 0, 1, 2, 3
+-----+-----+-----+-----+
| 2 ----- >< --- Frame 3 -- >< -| source symbols 4, 5, 6, 7
+-----+-----+-----+-----+
| Frame 4 - >< -F5- >| source symbols 8, 9 and 10
+-----+-----+-----+-----+
                                (incomplete)

```

Figure 1: Example of source symbol mapping, when the E value is relatively small.

3.2. Pseudo-Random Number Generator (PRNG)

The RLC FEC Schemes defined in this document rely on the TinyMT32 PRNG defined in [RLC].

3.3. Coding Coefficients Generation Function

The coding coefficients, used during the encoding process, are generated at the RLC encoder by the `generate_coding_coefficients()` function each time a new repair symbol needs to be produced. This specification uses the `generate_coding_coefficients()` defined in [RLC].

4. Sliding Window RLC FEC Scheme over $GF(2^{8})$ when Protecting a Single QUIC Stream

This fully-specified FEC Scheme defines the Sliding Window Random Linear Codes (RLC) over $GF(2^{8})$ when protecting a single QUIC stream.

4.1. Formats and Codes

4.1.1. Configuration Information

This section provides the RLC configuration information that needs to be shared during QUIC negotiation between the QUIC sender and receiver endpoints in order to synchronize them.

- o FEC Encoding ID: the value assigned to this fully specified FEC Scheme MUST be XXXX, as assigned by IANA (Section 7). This FEC Encoding ID is used during the QUIC negotiation to uniquely identify the RLC FEC Scheme for QUIC;
- o Encoding symbol size, E (in bytes): a non-negative integer that indicates the size of each source and repair symbol, in bytes.

This element is required both by the QUIC sender endpoint (RLC encoder) and the QUIC receiver endpoint(s) (RLC decoder).

4.1.2. REPAIR Frame Format

The RLC FEC Scheme does not use any explicit Source FEC Payload ID, meaning that QUIC STREAM frame format is not modified.

On the opposite, the RLC FEC Scheme requires QUIC REPAIR frames to convey enough information. This section specifies the REPAIR frame format specific to the RLC FEC Scheme and a single QUIC stream. Note that the notion of REPAIR frame format is equivalent to the notion of Repair FEC Payload ID in [RLC].

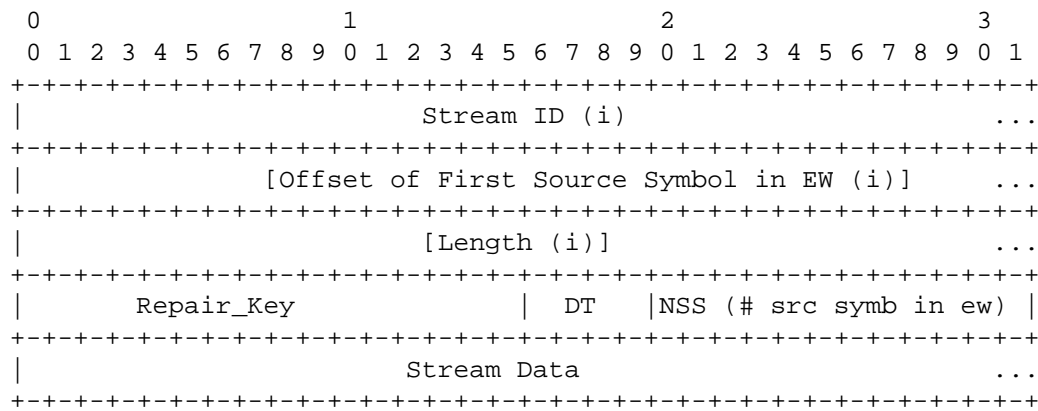


Figure 2: REPAIR frame format when protecting a single QUIC stream.

More precisely, the REPAIR frame format is composed of the following fields (Figure 2):

Stream ID (variable-size field): a variable-length integer indicating the stream ID of the stream. See [QUIC-transport]. The Stream ID for an RLC REPAIR frame MUST be equal to the Stream ID used for the data stream it protects;

Offset of First Source Symbol in the Encoding Window (variable-size field): a variable-length integer specifying the byte offset in the stream for the first source symbol of the encoding window.

Length (variable-size field): a variable-length integer specifying the length of the Stream Data field in this REPAIR frame. This length MUST be a non zero multiple of the source symbol size, E, since a REPAIR frame contains one or more repair symbols for this stream;

Repair_Key (16-bit field): this unsigned integer is used as a seed by the coefficient generation function ([Section 3.3](#)) in order to generate the desired number of coding coefficients. When a FEC Repair Packet contains several repair symbols, this repair key value is that of the first repair symbol. The remaining repair keys can be deduced by incrementing by 1 this value, up to a maximum value of 65535 after which it loops back to 0.

Density Threshold for the coding coefficients, DT (4-bit field): this unsigned integer carries the Density Threshold (DT) used by the coding coefficient generation function [Section 3.3](#). More precisely, it controls the probability of having a non zero coding coefficient, which equals $(DT+1) / 16$. When a FEC Repair Packet contains several repair symbols, the DT value applies to all of them;

Number of Source Symbols in the encoding window, NSS (12-bit field):

this unsigned integer indicates the number of source symbols in the encoding window when this repair symbol was generated. When a FEC Repair Packet contains several repair symbols, this NSS value applies to all of them;

Stream Data: data for this repair symbol(s).

4.1.3. Additional Procedures

4.2. FEC Code Specification

This RLC FEC Scheme relies on the FEC code specification defined in [\[RLC\]](#).

4.2.1. Encoding Side

[RLC] high level description of a Sliding Window RLC encoder also applies here to this FEC Scheme.

4.2.2. Decoding Side

[RLC] high level description of a Sliding Window RLC decoder also applies here to this FEC Scheme.

5. Sliding Window RLC FEC Scheme over $GF(2^{8})$ when Protecting Several QUIC Streams

This section focusses on the general case where FEC protection is globally applied across two or more QUIC streams.

TODO

6. Security Considerations

TBD

7. IANA Considerations

This document registers two values in the "QUIC FEC Encoding IDs" registry as follows:

- o XXXX refers to the Sliding Window Random Linear Codes (RLC) over $GF(2^{8})$ FEC Scheme for a Single QUIC Stream, as defined in [Section 4](#) of this document.
- o YYYY refers to the Sliding Window Random Linear Codes (RLC) over $GF(2^{8})$ FEC Scheme for a Several QUIC Stream, as defined in [Section 5](#) of this document.

8. Acknowledgments

TBD

9. References

9.1. Normative References

[Coding4QUIC]

Swett, I., Montpetit, M-J., and V. Roca, "Coding for QUIC", Work in Progress, NWCRG [draft-swett-nwcrg-coding-for-quic](#) (Work in Progress), June 2018, <<https://tools.ietf.org/html/draft-swett-nwcrg-coding-for-quic>>.

[QUIC-transport]

Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", [draft-ietf-quic-transport-12](#) (work in progress), May 2018, <<https://datatracker.ietf.org/doc/draft-ietf-quic-transport/>>.

[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/info/rfc2119>>.

[RLC] Roca, V., "Sliding Window Random Linear Code (RLC) Forward Erasure Correction (FEC) Scheme for FECFRAME", Work in Progress, Transport Area Working Group (TSVWG) [draft-ietf-tsvwg-rlc-fec-scheme](#) (Work in Progress), May 2018, <<https://tools.ietf.org/html/draft-ietf-tsvwg-rlc-fec-scheme>>.

9.2. Informative References

[nc-taxonomy] Roca et al., V., "Taxonomy of Coding Techniques for Efficient Network Communications", [draft-irtf-nwcrg-network-coding-taxonomy](#) (Work in Progress) (work in progress), June 2018, <<https://datatracker.ietf.org/doc/draft-irtf-nwcrg-network-coding-taxonomy/>>.

Authors' Addresses

Vincent Roca
INRIA
Univ. Grenoble Alpes
France

Email: vincent.roca@inria.fr

Ian Swett
Google
Cambridge, MA
US

Email: ianswett@google.com

Marie-Jose Montpetit
Triangle Video
Boston, MA
US

Email: marie@mjmontpetit.com