Reliable Multicast
Internet-Draft
Intended status: Informational
Expires: March 8, 2007

M. Watson
M. Luby
Digital Fountain
L. Vicisano
Cisco Systems, Inc.
September 4, 2006

Forward Error Correction (FEC) Building Block
draft-ietf-rmt-fec-bb-revised-04

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

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 draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

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

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on March 8, 2007.

Copyright Notice

Copyright (C) The Internet Society (2006).

Abstract

This document describes how to use Forward Error Correction (FEC) codes to efficiently provide and/or augment reliability for data transport. This document defines a framework for the definition of the information that needs to be communicated in order to use an FEC code for delivering content, in addition to the encoded data itself, and for definition of formats and codes for communcation of that information. Both information communicated with the encoded data itself and information that needs to be communicated 'out-of-band' are considered. The procedures for specifying new FEC codes, defining the information communication requirements associated with those codes and registering them with the Internet Assigned Numbers Authority (IANA) are also described. The requirements on Content Delivery Protocols which wish to use FEC codes defined within this framework are also defined. The companion document titled, "The Use of Forward Error Correction (FEC) in Reliable Multicast" describes some applications of FEC codes for delivering content.

Table of Contents

1. Introduction 4
2. Definitions/Abbreviations 5
3. Requirements notation 6
4. Rationale 7
5. Applicability Statement 9
6. Functionality 10
6.1. FEC Schemes 11
6.2. FEC Object Transmission Information 13
6.2.1. Transport of FEC Object Transmission Information . . . 14
6.2.2. Opacity of FEC Object Transmission Information 15
6.2.3. Mandatory FEC Object Transmission Information elements 15
6.2.4. Common FEC Object Transmission Information elements . 16
6.2.5. Scheme-specific FEC Object Transmission Information element 16
6.3. FEC Payload ID 17
7. FEC scheme specifications 18
8. CDP specifications 21
9. Common algorithms 22
9.1. Block partitioning algorithm 22
9.1.1. First Step 22
9.1.2. Second step 23
10. Requirements from other building blocks 24
11. Security Considerations 25
12. IANA Considerations 26
12.1. Explicit IANA Assignment Guidelines 26
13. Changes from RFC3452 28
14. Acknowledgments 29
15. References 30
15.1. Normative References 30
15.2. Informative References 30
Authors' Addresses 31
Intellectual Property and Copyright Statements 32

1. Introduction

This document describes how to use Forward Error Correction (FEC) codes to provide support for reliable delivery of content within the context of a Content Delivery Protocol (CDP). This document describes a building block as defined in [10]. This document is a product of the IETF RMT WG and follows the general guidelines provided in [8].

The purpose of this building block is to define a framework for forward error correction such that:

1. CDPs can be designed to operate with a range of different FEC codes/schemes, without needing to know details of the specific FEC code/scheme that may be used

2. FEC schemes can be designed to operate with a range of different CDPs, without needing to know details of the specific CDPs

Note that a 'CDP' in the context of this document may consist of several distinct protocol mechanisms and may support any kind of application requiring reliable transport - for example object delivery and streaming applications.

This document also provides detailed guidelines on how to write an RFC for an FEC scheme corresponding to a new FEC Encoding ID (for both Fully-Specified and Under-Specified FEC Schemes).

RFC3452 [3] contained a previous version of this specification, which was published in the "Experimental" category. It was the stated intent of the RMT working group to re-submit this specification as an IETF Proposed Standard in due course.

This Proposed Standard specification is thus based on RFC3452 [3] updated according to accumulated experience and growing protocol maturity since the publication of RFC3452 [3]. Said experience applies both to this specification itself and to congestion control strategies related to the use of this specification.

The differences between RFC3452 [3] and this document listed in Section 13

2. Definitions/Abbreviations

Object: An ordered sequence of bytes to be transferred by the transport protocol. For example, a file or stream.

Symbol: A unit of data processed by the Forward Error Correction code. A symbol is always considered as a unit i.e. it is either completely received or completely lost.

Source symbol: A symbol containing information from the original object.

Repair symbol: A symbol containing information generated by the FEC code which can be used to recover lost source symbols.

Encoding symbol: A source symbol or a repair symbol.

Encoder: The FEC scheme specific functions required to transform a object into FEC encoded data. i.e. the functions that produce repair symbols using source symbols.

Decoder: The FEC scheme specific functions required to transform received FEC encoded data into a copy of the original object

Receiver: A system supporting the receiving functions of a CDP and FEC scheme according to this specification

Sender: A system supporting the sending functions of a CDP and FEC scheme according to this specification

Source Block: A part of the object formed from a subset of the object's source symbols

CDP: Content Delivery Protocol

FEC: Forward Error Correction

3. 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 [1].

4. Rationale

An FEC code is a valuable basic component of any CDP that is to provide reliable delivery of an object. Using FEC codes is effective in the context of IP multicast and reliable delivery because FEC encoding symbols can be useful to all receivers for reconstructing an object even when the receivers have received different encoding symbols. Furthermore, FEC codes can ameliorate or even eliminate the need for feedback from receivers to senders to request retransmission of lost packets.

Central to this document is the concept of an 'FEC Scheme'. An FEC scheme defines ancilliary information and procedures which, combined with an FEC code specification, fully define how the FEC code can be used with CDPs. An FEC scheme may be associated with a single standardised FEC code (A 'Fully-Specified' FEC scheme) or may be applicable to many FEC codes (An 'Under-Specified' FEC scheme).

This document describes a framework for the definition of FEC schemes. Definition of actual FEC schemes is outside the scope of this document. This document also defines requirements for reliable CDPs that make use of FEC schemes. Any such CDP which is compliant to the requirements specified in this document can make use of any FEC scheme which is defined within the framework described here. Note that FEC schemes may place restrictions on the types of CDP they are intended to be used with. For example, some FEC schemes may be specific to particular types of application, such as file delivery or streaming.

The goal of the FEC building block is to describe functionality directly related to FEC codes that is common to all reliable CDPs and to all FEC schemes, and to leave out any additional functionality that is specific to particular CDPs or particular FEC schemes. The primary functionality described in this document that is common to all such CDPs that use FEC codes is the definition and transport of three kinds of information from sender to receiver(s): encoding symbols themselves, anciliary information associated with encoding symbols (or groups of such symbols, such as the group of symbols in a single packet, or the group of symbols related to a single source block) and anciliary information associated with the whole object being transfered. It is important to note that this ancilliary information is only required by the receiver if one or more of the encoding symbols to which it relates are received.

This document for example does not describe how receivers may request transmission of particular encoding symbols for an object. This is because although there are CDPs where requests for transmission are of use, there are also CDPs that do not require such requests.

The companion document [4] should be consulted for a full explanation of the benefits of using FEC codes for reliable content delivery using IP multicast. FEC codes are also useful in the context of unicast, and thus the scope and applicability of this document is not limited to IP multicast.

5. Applicability Statement

The FEC building block does not provide any support for congestion control. Any complete multicast CDP MUST provide congestion control that conforms to [6], and thus this MUST be provided by another building block when the FEC building block is used in a CDP.

A more complete description of the applicability of FEC codes can be found in the companion document [4].

6. Functionality

This section describes FEC information that is either to be sent in packets also containing FEC encoding symbols or 'out-of-band'. The FEC information is associated with transmission of encoding symbols related to a particular object. There are three classes of packets that may contain FEC information: data packets, session-control packets and feedback packets. They generally contain different kinds of FEC information. Note that some CDPs may not use session-control or feedback packets.

Data packets may sometimes serve as session-control packets as well; both data and session-control packets generally travel downstream from the sender towards receivers and are sent to a multicast channel or to a specific receiver using unicast. Session-control packets may additionally travel upstream from receivers to senders.

As a general rule, feedback packets travel upstream from receivers to the sender. Sometimes, however, they might be sent to a multicast channel or to another receiver or to some intermediate node or neighboring router that provides recovery services.

This document specifies the FEC information that must be carried in data packets and the other FEC information that must be communicated from sender to receiver(s) either out-of-band or in data packets. This document does not specify out-of-band methods nor does it specify the way out-of-band FEC information is associated with FEC information carried in data packets. These methods must be specified in CDP specifications that use this FEC building block.

FEC information is classified as follows:

1. FEC information associated with an object

   This is information that is essential for the FEC decoder to decode a specific object. An example of this information is the identity of the FEC scheme that is being used to encode the object, in the form of the FEC Encoding ID. The FEC Encoding ID is described further below. This information may also include FEC scheme-specific parameters for the FEC decoder.

2. FEC information associated with specific encoding symbols for an object

   This is information which is associated with one or more encoding symbols and is thus needed by the decoder whenever one or more of those encoding symbols have been received. Depending on the FEC

scheme, information may be associated with individual symbols and/or with groups of symbols. One common such grouping is the group of symbols included within a single packet. Many FEC schemes also segment the object being encoded into multiple 'source blocks', each of which is processed independently for FEC purposes. Information about each source block is another type of information associated with a group of encoding symbols, this time the group of symbols which are related to a given source block.

Two 'containers' are provided for communicating the FEC information described above, but there is not necessarily a one-to-one correspondance between the class of FEC information and the mechanism used. The two mechanisms are:

a. FEC Object Transmission Information

CDPs must provide a reliable mechanism for communicating certain FEC information from sender to receiver(s). This information is known as 'FEC Object Transmission Information' and its contents depend on the particular FEC scheme. It includes all information of the first class above and may include information of the second class. The FEC Object Transmission Information can be sent to a receiver within the data packet headers, within session control packets, or by some other means.

b. FEC Payload ID

CDPs must provide a mechanism for communicating information which identifies (for FEC purposes) the encoding symbols carried by a packet. This information is known as the FEC Payload ID and its contents depend on the FEC scheme. It includes only information of the second class above. A data packet that carries encoding symbols MUST include an FEC Payload ID.

6.1. FEC Schemes

Two types of FEC scheme are defined by this document: 'Fully-Specified' FEC schemes and 'Under-Specified' FEC schemes. An FEC scheme is a Fully-Specified FEC scheme if the encoding scheme is formally and Fully-Specified, in a way that independent implementors can implement both encoder and decoder from a specification that is an IETF RFC.

It is possible that an FEC scheme may not be a Fully-Specified FEC scheme, because either a specification is simply not available or a party exists that owns the encoding scheme and is not willing to disclose the algorithm or specification. We refer to such an FEC

encoding scheme as an Under-Specified FEC scheme.

FEC schemes are identified by an FEC Encoding ID, which is an integer identifier assigned by IANA. The FEC Encoding ID allows receivers to select the appropriate FEC decoder. The value of the FEC Encoding ID MUST be the same for all transmission of encoding symbols related to a particular object, but MAY vary across different transmissions of encoding symbols about different objects, even if transmitted to the same set of multicast channels and/or using a single upper-layer session.

The FEC Instance ID is an integer value that identifies a specific instance of an Under-Specified FEC scheme. This value is not used for Fully-Specified FEC schemes. The FEC Instance ID is scoped by the FEC Encoding ID, and FEC Instance ID values are subject to IANA registration.

The FEC Encoding ID for Fully-Specified FEC Schemes and both the FEC Encoding ID and FEC Instance ID for Under-Specified FEc Schemes are essential for the decoder to decode an object and thus are part of the FEC Object Transmission Information.

The following requirements apply to all FEC schemes, whether Fully-Specified or Under-Specified:

- The type, semantics and an encoding format for the FEC Payload ID and the FEC Object Transmission Information MUST be defined.

- A value for the FEC Encoding ID MUST be reserved and associated with the types, semantics and encoding format of the FEC Payload ID and the FEC Object Transmission Information.

The specification for an Under-Specified FEC Scheme MAY allocate a sub-field within the Scheme-specific FEC Object Transmission Information element which is for instance-specific information. Each specific instance of the Under-Specified FEC Scheme may then use this field in an instance-specific way. The FEC scheme should define the scheme-specific FEC Object Transmission Information element in such a way that receivers that do not support the received FEC Instance ID can still parse and interpret the scheme-specific FEC Object Transmission Information element with the exception of the instance-specific field.

An already defined Under-Specified FEC Scheme (i.e. FEC Encoding ID value) MUST be reused if the associated FEC Payload ID and FEC Object Transmission Information have the required fields and encoding formats for a new Under-Specified FEC scheme instance.

An instance of an Under-Specified FEC scheme is fully identified by the tuple (FEC Encoding ID, FEC Instance ID). The tuple MUST identify a single scheme instance that has at least one implementation. The party that owns this tuple MUST be able to provide information on how to obtain the Under-Specified FEC scheme instance identified by the tuple, e.g., a pointer to a publicly available reference-implementation or the name and contacts of a company that sells it, either separately or embedded in another product.

This specification reserves the range 0-127 for the values of FEC Encoding IDs for Fully-Specified FEC schemes and the range 128-255 for the values of Under-Specified FEC schemes.

6.2. FEC Object Transmission Information

The FEC Object Transmission Information contains information which is essential to the decoder in order to decode the encoded object. It may also contain information which is required to decode certain groups of encoding symbols, for example individual Source Blocks within the object. This information is communicated reliably by the CDP to the receiver(s) as described in Section 8.

The FEC Object Transmission Information may consist of several elements and each element may be one of three types, as follows:

Mandatory: These elements are defined in this specification and are each mandatory for at least one of the two types of FEC Scheme. Each FEC scheme specifies how the values of the Mandatory FEC Object Transmission Information elements are determined and each CDP specifies how this information is encoded and reliably communicated to the receiver(s). The Mandatory FEC Object Transmission Information includes the identification of the FEC Scheme, which is needed by the receiver to determine whether it supports the FEC Scheme.

Common: These elements are defined in this specification and are optional to be used by an FEC scheme. Each FEC scheme specifies which of the Common FEC Object Transmission Information elements it uses and how the values of these elements are determined.

Scheme-specific: An FEC scheme may specify a single Scheme-specific FEC Object Transmission Information element. The FEC scheme specifies the type, semantics and encoding format of the Scheme-specific FEC Object Transmission Information element. The resulting byte-string is known as the "encoded Scheme-specific FEC Object Transmission Information". Each CDP specifies how the encoded Scheme-specific FEC Object Transmission is communicated

reliably to the receiver(s) i.e. exactly where it shall be carried within packets of the CDP. Note that although from the point of view of this specification and of CDPs there is only a single Scheme-specific FEC Object Transmission Information element, the FEC scheme may specify this element to contain multiple distinct pieces of information.

Each FEC scheme specifies an encoding format for the Common and Scheme-specific FEC Object Transmission Information. Each CDP must specify at least one of the following:

1. A means to reliably communicate the Common FEC Object Transmission Information elements to the receiver(s) using the encoding format defined by the FEC scheme.

2. An alternative, CDP specific, encoding format for each of the Common FEC Object Transmission Information elements.

The Mandatory and Common FEC Object Transmission Information elements are defined in the sections below.

6.2.1. Transport of FEC Object Transmission Information

It is the responsibility of the CDP to reliably transport the FEC Object Transmission Information to the receiver(s).

It is important to note that the encoding format of the Mandatory FEC Object Transmission Information elements (The FEC Encoding ID) is defined by the CDP. This is so that the receiver can identify the FEC Scheme to be used for interpreting the remaining FEC Object Transmission Information elements. All CDPs must define encoding formats for the Mandatory FEC Object Transmission Information element.

Common FEC Object Transmission Information elements can be transported in two different ways: (a) the FEC Scheme defines an encoding format for the Common FEC Object Transmission Information elements that it uses and the CDP transports this encoded data block, or (b) the CDP defines an encoding format for each Common FEC Object Transmission Information element and transports the information in this format.

An FEC Scheme MUST define an encoding format for the Common FEC Object Transmission Information elements that it uses. The resulting byte string is known as the "encoded Common FEC Object Transmission Information". A CDP MAY define individual encoding formats for each of the Common FEC Object Transmission Information elements. The CDP determines which way the Common FEC Object Transmission Information

elements shall be transported, (a) or (b). Note that a CDP may provide support for one or both options.

In the case that the CDP uses the encoding format specified by the FEC scheme then it may simply concatenate the encoded Common FEC Object Transmission Information and the encoded Scheme-specific FEC Object Transmission Information or it may carry each in a seperate field or wrapper within the CDP. In the former case, the concatenated byte-string is known as the encoded FEC Object Transmission Information. The FEC scheme must define the encoding format for the Common FEC Object Transmission Information elements that it uses in such a way that the length of each element is either fixed or can be determined from the encoded data itself.

The encoding format of the Scheme-specific FEC Object Transmission Information element is defined by the FEC scheme. CDPs specify only how the resulting byte sequence is communicated. As with the encoding format for the Common FEC Oject Transmission Information elements the length of the Scheme-specific FEC Object Transmission Information must either be fixed or it must be possible to determine the length from the encoded data itself.

6.2.2. Opacity of FEC Object Transmission Information

The Scheme-specific FEC Object Transmission Information element is opaque to the CDP in the sense that inspecting the contents of this element can only be done if FEC scheme-specific logic is included in the CDP.

The encoding formats defined by the FEC scheme for the Common FEC Object Transmission Information elements are also opaque to the CDP in the same sense.

The encoding formats defined by the CDP for the Common FEC Object Transmission Information elements are not opaque in this sense, although it must be considered that different FEC Schemes may use different combinations of the Common FEC Object Transmission Information elements.

6.2.3. Mandatory FEC Object Transmission Information elements

The Mandatory FEC Object Transmission Information element is:

FEC Encoding ID: an integer between 0 and 255 inclusive identifying a specific FEC scheme (Fully-Specified or Under-Specified.)

6.2.4. Common FEC Object Transmission Information elements

The Common FEC Object Transmission Information elements are described below. Note that with the exception of the FEC Instance ID, this specification does not provide complete definitions of these fields. Instead only aspects of the abstract type are defined. The precise type and semantics are defined for each FEC scheme in the FEC scheme specification.

FEC Instance ID: an integer between 0 and 65535 inclusive identifying an instance of an Under-specifed FEC scheme

Transfer-Length: a non-negative integer indicating the length of the object in bytes

Encoding-Symbol-Length: a non-negative integer indicating the length of each encoding symbol in bytes

Maximum-Source-Block-Length: a non-negative integer indicating the maximum number of source symbols in a source block

Max-Number-of-Encoding-Symbols: a non-negative integer indicating the maximum number of encoding symbols (i.e. source plus repair symbols in the case of a systematic code)

The FEC Instance ID MUST be used by all Under-specified FEC schemes and MUST NOT be used by Fully-specified FEC Schemes.

FEC Schemes define the precise type of those of the above elements that they use and in particular may restrict the value range of each element. FEC Schemes also define an encoding format for the subset of the above elements that they use. CDPs may also provide an encoding format for each element in which case this encoding format MUST be capable of representing values up to (2^^16)-1 in the case of the FEC Instance ID, (2^^48)-1 in the case of the Transfer-Length and up to (2^^32)-1 for the other elements. CDPs may additionally or alternatively provide a mechanism to transport the encoded Common FEC Object Transmission information defined by the FEC scheme. For example, FLUTE [8] specifies an XML-based encoding format for these elements, but can also transport FEC scheme-specific encoding formats within the EXT-FTI LCT header extension.

6.2.5. Scheme-specific FEC Object Transmission Information element

The Scheme-specific FEC Object Transmission Information element may be used by an FEC Scheme to communicate information which is essential to the decoder which cannot adequately be represented within the Mandatory or Common FEC Object Transmission Information

elements.

From the point of view of a CDP, the Scheme-specific FEC Object Transmission Information element is an opaque, variable length, byte string. The FEC Scheme defines the structure of this byte string, which may contain multiple distinct elements.

6.3. FEC Payload ID

The FEC Payload ID contains information which indicates to the FEC decoder the relationships between the encoding symbols carried by a particular packet and the FEC encoding transformation. For example if the packet carries source symbols then the FEC Payload ID indicates which source symbols of the object are carried by the packet. If the packet carries repair symbols, then the FEC Payload ID indicates how those repair symbols were constructed from the object.

The FEC Payload ID may also contain information about larger groups of encoding symbols of which those contained in the packet are part. For example, the FEC Payload ID may contain information about the source block the symbols are related to.

The FEC Payload ID for a given packet is essential to the decoder if and only if the packet itself is received. Thus it must be possible to obtain the FEC Payload ID from the recieved packet. Usually, the FEC Payload ID is simply carried explicitly as a separate field within each packet. Some FEC schemes may specify means for deriving the relationship between the carried encoding symbols and the object implicitly from other information within the packet, such as protocol headers already present. Such FEC schemes could obviously only be used with CDPs which provided the appropriate information from which the FEC Payload ID could be derived.

The encoding format of the FEC Payload ID is defined by the FEC Scheme. CDPs specify how the FEC Payload ID is carried within data packets i.e. the position of the FEC Payload ID within the CDP packet format and the how it is associated with encoding symbols.

FEC schemes for systematic FEC codes may specify two FEC Payload ID formats, one for packets carrying only source symbols and another for packets carrying at least one repair symbol. CDPs must include an indication of which of the two FEC Payload ID formats is included in each packet if they wish to support such FEC Schemes.

7. FEC scheme specifications

A specification for a new FEC scheme MUST include the following things:

1. The FEC Encoding ID value that uniquely identifies the FEC scheme. This value MUST be registered with IANA as described in Section 12.

2. The type, semantics and encoding format of one or two FEC Payload IDs. Where two FEC Payload ID formats are specified, then the FEC scheme MUST be a systematic FEC code and one FEC Payload ID format MUST be designated for use with packets carrying only source symbols and the other FEC Payload ID format MUST be designated for use with packets carrying at least one repair symbol.

3. The type and semantics of the FEC Object Transmission Information. The FEC Scheme MAY define additional restrictions on the type (including value range) of the Common FEC Object Transmission Information elements.

4. An encoding format for the Common FEC Object Transmission Information elements used by the FEC Scheme.

Fully-Specified FEC schemes MUST further specify:

1. A full specification of the FEC code.

   This specification MUST precisely define the valid FEC Object Transmission Information values, the valid FEC Payload ID values and the valid packet payload sizes for any given object (where packet payload refers to the space - not necessarily contiguous - within a packet dedicated to carrying encoding symbol bytes).

   Furthermore, given an object, valid values for each of the FEC Object Transmission Information elements used by the FEC Scheme, a valid FEC Payload ID value and a valid packet payload size, the specification MUST uniquely define the values of the encoding symbol bytes to be included in the packet payload of a packet with the given FEC Payload ID value.

   A common and simple way to specify the FEC code to the required level of detail is to provide a precise specification of an encoding algorithm which, given an object, valid values for each of the FEC Object Transmission Information elements used by the FEC Scheme for the object, a valid FEC Payload ID and packet payload length as input produces the exact value of the encoding

symbol bytes as output.

2. A description of practical encoding and decoding algorithms.

   This description need not be to the same level of detail as for (1) above, however it must be sufficient to demonstrate that encoding and decoding of the code is both possible and practical.

FEC scheme specifications MAY additionally define the following:

1. Type, semantics and encoding format of a Scheme-specific FEC Object Transmission Information element.

Note that if an FEC sheme does not define a Scheme-specific FEC Object Transmission Information then such an element MUST NOT be introduced in future versions of the FEC Scheme. This requirement is included to ensure backwards-compatibility of CDPs designed to support only FEC Schemes which do not use the Scheme-specific FEC Object Transmission Information element.

Whenever an FEC scheme specification defines an 'encoding format' for an element, this must be defined in terms of a sequence of bytes which can be embedded within a protocol. The length of the encoding format MUST either be fixed or it must be possible to derive the length from examining the encoded bytes themselves. For example, the initial bytes may include some kind of length indication.

FEC schemes SHOULD make use of the Common FEC Object Transmission Information elements in preference to including infomation in a Scheme-specific FEC Object Transmission Information element.

FEC scheme specifications SHOULD use the terminology defined in this document and SHOULD follow the following format:

1. Introduction <define whether the scheme is Fully-Specified or Under-Specified>

   <describe the use-cases addressed by this FEC scheme>

2. Formats and Codes

   2.1 FEC Payload ID(s) <define the type and format of one or two FEC Payload IDs>

2.2 FEC Object Transmission Information

2.2.1 Mandatory <define the value of the FEC Encoding ID for this FEC scheme>

2.2.2 Common <describe which Common FEC Object Transmission Information elements are used by this FEC scheme, define their value ranges and define an encoding format for them>

2.2.3 Scheme-Specific <define the Scheme-specific FEC Object Transmission Information, including an encoding format, if required>

3. Procedures <describe any procedures which are specific to this FEC scheme, in particular derivation and interpretation of the fields in the FEC Payload ID and FEC Object Transmission Information.>

4. FEC code specification (for Fully-Specified FEC schemes only) <provide a complete specification of the FEC Code>

Specifications MAY include additional sections, for example, examples.

Each FEC scheme MUST be specified independently of all other FEC schemes; for example, in a separate specification or a completely independent section of larger specification.

8. CDP specifications

A specification for a CDP that uses this building block MUST include the following things:

1. Definitions of an encoding formats for the Mandatory FEC Object Transmission Information element.

2. A means to reliably communicate the Mandatory FEC Object Transmission Information element from sender to receiver(s) using the encoding format defined in (1).

3. Means to reliably communicate the Common FEC Object Transmission Information element from sender to receiver(s) using either or both of (a) the encoding format defined by the FEC Scheme or (b) encoding formats defined by the CDP

4. A means to reliably communicate the Scheme-specific FEC Object Transmission Information element from sender to receiver(s) using the encoding format of the Scheme-specific FEC Object Transmission Information element defined by the FEC scheme.

5. A means to communicate the FEC Payload ID in association with a data packet. Note that the encoding format of the FEC Payload ID is defined by the FEC Scheme.

If option (b) of (3) above is used, then the CDP MUST specify an encoding format for the Common FEC Object Transmission Information elements.

CDPs MAY additionally specify the following things:

1. A means to indicate whether the FEC Payload ID within a packet is encoded according to the format for packets including only source symbols or according to the format for packets including at least one repair symbol.

9. Common algorithms

This section describes certain algorithms which are expected to be commonly required by FEC schemes or by CDPs. FEC Schemes and CDPs SHOULD use these algorithms in preference to scheme or protocol specific algorithms where appropriate.

9.1. Block partitioning algorithm

This algorithm computes a partitioning of a object into source blocks so that all source blocks are as close to being equal length as possible. A first number of source blocks are of the same larger length, and the remaining second number of source blocks of the same smaller length.

This algorithm is described in two steps, the second of which may be useful in itself as an independent algorithm in some cases. In the first step, the number of source symbols (T) and the number of source blocks (N) are derived from the Object transfer length (L), Maximum Source Block Length (B) and Symbol Length (E).

In the second step, the partitioning of the object is derived from the number of source symbols (T) and the number of source blocks (N). The partitioning is defined in terms of a first number of source blocks (I), a second number of source blocks (N-I), the length of each of the first source blocks (A_large) and the length of each of the second source blocks (A_small).

The following notation is used in the description below:

ceil[x] denotes x rounded up to the nearest integer.

floor[x] denotes x rounded down to the nearest integer.

9.1.1. First Step

Input:

B -- Maximum Source Block Length, i.e., the maximum number of source symbols per source block

L -- Transfer Length in bytes

E -- Encoding Symbol Length in bytes

Output:

T -- the number of source symbols in the object.

N -- The number of source blocks into which the object shall be partitioned.

Algorithm:

1. The number of source symbols in the transport object is computed as T = ceil[L/E].

2. The transport object shall be partitioned into N = ceil[T/B] source blocks.

9.1.2. Second step

Input:

T -- the number of source symbols in the object.

N -- The number of source blocks into which the object is partitioned.

Output:

I -- the number of larger source blocks.

A_large -- the length of each of the larger source blocks in symbols.

A_small -- the length of each of the smaller source blocks in symbols.

Algorithm:

1. A_large = ceil[T/N]

2. A_small = floor[T/N]

3. I = T - A_small * N

Each of the first I source blocks then consists of A_large source symbols, each source symbol is E bytes in length. Each of the remaining N-I source blocks consist of A_small source symbols, each source symbol is E bytes in length except that the last source symbol of the last source block is L-((L-1)/E) rounded down to the nearest integer)*E bytes in length.

10. Requirements from other building blocks

The FEC building block does not provide any support for congestion control. Any complete CDP MUST provide congestion control that conforms to [6], and thus this MUST be provided by another building block when the FEC building block is used in a CDP.

There are no other specific requirements from other building blocks for the use of this FEC building block. However, any CDP that uses the FEC building block may use other building blocks for example to provide support for sending higher level session information within data packets containing FEC encoding symbols.

11. Security Considerations

Data delivery can be subject to denial-of-service attacks by attackers which send corrupted packets that are accepted as legitimate by receivers. This is particularly a concern for multicast delivery because a corrupted packet may be injected into the session close to the root of the multicast tree, in which case the corrupted packet will arrive to many receivers. This is particularly a concern for the FEC building block because the use of even one corrupted packet containing encoding data may result in the decoding of an object that is completely corrupted and unusable. It is thus RECOMMENDED that the decoded objects be checked for integrity before delivering objects to an application. For example, an MD5 hash [7] of an object may be appended before transmission, and the MD5 hash is computed and checked after the object is decoded but before it is delivered to an application. Moreover, in order to obtain strong cryptographic integrity protection a digital signature verifiable by the receiver SHOULD be computed on top of such a hash value. It is also RECOMMENDED that a packet authentication protocol such as TESLA [9] be used to detect and discard corrupted packets upon arrival. Furthermore, it is RECOMMENDED that Reverse Path Forwarding checks be enabled in all network routers and switches along the path from the sender to receivers to limit the possibility of a bad agent successfully injecting a corrupted packet into the multicast tree data path.

Another security concern is that some FEC information may be obtained by receivers out-of-band in a session description, and if the session description is forged or corrupted then the receivers will not use the correct protocol for decoding content from received packets. To avoid these problems, it is RECOMMENDED that measures be taken to prevent receivers from accepting incorrect session descriptions, e.g., by using source authentication to ensure that receivers only accept legitimate session descriptions from authorized senders.

12. IANA Considerations

Values of FEC Encoding IDs and FEC Instance IDs are subject to IANA registration. FEC Encoding IDs and FEC Instance IDs are hierarchical: FEC Encoding IDs scope independent ranges of FEC Instance IDs. Only FEC Encoding IDs that correspond to Under-Specified FEC schemes scope a corresponding set of FEC Instance IDs.

The FEC Encoding ID and FEC Instance IDs are non-negative integers. In this document, the range of values for FEC Encoding IDs is 0 to 255. Values from 0 to 127 are reserved for Fully-Specified FEC schemes and Values from 128 to 255 are reserved for Under-Specified FEC schemes, as described in more detail in Section 6.1.

12.1. Explicit IANA Assignment Guidelines

This document defines a name-space for FEC Encoding IDs named:

ietf:rmt:fec:encoding

The values that can be assigned within the "ietf:rmt:fec:encoding" name-space are numeric indexes in the range [0, 255], boundaries included. Assignment requests are granted on a "IETF Consensus" basis as defined in [2].

This document also defines a name-space for FEC Instance IDs named:

ietf:rmt:fec:encoding:instance

The "ietf:rmt:fec:encoding:instance" name-space is a sub-name-space associated with the "ietf:rmt:fec:encoding" name-space. Each value of "ietf:rmt:fec:encoding" assigned in the range [128, 255] has a separate "ietf:rmt:fec:encoding:instance" sub-name-space that it scopes. Values of "ietf:rmt:fec:encoding" in the range [0, 127] do not scope a "ietf:rmt:fec:encoding:instance" sub-name-space.

The values that can be assigned within each "ietf:rmt:fec:encoding:instance" sub-name-space are non-negative integers less than 65536. Assignment requests are granted on a "First Come First Served" basis as defined in [2]. The same value of "ietf:rmt:fec:encoding:instance" can be assigned within multiple distinct sub-name-spaces, i.e., the same value of "ietf:rmt:fec:encoding:instance" can be used for multiple values of "ietf:rmt:fec:encoding".

Requestors of "ietf:rmt:fec:encoding:instance" assignments MUST provide the following information:

- The value of "ietf:rmt:fec:encoding" that scopes the "ietf:rmt:fec:encoding:instance" sub-name-space. This must be in the range [128, 255].

- Point of contact information

- A pointer to publicly accessible documentation describing the Under-Specified FEC scheme, associated with the value of "ietf: rmt:fec:encoding:instance" assigned, and a way to obtain it (e.g., a pointer to a publicly available reference-implementation or the name and contacts of a company that sells it, either separately or embedded in a product).

It is the responsibility of the requestor to keep all the above information up to date.

13. Changes from RFC3452

This section lists the changes between the Experimental version of this specification, [3], and this version:

- The requirements for definition of a new FEC Scheme and the requirements for specification of new Content Delivery Protocols which use FEC Schemes are made more explicit to permit independent definition of FEC Schemes and Content Delivery Protocols.

- The definitions of basic FEC Schemes have been removed with the intention of publishing these separately.

- The FEC Object Transmission Information is more explicitly defined and in particular three classes of FEC OTI (Mandatory, Common and Scheme-specific) are introduced to permit re-usable definition of explicit fields in Content Delivery Protocols to carry these elements.

- FEC Schemes are required to specify a complete encoding for the FEC Object Transmission which can be carried transparently by Content Delivery protocols (instead of defining explicit elements).

- The possibility for FEC Schemes to define two FEC Payload ID formats for use with souce and repair packets respectively in the case of systematic FEC codes is introduced.

- The file blocking algorithm from FLUTE is included here as a common algorithm which is recommended to be re-used by FEC Schemes when appropriate.

14. Acknowledgments

This document is largely based on RFC3452 [3] and thus thanks are due to the additional authors of that document: J. Gemmell, L. Rizzo, M. Handley, J. Crowcroft.

15. References

15.1. Normative References

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

[2] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

15.2. Informative References

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

[4] 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.

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

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

[7] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.

[8] Paila, T., Luby, M., Lehtonen, R., Roca, V., and R. Walsh, "FLUTE - File Delivery over Unidirectional Transport", RFC 3926, October 2004.

[9] Perrig, A., Song, D., Canetti, R., Tygar, J., and B. Briscoe, "Timed Efficient Stream Loss-Tolerant Authentication (TESLA): Multicast Source Authentication Transform Introduction", RFC 4082, June 2005.

[10] 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.

Authors' Addresses

Mark Watson
Digital Fountain
39141 Civic Center Drive
Suite 300
Fremont, CA 94538
U.S.A.

Email: mark@digitalfountain.com

Michael Luby
Digital Fountain
39141 Civic Center Drive
Suite 300
Fremont, CA 94538
U.S.A.

Email: luby@digitalfountain.com

Lorenzo Vicisano
Cisco Systems, Inc.
170 West Tasman Dr.
San Jose, CA 95134
U.S.A.

Email: lorenzo@cisco.com

Full Copyright Statement

Copyright (C) The Internet Society (2006).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgment

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