Network Working Group
Request for Comments: 5445
Obsoletes: 3452, 3695
Category: Standards Track

M. Watson
Digital Fountain

March 2009

Basic Forward Error Correction (FEC) Schemes

Status of This Memo

Copyright Notice

Abstract

This document provides Forward Error Correction (FEC) Scheme specifications according to the Reliable Multicast Transport (RMT) FEC building block for the Compact No-Code FEC Scheme, the Small Block, Large Block, and Expandable FEC Scheme, the Small Block Systematic FEC Scheme, and the Compact FEC Scheme. This document obsoletes RFC 3695 and assumes responsibility for the FEC Schemes defined in RFC 3452.

Table of Contents

1. Introduction

The document specifies the following FEC Schemes according to the specification requirements of the FEC building block [RFC5052]:

- Compact No-Code FEC Scheme
- Small Block, Large Block, and Expandable FEC Scheme
- Small Block Systematic FEC Scheme
- Compact FEC Scheme

This document inherits the context, language, declarations and restrictions of the FEC building block [RFC5052]. This document also uses the terminology of the companion document [RFC3453], which describes the use of FEC codes within the context of reliable IP multicast transport and provides an introduction to some commonly used FEC codes.

Building blocks are defined in [RFC3048]. This document follows the general guidelines provided in [RFC3269].

[RFC3452] and [RFC3695] contain previous versions of the FEC Schemes defined in this specification. These RFCs were published in the "Experimental" category. It was the stated intent of the RMT working group to re-submit these specifications as an IETF Proposed Standard in due course. This document obsoletes [RFC3695]. [RFC3452] has already been obsoleted by [RFC5052], and this document assumes responsibility for aspects of [RFC3452] that were not included in [RFC5052].

This Proposed Standard specification is thus based on and backwards compatible with the FEC Schemes defined in [RFC3452] and [RFC3695], updated according to accumulated experience and growing protocol maturity since their original publication. Said experience applies both to this specification itself and to congestion control strategies related to the use of this specification.

The differences between the FEC Scheme specifications in [RFC3452] and [RFC3695] and this document are listed in Section 10.

Integer fields specified in this document are all encoded in network byte order.

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. Compact No-Code FEC Scheme

3.1. Introduction

The Compact No-code FEC Scheme is a Fully-Specified FEC Scheme. The scheme requires no FEC coding and is specified primarily to allow simple interoperability testing between different implementations of protocol instantiations that use the FEC building block.

3.2. Formats and Codes

3.2.1. FEC Payload ID(s)

The FEC Payload ID for the Compact No-Code FEC Scheme is composed of a Source Block Number and an Encoding Symbol ID as shown in Figure 1.

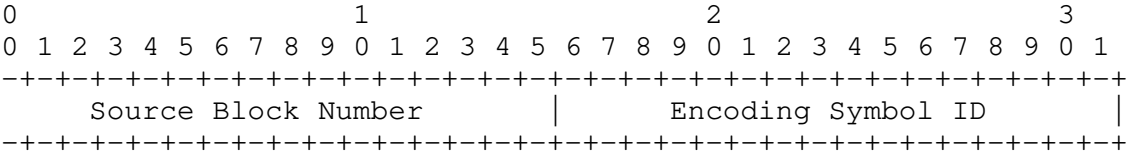

Figure 1: FEC Payload ID Format for Compact No-Code FEC Scheme

The Source Block Number (SBN) is a 16-bit unsigned integer that is used to identify from which source block of the object the encoding symbol in the payload of the packet is generated. There are two possible modes: in the unique SBN mode, each source block within the object has a unique Source Block Number associated with it, and in the non-unique SBN mode, the same Source Block Number may be used for more than one source block within the object. Which mode is being used for an object is outside the scope of this document and MUST be communicated, either explicitly or implicitly, out-of-band to receivers.

If the unique SBN mode is used, then successive Source Block Numbers are associated with consecutive source blocks of the object starting with Source Block Number 0 for the first source block of the object. In this case, there are at most 2^^16 source blocks in the object.

If the non-unique SBN mode is used, then the mapping from source blocks to Source Block Numbers MUST be communicated out-of-band to receivers, and how this is done is outside the scope of this document. This mapping could be implicit, for example, determined by the transmission order of the source blocks. In non-unique SBN mode, packets for two different source blocks mapped to the same Source Block Number SHOULD NOT be sent within an interval of time that is shorter than the transport time of a source block. The transport time of a source block includes the amount of time needed to process the source block at the sender transport layer, the network transit time for packets, and the amount of time needed to process the source block at the receiver transport. This allows the receiver to clearly decide which packets belong to which source block.

The Encoding Symbol ID is a 16-bit unsigned integer that identifies which specific encoding symbol generated from the source block is carried in the packet payload. The exact details of the correspondence between Encoding Symbol IDs and the encoding symbols in the packet payload are specified in Section 3.4.

3.2.2. FEC Object Transmission Information

3.2.2.1. Mandatory

The mandatory FEC Object Transmission Information element for the Compact No-Code FEC Scheme is:

- FEC Encoding ID: zero (0)

3.2.2.2. Common

The Common FEC Object Transmission Information elements and their value ranges for the Compact No-Code FEC Scheme are:

Transfer-Length: a non-negative integer, less than 2^^48, indicating the length of the object in octets.

Encoding-Symbol-Length: a non-negative integer, less than 2^^16, indicating the length of each encoding symbol in octets.

Maximum-Source-Block-Length: a non-negative integer, less than 2^^32, indicating the maximum number of source symbols in a source block.

The encoded Common FEC Object Transmission Information is defined in Figure 2.

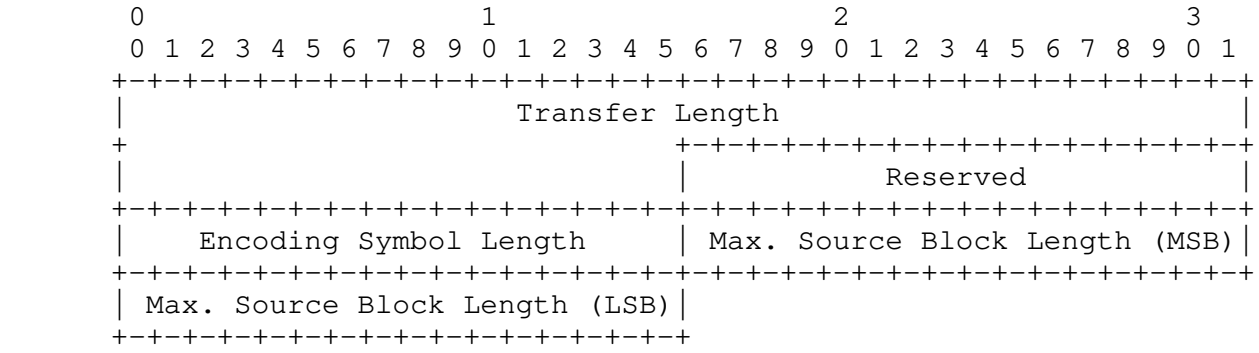

Figure 2: Encoded Common FEC Object Transmission Information (OTI) for Compact No-Code FEC Scheme

The Transfer Length, Encoding Symbol Length, and Maximum Source Block Length are encoded as unsigned integers, of length 48 bits, 16 bits, and 32 bits, respectively.

All Encoding Symbols of a transport object MUST have length equal to the length specified in the Encoding Symbol Length element, with the optional exception of the last source symbol of the last source block (so that redundant padding is not mandatory in this last symbol). This last source symbol MUST be logically padded out with zeroes when another Encoding Symbol is computed based on this source symbol to ensure the same interpretation of this Encoding Symbol value by the sender and receiver. However, this padding does not actually need to be sent with the data of the last source symbol.

The "Reserved" field in the Encoded FEC Object Transmission Information MUST be set to zero by senders and its value MUST be ignored by receivers.

> Note: this FEC Scheme was first defined in [RFC3695], which did not require that the Encoding Symbol Length should be the same for every source block. This document introduces a general requirement that the Encoding Symbol Length be the same across source blocks. Since no protocols were defined that support variation in the Encoding Symbol Length between source blocks, this can be done without introducing backwards compatibility issues.

3.2.2.3. Scheme-Specific

No Scheme-Specific FEC Object Transmission Information elements are defined by this FEC Scheme.

3.3. Procedures

The algorithm defined in Section 9.1. of [RFC5052] MUST be used to partition the file into source blocks.

3.4. FEC Code Specification

The Compact No-Code FEC Scheme does not require FEC encoding or decoding. Instead, each encoding symbol consists of consecutive bytes of a source block of the object.

The following two subsections describe the details of how the Compact No-Code FEC Scheme operates for each source block of an object.

3.4.1. Source Block Logistics

Let X > 0 be the length of a source block in bytes. Let L > 0 be the length of the encoding symbol contained in the payload of each packet. The value of X and L are part of the FEC Object Transmission Information, and how this information is communicated to a receiver is outside the scope of this document.

For a given source block X bytes in length with Source Block Number I, let N = X/L rounded up to the nearest integer. The encoding symbol carried in the payload of a packet consists of a consecutive portion of the source block. The source block is logically partitioned into N encoding symbols, each L bytes in length, and the corresponding Encoding Symbol IDs range from 0 through N-1 starting at the beginning of the source block and proceeding to the end. Thus, the encoding symbol with Encoding Symbol ID Y consists of bytes L*Y through L*(Y+1)-1 of the source block, where the bytes of the source block are numbered from 0 through X-1. If X/L is not integral then the last encoding symbol with Encoding Symbol ID = N-1 consists of bytes L*(N-1) through the last byte X-1 of the source block, and the remaining L*N - X bytes of the encoding symbol can by padded out with zeroes.

As an example, suppose that the source block length X = 20,400 and encoding symbol length L = 1,000. The encoding symbol with Encoding Symbol ID = 10 contains bytes 10,000 through 10,999 of the source block, and the encoding symbol with Encoding Symbol ID = 20 contains bytes 20,000 through the last byte 20,399 of the source block and the remaining 600 bytes of the encoding symbol can be padded with zeroes.

There are no restrictions beyond the rules stated above on how a sender generates encoding symbols to send from a source block. However, it is recommended that an implementor refer to the companion document [RFC3452] for general advice.

In the next subsection, a procedure is recommended for sending and receiving source blocks.

3.4.2. Sending and Receiving a Source Block

The following carousel procedure is RECOMMENDED for a sender to generate packets containing FEC Payload IDs and corresponding encoding symbols for a source block with Source Block Number I. Set the length in bytes of an encoding symbol to a fixed value L, which is reasonable for a packet payload (e.g., ensure that the total packet size does not exceed the MTU) and that is smaller than the source block length X, e.g., L = 1,000 for X >= 1,000. Initialize Y to a value randomly chosen in the interval [0..N-1]. Repeat the following for each packet of the source block to be sent.

- If Y <= N-1, then generate the encoding symbol Y.
- Within the FEC Payload ID, set the Source Block Length to X, set the Source Block Number = I, set the Encoding Symbol ID = Y, place the FEC Payload ID and the encoding symbol into the packet to send.
- In preparation for the generation of the next packet: if Y < N-1 then increment Y by one else if Y = N-1 then reset Y to zero.

The following procedure is RECOMMENDED for a receiver to recover the source block based on receiving packets for the source block from a sender that is using the carousel procedure described above. The receiver can determine from which source block a received packet was generated by the Source Block Number carried in the FEC Payload ID. Upon receipt of the first FEC Payload ID for a source block, the receiver uses the Source Block Length and Encoding Symbol Length received out-of-band as part of the FEC Object Transmission Information to determine the length X in bytes of the source block and length L in bytes of each encoding symbol. The receiver allocates space for the X bytes that the source block requires. The receiver also computes the length of the encoding symbol in the payload of the packet by subtracting the packet header length from the total length of the received packet. The receiver checks that this symbol length is equal to L, except in the case that this is the last symbol of the source block in which case the symbol length in the packet may be less than L. After calculating N = X/L rounded up to the nearest integer, the receiver allocates a boolean array RECEIVED[0..N-1] with all N entries initialized to false to track received encoding symbols. The receiver keeps receiving packets for the source block as long as there is at least one entry in RECEIVED still set to false or until the application decides to give up on this source block and move on to other source blocks. For each

received packet for the source block (including the first packet), the steps to be taken to help recover the source block are as follows. Let Y be the value of the Encoding Symbol ID within the FEC Payload ID of the packet. If Y <= N-1, then the receiver copies the encoding symbol into the appropriate place within the space reserved for the source block and sets RECEIVED[Y] = true. If all N entries of RECEIVED are true, then the receiver has recovered the entire source block.

4. Small Block, Large Block, and Expandable FEC Scheme

4.1. Introduction

This section defines an Under-Specified FEC Scheme for Small Block FEC codes, Large Block FEC codes, and Expandable FEC codes as described in [RFC3453].

4.2. Formats and Codes

4.2.1. FEC Payload ID(s)

The FEC Payload ID is composed of a Source Block Number and an Encoding Symbol ID structured as shown in Figure 3.

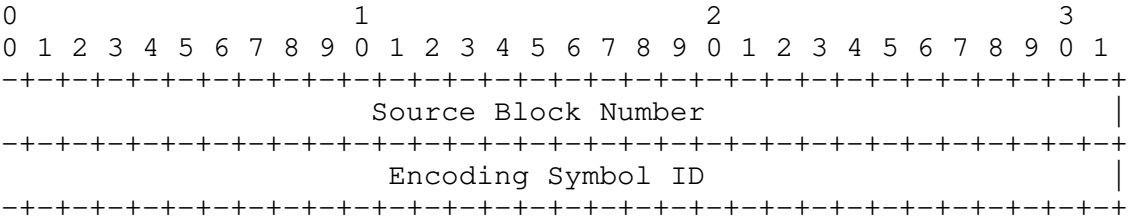

Figure 3: FEC Payload ID Format for Small Block, Large Block, and Expandable FEC Codes

The Source Block Number is a 32-bit unsigned integer that identifies from which source block of the object the encoding symbol(s) in the payload are generated. These blocks are numbered consecutively from 0 to N-1, where N is the number of source blocks in the object.

The Encoding Symbol ID is a 32-bit unsigned integer that identifies which specific encoding symbol(s) generated from the source block are carried in the packet payload. The exact details of the correspondence between Encoding Symbol IDs and the encoding symbol(s) in the packet payload are dependent on the particular FEC Scheme instance used as identified by the FEC Encoding ID and by the FEC Instance ID, and these details may be proprietary.

4.2.2. FEC Object Transmission Information

4.2.2.1. Mandatory

The mandatory FEC Object Transmission Information element for the Small Block, Large Block, and Expandable FEC Scheme are:

- FEC Encoding ID: 128

4.2.2.2. Common

The Common FEC Object Transmission Information elements and their value ranges for the Small Block, Large Block, and Expandable FEC Scheme are:

FEC Instance ID: a non-negative integer less than 2^^16.

Transfer-Length: a non-negative integer less than 2^^48, indicating the length of the object in octets.

Encoding-Symbol-Length: a non-negative integer less than 2^^16, indicating the length of each encoding symbol in octets.

Maximum-Source-Block-Length: a non-negative integer less than 2^^32, indicating the maximum number of source symbols in a source block.

The encoded Common FEC Object Transmission Information is defined in Figure 4.

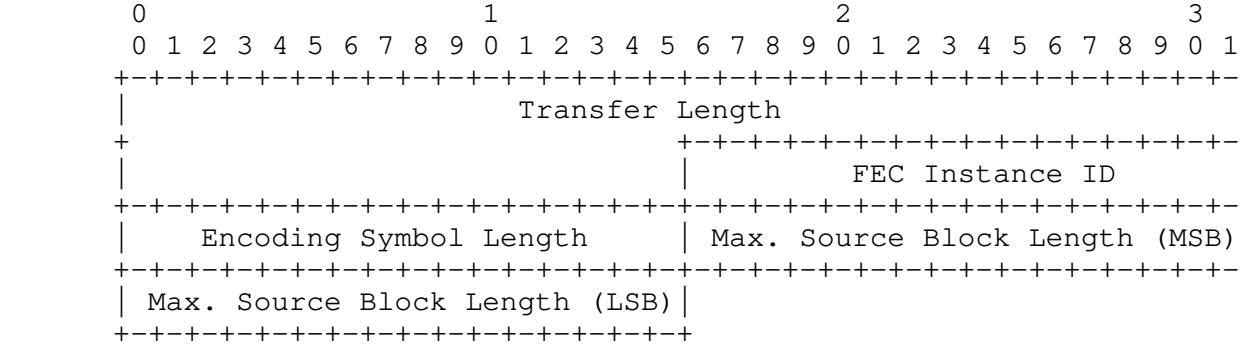

Figure 4: Encoded Common FEC OTI for Small Block, Large Block, and Expandable FEC Scheme

The Transfer Length (48 bits), FEC Instance ID (16 bits), Encoding Symbol Length (16 bits), and Maximum Source Block Length (32 bits) are encoded as unsigned integers.

4.2.2.3. Scheme-Specific

The Scheme-Specific FEC Object Transmission Information field for the Small Block, Large Block, and Expandable FEC Scheme provides for the possibility of Instance-specific FEC Object Transmission Information. The format of the Scheme-Specific FEC Object Transmission Information for this FEC Scheme is defined in Figure 5.

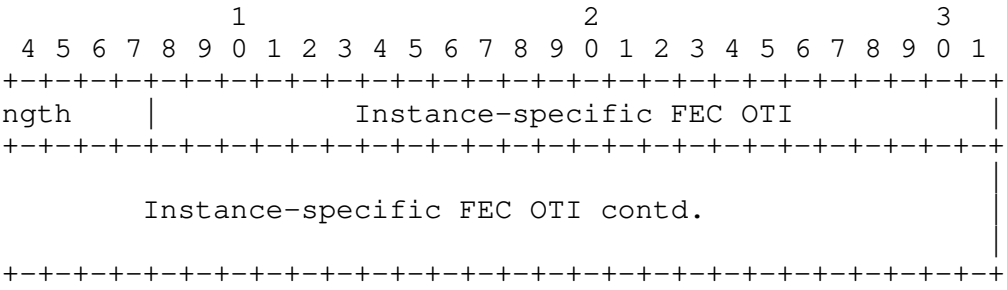

Figure 5: Encoded Scheme-Specific FEC OTI for Small Block, Large Block, and Expandable FEC Scheme

The Scheme-Specific FEC Object Transmission Information field contains the following sub-fields:

Length (1 octet): an unsigned integer that specifies the length of the Scheme-Specific FEC OTI in four-octet words (including this length field), except that the value zero indicates that no Instance-specific FEC OTI Information is provided. When the Length is zero, three padding bytes containing value zero SHALL follow the Length field to maintain 4-octet alignment.

Instance-specific FEC OTI Information: the contents of this field are FEC Scheme Instance-specific.

Note that in the case of a Content Delivery protocol that supports external signaling of the total FEC Object Transmission Information length, then the Scheme-Specific FEC OTI field defined here is optional. Otherwise, this field MUST be included.

4.3. Procedures

The algorithm defined in Section 9.1. of [RFC5052] MUST be used to partition the file into source blocks.

4.4. FEC Code Specification

The FEC code specification and the correspondence of Encoding Symbols IDs to encoding symbols are defined by specific instances of this scheme and so are out of scope of this document.

5. Small Block Systematic FEC Scheme

5.1. Introduction

This section defines an Under-Specified FEC Scheme for Small Block Systematic FEC codes as described in [RFC3453]. For Small Block Systematic FEC codes, each source block is of length at most 65535 source symbols.

Although these codes can generally be accommodated by the FEC Encoding ID described in Section 4, a specific FEC Encoding ID is defined for Small Block Systematic FEC codes to allow more flexibility and to retain header compactness. The small source block length and small expansion factor that often characterize systematic codes may require the data source to frequently change the source block length. To allow the dynamic variation of the source block length and to communicate it to the receivers with low overhead, the block length is included in the FEC Payload ID.

5.2. Formats and Codes

5.2.1. FEC Payload ID(s)

The FEC Payload ID is composed of the Source Block Number, Source Block Length, and the Encoding Symbol ID structured as shown in Figure 6.

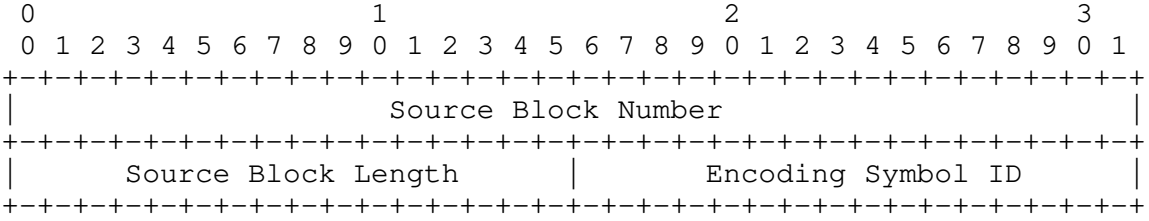

Figure 6: FEC Payload ID Format for Small Block Systematic FEC Scheme

The Source Block Number is a 32-bit unsigned integer that identifies from which source block of the object the encoding symbol(s) in the payload are generated. These blocks are numbered consecutively from 0 to N-1, where N is the number of source blocks in the object.

The Source Block Length is a 16-bit unsigned integer that specifies the length in units of source symbols of the source block identified by the Source Block Number.

The Encoding Symbol ID is a 16-bit unsigned integer that identifies which specific encoding symbol(s) generated from the source block are carried in the packet payload. Each encoding symbol is either an original source symbol or a redundant symbol generated by the encoder. The exact details of the correspondence between Encoding Symbol IDs and the encoding symbol(s) in the packet payload are dependent on the particular FEC Scheme instance used as identified by the FEC Instance ID, and these details may be proprietary.

5.2.2. FEC Object Transmission Information

5.2.2.1. Mandatory

The mandatory FEC Object Transmission Information element for the Small Block Systematic FEC Scheme is:

- FEC Encoding ID: 129

5.2.2.2. Common

The Common FEC Object Transmission Information elements and their value ranges for the Small Block Systematic FEC Scheme are:

FEC Instance ID: a non-negative integer less than 2^^16.

Transfer-Length: a non-negative integer less than 2^^48, indicating the length of the object in octets.

Encoding-Symbol-Length: a non-negative integer less than 2^^16, indicating the length of each encoding symbol in octets.

Maximum-Source-Block-Length: a non-negative integer less than 2^^16, indicating the maximum number of source symbols in a source block.

Max-Number-of-Encoding-Symbols: a non-negative integer less than 2^^16, indicating the maximum number of encoding symbols per block (i.e., source plus repair symbols in the case of a systematic code).

The encoded Common FEC Object Transmission Information is defined in Figure 7.

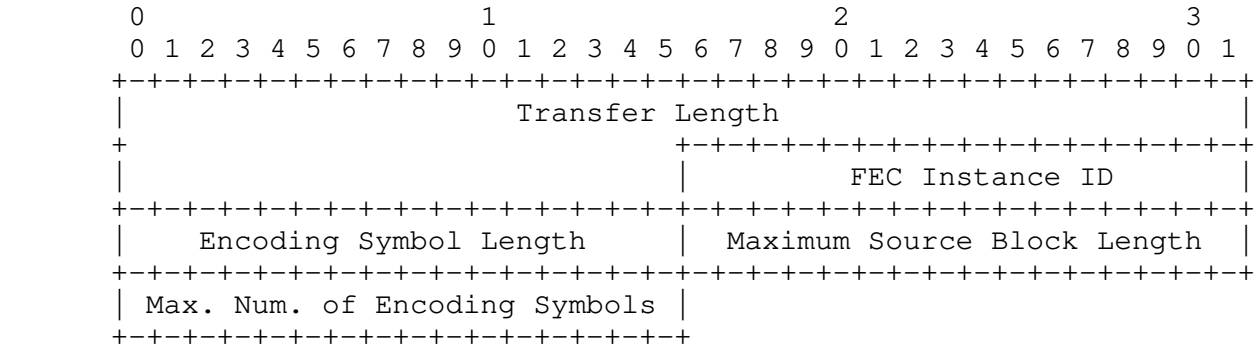

Figure 7: FEC OTI Format for Small Block Systematic FEC Scheme

The Transfer Length (48 bits), FEC Instance ID (16 bits), Encoding Symbol Length (16 bits), Maximum Source Block Length (16 bits), and Maximum Number of Encoding Symbols (16 bits) are encoded as unsigned integers.

All Encoding Symbols of a transport object MUST have length equal to the length specified in the Encoding Symbol Length field, with the optional exception of the last source symbol of the last source block (so that redundant padding is not mandatory in this last symbol). This last source symbol MUST be logically padded out with zeroes when another Encoding Symbol is computed based on this source symbol to ensure the same interpretation of this Encoding Symbol value by the sender and receiver. However, this padding need not be actually sent with the data of the last source symbol.

Note: this FEC Scheme was first defined in [RFC3452], which did not require that the Encoding Symbol Length should be the same for every source block. However, no protocols have been defined that support variation in the Encoding Symbol Length between source blocks, and thus introduction of a general requirement that the Encoding Symbol Length be the same across source blocks (as defined here) should not cause backwards compatibility issues and will aid interoperability.

5.2.2.3. Scheme-Specific

The Scheme-Specific FEC Object Transmission Information format defined in Section 4.2.2.3 SHALL be used.

5.3. Procedures

The algorithm defined in Section 9.1. of [RFC5052] MAY be used to partition the file into source blocks. Otherwise, the FEC Scheme instance MUST specify the algorithm that is used.

5.4. FEC Code Specification

The FEC code specification and the correspondence of Encoding Symbols IDs to encoding symbols are defined by specific instances of this scheme and so are out of scope of this document.

6. Compact FEC Scheme

6.1. Introduction

The Compact FEC Scheme is an Under-Specified FEC Scheme. This FEC Scheme is similar in spirit to the Compact No-Code FEC Scheme, except that a non-trivial FEC encoding (that is Under-Specified) may be used to generate encoding symbol(s) placed in the payload of each packet and a corresponding FEC decoder may be used to produce the source block from received packets.

6.2. Formats and Codes

6.2.1. FEC Payload ID(s)

The FEC Payload ID format defined in Section 3.2.1 SHALL be used.

6.2.2. FEC Object Transmission Information

6.2.2.1. Mandatory

The mandatory FEC Object Transmission Information element for the Compact No-Code FEC Scheme is:

- FEC Encoding ID: 130

6.2.2.2. Common

The Common FEC Object Transmission Information elements and their encoding are the same as defined for the Small Block, Large Block, and Expandable FEC Scheme in Figure 4.

6.2.2.3. Scheme-Specific

The Scheme-Specific FEC Object Transmission Information format defined in Section 4.2.2.3 SHALL be used.

6.3. Procedures

The algorithm defined in Section 9.1. of [RFC5052] MUST be used to partition the file into source blocks.

6.4. FEC Code Specification

The FEC code specification and the correspondence of Encoding Symbols IDs to encoding symbols are defined by specific instances of this scheme and so are out of scope of this document.

7. Security Considerations

This specification does not introduce any further security considerations beyond those described in [RFC5052].

8. Acknowledgements

This document is substantially based on [RFC3695] by Michael Luby and Lorenzo Vicisano and [RFC3452] by Michael Luby, Lorenzo Vicisano, Jim Gemmell, Luigi Rizzo, Mark Handley, and Jon Crowcroft.

9. IANA Considerations

FEC Encoding IDs 0 and 130 were first defined and registered in the ietf:rmt:fec:encoding namespace by [RFC3695]. This document updates and obsoletes the definitions from that specification. References to that specification should be replaced with references to this document.

FEC Encoding IDs 128 and 129 were first defined and registered in the ietf:rmt:fec:encoding namespace by [RFC3452]. This document updates and obsoletes the definitions from that specification. References to that specification should be replaced with references to this document.

Values of FEC Encoding IDs and FEC Instance IDs are subject to IANA registration. For general guidelines on IANA considerations as they apply to this document, see [RFC5052].

This document assigns the Fully-Specified FEC Encoding ID 0 under the ietf:rmt:fec:encoding name-space (which was previously assigned by [RFC3695], which is obsoleted by this document) to "Compact No-Code" as specified in Section 3 above.

This document assigns the Under-Specified FEC Encoding ID 128 under the ietf:rmt:fec:encoding name-space (which was previously assigned by [RFC3452]) to "Small Block, Large Block, and Please note that we have added a comma between large block and expandable throughout this document (RFC Editor style is to include a comme before the last item of a series). If you do not object, we will ask IANA to include this comma in their registry for consistency. --> Expandable FEC Codes" as specified in Section 4 above.

This document assigns the Under-Specified FEC Encoding ID 129 under the ietf:rmt:fec:encoding name-space (which was previously assigned by [RFC3452]) to "Small Block Systematic FEC Codes" as specified in Section 5 above.

This document assigns the Under-Specified FEC Encoding ID 130 under the ietf:rmt:fec:encoding name-space (which was previously assigned by [RFC3695], which is obsoleted by this document) to "Compact FEC" as specified in Section 6 above.

As FEC Encoding IDs 128, 129, and 130 are Under-Specified, "FEC Instance ID" sub-name-spaces must be established, in accordance to [RFC5052]. Hence, this document also assumes responsibility for the "FEC Instance ID" registries named.

ietf:rmt:fec:encoding:instance:128, scoped by ietf:rmt:fec: encoding = 128

ietf:rmt:fec:encoding:instance:129, scoped by ietf:rmt:fec: encoding = 129

ietf:rmt:fec:encoding:instance:130, scoped by ietf:rmt:fec: encoding = 130

The values that can be assigned within these namespaces are non-negative numeric indices. Assignment requests are granted on a "First Come First Served" basis. [RFC5052] specifies additional criteria that MUST be met for the assignment within the generic ietf: rmt:fec:encoding:instance name-space. These criteria also apply to ietf:rmt:fec:encoding:instance:128, ietf:rmt:fec:encoding:instance: 129, and ietf:rmt:fec:encoding:instance:130.

10. Changes from Schemes Defined in RFC 3452 and RFC 3695

This section describes the changes between the Experimental versions of these FEC Scheme specifications contained in RFC 3452 [RFC3452] and RFC 3695 [RFC3695] and those defined in this specification:

- Scheme definitions have been updated to meet the requirements of [RFC5052].
- Complete encoding formats for the FEC Object Transmission Information for each scheme are defined here, instead of within content delivery protocol specifications, since the exact format depends on the FEC Scheme.

- The previous specifications for the Compact No-Code and Small Block Systematic FEC Schemes did not require that all encoding symbols of the object should have the same length. This requirement is introduced in this specification. Since no protocols have been defined that support variation of the encoding symbol length within an object this should not cause backwards compatibility issues.

11. References

11.1. Normative References

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

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

11.2. Informative References

[RFC3452] Luby, M., Vicisano, L., Gemmell, J., Rizzo, L., Handley, M., and J. Crowcroft, "Forward Error Correction (FEC) Building Block", RFC 3452, December 2002.

[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.

[RFC3269] Kermode, R. and L. Vicisano, "Author Guidelines for Reliable Multicast Transport (RMT) Building Blocks and Protocol Instantiation documents", RFC 3269, April 2002.

[RFC3048] Whetten, B., Vicisano, L., Kermode, R., Handley, M., Floyd, S., and M. Luby, "Reliable Multicast Transport Building Blocks for One-to-Many Bulk-Data Transfer", RFC 3048, January 2001.

[RFC3695] Luby, M. and L. Vicisano, "Compact Forward Error Correction (FEC) Schemes", RFC 3695, February 2004.

Author's Address

Mark Watson
Digital Fountain
39141 Civic Center Drive
Suite 300
Fremont, CA 94538
USA

EMail: mark@digitalfountain.com