Internet Engineering Task Force
INTERNET-DRAFT
draft-ietf-rmt-bb-fec-supp-compact-01.ps

RMT WG
M. Luby
Digital Fountain
L. Vicisano
Cisco

12 May 2003
Expires: November 2003

Compact Forward Error Correction (FEC) Schemes

Status of this Document

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC 2026 [1]. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are 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 a "work in progress".

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

To view the list Internet-Draft Shadow Directories, see http://www.ietf.org/shadow.html.

This document is a product of the IETF RMT WG. Comments should be addressed to the authors, or the WG's mailing list at rmt@lbl.gov.

Abstract

This document introduces some Forward Error Correction (FEC) schemes that supplement the FEC schemes described in RFC 3452 [6]. The primary benefits of these additional FEC schemes are that they are designed for reliable bulk delivery of large objects using a more compact FEC Payload ID, and they can be used to sequentially deliver blocks of an object of indeterminate length. Thus, they more flexibly support different delivery models with less packet header overhead.

This document also describes the Fully-Specified FEC scheme corresponding to FEC Encoding ID 0. This Fully-Specified FEC scheme requires no FEC coding and is introduced primarily to allow simple interoperability testing between different implementations of protocol instantiations that use the FEC building block.

Table of Contents

1. Introduction

This document describes two new Forward Error Correction (FEC) schemes corresponding to FEC Encoding IDs 0 and 130 which supplement the FEC schemes corresponding to FEC Encoding IDs 128 and 129 described in the FEC Building Block [6].

The new FEC schemes are particularly applicable when an object is partitioned into equal-length source blocks. In this case, the source block length common to all source blocks can be communicated out-of-band, thus saving the additional overhead of carrying the source block length within the FEC Payload ID of each packet. The new FEC schemes are similar to the FEC schemes with FEC Encoding ID 128 defined in RFC 3452 [6], except that the FEC Payload ID is half as long. This is the reason that these new FEC schemes are called Compact FEC schemes.

The primary focus of FEC Encoding IDs 128 and 129 is to reliably deliver bulk objects of known length. The FEC schemes described in this document are designed to be used for both reliable delivery of bulk objects of known length, and for the delivery of a stream of source blocks for an object of indeterminate length. Within the block-stream delivery model, reliability guarantees can range from acknowledged reliable delivery of each block to unacknowledged enhanced-reliability delivery of time-sensitive blocks, depending on the properties of the protocol instantiation in which the FEC scheme is used. Acknowledged reliable block-stream delivery is similar in spirit to the byte-stream delivery that TCP offers, except that the unit of delivery is a block of data instead of a byte of data. In the spirit of a building block (see RFC 3048 [9]), the FEC schemes described in this document can be used to provide reliability for other service models as well.

The two new FEC Encoding IDs 0 and 130 are described in Section 2, and this supplements Section 5 of the FEC building block [6]. Section 3 of this document describes the Fully-Specified FEC scheme corresponding to the FEC Encoding ID 0. This Fully-Specified FEC 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.

This document inherits the context, language, declarations and restrictions of the FEC building block [6]. This document also uses the terminology of the companion document [7] 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 RFC 3048 [9]. This document is a product of the IETF RMT WG and follows the general guidelines provided in RFC 3269 [3].

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 [2].

2. Packet Header Fields

This section specifies FEC Encoding IDs 0 and 130 and the associated FEC Payload ID formats and the specific information in the corresponding FEC Object Transmission Information. The FEC scheme associated with FEC Encoding ID 0 is Fully-Specified whereas the FEC schemes associated with FEC Encoding ID 130 are Under-Specified.

FEC Encoding IDs 0 and 130 have the same FEC Payload ID format, which is described in the following subsection. The FEC Object Transmission Information for FEC Encoding IDs 0 and 130 is different, and is described in the subsequent two subsections.

2.1. FEC Payload ID for FEC Encoding IDs 0 and 130

The FEC Payload ID for FEC Encoding IDs 0 and 130 is composed of a Source Block Number and an Encoding Symbol ID structured as follows:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Source Block Number       |      Encoding Symbol ID       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The 16-bit Source Block Number 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 the source block is processed at the transport layer by the sender, the network transit time for packets, and the amount of time the source block is processed at the transport layer by a receiver. This is so that the receiver can clearly decide which packets belong to which source block.

The 16-bit Encoding Symbol ID 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 for FEC Encoding ID 0 are specified in Section 3. The exact details of the correspondence between Encoding Symbol IDs and the encoding symbol(s) in the packet payload for FEC Encoding ID 130 are dependent on the particular encoding algorithm used as identified by the FEC Encoding ID and by the FEC Instance ID.

2.2. Compact No-Code FEC scheme

This subsection reserves FEC Encoding ID 0 for the Compact No-Code FEC scheme that is described in this subsection and in Section 3. This is a Fully-Specified FEC scheme that is primarily intended to be used for simple interoperability testing between different implementations of protocol instantiations that use the FEC building block. The value of this

FEC scheme is that no FEC encoding or decoding is required to implement it and therefore it is easy to test interoperability between protocols that may use different proprietary FEC schemes in production in their first implementations.

The FEC Payload ID format for FEC Encoding ID 0 is described in Subsection 2.1. The FEC Object Transmission Information has the following specific information:

- o The FEC Encoding ID 0.
- o For each source block of the object, the length in bytes of the encoding symbol carried in the packet payload. This length MUST be the same for all packets sent for the same source block, but MAY be different for different source blocks in the same object.
- o For each source block of the object, the length of the source block in bytes. Typically, each source block for the object has the same length and thus only one length common to all source blocks need be communicated, but this is not a requirement. For convenience, the source block length MAY be a multiple of the length of the encoding symbol carried in one packet payload.

How this out-of-band information is communicated is outside the scope of this document.

Other information, such as the object length and the number of source blocks of the object for an object of known length may be needed by a receiver to support some delivery models such as reliable bulk data delivery.

2.3. Compact FEC scheme

This subsection reserves FEC Encoding ID 130 for the Compact FEC scheme that is described in this subsection. This 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.

The FEC Payload ID format for FEC Encoding ID 0 is described in Subsection 2.1. The FEC Object Transmission Information has the following specific information:

- o The FEC Encoding ID 130.
- o The FEC Instance ID associated with the FEC Encoding ID 130 to be used.
- o For each source block of the object, the aggregate length of the encoding symbol(s) carried in one packet payload. This length MUST be the same for all packets sent for the same source block, but MAY be different for different source blocks in the same object.
- o For each source block of the object, the length of the source block in bytes. Typically, each source block for the object has the same length and thus only one length common to all source blocks need be communicated, but this is not a requirement. For convenience, the source block length MAY be a multiple of the aggregate length of the encoding symbol(s) carried in one packet payload.

How this out-of-band information is communicated is outside the scope of this document.

Other information, such as the object length and the number of source blocks of the object for an object of known length may be needed by a receiver to support some delivery models such as reliable bulk data delivery.

3. Compact No-Code FEC scheme

In this section we describe a Fully-Specified FEC scheme corresponding to FEC Encoding ID 0. The primary purpose for introducing this FEC schemes is to allow simple interoperability testing between different implementations of the same protocol instantiation that uses the FEC building block.

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 FEC Payload ID consists of two fields, the 16-bit Source Block Number and the 16-bit Encoding Symbol ID, as described in Subsection 2.1. The relative lengths of these fields were chosen for their similarity with the corresponding fields of the FEC Payload ID associated with FEC Encoding ID 130, and because of this testing interoperability of the FEC scheme associated with FEC Encoding ID 0 provides a first basic step to testing interoperability of an FEC scheme associated with FEC Encoding ID 130.

The mapping between source blocks of an object and Source Block Numbers is as described in Subsection 2.1. The next two subsections describe the details of how the Compact No-Code FEC scheme operates for each source block of an object. These subsections are not meant to suggest a particular implementation, but just to illustrate the general algorithm through the description of a simple, non-optimized implementation.

3.1. Source Block Logistics

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

Let L > 0 be the length of the encoding symbol contained in the payload of each packet. There are several possible ways the length of the encoding symbol L can be communicated to the receiver, and how this is done is outside the scope of this document. As an example, a sender could fix the packet payload length to be L in order to place the encoding symbol of length L into the packet, and then a receiver could infer the value of L from the length of the received packet payload. It is REQUIRED that L be the same for all packets sent for the same source block but MAY be different for different source blocks within the same object.

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 of refer to the companion document [7] for general advice.

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

3.2. 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.

- o If Y < N-1 then generate the encoding symbol consisting of the L bytes of the source block numbered L*Y through L*(Y+1)-1.
- o If Y = N-1 then generate the encoding symbol consisting of the last X - L*(N-1) bytes of the source block numbered L*(N-1) through X-1 followed by L*N - X padding bytes of zeroes.
- o 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.
- o In preparation for the generation of the next packet: if Y < N-1 then increment Y by one elseif 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 describe 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 received out-of-band as part of the FEC Object Transmission Information to determine the length X in bytes of the source block, and allocates space for the X bytes that the source block requires. The receiver also computes the length L of the encoding symbol(s) in the payload of the packet by subtracting the packet header length from the total length of the received packet (and the receiver checks that this length is the same in each subsequent received packet from the same source block). 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 FEC Payload ID of the packet. If Y < N-1 then the receiver copies the L bytes of the encoding symbol into bytes numbered L*Y through L*(Y+1)-1 of the space reserved for the source block. If Y = N-1 then the receiver copies the first X - L*(N-1) bytes of the encoding symbol into bytes numbered L*(N-1) through X-1 of the space reserved for the source block. In either case, the receiver sets RECEIVED[Y] = true. At each point in time, the receiver has successfully recovered bytes L*Y through L*(Y+1)-1 of the source block for all Y in the interval [0..N-1] for which RECEIVED[Y] is true. If all all N entries of RECEIVED are true then the receiver has recovered the entire source block.

4. Usage Examples

The following subsections outline some usage examples for FEC Encoding IDs 0 and 130.

4.1. Reliable Bulk Data Delivery

One possible delivery model that can be supported using any FEC scheme described in this document is reliable bulk data delivery. In this model, one or more potentially large objects are to be reliably delivered to potentially multiple receivers using multicast. For this delivery model the unique SBN mode is often used. Using this mode the maximum length of an object that can be delivered is at most the number of possible source blocks times the maximum length of a source block. If the aggregate length of encoding symbols carried in a packet payload is L bytes then the maximum length of a source block is the number of distinct Encoding Symbol IDs times L, or 2^16 * L for FEC Encoding IDs 0 and 130. If for example L = 1 KB then the length of a source block can be up to around 65 MB. However, in practice the length of the source block is usually shorter than the number of distinct Encoding Symbol IDs times L, and thus generally the length of a source block is a fraction of 65 MB. Since the number of distinct Source Block Numbers is 2^16, for this example an object can be more than a terabyte.

The non-unique SBN mode of delivery can also be effectively used for reliably delivering large object. In this case, the length of the object delivered could be arbitrarily large, depending on the out-of-band mapping between source blocks and Source Block Numbers.

4.2. Block-Stream Delivery

Another possible delivery model that can be supported using FEC Encoding ID 0 or 130 is block-stream delivery of an object. In this model, the source blocks of a potentially indeterminate length object are to be reliably delivered in sequence to one or multiple receivers. Thus, the object could be partitioned into source blocks on-the-fly at the sender as the data arrives, and all packets generated for one source block are sent before any packets are sent for the subsequent source block. In this example, all source blocks could be of the same length and this length could be communicated out-of-band to a receiver before the receiver joins the session. For this delivery model it is not required that the Source Block Numbers for all source blocks are unique. However, a suggested usage is to use all 2^16 Source Block Numbers for consecutive source blocks of the object, and thus the time between reuse of a Source Block Number is the time it takes to send the packets for 2^16 source blocks.

This delivery model can be used to reliably deliver an object to one or multiple receivers, using either an ACK or NACK based acknowledgement based scheme for each source block. As another example the sender could send a fixed number of packets for each source block without any acknowledgements from receivers, for example in a live streaming without feedback application.

5. Security Considerations

The security considerations for this document are the same as they are for RFC 3452 [6].

6. IANA Considerations

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 RFC 3452 [6]. This document assigns the Fully-Specified FEC Encoding ID 0 under the ietf:rmt:fec:encoding name-space to "Compact No-Code". The FEC Payload ID format and corresponding FEC Object Transmission Information associated with FEC Encoding ID 0 is described in Subsections 2.1 and 2.2, and the corresponding FEC scheme is described in Section 3.

This document assigns the Under-Specified FEC Encoding ID 130 under the ietf:rmt:fec:encoding name-space to "Compact FEC". This document also establishes a new "FEC Instance ID" registry

ietf:rmt:fec:encoding:instance:130

ietf:rmt:fec:encoding = 130 (Compact FEC)

The FEC Payload ID format and corresponding FEC Object Transmission Information associated with FEC Encoding ID 130 is described in Subsections 2.1 and 2.3.

7. References

[1] Bradner, S., "The Internet Standards Process -- Revision 3", RFC 2026, October 1996.

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

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

[4] Luby, M., Gemmell, J., Vicisano, L., Rizzo, L. and J. Crowcroft, "Asynchronous Layered Coding (ALC) Protocol Instantiation", RFC 3450 December 2002.

[5] Luby, M., Gemmell, J., Vicisano, L., Rizzo, L., Handley, M. and J. Crowcroft, "Layered Coding Transport (LCT) Building Block", RFC 3451 December 2002.

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

[7] 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.

[8] Mankin, A., Romanow, A., Bradner, S., Paxson V., "IETF Criteria for Evaluating Reliable Multicast Transport and Application Protocols", RFC 2357, June 1998.

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

8. Authors' Addresses

Michael Luby
luby@digitalfountain.com
Digital Fountain, Inc.
39141 Civic Center Drive
Suite 300
Fremont, CA 94538

Lorenzo Vicisano
lorenzo@cisco.com
cisco Systems, Inc.
170 West Tasman Dr.,
San Jose, CA, USA, 95134

9. Full Copyright Statement