FecFrame
Internet-Draft
Intended status: Experimental
Expires: April 25, 2011

V. Roca
INRIA
M. Cunche
NICTA
J. Lacan
ISAE/LAAS-CNRS

October 22, 2010

Simple LDPC-Staircase Forward Error Correction (FEC) Scheme for FECFRAME
draft-roca-fecframe-ldpc-01

Abstract

This document describes a fully-specified simple FEC scheme for LDPC-staircase codes that can be used to protect media streams along the lines defined by the FECFRAME framework. These codes have many interesting properties: they are systematic codes, they perform close to ideal codes in many use-cases and they also feature very high encoding and decoding throughputs. LDPC-Staircase codes are therefore a good solution to protect a single high bitrate source flow, or to protect globally several mid-rate flows within a single FECFRAME instance. They are also a good solution whenever the processing load of a software encoder or decoder must be kept to a minimum.

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 http://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 April 25, 2011.

Copyright Notice

Table of Contents

1. Introduction

The use of Forward Error Correction (FEC) codes is a classic solution to improve the reliability of unicast, multicast and broadcast Content Delivery Protocols (CDP) and applications [RFC3453]. The [FECFRAME-FRAMEWORK] document describes a generic framework to use FEC schemes with media delivery applications, and for instance with real-time streaming media applications based on the RTP real-time protocol. Similarly the [RFC5052] document describes a generic framework to use FEC schemes with with objects (e.g., files) delivery applications based on the ALC [RFC5775] and NORM [RFC5740] reliable multicast transport protocols.

More specifically, the [RFC5053] (Raptor) and [RFC5170] (LDPC-Staircase and LDPC-Triangle) FEC schemes introduce erasure codes based on sparse parity check matrices for object delivery protocols like ALC and NORM. Similarly, the [RFC5510] document introduces Reed-Solomon codes based on Vandermonde matrices for the same object delivery protocols. All these codes are systematic codes, meaning that the k source symbols are part of the n encoding symbols. Additionally, the Reed-Solomon FEC codes belong to the class of Maximum Distance Separable (MDS) codes that are optimal in terms of erasure recovery capabilities. It means that a receiver can recover the k source symbols from any set of exactly k encoding symbols out of n. This is not the case with either Raptor or LDPC-Staircase codes, and these codes require a certain number of encoding symbols in excess to k. However, this number is small in practice when an appropriate decoding scheme is used at the receiver [SPSC08]. Another key difference is the high encoding/decoding complexity of Reed-Solomon codecs compared to Raptor or LDPC-Staircase codes. A difference of one or more orders of magnitude or more in terms of encoding/decoding speed exists between the Reed-Solomon and LDPC-Staircase software codecs [SPSC08][CunchePHD10]. Finally, Raptor and LDPC-Staircase codes are large block FEC codes, in the sense of [RFC3453], since they can efficiently deal with a large number of source symbols.

The present document focuses on LDPC-Staircase codes, that belong to the well-known class of "Low Density Parity Check" codes. Because of their key features, these codes are a good solution to protect a single high bitrate source flow as in [LCN10], or to protect globally several mid-rate source flows within a single FECFRAME instance. They are also a good solution whenever processing requirements at a software encoder or decoder must be kept to a minimum, independently of the ADU flow(s) bitrate.

This documents inherits from [RFC5170] the specifications of the core LDPC-Staircase codes. Therefore this document specifies only the

information specific to the FECFRAME context and refers to [RFC5170] for the core specifications of the codes. To that purpose, the present document introduces:

- the Fully-Specified FEC Scheme with FEC Encoding ID XXX that specifies a simple way of using LDPC-Staircase codes in order to protect arbitrary ADU flows.

Finally, a publicly available reference implementation of these codes is available and distributed under a GNU/LGPL (Lesser General Public License) [LDPC-codec].

2. Terminology

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 RFC 2119 [RFC2119].

3. Definitions Notations and Abbreviations

3.1. Definitions

This document uses the following terms and definitions. Some of them are FEC scheme specific and are in line with [RFC5052]:

Source symbol: unit of data used during the encoding process. In this specification, there is always one source symbol per ADU.

Encoding symbol: unit of data generated by the encoding process. With systematic codes, source symbols are part of the encoding symbols.

Repair symbol: encoding symbol that is not a source symbol.

Code rate: the k/n ratio, i.e., the ratio between the number of source symbols and the number of encoding symbols. By definition, the code rate is such that: 0 < code rate <= 1. A code rate close to 1 indicates that a small number of repair symbols have been produced during the encoding process.

Systematic code: FEC code in which the source symbols are part of the encoding symbols. The Reed-Solomon codes introduced in this document are systematic.

Source block: a block of k source symbols that are considered together for the encoding.

Packet Erasure Channel: a communication path where packets are either dropped (e.g., by a congested router, or because the number of transmission errors exceeds the correction capabilities of the physical layer codes) or received. When a packet is received, it is assumed that this packet is not corrupted.

Some of them are FECFRAME framework specific and are in line with

[FECFRAME-FRAMEWORK]:

Application Data Unit (ADU): a unit of data coming from (sender) or given to (receiver) the media delivery application. Depending on the use-case, an ADU can use an RTP encapsulation. In this specification, there is always one source symbol per ADU.

(Source) ADU Flow: a flow of ADUs from a media delivery application and to which FEC protection is applied. Depending on the use-case, several ADU flows can be protected together by the FECFRAME framework.

ADU Block: a set of ADUs that are considered together by the FECFRAME instance for the purpose of the FEC scheme. Along with the F[], L[], and Pad[] fields, they form the set of source symbols over which FEC encoding will be performed.

ADU Information (ADUI): a unit of data constituted by the ADU and the associated Flow ID, Length and Padding fields (Section 4.3). This is the unit of data that is used as source symbol.

FEC Framework Configuration Information: the FEC scheme specific information that enables the synchronization of the FECFRAME sender and receiver instances.

FEC Source Packet: a data packet submitted to (sender) or received from (receiver) the transport protocol. It contains an ADU along with its optional Explicit Source FEC Payload ID.

FEC Repair Packet: a repair packet submitted to (sender) or received from (receiver) the transport protocol. It contains a repair symbol along with its Repair FEC Payload ID.

The above terminology is illustrated in Figure 1 (sender’s point of view):

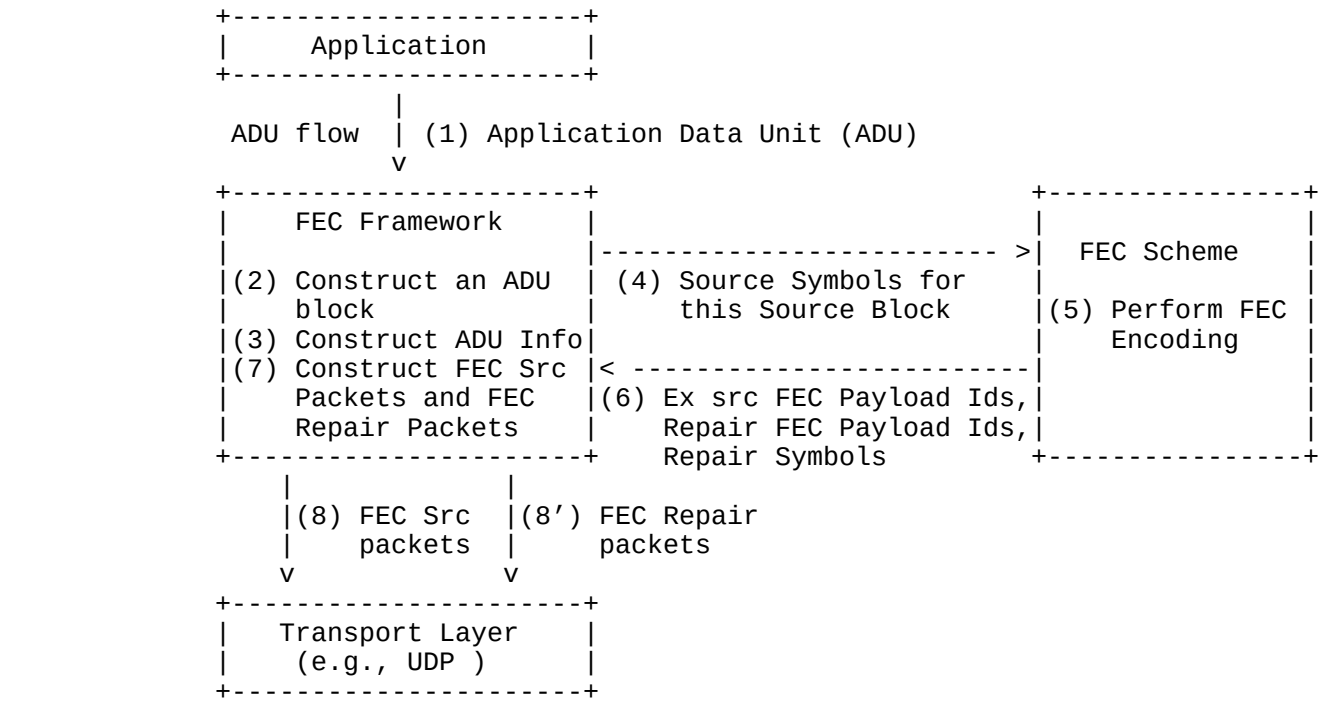

Figure 1: Terminology used in this document (sender).

3.2. Notations

This document uses the following notations: Some of them are FEC scheme specific:

k	denotes the number of source symbols in a source block.
max_k	denotes the maximum number of source symbols for any source block.
n	denotes the number of encoding symbols generated for a source block.
E	denotes the encoding symbol length in bytes.
CR	denotes the "code rate", i.e., the k/n ratio.
N1	denotes the target number of "1s" per column in the left side of the parity check matrix.
N1m3	denotes the value N1 - 3.
a^^b	denotes a raised to the power b.

Some of them are FECFRAME framework specific:

B denotes the number of ADUs per ADU block.
max_B denotes the maximum number of ADUs for any ADU block.

3.3. Abbreviations

This document uses the following abbreviations:
ADU stands for Application Data Unit.
ESI stands for Encoding Symbol ID.
FEC stands for Forward Error (or Erasure) Correction code.
FFCI stands for FEC Framework Configuration Information.
LDPC stands for Low Density Parity Check.
RS stands for Reed-Solomon.
MDS stands for Maximum Distance Separable code.

4. Common Procedures Related to the ADU Block and Source Block Creation

This section introduces the procedures that are used during the ADU block and the related source block creation, for the FEC scheme considered.

4.1. Restrictions

This specification has the following restrictions:
- there MUST be exactly one source symbol per ADUI, and therefore per ADU;
- there MUST be exactly one repair symbol per FEC Repair Packet;
- there MUST be exactly one source block per ADU block;
- the use of the LDPC-Staircase scheme is such that there MUST be exactly one encoding symbol per group, i.e., G MUST be equal to 1 [RFC5170];

4.2. ADU Block Creation

Several aspects must be considered, that impact the ADU block creation:
- the maximum source block size (max_k parameter);
- the potential real-time constraints, that impact the maximum ADU block size, since the larger the block size, the larger the decoding delay;

We now detail each of these aspects.

The maximum source block length in symbols, max_k, depends on several parameters: the code rate (CR), the Encoding Symbol ID (ESI) field length in the Explicit Source/Repair FEC Payload ID (16 bits), as well as possible internal codec limitations. More specifically, max_k cannot be larger than the following values, derived from the ESI field size limitation, for a given code rate:

max1_k = 2^^(16 - ceil(Log2(1/CR)))

Some common max1_k values are:

- CR == 1 (no repair symbol): max1_k = 2^^16 = 65536 symbols
- 1/2 <= CR < 1: max1_k = 2^^15 = 32,768 symbols
- 1/4 <= CR < 1/2: max1_k = 2^^14 = 16,384 symbols

Additionally, a codec MAY impose other limitations on the maximum block size, for instance, because of a limited working memory size. This decision MUST be clarified at implementation time, when the target use-case is known. This results in a max2_k limitation.

Then, max_k is given by:

max_k = min(max1_k, max2_k)

Note that this calculation is only required at the encoder (sender), since the actual k parameter (k <= max_k) is communicated to the decoder (receiver) through the Explicit Source/Repair FEC Payload ID.

The source ADU flows usually have real-time constraints. It means that the maximum number of ADUs of an ADU block must not exceed a certain threshold since it directly impacts the decoding delay. It is the role of the developer, who knows the flow real-time features, to define an appropriate upper bound to the ADU block size, max_rt.

If we take into account these constraints, we find: max_B = min(max_k, max_rt). Then max_B gives an upper bound to the number of ADUs that can constitute an ADU block.

4.3. Source Block Creation

In its most general form the FECFRAME framework and the LDPC-Staircase FEC scheme are meant to protect a set of independent flows. Since the flows have no relationship to one another, the ADU size of each flow can potentially vary significantly. Even in the special case of a single flow, the ADU sizes can largely vary (e.g., the various frames of a "Group of Pictures (GOP) of an H.264 flow). This diversity must be addressed since the RS FEC scheme requires a constant encoding symbol size (E parameter) per source block. Since this specification requires that there is only one source symbol per ADU, E must be large enough to contain all the ADUs of an ADU block along with their prepended 3 bytes (see below).

In situations where E is determined per source block (default, specified by the FCCI/FSSI with S = 0, Section 5.1.1.2), E is equal to the size of the largest ADU of this source block plus three (for the prepended 3 bytes, see below). In this case, upon receiving the first FEC Repair Packet for this source block, since this packet MUST contain a single repair symbol (Section 5.1.3), a receiver determines the E parameter used for this source block.

In situations where E is fixed (specified by the FCCI/FSSI with S = 1, Section 5.1.1.2), then E must be greater or equal to the size of the largest ADU of this source block plus three (for the prepended 3 bytes, see below). If this is not the case, an error is returned. How to handle this error is use-case specific (e.g., a larger E parameter may be communicated to the receivers in an updated FFCI message, using an appropriate mechanism) and is not considered by this specification.

The ADU block is always encoded as a single source block. There are a total of B <= max_B ADUs in this ADU block. For the ADU i, with 0 <= i <= B-1, 3 bytes are prepended (Figure 2):

- The first byte, FID[i] (Flow ID), contains the integer identifier associated to the source ADU flow to which this ADU belongs to. It is assumed that a single byte is sufficient, or said differently, that no more than 256 flows will be protected by a single instance of the FECFRAME framework.
- The following two bytes, L[i] (Length), contain the length of this ADU, in network byte order (i.e., big endian). This length is for the ADU itself and does not include the FID[i], L[i], or Pad[i] fields.

Then zero padding is added to ADU i (if needed) in field Pad[i], for alignment purposes up to a size of exactly E bytes. The data unit resulting from the ADU i and the F[i], L[i] and Pad[i] fields, is called ADU Information (or ADUI). Each ADUI contributes to exactly one source symbol to the source block.

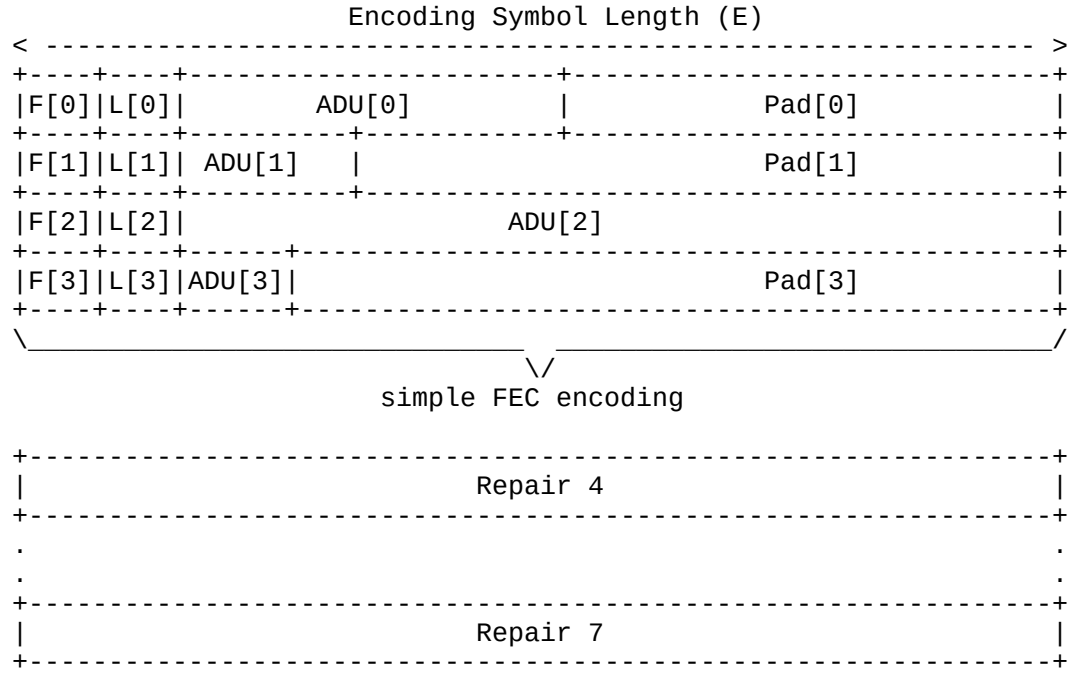

Figure 2: Source block creation, for code rate 1/2 (equal number of source and repair symbols, 4 in this example), and S = 0.

Note that neither the initial 3 bytes nor the optional padding are sent over the network. However, they are considered during FEC encoding. It means that a receiver who lost a certain FEC source packet (e.g., the UDP datagram containing this FEC source packet) will be able to recover the ADUI if FEC decoding succeeds. Thanks to the initial 3 bytes, this receiver will get rid of the padding (if any) and identify the corresponding ADU flow.

5. LDPC-Staircase FEC Scheme for Arbitrary ADU Flows

5.1. Formats and Codes

5.1.1. FEC Framework Configuration Information

The FEC Framework Configuration Information (or FFCI) includes information that MUST be communicated between the sender and receiver(s). More specifically, it enables the synchronization of the FECFRAME sender and receiver instances. It includes both mandatory elements and scheme-specific elements, as detailed below.

5.1.1.1. Mandatory Information

FEC Encoding ID: the value assigned to this fully-specified FEC scheme MUST be XXX, as assigned by IANA (Section 7).
When SDP is used to communicate the FFCI, this FEC Encoding ID is carried in the 'encoding-id' parameter.

5.1.1.2. FEC Scheme-Specific Information

The FEC Scheme Specific Information (FSSI) includes elements that are specific to the present FEC scheme. More precisely:
PRNG seed (seed): a non-negative 32 bit integer used as the seed of the Pseudo Random Number Generator, as defined in [RFC5170].
Encoding symbol length (E): a non-negative integer that indicates either the length of each encoding symbol in bytes (strict mode, i.e., if S = 1), or the maximum length of any encoding symbol (i.e., if S = 0).
Strict (S) flag: when set to 1 this flag indicates that the E parameter is valid for the whole session, unless otherwise notified. When set to 0 this flag indicates that the E parameter is only the maximum length of each encoding symbol, for the whole session, unless otherwise notified.
N1 minus 3 (n1m3): an integer between 0 (default) and 7, inclusive. The number of "1s" per column in the left side of the parity check matrix, N1, is then equal to N1m3 + 3, as specified in [RFC5170].
These elements are required both by the sender (LDPC-Staircase encoder) and the receiver(s) (LDPC-Staircase decoder).

When SDP is used to communicate the FFCI, this FEC scheme-specific information is carried in the 'fssi' parameter in textual representation as specified in [SDP_ELEMENTS]. For instance:

```
fssi = seed:1234,E:1400,S:0,n1m3:0
```

If another mechanism requires the FSSI to be carried as an opaque octet string (for instance after a Base64 encoding), the encoding format consists of the following 7 octets:
- PRNG seed (seed): 32 bit field.
- Encoding symbol length (E): 16 bit field.
- Strict (S) flag: 1 bit field.
- N1m3 parameter (n1m3): 7 bit field.

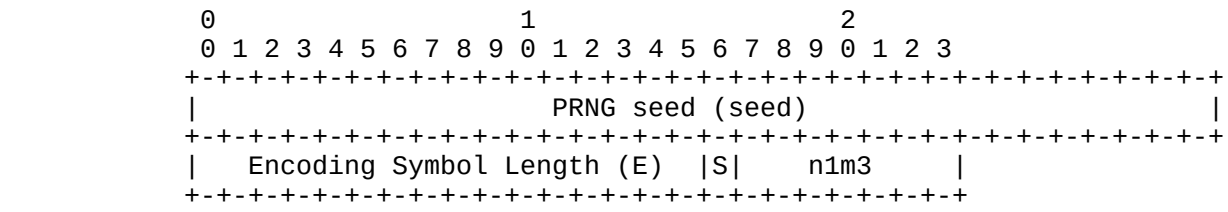

Figure 3: FSSI encoding format.

5.1.2. Explicit Source FEC Payload ID

A FEC source packet MUST contain an Explicit Source FEC Payload ID that is appended to the end of the packet as illustrated in Figure 4.

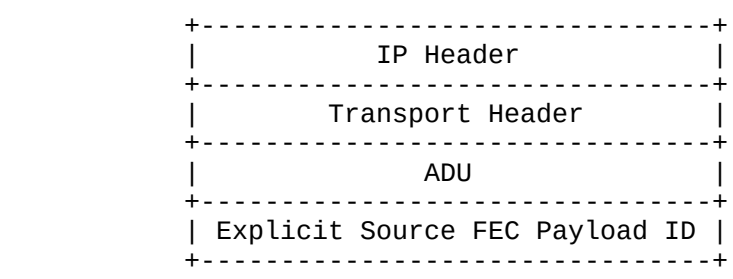

Figure 4: Structure of a FEC Source Packet with the Explicit Source FEC Payload ID.

More precisely, the Explicit Source FEC Payload ID is composed of the following fields (Figure 5):

Source Block Number (SBN) (16 bit field): this field identifies the source block to which this FEC source packet belongs.

Encoding Symbol ID (ESI) (16 bit field): this field identifies the source symbol contained in this FEC source packet. This value is such that 0 <= ESI <= k - 1 for source symbols.

Source Block Length (k) (16 bit field): this field provides the number of source symbols for this source block, i.e., the k parameter.

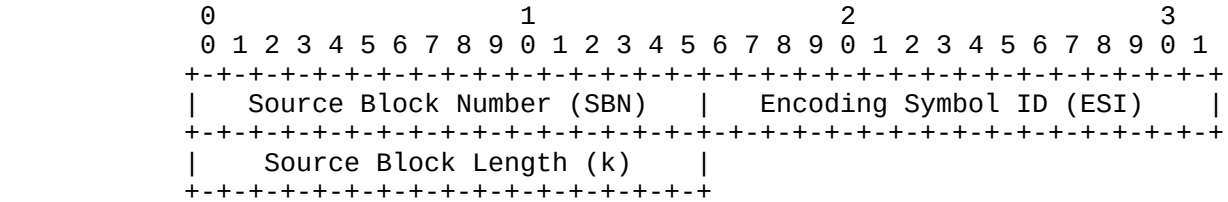

Figure 5: Source FEC Payload ID encoding format.

5.1.3. Repair FEC Payload ID

A FEC repair packet MUST contain a Repair FEC Payload ID that is prepended to the repair symbol(s) as illustrated in Figure 6. There MUST be a single repair symbol per FEC repair packet.

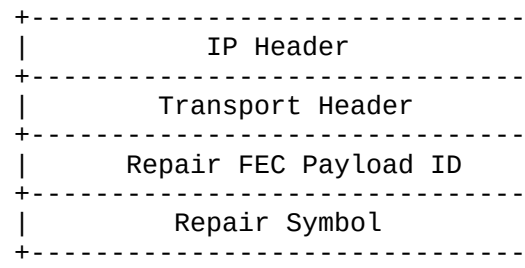

Figure 6: Structure of a FEC Repair Packet with the Repair FEC Payload ID.

More precisely, the Repair FEC Payload ID is composed of the following fields: (Figure 7):

Source Block Number (SBN) (16 bit field): this field identifies the source block to which the FEC repair packet belongs.

Encoding Symbol ID (ESI) (16 bit field) this field identifies the repair symbol contained in this FEC repair packet. This value is such that k <= ESI <= n - 1 for repair symbols.

Source Block Length (k) (16 bit field): this field provides the number of source symbols for this source block, i.e., the k parameter.

Number of Encoding Symbols (n) (16 bit field): this field provides the number of encoding symbols for this source block, i.e., the n parameter.

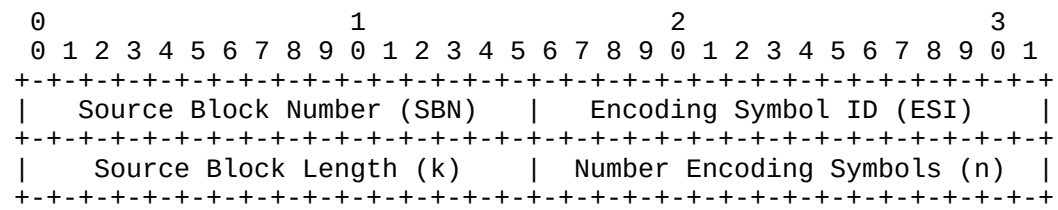

Figure 7: Repair FEC Payload ID encoding format.

5.2. Procedures

The following procedures apply:

- The source block creation procedures are specified in Section 4.3.
- The SBN value is incremented for each new source block, starting at 0 for the first block of the ADU flow. Wrapping to zero will happen for long sessions, after value 2^^16 - 1.
- The ESI of encoding symbols is managed sequentially, starting at 0 for the first symbol. The first k values (0 <= ESI <= k - 1) identify source symbols, whereas the last n-k values (k <= ESI <= n - 1) identify repair symbols.
- The FEC repair packet creation procedures are specified in Section 5.1.3.

5.3. FEC Code Specification

The present document inherits from [RFC5170] the specification of the core LDPC-Staircase codes for a packet erasure transmission channel.

Because of the requirement to have exactly one encoding symbol per group, i.e., because G MUST be equal to 1 (Section 4.1), several parts of [RFC5170] are useless. In particular, this is the case of Section 5.6. "Identifying the G Symbols of an Encoding Symbol Group".

6. Security Considerations

6.1. Problem Statement

A content delivery system is potentially subject to many attacks. Some of them target the network (e.g., to compromise the routing infrastructure, by compromising the congestion control component), others target the Content Delivery Protocol (CDP) (e.g., to compromise its normal behavior), and finally some attacks target the content itself. Since this document focuses on various FEC schemes, this section only discusses the additional threats that their use within the FECFRAME framework can create to an arbitrary CDP.

More specifically, these attacks may have several goals:
- those that are meant to give access to a confidential content (e.g., in case of a non-free content),
- those that try to corrupt the ADU Flows being transmitted (e.g., to prevent a receiver from using it),
- and those that try to compromise the receiver’s behavior (e.g., by making the decoding of an object computationally expensive).

These attacks can be launched either against the data flow itself (e.g., by sending forged FEC Source/Repair Packets) or against the FEC parameters that are sent either in-band (e.g., in the Repair FEC Payload ID) or out-of-band (e.g., in a session description).

6.2. Attacks Against the Data Flow

First of all, let us consider the attacks against the data flow.

6.2.1. Access to Confidential Contents

Access control to the ADU Flow being transmitted is typically provided by means of encryption. This encryption can be done within the content provider itself, by the application (for instance by using the Secure Real-time Transport Protocol (SRTP) [RFC3711]), or at the Network Layer, on a packet per packet basis when IPSec/ESP is used [RFC4303]. If confidentiality is a concern, it is RECOMMENDED that one of these solutions be used. Even if we mention these attacks here, they are not related nor facilitated by the use of FEC.

6.2.2. Content Corruption

Protection against corruptions (e.g., after sending forged FEC Source/Repair Packets) is achieved by means of a content integrity verification/sender authentication scheme. This service is usually provided at the packet level. In this case, after removing all forged packets, the ADU Flow may be sometimes recovered. Several techniques can provide this source authentication/content integrity service:

- at the application level, the Secure Real-time Transport Protocol (SRTP) [RFC3711] provides several solutions to authenticate the source and check the integrity of RTP and RTCP messages, among other services. For instance, associated to the Timed Efficient Stream Loss-Tolerant Authentication (TESLA) [RFC4383], SRTP is an attractive solution that is robust to losses, provides a true authentication/integrity service, and does not create any prohibitive processing load or transmission overhead. Yet, checking a packet requires a small delay (a second or more) after its reception with TESLA. Other building blocks can be used within SRTP to provide authentication/content integrity services.
- at the Network Layer, IPSec/ESP offers (among other services) an integrity verification mechanism that can be used to provide authentication/content integrity services.

It is up to the developer and the person in charge of deployment, who know the security requirements and features of the target application area, to define which solution is the most appropriate. Nonetheless it is RECOMMENDED that at least one of these techniques be used.

6.3. Attacks Against the FEC Parameters

Let us now consider attacks against the FEC parameters included in the FFCI that are usually sent out-of-band (e.g., in a session

description). Attacks on these FEC parameters can prevent the decoding of the associated object. For instance modifying the PRNG seed or N1m3 fields will lead a receiver to consider a different parity check matrix, i.e., a different code. Modifying the E parameter will lead a receiver to consider bad Repair Symbols for a received FEC Repair Packet.

It is therefore RECOMMENDED that security measures be taken to guarantee the FFCI integrity. When the FFCI is sent out-of-band in a session description, this latter SHOULD be protected, for instance by digitally signing it.

Attacks are also possible against some FEC parameters included in the Explicit Source FEC Payload ID and Repair FEC Payload ID. For instance modifying the Source Block Number of a FEC Source of Repair Packet will lead a receiver to assign this packet to a wrong block.

It is therefore RECOMMENDED that security measures be taken to guarantee the Explicit Source FEC Payload ID and Repair FEC Payload ID integrity. To that purpose, one of the packet-level source authentication/content integrity techniques of Section 6.2.2 can be used.

7. IANA Considerations

The FEC Encoding ID value is subject to IANA registration.

TBD

8. Acknowledgments

TBD

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119.

[RFC5170] Roca, V., Neumann, C., and D. Furodet, "Low Density Parity Check (LDPC) Forward Error Correction", RFC 5170, June 2008.

[FECFRAME-FRAMEWORK]

Watson, M., "Forward Error Correction (FEC) Framework", draft-ietf-fecframe-framework-10 (Work in Progress), September 2010.

[SDP_ELEMENTS]
Begen, A., "SDP Elements for FEC Framework", draft-ietf-fecframe-sdp-elements-10 (Work in Progress), October 2010.

9.2. Informative References

[RFC3453] Luby, M., Vicisano, L., Gemmell, J., Rizzo, L., Handley, M., and J. Crowcroft, "The Use of Forward Error Correction (FEC) in Reliable Multicast", RFC 3453, December 2002.

[RFC5052] Watson, M., Luby, M., and L. Vicisano, "Forward Error Correction (FEC) Building Block", RFC 5052, August 2007.

[RFC5510] Lacan, J., Roca, V., Peltotalo, J., and S. Peltotalo, "Reed-Solomon Forward Error Correction (FEC) Schemes", RFC 5510, April 2009.

[RFC5053] Luby, M., Shokrollahi, A., Watson, M., and T. Stockhammer, "Raptor Forward Error Correction Scheme", RFC 5053, June 2007.

[RFC5740] Adamson, B., Bormann, C., Handley, M., and J. Macker, "NACK-Oriented Reliable Multicast (NORM) Transport Protocol", RFC 5740, November 2009.

[RFC5775] Luby, M., Watson, M., and L. Vicisano, "Asynchronous Layered Coding (ALC) Protocol Instantiation", RFC 5775, April 2010.

[SPSC08] Cunche, M. and V. Roca, "Optimizing the Error Recovery Capabilities of LDPC-staircase Codes Featuring a Gaussian Elimination Decoding Scheme", 10th IEEE International Workshop on Signal Processing for Space Communications (SPSC’08), October 2008.

[CunchePHD10]
Cunche, M., "High performances AL-FEC codes for the erasure channel : variation around LDPC codes", PhD dissertation (in French) (http://tel.archives-ouvertes.fr/tel-00451336/en/), June 2010.

[LCN10] Matsuzono, K., Detchart, J., Cunche, M., Roca, V., and H.

Asaeda, "Performance Analysis of a High-Performance Real-Time Application with Several AL-FEC Schemes", 35th Annual IEEE Conference on Local Computer Networks 2010 (LCN 2010), October 2010.

[LDPC-codec] Cunche, M., Roca, V., Neumann, C., and J. Laboure, "LDPC-Staircase/LDPC-Triangle Codec Reference Implementation", INRIA Rhone-Alpes and STMicroelectronics, <http://planete-bcast.inrialpes.fr/>.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

[RFC4383] Baugher, M. and E. Carrara, "The Use of Timed Efficient Stream Loss-Tolerant Authentication (TESLA) in the Secure Real-time Transport Protocol (SRTP)", RFC 4383, February 2006.

Authors' Addresses

Vincent Roca
INRIA
655, av. de l'Europe
Inovallee; Montbonnot
ST ISMIER cedex 38334
France

Email: vincent.roca@inria.fr
URI: http://planete.inrialpes.fr/people/roca/

Mathieu Cunche
NICTA
Australia

Email: mathieu.cunche@nicta.com.au
URI: http://mathieu.cunche.free.fr/

Jerome Lacan
ISAE/LAAS-CNRS
1, place Emile Blouin
Toulouse 31056
France

Email: jerome.lacan@isae.fr
URI: http://dmi.ensica.fr/auteur.php3?id_auteur=5