FEC Framework A. Begen
Internet-Draft Cisco Systems
Intended status: Standards Track July 7, 2008
Expires: January 8, 2009

1-D Interleaved Parity FEC Scheme for FEC Framework
draft-begen-fecframe-interleaved-fec-scheme-00

Status of this Memo

Copyright Notice

Abstract

This document describes a Fully-Specified Forward Error Correction (FEC) Scheme for the one-dimensional (1-D) interleaved parity code and its application to reliable delivery of media streams in the context of FEC Framework. The 1-D interleaved parity code is a systematic code, where a number of repair symbols are generated from a set of source symbols and sent in one or more repair flows in addition to the source symbols that are sent to the receiver(s) within a source flow. The 1-D interleaved parity code offers a good

protection against bursty packet losses at a cost of decent complexity. This document extends the FEC header defined in RFC 2733 and registers a new RTP payload format for the FEC that is generated by the 1-D interleaved parity code from a source media encapsulated in RTP. This new payload format is compatible with and used as a part of the DVB Application-layer FEC Specification.

Table of Contents

1. Introduction

This document extends the FEC header defined in [RFC2733] and registers a new RTP payload format for the FEC that is generated by the 1-D interleaved parity code from a source media encapsulated in RTP [RFC3550]. The type of the protected source media can be audio, video, text or application. The FEC data is generated by an instance of the FEC Framework, which is configured by the FEC Framework Configuration Information. This configuration information, which is communicated through out-of-band means, plus the information contained in the payload format let the receiver(s) know the exact associations/relationships between the source and repair packets.

The 1-D interleaved parity FEC uses the exclusive OR (XOR) operation to generate the repair symbols. In a nutshell, the following steps take place:

- o The sender determines a set of source packets to be protected together based on the FEC Framework Configuration Information.

- o The sender applies the XOR operation on the source symbols to generate the required number of repair symbols.

- o The sender packetizes the repair symbols and sends the repair packet(s) along with the source packets to the receiver(s).

Per the FEC Framework requirements, the sender MUST transmit the source and repair packets in different source and repair flows, respectively. At the receiver side, if all of the source packets are successfully received, there is no need for FEC recovery and the repair packets are discarded. However, if there are missing source packets, the repair packets can be used to recover the missing information. Block diagrams for the systematic parity FEC encoder and decoder are sketched in Figure 1 and Figure 2, respectively.

```
                                 +------------+
    +--+  +--+  +--+  +--+ --> | Systematic | --> +--+  +--+  +--+  +--+
    +--+  +--+  +--+  +--+     | Parity FEC |     +--+  +--+  +--+  +--+
                                 |  Encoder   |
                                 |  (Sender)  | --> +==+  +==+
                                 +------------+     +==+  +==+

    Source Packet: +--+    Repair Packet: +==+
                   +--+                   +==+
```

Figure 1: Block diagram for systematic parity FEC encoder

```
                                 +------------+
    +--+    X    X    +--+ --> | Systematic | --> +--+  +--+  +--+  +--+
    +--+              +--+     | Parity FEC |     +--+  +--+  +--+  +--+
                                 |  Decoder   |
                +==+  +==+ --> | (Receiver) |
                +==+  +==+     +------------+

    Source Packet: +--+    Repair Packet: +==+    Lost Packet: X
                   +--+                   +==+
```

Figure 2: Block diagram for systematic parity FEC decoder

Suppose that we have a group of D x L source packets that have sequence numbers starting from 1 running to D x L. If we apply the XOR operation to the group of the source packets whose sequence numbers are L apart from each other as sketched in Figure 3, we generate L repair packets. This process is referred to as 1-D interleaved FEC protection, and the resulting L repair packets are referred to as interleaved (or column) FEC packets.

```
       +-------------+ +-------------+ +-------------+     +-------+
       | S_1         | | S_2         | | S3          | ... | S_L   |
       | S_L+1       | | S_L+2       | | S_L+3       | ... | S_2xL |
       | .           | | .           | |             |     |       |
       | .           | | .           | |             |     |       |
       | .           | | .           | |             |     |       |
       | S_(D-1)xL+1 | | S_(D-1)xL+2 | | S_(D-1)xL+3 | ... | S_DxL |
       +-------------+ +-------------+ +-------------+     +-------+
              +               +               +                +
        -------------   -------------   -------------       -------
       |     XOR     | |     XOR     | |     XOR     | ... |  XOR  |
        -------------   -------------   -------------       -------
              =               =               =                =
            +===+           +===+           +===+            +===+
            |C_1|           |C_2|           |C_3|      ...   |C_L|
            +===+           +===+           +===+            +===+
```

Figure 3: Generating interleaved (column) FEC packets

In Figure 3, S_n and C_m denote the source packet with a sequence number n and the interleaved (column) FEC packet with a sequence number m, respectively.

1.1. Use Cases

We generate one interleaved repair packet out of D non-consecutive source packets. This repair packet can provide a full recovery of the missing information if there is only one packet missing among the corresponding source packets. This implies that 1-D interleaved FEC protection performs well under bursty loss conditions provided that L is chosen large enough, i.e., L-packet duration SHOULD NOT be shorter than the duration of the burst that is intended to be repaired.

For example, consider the scenario depicted in Figure 4 where the sender generates interleaved FEC packets and a bursty loss hits the source packets. Since the number of columns is larger than the number of packets lost due to the bursty loss, the repair operation succeeds.

```
                         +---+
                         | 1 |    X      X      X
                         +---+

                         +---+  +---+  +---+  +---+
                         | 5 |  | 6 |  | 7 |  | 8 |
                         +---+  +---+  +---+  +---+

                         +---+  +---+  +---+  +---+
                         | 9 |  | 10|  | 11|  | 12|
                         +---+  +---+  +---+  +---+

                         +===+  +===+  +===+  +===+
                         |C_1|  |C_2|  |C_3|  |C_4|
                         +===+  +===+  +===+  +===+
```

Figure 4: Example scenario where 1-D interleaved FEC protection succeeds error recovery

The sender may generate interleaved FEC packets to combat with the bursty packet losses. However, two or more random packet losses may hit the source and repair packets in the same column. In that case, the repair operation fails. This is illustrated in Figure 5. Note that it is possible that two or more bursty losses may occur in the same source block, in which case interleaved FEC packets may still fail to recover the lost data.

```
                         +---+         +---+  +---+
                         | 1 |    X    | 3 |  | 4 |
                         +---+         +---+  +---+

                         +---+         +---+  +---+
                         | 5 |    X    | 7 |  | 8 |
                         +---+         +---+  +---+

                         +---+  +---+  +---+  +---+
                         | 9 |  | 10|  | 11|  | 12|
                         +---+  +---+  +---+  +---+

                         +===+  +===+  +===+  +===+
                         |C_1|  |C_2|  |C_3|  |C_4|
                         +===+  +===+  +===+  +===+
```

Figure 5: Example scenario where 1-D interleaved FEC protection fails error recovery

1.2. Overhead Computation

The overhead is defined as the ratio of the number of bytes belonging to the repair packets to the number of bytes belonging to the protected source packets.

Assuming that each repair packet carries an equal number of bytes carried by a source packet, we can compute the overhead as follows:

Overhead = 1/D

where D is the number of rows in the source block.

1.3. Relation to Existing Specifications

This section discusses the relation of the current specification to other existing specifications.

1.3.1. RFC 2733 and RFC 3009

The current specification extends the FEC header defined in [RFC2733] and registers a new RTP payload format. This new payload format is not backward compatible with the payload format that was registered by [RFC3009].

1.3.2. SMPTE 2022-1

In 2007, the Society of Motion Picture and Television Engineers (SMPTE) - Technology Committee N26 on File Management and Networking Technology - decided to revise the Pro-MPEG Code of Practice (CoP) #3 Release 2 specification, which (was initially produced by the Pro-MPEG Forum in 2004) discussed the several aspects of the transmission of MPEG-2 transport streams over IP networks. The new SMPTE specification is referred to as [SMPTE2022-1].

The Pro-MPEG CoP #3 r2 document was originally based on [RFC2733]. SMPTE revised the document by extending the FEC header (by setting the E bit) proposed in [RFC2733]. This extended header offers some improvements.

For example, instead of utilizing the bitmap field used in [RFC2733], [SMPTE2022-1] introduces separate fields to convey the number of rows (D) and columns (L) of the source block as well as the type of the repair packet (i.e., whether the repair packet is an interleaved FEC packet computed over a column or a non-interleaved FEC packet computed over a row). These fields plus the base sequence number allow the receiver side to establish the associations between the source and repair packets. Note that although the bitmap field is

not utilized, the FEC header of [SMPTE2022-1] inherently carries over the bitmap field from [RFC2733].

On the other hand, some parts of [SMPTE2022-1] are not in compliant with RTP [RFC3550]. For example, [SMPTE2022-1] sets the SSRC field to zero and does not use the timestamp field in the RTP headers of the repair packets (Receivers ignore the timestamps of the repair packets). Furthermore, [SMPTE2022-1] also sets the CC field in the RTP header to zero and does not allow any Contributing Source (CSRC) entry in the RTP header.

The current document adopts the extended FEC header of [SMPTE2022-1] and registers a new RTP payload format. At the same time, this document fixes the parts of [SMPTE2022-1] that are not in compliant with RTP [RFC3550].

1.3.3. ETSI TS 102 034

In 2007, the Digital Video Broadcasting (DVB) consortium published a technical specification [DVB-AL-FEC] through European Telecommunications Standards Institute (ETSI). This specification covers several areas related to the transmission of MPEG-2 transport stream-based services over IP networks.

The Annex E of [DVB-AL-FEC] defines an optional protocol for Application-layer FEC (AL-FEC) protection of streaming media for DVB-IP services carried over RTP [RFC3550] transport. AL-FEC protocol uses two layers for protection: a base layer that is produced by a packet-based interleaved parity code, and an enhancement layer that is produced by a Raptor code. While the use of the enhancement layer is optional, the use of the base layer is mandatory wherever AL-FEC is used.

The interleaved parity code that is used in the base layer is a subset of [SMPTE2022-1]. In particular, AL-FEC base layer uses the 1-D interleaved FEC protection only from [SMPTE2022-1]. The new RTP payload format that is defined and registered in this document is compatible with the AL-FEC base layer.

1.4. Document Outline

This FEC scheme specification follows the document structure defined in [I-D.ietf-fecframe-framework].

2. Requirements Notation

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].

3. Definitions, Notations and Abbreviations

The definitions, notations and abbreviations commonly used in this document are summarized in this section.

3.1. Definitions

This document uses the following definitions. For further definitions that apply to FEC Framework in general, see [I-D.ietf-fecframe-framework].

Source Flow: The packet flow(s) carrying the source data and to which FEC protection is to be applied.

Repair Flow: The packet flow(s) carrying the repair data.

Symbol: A unit of data. Its size, in bytes, is referred to as the symbol size.

Source Symbol: The smallest unit of data used during the encoding process.

Repair Symbol: Repair symbols are generated from the source symbols.

Source Packet: Data packets that contain only source symbols.

Repair Packet: Data packets that contain only repair symbols.

Source Block: A block of source symbols that are considered together in the encoding process.

FEC Framework Configuration Information: Information that controls the operation of the FEC Framework. Each FEC Framework instance has its own configuration information.

FEC Payload ID: Information that identifies the contents of a packet with respect to the FEC scheme.

Source FEC Payload ID: An FEC Payload ID specifically used with source packets.

Repair FEC Payload ID: An FEC Payload ID specifically used with repair packets.

3.2. Notations

- L: Number of columns of the source block.

- D: Number of rows of the source block.

3.3. Abbreviations

- XOR: Bitwise exclusive OR operation.
  0 XOR 0 = 0
  0 XOR 1 = 1
  1 XOR 0 = 1
  1 XOR 1 = 0

- FSSI: FEC-Scheme-Specific Information.

- SS-FSSI: Sender-Side FEC-Scheme-Specific Information.

- RS-FSSI: Receiver-Side FEC-Scheme-Specific Information.

4. Formats and Codes

This section defines the formats of the source and repair packets as well as the configuration information for the FEC scheme.

4.1. Source FEC Payload ID

The source packets MUST contain the information that identifies the source block and the position within the source block occupied by the packet. This information is referred to as the Source FEC Payload ID. In some cases, Source FEC Payload ID may be inferred from the fields already existing in the packet. In other cases, however, the required information is explicitly encoded into a specific field called Explicit Source FEC Payload ID, which is appended to the end of the source packets [I-D.ietf-fecframe-framework].

Since the source packets that are carried within an RTP stream already contain unique sequence numbers in their RTP headers [RFC3550], the Source FEC Payload ID can be derived in a straightforward manner. Thus, there is no need to use the Explicit Source FEC Payload ID field. The primary advantage of this approach is that the source packets are not modified in anyway. This provides backward compatibility for the receivers that do not support FEC at all. In multicast scenarios, this backward compatibility becomes

quite useful as it allows the non-FEC-capable receivers to receive and interpret the source packets.

The derivation of the Source FEC Payload ID from the RTP sequence number is discussed in Section 5.

Editor's note: This section should specify the additional requirements (if any) that are relevant to grouping multiple source flows together before applying FEC protection.

4.2. Repair FEC Payload ID

The repair packets MUST contain information that identifies the source block they pertain to and the relationship between the contained repair symbols and the original source block. This information is referred to as the Repair FEC Payload ID. This information MUST be encoded into a specific field between the transport header and the repair symbols within a repair packet, as shown in Figure 7 [I-D.ietf-fecframe-framework].

```
+------------------------------+
|          IP Header           |
+------------------------------+
|       Transport Header       |
+------------------------------+
|    Repair FEC Payload ID     |
+------------------------------+
|        Repair Symbols        |
+------------------------------+
```

Figure 7: Format of repair packets

Since the repair packets are carried within an RTP stream, the Repair FEC Payload ID consists of an RTP header and an FEC header. This is shown in Figure 8.

```
+------------------------------+
|          IP Header           |
+------------------------------+
|       Transport Header       |  ,--+------------------------------+
+------------------------------+-'   |         RTP Header           |
|    Repair FEC Payload ID     |     +------------------------------+
+------------------------------+-.   |         FEC Header           |
|        Repair Symbols        |  `--+------------------------------+
+------------------------------+
```

Figure 8: Format of Repair FEC Payload ID

The RTP header is formatted according to [RFC3550] with some further clarifications listed below:

- Version: The version field is set to 2.

- Padding (P) Bit: This bit is obtained by applying protection to the corresponding P bits from the RTP headers of the source packets protected by this repair packet.

- Extension (X) Bit: This bit is obtained by applying protection to the corresponding X bits from the RTP headers of the source packets protected by this repair packet. However, an RTP header extension is never present in a repair packet, independent of the value of the X bit.

- CSRC Count (CC): This field is obtained by applying protection to the corresponding CC values from the RTP headers of the source packets protected by this repair packet. However, a CSRC list is never present in a repair packet, independent of the value of the CC field.

- Marker (M) Bit: This bit is obtained by applying protection to the corresponding M bits from the RTP headers of the source packets protected by this repair packet..

- Payload Type: The payload type for the repair packets is determined through the payload format specified in the FEC Framework Configuration Information. Note that this document registers a new payload format for the repair packets (Refer to Section 10 for details). According to [RFC3550], an RTP receiver that cannot recognize a payload type must discard it. This provides backward compatibility. The FEC mechanisms can then be used in a multicast group with mixed FEC-capable and non-FEC-capable receivers. If a non-FEC-capable receiver receives a repair packet, it will not recognize the payload type, and hence, discards the repair packet.

o Sequence Number (SN): The sequence number has the standard definition. It MUST be one higher than the sequence number in the previously transmitted repair packet.

o Timestamp (TS): The timestamp MUST be set to the timestamp of the source packet whose sequence number is the lowest among the source packets protected by this repair packet.

o Synchronization Source (SSRC): The SSRC value SHALL be randomly assigned as suggested by [RFC3550]. This allows the sender to multiplex the source and repair flows on the same port, or multiplex multiple repair flows on a single port. The repair flows SHOULD use the RTCP CNAME field to associate themselves with the source flow. Note that due to the randomness of the SSRC assignments, there is a possibility of SSRC collision. In such cases, the collisions MUST be resolved as described in [RFC3550].

Note that the P bit, X bit, CC field and M bit of the source packets are protected by the corresponding bits/fields in the RTP header of the repair packet. On the other hand, the payload of a repair packet protects the concatenation of the CSRC list, RTP extension, payload and padding of the source RTP packets associated with this repair packet.

The FEC header is 16 octets. The format of the FEC header is shown in Figure 9.

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |           SN base low         |        Length recovery        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |E| PT recovery |                    Mask                       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                          TS recovery                          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |N|D|Type |Index|     Offset    |       NA      |  SN base ext  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 9: Format of the FEC header

The FEC header consists of the following fields:

o The SN base low field is used to indicate the lowest sequence number, taking wrap around into account, of those source packets protected by this repair packet.

- o The Length recovery field is used to determine the length of any recovered packets.

- o The E bit is the extension flag introduced in [RFC2733] and used to extend the [RFC2733] FEC header.

- o The PT recovery field is used to determine the payload type of the recovered packets.

- o The Mask field is not used.

- o The TS recovery field is used to determine the timestamp of the recovered packets.

- o The N bit is the extension flag that is reserved for future uses.

- o The D bit is not used.

- o The Type field indicates the type of the error-correcting code used. This document defines only one error-correcting code.

- o The Index field is not used.

- o The Offset and NA fields are used to indicate the number of columns (L) and rows (D) of the source block, respectively.

- o The SN base ext field is not used.

The details on setting the fields in the FEC header are provided in Section 6.2.

It should be noted that a mask-based approach (similar to the one specified in [RFC2733]) may not be very efficient to indicate which source packets in the current source block are associated with a given repair packet. In particular, for the applications that would like to use large source block sizes, the size of the mask that is required to describe the source-repair packet associations may be prohibitively large. Instead, a systematic approach is inherently more efficient.

4.3. FEC Framework Configuration Information

The FEC Framework defines a minimum set of information that MUST be communicated between the sender and receiver(s) for a proper operation of the FEC scheme. This information is called the FEC Framework Configuration Information. This information specifies how the sender applies protection to the source flow(s) and how the repair flow(s) can be used to recover the lost data. In other words,

this information specifies the relationship(s) between the source and repair flows. The FEC Framework requires every FEC Framework instance to provide its own configuration information.

From the FEC scheme point of view, the FEC Framework Configuration Information consists of mandatory and scheme-specific elements. We describe these elements below.

4.3.1. Mandatory Elements

- o FEC Encoding ID: The value of the FEC Encoding ID for the fully-specified FEC scheme defined in this document MUST be TBD as assigned by IANA. Refer to Section 10.

4.3.2. Scheme-Specific Elements

FEC-Scheme-Specific Information (FSSI) includes the information that is specific to the FEC scheme used by the Content Delivery Protocol. FSSI is used to communicate the information that cannot be adequately represented otherwise and is essential for proper FEC encoding and decoding operations.

The FSSI is carried in two opaque containers. The first container contains the FSSI required only by the sender. This information is referred to as the Sender-Side FEC-Scheme-Specific Information (SS-FSSI). Rest of the FSSI is referred to as the Receiver-Side FEC-Scheme-Specific Information (RS-FSSI) and carried in the second container.

The following parameters are carried in the FEC Scheme-Specific Information element:

- o L: Number of columns of the source block. L is a positive integer.

- o D: Number of rows of the source block. D is a positive integer.

All of the parameters listed above MUST be included in the FSSI. The parameters L and D are carried within the SS-FSSI container.

4.3.3. Encoding Format

TBC.

5. Procedures

This section describes the procedures that are specific to the 1-D

interleaved parity FEC scheme.

5.1. Configuration Information Signaling Procedures

This specification makes use of the signaling protocol to signal the FEC Framework Configuration Information between the sender and receiver(s). This enables the sender and receiver(s) to be in sync with respect to the information needed for the operation of FEC Framework.

5.2. Content Delivery Protocol Requirements

Content Delivery Protocol (CDP) is a complete application-protocol specification that provides FEC capabilities by making use of the FEC Schemes through the use of FEC Framework defined in [I-D.ietf-fecframe-framework].

The parity FEC encoder and decoder require the following from the CDP:

- o The size of the source block, namely the number of columns (L) and the number of rows (D).

This information is transmitted to the receiver side by the CDP through the FEC Framework Configuration Information. The parity encoder additionally requires:

- o The data to be protected.

The parity encoder provides the following information to the CDP:

- o An interleaved (column) FEC packet that is generated by applying protection over each column in the current source block.

The source packets as well as the repair packets are then transmitted to the receiver(s) by the transport protocol chosen by the CDP.

5.3. Determination of Source Block Size and Repair Window

TBC.

Editor's note: This section should discuss the derivation of the Source FEC Payload ID from the RTP sequence number.

6. 1-D Interleaved Parity FEC Code Specification

This section provides a complete specification of the 1-D interleaved

parity FEC scheme.

6.1. Overview

The following sections specify the steps involved in generating the repair packets and reconstructing the missing source packets from the repair packets.

6.2. Repair Packet Construction

The Repair FEC Payload ID consists of an RTP header and an FEC header. The RTP header of an repair packet is formed based on the guidelines given in Section 4.2.

The FEC header includes 16 octets. It is constructed by applying the XOR operation on the bit strings that are generated from the individual source packets protected by this particular repair packet. The set of the source packets that are associated with a given repair packet can be computed by the formula given in Section 6.3.1.

The bit string is formed for each source packet by concatenating the following fields together in the order specified:

- o Padding bit (1 bit) (This is the most significant bit of the bit string)
- o Extension bit (1 bit)
- o CC field (4 bits)
- o Marker bit (1 bit)
- o PT field (7 bits)
- o Timestamp (32 bits)
- o Unsigned network-ordered 16-bit representation of the source packet length in bytes minus 12 (for the fixed RTP header), i.e., the sum of the lengths of all the following if present: the CSRC list, header extension, RTP payload, and RTP padding (16 bits)
- o If CC is nonzero, the CSRC list (variable length)
- o If X is 1, the header extension (variable length)
- o Payload (variable length)

- Padding, if present (variable length)

Note that if the payload lengths of the source packets are not equal, each shorter packet MUST be padded to the length of the longest packet by adding octet 0's at the end. Due to this possible padding and mandatory FEC header, a repair packet usually has a larger size than the source packets it protects. This may cause problems if the resulting repair packet size exceeds the Maximum Transmission Unit (MTU) size of the path over which the repair flow is sent.

By applying the parity operation on the bit strings produced from the source packets, we generate the FEC bit string. Some parts of the RTP header and the FEC header of the repair packet are generated from the FEC bit string as follows:

- The first (most significant) bit in the FEC bit string is written into the Padding bit in the RTP header of the repair packet.

- The next bit in the FEC bit string is written into the Extension bit in the RTP header of the repair packet.

- The next 4 bits of the FEC bit string are written into the CC field in the RTP header of the repair packet.

- The next bit of the FEC bit string is written into the Marker bit in the RTP header of the repair packet.

- The next 7 bits of the FEC bit string are written into the PT recovery field in the FEC header.

- The next 32 bits of the FEC bit string are written into the TS recovery field in the FEC header.

- The next 16 bits are written into the Length recovery field in the FEC header. This allows the FEC procedure to be applied even when the lengths of the protected source packets are not identical.

- The remaining bits are set to be the payload of the repair packet.

The remaining parts of the FEC header are set as follows:

- The SN base low field MUST be set to the lowest sequence number, taking wrap around into account, of those source packets protected by this repair packet.

- The E bit MUST be set to 1 to extend the [RFC2733] FEC header.

- The Mask field SHALL be set to 0 and ignored by the receiver.
- The N bit SHALL be set to 0 and ignored by the receiver.
- The D bit SHALL be set to 0 and ignored by the receiver.
- The Type field MUST be set to 0.
- The Index field SHALL be set to 0 and ignored by the receiver.
- The Offset field MUST be set to the number of columns of the source block (L).
- The NA field MUST be set to the number of rows of the source block (D).
- The SN base ext field SHALL be set to 0 and ignored by the receiver.

6.3. Source Packet Reconstruction

This section describes the recovery procedures that are required to reconstruct the missing packets. The recovery process has two steps. In the first step, the FEC decoder determines which source and repair packets should be used in order to recover a missing packet. In the second step, the decoder recovers the missing packet, which consists of an RTP header and RTP payload.

In the following, we describe the RECOMMENDED algorithms for the first and second steps. Based on the implementation, different algorithms MAY be adopted. However, the end result MUST be identical to the one produced by the following algorithms.

6.3.1. Associating the Source and Repair Packets

The first step is associating the source and repair packets. The SN base low field in the FEC header shows the lowest sequence number of the source packets that form the particular column. In addition, the information of how many source packets are available in each column and row is available from the FEC Framework Configuration Information. This set of information uniquely identifies all of the source packets associated with a given repair packet.

Mathematically, for any received repair packet, p*, we can determine the sequence numbers of the source packets that are protected by this repair packet as follows:

p*_snb + i * L

where p*_snb denotes the value in the SN base low field of p*'s FEC header, L is the number of columns of the source block (conveyed in the Offset field) and

0 <= i < D

where D is the number of rows of the source block (conveyed in the NA field).

We denote the set of the source packets associated with repair packet p* by set T(p*). Note that in a source block whose size is L columns by D rows, set T includes D source packets. Recall that 1-D interleaved FEC protection can fully recover the missing information if there is only source packet is missing in set T. If the repair packet that protects the source packets in set T is missing, or the repair packet is available but two or more source packets are missing, then missing source packets in set T cannot be recovered by 1-D interleaved FEC protection.

6.3.2. Recovering the RTP Header and Payload

For a given set T, the procedure for the recovery of the RTP header of the missing packet, whose sequence number is denoted by SEQNUM, is as follows:

1. For each of the source packets that are successfully received in set T, compute the bit string as described in Section 6.2.

2. For the repair packet associated with set T, compute the bit string in the same fashion except use the PT recovery field instead of the PT field and TS recovery field instead of the Timestamp field, and set the CSRC list, header extension, and padding to null regardless of the values of the X bit and CC field.

3. If any of the bit strings generated from the source packets are shorter than the bit string generated from the repair packet, pad them to be the same length as the bit string generated from the repair packet. For padding, the padding of octet 0 MUST be added at the end of the bit string.

4. Calculate the recovered bit string as the XOR of the bit strings generated from all source packets in set T and the FEC bit string generated from the repair packet associated with set T.

5. Create a new packet with the standard 12-byte RTP header and no payload.

6. Set the version of the new packet to 2.

7. Set the Padding bit in the new packet to the first bit in the recovered bit string.

8. Set the Extension bit in the new packet to the next bit in the recovered bit string.

9. Set the CC field to the next 4 bits in the recovered bit string.

10. Set the Marker bit in the new packet to the next bit in the recovered bit string.

11. Set the Payload type in the new packet to the next 7 bits in the recovered bit string.

12. Set the SN field in the new packet to SEQNUM.

13. Set the TS field in the new packet to the next 32 bits in the recovered bit string.

14. Take the next 16 bits of the recovered bit string and set Y to whatever unsigned integer this represents (assuming network-order). Take Y bytes from the recovered bit string and append them to the new packet. Y represents the length of the new packet in bytes minus 12 (for the fixed RTP header), i.e., the sum of the lengths of all the following if present: the CSRC list, header extension, RTP payload and RTP padding.

15. Set the SSRC of the new packet to the SSRC of the source RTP stream.

This procedure completely recovers both the header and payload of an RTP packet.

7. Session Description Protocol (SDP) Signaling

This section provides an SDP [RFC4566] example. The following example uses the SDP elements for FEC Framework, which were introduced in [I-D.ietf-fecframe-sdp-elements], and the FEC grouping semantics [RFC4756].

Editor's note: No FEC Encoding ID has been registered with IANA for the FEC scheme proposed in this document. In the example below, an

FEC Encoding ID of zero will be used.

In this example, we have one source video stream (mid:S1) and one FEC repair stream (mid:R1). We form one FEC group with the "a=group:FEC S1 R1" line. The source and repair streams are sent to the same port on different multicast groups. The repair window is set to 200 ms.

```
v=0
o=ali 1122334455 1122334466 IN IP4 fec.rocks.com
s=Interleaved Parity FEC Example
t=0 0
a=group:FEC S1 R1
m=video 30000 RTP/AVP 100
c=IN IP4 224.1.1.1/127
a=rtpmap:100 MP2T/90000
a=fec-source-flow: id=0
a=mid:S1
m=application 30000 RTP/AVP 110
c=IN IP4 224.1.2.1/127
a=rtpmap:110 1d-interleaved-parityfec/90000
a=fec-repair-flow: encoding-id=0; ss-fssi=L:5 D:10
a=repair-window: 200
a=mid:R1
```

8. Congestion Control Considerations

For the general congestion control considerations related to the use of FEC, refer to [I-D.ietf-fecframe-framework].

9. Security Considerations

For the general security considerations related to the use of FEC, refer to [I-D.ietf-fecframe-framework].

10. IANA Considerations

10.1. Registration of FEC Encoding ID

The value of FEC Encoding ID is subject to IANA registration. For general guidelines on IANA considerations as they apply to this document, refer to [I-D.ietf-fecframe-framework].

This document assigns the Fully-Specified FEC Encoding ID TBD under the ietf:fecframe:fec:encoding name-space to "1-D Interleaved Parity FEC Code."

10.2. Registration of audio/1d-interleaved-parityfec

TBC.

10.3. Registration of video/1d-interleaved-parityfec

TBC.

10.4. Registration of text/1d-interleaved-parityfec

TBC.

10.5. Registration of application/1d-interleaved-parityfec

TBC.

11. Acknowledgments

A major part of this document is borrowed from [RFC2733] and [SMPTE2022-1]. Thus, the author would like to thank the authors and editors of these earlier specifications.

12. References

12.1. Normative References

[I-D.ietf-fecframe-framework]
Watson, M., "Forward Error Correction (FEC) Framework", draft-ietf-fecframe-framework-01 (work in progress), November 2007.

[I-D.ietf-fecframe-sdp-elements]
Begen, A., "SDP Elements for FEC Framework", draft-ietf-fecframe-sdp-elements-00 (work in progress), February 2008.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

[RFC4756] Li, A., "Forward Error Correction Grouping Semantics in Session Description Protocol", RFC 4756, November 2006.

12.2. Informative References

[RFC2733] Rosenberg, J. and H. Schulzrinne, "An RTP Payload Format for Generic Forward Error Correction", RFC 2733, December 1999.

[RFC3009] Rosenberg, J. and H. Schulzrinne, "Registration of parityfec MIME types", RFC 3009, November 2000.

[DVB-AL-FEC] DVB Document A086 Rev. 4 (ETSI TS 102 034 V1.3.1), "Transport of MPEG 2 Transport Stream (TS) Based DVB Services over IP Based Networks", March 2007.

[SMPTE2022-1] SMPTE 2022-1-2007, "Forward Error Correction for Real-Time Video/Audio Transport over IP Networks", 2007.

Author's Address

Ali Begen
Cisco Systems
170 West Tasman Drive
San Jose, CA 95134
USA

Email: abegen@cisco.com

Full Copyright Statement

Intellectual Property

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).