

FECFRAME Working Group
Internet Draft
Intended status: Informational
Expires: July 2012

Rajiv Asati
Cisco Systems

June 8, 2012

Methods to convey FEC Framework Configuration Information
draft-ietf-fecframe-config-signaling-09.txt

Abstract

FEC Framework document [RFC6363] defines the FEC Framework Configuration Information necessary for the FEC framework operation. This document describes how to use signaling protocols such as Session Announcement Protocol (SAP), Session Initiation Protocol (SIP), Real Time Stream Protocol (RTSP) etc. for determining and communicating the Configuration information between sender(s) and receiver(s).

This document doesn't define any new signaling protocol.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79. This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

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/lid-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 November 8, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction.....	3
2. Specification Language.....	4
3. Terminology/Abbreviations.....	4
4. FEC Framework Configuration Information.....	5
4.1. Encoding Format.....	6
5. Signaling Protocol Usage.....	7
5.1. Signaling Protocol for Multicasting.....	8
5.1.1. Sender Procedure.....	9
5.1.2. Receiver Procedure.....	12
5.2. Signaling Protocol for Unicasting.....	13
5.2.1. SIP.....	13
5.2.2. RTSP.....	14
6. Security Considerations.....	15
7. IANA Considerations.....	15

8. Acknowledgments.....	15
9. References.....	16
9.1. Normative References.....	16
9.2. Informative References.....	16
Author's Addresses.....	17

1. Introduction

FEC Framework document [RFC6363] defines the FEC Framework Configuration Information that governs the overall FEC framework operation common to any FEC scheme. This information must be available at both sender and receiver(s).

This document describes how various signaling protocols such as Session Announcement Protocol (SAP)[RFC2974], Session Initiation Protocol (SIP)[RFC3261], Real Time Stream Protocol (RTSP)[RFC2326] etc. could be used by the FEC scheme (and/or Content Delivery Protocol (CDP)) to communicate the Configuration information between sender and receiver(s). The configuration information may be encoded in any compatible format such as SDP [RFC4566], XML etc., though this document references to SDP encoding usage quite extensively.

Note that this document doesn't define any new signaling protocol; rather it just provides examples of how existing protocols should be used. Also, the list of signaling protocols for unicast is not intended to be a complete list.

This document doesn't describe any FEC scheme specific information (FSSI) (for example, how source blocks are constructed) or any sender or receiver side operation for a particular FEC scheme (for example, whether the receiver makes use of one or more repair flows that are received). Such FEC scheme specifics should be covered in separate document(s). This document doesn't mandate a particular encoding format for the configuration information either.

This document is structured such that Section 2 describes the terms used in this document, section 4 describes the FEC Framework Configuration Information, section 5 describes how to use signaling protocol for the multicast and unicast applications, and section 6 describes security consideration.

2. Specification Language

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. Terminology/Abbreviations

This document makes use of the terms/abbreviations defined in the FEC Framework document [RFC6363] and defines the following additional terms:

- o Media Sender - Node providing original media flow(s) to the 'FEC Sender'
- o Media Receiver - Node performing the Media decoding;
- o FEC Sender - Node performing the FEC encoding on the original media flow(s) to produce the FEC repair flow(s)
- o FEC Receiver - Node performing the FEC decoding, as needed, and providing the original media flow(s) to the Media receiver.
- o Sender - Same as FEC Sender
- o Receiver - Same as FEC Receiver
- o (Media) Flow - A single media instance i.e., an audio stream or a video stream.

This document deliberately refers to the 'FEC Sender' and 'FEC Receiver' as the 'Sender' and 'Receiver' respectively.

4. FEC Framework Configuration Information

The FEC Framework [RFC6363] defines a minimum set of information that is communicated between the sender and receiver(s) for a proper operation of an FEC scheme. This information is referred to as "FEC Framework Configuration Information". This is the information that the FEC Framework needs in order to apply FEC protection to the transport flows.

A single instance of the FEC Framework provides FEC protection for all packets of a specified set of source packet flows, by means of one or more packet flows consisting of repair packets. As per the FEC Framework document [RFC6363] section 6.5, the FEC Framework Configuration Information includes the following for each FEC Framework instance:

1. Identification of the repair flow(s)
2. Identification of Source Flow(s)
3. Identification of FEC Scheme
4. Length of Explicit Source FEC payload ID
5. FEC Scheme Specific Information (FSSI)

FSSI basically provides an opaque container to encode FEC scheme specific configuration information such as buffer size, decoding wait-time etc. Please refer to the FEC Framework document [RFC6363] for more details.

The usage of signaling protocols described in this document requires that the application layer responsible for the FEC Framework instance provide the value for each of the configuration information parameter (listed above) encoded as per the chosen encoding format. In case of failure to receive the complete information, the signaling protocol module must return an error for the Operation, Administration and Maintenance (OAM) purposes and optionally convey this error to the application layer. Please refer to the figure 1 of the FEC Framework document [RFC6363] for further illustration.

This document does not make any assumption that the 'FEC sender' and 'Media Sender' functionalities are implemented on the same device,

though that may be the case. Similarly, this document does not make any assumption that 'FEC receiver' and 'Media Receiver' functionalities are implemented on the same device, though that may be the case. There may also be more than one Media Sender.

4.1. Encoding Format

The FEC Framework Configuration Information (listed above in section 4) may be encoded in any format such as SDP, XML etc. as chosen or preferred by a particular FEC Framework instance. The selection of such encoding format or syntax is independent of the signaling protocol and beyond the scope of this document.

Whatever encoding format is selected for a particular FEC framework instance, it must be known to the signaling protocol. This is to provide a means (e.g. a field such as Payload Type) in the signaling protocol message(s) to convey the chosen encoding format for the configuration information so that the Payload i.e., configuration information can be correctly parsed as per the semantics of the chosen encoding format at the receiver. Please note that the encoding format is not a negotiated parameter, but rather a property of a particular FEC Framework instance and/or its implementation.

Additionally, the encoding format for each FEC Framework configuration parameter must be defined in terms of a sequence of octets that can be embedded within the payload of the signaling protocol message(s). The length of the encoding format must either be fixed, or derived by examining the encoded octets themselves. For example, the initial octets may include some kind of length indication.

Independent of the encoding formats supported by an FEC scheme, each instance of the FEC Framework must use a single encoding format to describe all of the configuration information associated with that instance. The signaling protocol specified in this document should not validate the encoded information, though it may validate the syntax or length of the encoded information.

The reader may refer to the SDP elements document [RFC6364], which describes the usage of 'SDP' encoding format as an example encoding format for FEC Framework Configuration Information.

5. Signaling Protocol Usage

FEC Framework [RFC6363] requires certain FEC Framework Configuration Information to be available to both sender and receiver(s). This configuration information is almost always formulated at the sender (or on behalf of a sender), and somehow made available at the receiver(s). While one may envision a static method to populate the configuration information at both sender and receiver(s), it would not be optimal since it would (a) require the knowledge of every receiver in advance, (b) require the time and means to configure each receiver and sender, and (c) increase the misconfiguration possibility. Hence, there is a benefit in using a dynamic method i.e., signaling protocol to convey the configuration information between sender and one or more receivers.

Since the configuration information may be needed at a particular receiver versus many receivers (depending on the multimedia stream being unicast e.g. Video on Demand, or multicast e.g. Broadcast or IPTV), we need two types of signaling protocols - one to deliver the configuration information to many receivers via multicasting (described in section 5.1), and the other to deliver the configuration information to one and only one receiver via unicasting (described in section 5.2).

Figure 1 below illustrates a sample topology showing the FEC sender and FEC receiver (that may or may not be the Media Sender and Media Receiver respectively) such that FEC_Sender1 is serving FEC_Receiver11,12,13 via the multicast signaling protocol, whereas the FEC_Sender2 is serving only FEC_Receiver2 via the unicast signaling protocol.

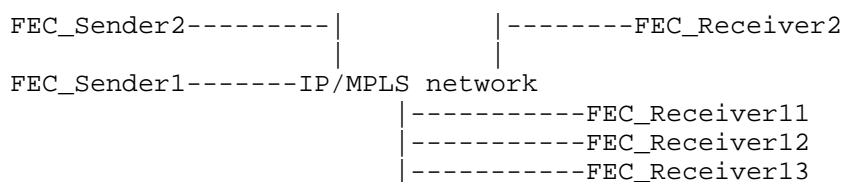


Figure 1 Topology using Sender and Receiver

The rest of the document continues to use the terms 'Sender' and 'Receiver' to refer to the 'FEC Sender' and 'FEC Receiver' respectively.

5.1. Signaling Protocol for Multicasting

This specification describes using Session Announcement Protocol (SAP) version 2 [RFC2974] as the signaling protocol to multicast the configuration information from one sender to many receivers. The apparent advantage is that the server doesn't need to maintain any state for any receiver using SAP.

SAP messages are carried over UDP over IP with destination UDP port being 9875 and source UDP port being any available number, as described in RFC2974. The SAP message(s) MUST contain an authentication header using GPG authentication.

At the high level, a sender, acting as the SAP announcer, signals the FEC Framework Configuration Information for each FEC Framework instance available at the sender, using the SAP message(s). The configuration information, encoded in a suitable format as per the section 4.1, is carried in the Payload of the SAP message(s). A receiver, acting as the SAP listener, listens on a well-known UDP port and at least one well known multicast group IP address (as explained in the section 5.1.1). This enables the receiver to receive the SAP message(s) and obtains the FEC Framework Configuration Information for each FEC Framework Instance.

Using the configuration information, the receiver becomes aware of available FEC protection options, corresponding multicast trees (S,G or *,G addresses) etc. The receiver may subsequently subscribe to one or more multicast trees to receive the FEC streams using out-of-band multicasting techniques such as PIM [RFC4601]. This, however, is outside the scope of this document.

Figure 2 below illustrates the SAP packet format (it is reprinted from the RFC2974) -

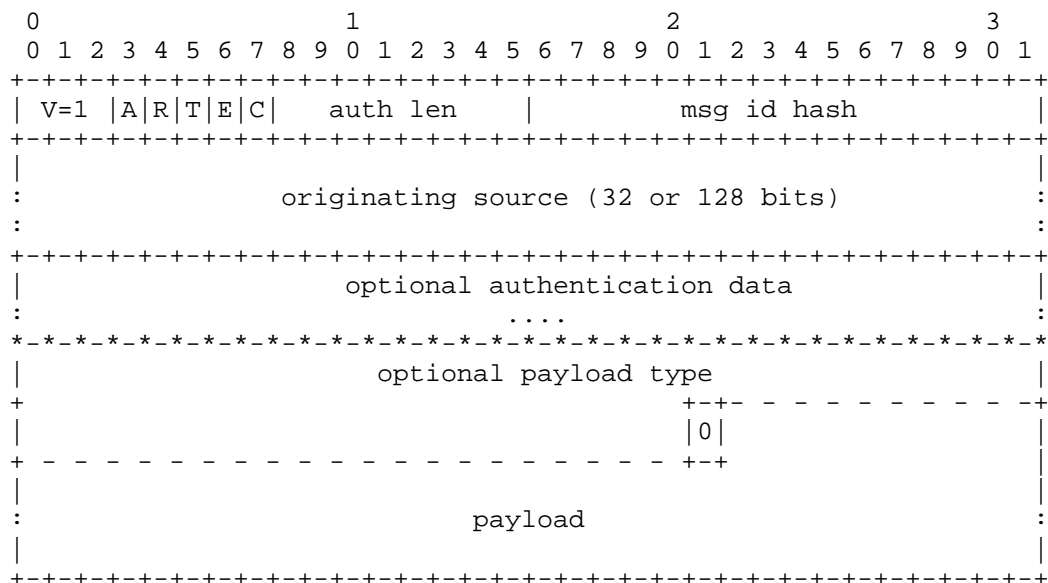


Figure 2 SAP Message format

While the RFC2974 includes explanation for each field, it is worth discussing the 'Payload' and 'Payload Type' fields. The 'Payload' field is used to carry the FEC Framework Configuration Information. Subsequently, the optional 'Payload Type' field, which is a MIME content type specifier, is used to describe the encoding format used to encode the Payload.

For example, the 'Payload Type' field may be application/sdp if the FEC Framework Configuration Information is encoded in SDP format and carried in the SAP payload. Similarly, it would be application/xml if the FEC Framework Configuration Information was encoded in XML format.

Section 5.1.1 describes the sender procedure, whereas the section 5.1.2 describes the receiver procedure in the context of config signaling using RFC2974.

5.1.1. Sender Procedure

The sender signals the FEC framework configuration for each FEC framework instance in a periodic SAP announcement message [RFC2974]. The SAP announcement message is sent to a well known multicast IP

address and UDP port, as specified in [RFC2974]. The announcement is multicast with the same scope as the session being announced.

The SAP module at the sender obtains the FEC Framework Configuration Information per Instance from the 'FEC Framework' module and places that in the SAP payload accordingly. A single SAP (announcement) message must carry the FEC Framework Configuration Information for a single FEC Framework Instance. The SAP message is then sent over UDP over IP.

While it is possible to aggregate multiple SAP (announcement) messages in a single UDP datagram as long as the resulting UDP datagram length is less than the IP MTU of the outgoing interface, this specification does not recommend it since there is no length field in the SAP header to identify SAP message boundary. Hence, this specification recommends single SAP announcement message to be sent in a UDP datagram.

The IP packet carrying the SAP message must be sent to destination IP address of one of the following depending on the selected scope:

- 224.2.127.254 (if IPv4 global scope 224.0.1.0-238.255.255.255 is selected for the FEC stream), or
- FF0X:0:0:0:0:0:2:7FFE (if IPv6 multicasting is selected for the FEC stream, where X is the 4-bit scope value), or
- the highest multicast address (239.255.255.255, for example) in the relevant administrative scope zone (if IPv4 administrative scope 239.0.0.0-239.255.255.255 is selected for the FEC stream)

As defined in RFC2974, the IP packet carrying SAP message must use destination UDP port being 9875 and source UDP port being any available number. The default IP TTL value (or Hop Limit value) should be 255 at the sender, though the sender implementation may allow it to be any other value to implicitly create the multicast boundary for SAP announcements. The IP DSCP field may be set to any value that indicates a desired QoS treatment in the IP network.

The IP packet carrying the SAP message must be sent with source IP address that is reachable by the receiver. The sender may assign the same IP address in the "originating source" field of the SAP message, as the one used in the source IP address of the IP packet.

Furthermore, the FEC Framework Configuration Information must not include any of the reserved multicast group IP addresses for the FEC streams (i.e., source or repair flows), though it may use the same

IP address as the 'originating source' address to identify the FEC streams (i.e., source or repair flows). Please refer to IANA assignments for multicast addresses.

The sender must periodically send the 'SAP announcement' message to ensure that the receiver doesn't purge the cached entry(s) from the database and doesn't trigger the deletion of FEC Framework Configuration Information.

While the time interval between repetitions of an announcement can be calculated as per the very sophisticated but complex method explained in [RFC2974], this document recommends a simpler method in which the user specifies the time interval in the range of 1-200 seconds with suggested default value being 60 seconds. In this method, the 'time interval' may be signaled in the SAP message payload e.g. within the FEC Framework Configuration Information.

Note that SAP doesn't allow the time-interval to be signaled in the SAP header. Hence, the usage of simpler method requires the time-interval to be included in the FEC Framework Configuration Information, if the default time interval (=60 seconds) for SAP message repetitions is not used. For example, the usage of "r=" (repeat time) field in SDP may convey the time-interval value, if SDP encoding format is used.

The time interval must be chosen to ensure that SAP announcement messages are sent out before the corresponding multicast routing entry e.g. (S,G) or (*,G) (corresponding to the SAP multicast tree(s)) on the router(s) times out. (It is worth noting that the default time-out period for the multicast routing entry is 210 seconds, per the PIM specification [RFC4601], though the time-out period may be set to another value as allowed by the router implementation.)

A SAP implementation may also support the complex method for determining the SAP announcement time interval, and provide the option to select it.

The sender may choose to delete the announced FEC Framework Configuration Information, as defined in section 4 of RFC2974. The explicit deletion is useful if the sender no longer desires to send anymore FEC streams.

If the sender needs to modify the announced FEC Framework Configuration Information for one or more FEC instances, then the sender must send a new announcement message with a different 'Message Identifier Hash' value as per the rules described in

section 5 of RFC2974 [RFC2974]. Such announcement message should be sent immediately (without having to wait for the time-interval) to ensure that the modifications are received by the receiver as soon as possible. The sender must also send the SAP deletion message to delete the previous SAP announcement message (i.e., with the previous 'Message Identifier Hash' value).

5.1.2. Receiver Procedure

The receiver must listen on UDP port 9875 for packets arriving with IP destination address of either 224.2.127.254 (if IPv4 global scope session is used for the FEC stream), or FF0X:0:0:0:0:0:2:7FFE (if IPv6 is selected, where X is the 4-bit scope value), or the highest IP address (239.255.255.255, for example) in the relevant administrative scope zone (if IPv4 administrative scope 239.0.0.0-239.255.255.255 is selected for the FEC stream). These IP addresses are mandated for SAP usage by RFC2974 [RFC2974].

The receiver, upon receiving a SAP announcement message, creates an entry, if it doesn't already exist, in a local database and passes the FEC Framework Configuration Information from the SAP Payload field to the 'FEC Framework' module. Each entry also maintains a time-out value, which is (re)set to five times the time-interval value, which is either the default = 60 seconds, or the value signaled by the sender.

Note that SAP doesn't allow the time-interval to be signaled in the SAP header. Hence, the time-interval should be included in the FEC Framework Configuration Information. For example, the usage of "r=" (repeat time) field in SDP to convey the time-interval value, if SDP encoding format is used.

The time-out value associated with each entry is reset when the corresponding announcement (please see section 5 of [RFC2974]) is received. If the time-out value for any entry reaches zero, then that entry must be deleted from the database, as described in section 4 of [RFC2974]. The receiver, upon receiving a SAP delete message, must delete the matching SAP entry in its database, as described in section 4 of [RFC2974].

The deletion of SAP entry must result in the receiver no longer using the relevant FEC Framework Configuration Information for the corresponding instance, and must no longer subscribe to any related FEC streams.

5.2. Signaling Protocol for Unicasting

This document describes leveraging any signaling protocol that is already used by the unicast application, for exchanging the FEC Framework Configuration Information between two nodes.

For example, a multimedia (VoD) client may send a request via unicasting for a particular content to the multimedia (VoD) server, which may offer various options such as encodings, bitrates, transport etc. for the content. The client selects the suitable options and answers to the server, paving the way for the content to be unicast on the chosen transport from server to the client. This offer/answer signaling, described in [RFC3264], is commonly utilized by many application protocols such as SIP, RTSP etc.

The fact that two nodes desiring unicast communication almost always rely on an application to first exchange the application related parameters via the signaling protocol makes it logical to enhance such signaling protocol(s) to (a) convey the desire for the FEC protection and (b) subsequently also exchange FEC parameters i.e., FEC Framework Configuration Information. This enables the node acting as the offerer to offer 'FEC Framework Configuration Information' for each of available FEC instances, and the node acting as the answerer conveying the chosen FEC Framework instance(s) to the offerer. The usage of FEC framework instance is explained the FEC Framework document [RFC6363].

While enhancing an application's signaling protocol to exchange FEC parameters is one method (briefly explained above), an alternative method would be to have a unicast based generic protocol that could be used by two nodes independent of the application's signaling protocol. The latter is not covered by this document, of course.

The remainder of this section provides example signaling protocols and explains how they can be used to exchange FEC Framework Configuration Information.

5.2.1. SIP

SIP [RFC3261] is an application-level signaling protocol to create, modify, and terminate multimedia sessions with one or more participants. SIP also enables the participants to discover one another and to agree on a characterization of a multimedia session

they would like to share. SIP runs on either TCP or UDP or SCTP transport, and uses SDP as the encoding format to describe multimedia session attributes.

SIP already uses an offer/answer model with SDP, described in [RFC3264], to exchange the information between two nodes to establish unicast sessions between them. This document extends the usage of this model for exchanging the FEC Framework Configuration Information, explained in section 4. Any SDP specific enhancements to accommodate the FEC Framework are covered in the SDP Elements specification [RFC6364].

5.2.2. RTSP

Real-Time Streaming Protocol (RTSP) [RFC2326] is an application-level signaling protocol for control over the delivery of data with real-time properties. RTSP provides an extensible framework to enable controlled, on-demand delivery of real-time data, such as audio and video. RTSP runs on either TCP or UDP transports.

RTSP already provides an ability to extend the existing method with new parameters. This specification defines 'FEC Protection Needed' option-tag (please see section 7 for IANA Considerations) and prescribes including it in the Require (or Proxy-Require) header of SETUP (method) request message, so as to request for FEC protection for the data.

The node receiving such request either responds with "200 OK" message that includes offers i.e., available FEC options (e.g. FEC Framework Configuration Information for each Instance) or "551 Option not supported" message. A sample of related message exchange is shown below -

```
Node1->Node2:  SETUP < ... > RTSP/1.0
                  CSeq: 1
                  Transport: <omitted for simplicity>
                  Require: FEC-protection-needed

Node2->Node1:  RTSP/1.0 200 OK
                  CSeq: 1
                  Transport: <omitted for simplicity>
```

The requesting node (Node1) may then send a new SETUP message to convey the selected FEC protection to Node2, and proceed with regular RTSP messaging.

Suffice to say, if the requesting node (Node1) received '551 Option not supported' response from Node2, then the requesting node (Node1) may send the SETUP message without using the Require header.

6. Security Considerations

This document recommends SAP message(s) be authenticated to ensure sender authentication, as described in section 5.1.

There is no additional security consideration other than what's already covered in [RFC2974] for SAP, [RFC2326] for RTSP, and [RFC3261] for SIP.

7. IANA Considerations

This document requests IANA to register a new RTSP Option tag (option-tag) listed below in the RTSP/1.0 Option Tags table of the "Real Time Streaming Protocol (RTSP)/1.0 Parameters" registry available from <http://www.iana.org/>, and provides the following information in compliance with section 3.8.1 in [RFC2326]:

- . Name of option-tag = FEC-protection-needed
- . Description = See section 5.2.2
- . Change of Control = IETF

8. Acknowledgments

Thanks to Colin Perkins for pointing out the issue with the time-interval for the SAP messages. Additionally, thanks to Vincent Roca, Ali Begen, Mark Watson, Ulas Kozat and David Harrington for greatly improving this document.

This document was prepared using 2-Word-v2.0.template.dot.

9. References

9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC6363] Watson, M., "Forward Error Correction (FEC) Framework", RFC6363, March 2011.
- [RFC6364] Begen, A., "Session Description Protocol Elements for FEC Framework ", RFC6364, October 2011.
- [RFC2974] Handley, M., Perkins, C. and E. Whelan, "Session Announcement Protocol", RFC 2974, October 2000.

9.2. Informative References

- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.
- [RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.
- [RFC2326] Schulzrinne, H., Rao, A. and R. Lanphier, "Real Time Streaming Protocol (RTSP)", RFC 2326, April 1998.
- [RFC3261] Handley, M., Schulzrinne, H., Schooler, E. and J. Rosenberg, "SIP: Session Initiation Protocol", RFC 3261, June 2002.
- [RFC4601] Fenner, etc., "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification", RFC 4601, August 2006.
- [RFC3547] Baugher, etc., "The Group Domain of Interpretation", RFC 3547, July 2003.

Author's Addresses

Rajiv Asati
Cisco Systems,
7025-6 Kit Creek Rd, RTP, NC, 27709-4987
Email: rajiva@cisco.com

FEC Framework
Internet-Draft
Intended status: Informational
Expires: June 18, 2010

A. Begen
Cisco
T. Stockhammer
Nomor Research
December 15, 2009

Guidelines for Implementing DVB-IPTV Application-Layer Hybrid FEC
Protection
draft-ietf-fecframe-dvb-al-fec-04

Abstract

The Annex E of the Digital Video Broadcasting (DVB)-IPTV technical specification defines an optional Application-layer Forward Error Correction (AL-FEC) protocol to protect the streaming media carried over RTP transport. The DVB-IPTV AL-FEC protocol uses two layers for FEC protection. The first (base) layer is based on the 1-D interleaved parity code. The second (enhancement) layer is based on the Raptor code. By offering a layered approach, the DVB-IPTV AL-FEC protocol offers a good protection against both bursty and random packet losses at a cost of decent complexity. This document describes how one can implement the DVB-IPTV AL-FEC protocol by using the 1-D interleaved parity code and Raptor code that have already been specified in separate documents.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and 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/lid-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 June 18, 2010.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the BSD License.

Table of Contents

1. Introduction	3
2. DVB-IPTV AL-FEC Specification	5
2.1. Base-Layer FEC	5
2.2. Enhancement-Layer FEC	7
2.3. Hybrid Decoding Procedures	8
3. Session Description Protocol (SDP) Signaling	8
4. Security Considerations	9
5. IANA Considerations	10
6. Acknowledgments	10
7. References	10
7.1. Normative References	10
7.2. Informative References	11
Authors' Addresses	11

1. Introduction

In 2007, the IP Infrastructure (IPI) Technical Module (TM) of the Digital Video Broadcasting (DVB) consortium published a technical specification [ETSI-TS-102-034v1.3.1] through European Telecommunications Standards Institute (ETSI). [ETSI-TS-102-034v1.3.1] covers several areas related to the transmission of MPEG2 transport stream-based services over IP networks.

The Annex E of [ETSI-TS-102-034v1.3.1] defines an optional protocol for Application-layer Forward Error Correction (AL-FEC) to protect the streaming media for DVB-IP services carried over RTP [RFC3550] transport. In 2009, DVB updated the specification in a new revision that is available as [ETSI-TS-102-034v1.4.1]. Among others, some updates and modifications to the AL-FEC protocol have been made. This document describes how one can implement the DVB-IPTV AL-FEC protocol by using the 1-D interleaved parity code [I-D.ietf-fecframe-interleaved-fec-scheme] and Raptor code specifications [I-D.ietf-fecframe-raptor], [I-D.watson-fecframe-rtp-raptor].

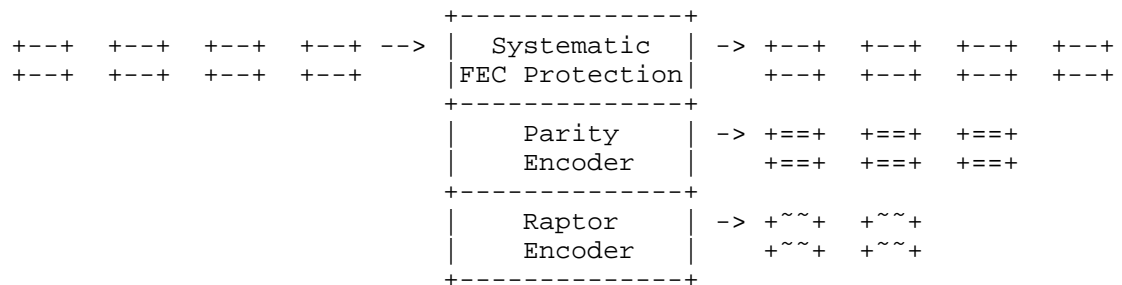
The DVB-IPTV AL-FEC protocol uses two layers for protection: a base layer that is produced by the 1-D interleaved parity code (also simply referred to as parity code in the remainder of this document), and an enhancement layer that is produced by the Raptor code. Whenever a receiver supports the DVB-IPTV AL-FEC protocol, the decoding support for the base-layer FEC is mandatory while the decoding support for the enhancement-layer FEC is optional. Both the interleaved parity code and the Raptor code are systematic FEC codes, meaning that source packets are not modified in any way during the FEC encoding process.

The DVB-IPTV AL-FEC protocol considers protection of single-sequence source RTP flows only. In the AL-FEC protocol, the source stream can only be an MPEG-2 transport stream. The FEC data at each layer are generated based on some configuration information, which also determines the exact associations and relationships between the source and repair packets. This document shows how this configuration may be communicated out-of-band in the Session Description Protocol (SDP) [RFC4566].

In DVB-IPTV AL-FEC, the source packets are carried in the source RTP stream and the generated FEC repair packets at each layer are carried in separate streams. At the receiver side, if all of the source packets are successfully received, there is no need for FEC recovery and the repair packets may be discarded. However, if there are missing source packets, the repair packets can be used to recover the

missing information.

The block diagram of the encoder side for the systematic DVB-IPTV AL-FEC protection is sketched in Figure 1. Here, the source packets are fed into the parity encoder to produce the parity repair packets. The source packets may also be fed to the Raptor encoder to produce the Raptor repair packets. Source packets as well as the repair packets are then sent to the receiver(s) over an IP network.



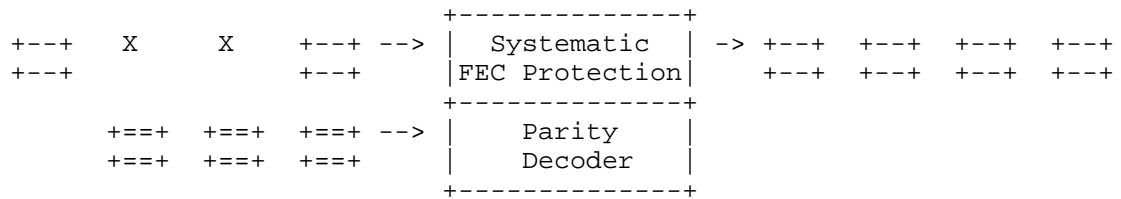
Source Packet: +---+
+---+

Base-layer Repair Packet: +==+
+==+

Enhancement-layer Repair Packet: +~~+
+~~+

Figure 1: Block diagram for the DVB-IPTV AL-FEC encoder

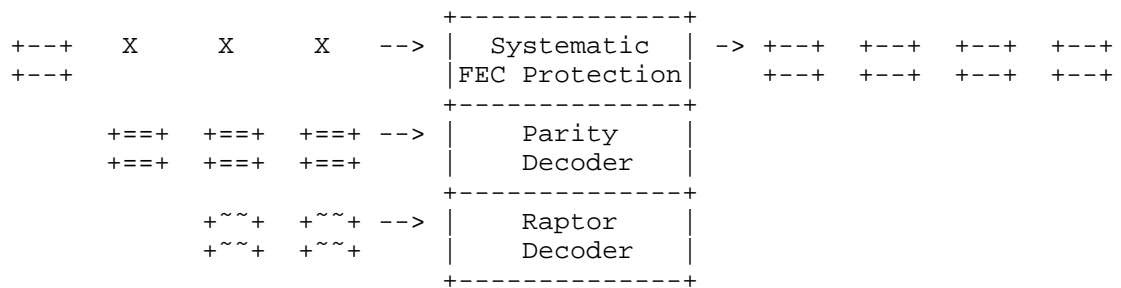
The block diagram of the decoder side for the systematic DVB-IPTV AL-FEC protection is sketched in Figure 2. This is a Minimum Performance Decoder since the receiver only supports decoding the base-layer repair packets. If there is a loss among the source packets, the parity decoder attempts to recover the missing source packets by using the base-layer repair packets.



Lost Packet: X

Figure 2: Block diagram for the DVB-IPTV AL-FEC minimum performance decoder

On the other hand, if the receiver supports decoding both the base-layer and enhancement-layer repair packets, a combined (hybrid) decoding approach is employed to improve the recovery rate of the lost packets. In this case, the decoder is called an Enhanced Decoder. Section 2.3 outlines the procedures for hybrid decoding.



Lost Packet: X

Figure 3: Block diagram for the DVB-IPTV AL-FEC enhanced decoder

2. DVB-IPTV AL-FEC Specification

The DVB-IPTV AL-FEC protocol comprises two layers of FEC protection: 1-D interleaved parity FEC for the base layer and Raptor FEC for the enhancement layer. The performance of these FEC codes has been examined in detail in [DVB-A115].

2.1. Base-Layer FEC

The 1-D interleaved parity FEC uses the exclusive OR (XOR) operation to generate the repair symbols. In a group of $D \times L$ source packets, the XOR operation is applied to each group of D source packets whose

sequence numbers are L apart from each other to generate a total of L repair packets. Due to interleaving, this FEC is effective against bursty packet losses up to burst sizes of length L .

The DVB-IPTV AL-FEC protocol requires the $D \times L$ block of the source packets protected by the 1-D interleaved FEC code to be wholly contained within a single source block of the Raptor code, if both FEC layers are used.

Originally, the DVB-IPTV AL-FEC protocol had adopted the 1-D interleaved FEC payload format from [SMPTE2022-1] in [ETSI-TS-102-034v1.3.1]. However, some incompatibilities with RTP [RFC3550] have been discovered in this specification. These issues have all been addressed in [I-D.ietf-fecframe-interleaved-fec-scheme] (For details, refer to Section 1 of [I-D.ietf-fecframe-interleaved-fec-scheme]). Some of the changes required by [I-D.ietf-fecframe-interleaved-fec-scheme] are, however, not backward compatible with the existing implementations that were based on [SMPTE2022-1].

In a recent liaison from IETF AVT WG to DVB TM-IPI, it has been recommended that DVB TM-IPI defines a new RTP profile for the AL-FEC protocol since in the new profile, several of the issues could easily be addressed without jeopardizing the compliance to RTP [RFC3550].

At the writing of this document, it was not clear whether or not a new RTP profile would be defined for the AL-FEC protocol. DVB TM-IPI attempted to address some of the issues in the updated specification [ETSI-TS-102-034v1.4.1], however, there are still outstanding issues.

The following is a list of the exceptions that need to be considered by an implementation adopting [I-D.ietf-fecframe-interleaved-fec-scheme] to be in compliant with the DVB-IPTV AL-FEC protocol as specified in [ETSI-TS-102-034v1.4.1].

- o SSRC

The DVB-IPTV AL-FEC protocol requires the SSRC fields of the FEC packets to be set to zero.

This requirement conflicts with RTP [RFC3550]. Unless signaled otherwise, RTP uses random SSRC values with collision detection. An explicit SSRC signaling mechanism is currently defined in [RFC5576] and can be used for this purpose.

- o CSRC

The DVB-IPTV AL-FEC protocol does not support the protection of the CSRC entries in the source packets. Thus, in an implementation compliant to DVB-IPTV AL-FEC protocol, the source

stream must not have any CSRC entries in its packets and subsequently the CC fields of the source RTP packets will be zero.

Note that if there are no RTP mixers used in a system running the DVB-IPTV AL-FEC protocol, the CC field of the source RTP packets will be zero and this is no longer an issue. Thus, if defined, the new RTP profile for the DVB-IPTV AL-FEC protocol should forbid the use of any RTP mixers.

- o **Timestamp**
In the DVB-IPTV AL-FEC protocol, the timestamp fields of the FEC packets are ignored by the receivers.
- o **Payload Type**
The DVB-IPTV AL-FEC protocol sets the PT fields of the FEC packets to 96.

A static payload type assignment for the base-layer FEC packets is outside the scope of [I-D.ietf-fecframe-interleaved-fec-scheme]. If defined, the new RTP profile for the DVB-IPTV AL-FEC protocol may assign 96 as the payload type for the base-layer FEC packets.

In implementations that are based on [I-D.ietf-fecframe-interleaved-fec-scheme] and are willing to be in compliant with the DVB-IPTV AL-FEC protocol as specified in [ETSI-TS-102-034v1.3.1], all these exceptions must be considered as well, however, in this case, the sender does not have to select a random initial sequence number for the FEC stream as suggested by [RFC3550].

Note that neither [ETSI-TS-102-034v1.3.1] nor [ETSI-TS-102-034v1.4.1] implements the 1-D interleaved parity code as specified in [I-D.ietf-fecframe-interleaved-fec-scheme]. Thus, the payload format registered in [I-D.ietf-fecframe-interleaved-fec-scheme] must not be used by the implementations that are compliant with the [ETSI-TS-102-034v1.3.1] or [ETSI-TS-102-034v1.4.1] specification.

2.2. Enhancement-Layer FEC

The Raptor code is a fountain code where as many encoding symbols as needed can be generated by the encoder on-the-fly from source data. Due to the fountain property of the Raptor code, multiple enhancement layers may also be specified, if needed.

The details of the Raptor code are provided in [I-D.ietf-fecframe-raptor]. The RTP payload format for Raptor FEC is specified in [I-D.watson-fecframe-rtp-raptor].

It is important to note that the DVB-IPTV AL-FEC protocol in the latest specification [ETSI-TS-102-034v1.4.1] allows both UDP-only and RTP-over-UDP encapsulations for the enhancement-layer FEC stream. The initial specification [ETSI-TS-102-034v1.3.1] exclusively permits UDP-only encapsulation for the enhancement-layer FEC stream.

When SDP is used for signaling, the transport protocol identifier permits to distinguish whether an RTP-over-UDP or UDP-only encapsulation is used. In case of any other signaling framework, the differentiation of the protocol for the enhancement-layer stream is achieved either explicitly through a protocol identifier or implicitly by the version number of the DVB IPTV Handbook. If none of the above signaling is provided, the receiver shall concur from the packet size of the repair packets if RTP-over-UDP or UDP-only encapsulation is used.

2.3. Hybrid Decoding Procedures

The receivers that support receiving and decoding both the base and enhancement-layer FEC perform hybrid decoding to improve the repair performance. The following steps may be followed to perform hybrid decoding:

1. Base-layer (Parity) Decoding: In this step, the repair packets that are encoded by the parity encoder are processed as usual to repair as many missing source packets as possible.
2. Enhancement-layer (Raptor) Decoding: If there are still missing source packets after the first step, the repair packets that are Raptor encoded are processed with the source packets already received and the source packets that are recovered in the first step.
3. Hybrid Decoding: If there are still missing source packets after the second step, the unprocessed base-layer (parity) repair packets are converted to a form in which they can be added to the Raptor decoding process. With this additional information, Raptor decoding may potentially recover any remaining missing source packet.

The procedure that should be followed to benefit from the base-layer repair packets in the Raptor decoding process is explained in detail in Section E.5.2 of [ETSI-TS-102-034v1.4.1].

3. Session Description Protocol (SDP) Signaling

This section provides an SDP [RFC4566] example for

[ETSI-TS-102-034v1.4.1]. The example uses the FEC grouping semantics [I-D.ietf-mmusic-rfc4756bis].

In the example, we have one source video stream (mid:S1), one FEC repair stream (mid:R1) that is produced by the 1-D interleaved parity FEC code as well as another FEC repair stream (mid:R2) that is produced by the Raptor FEC code. We form one FEC group with the "a=group:FEC-XR S1 R1 R2" line. The source and repair streams are sent to the same port on different multicast groups. The source, base-layer FEC and enhancement-layer FEC streams are all encapsulated in RTP.

Due to the exceptions described in Section 2.1, a [ETSI-TS-102-034v1.4.1]-compliant implementation must not use the RTP payload format defined in [I-D.ietf-fecframe-interleaved-fec-scheme]. Instead, it may use the payload format that has been registered by DVB TM-IPI for [ETSI-TS-102-034v1.3.1].

```
v=0
o=ali 1122334455 1122334466 IN IP4 fec.example.com
s=DVB-IPTV AL-FEC Example
t=0 0
a=group:FEC-XR S1 R1 R2
m=video 30000 RTP/AVP 100
c=IN IP4 233.252.0.1/127
a=rtpmap:100 MP2T/90000
a=mid:S1
m=application 30000 RTP/AVP 96
c=IN IP4 233.252.0.2/127
a=rtpmap:96 vnd.dvb.iptv.alfec-base/90000
a=mid:R1
m=application 30000 RTP/AVP 111
c=IN IP4 233.252.0.3/127
a=rtpmap:111 vnd.dvb.iptv.alfec-enhancement/90000
a=mid:R2
```

Note that in the example above, the payload type has been chosen as 96 for the base-layer FEC stream and there is no "a=fmtp:" line to specify the format parameters. Due to the lack of the format parameters for "vnd.dvb.iptv.alfec-base", it is not possible to learn the FEC parameters from the SDP description.

4. Security Considerations

This specification adds no new security considerations to the DVB-IPTV AL-FEC protocol.

5. IANA Considerations

There are no IANA considerations in this document.

6. Acknowledgments

This document is based on [ETSI-TS-102-034v1.3.1] and [ETSI-TS-102-034v1.4.1]. Thus, the authors would like to thank the editors of [ETSI-TS-102-034v1.3.1] and [ETSI-TS-102-034v1.4.1]. The authors also would like to thank those who reviewed earlier versions of this document.

7. References

7.1. Normative References

- [ETSI-TS-102-034v1.3.1]
ETSI TS 102 034 V1.3.1, "Transport of MPEG 2 TS Based DVB Services over IP Based Networks", October 2007.
- [ETSI-TS-102-034v1.4.1]
ETSI TS 102 034 V1.4.1, "Transport of MPEG 2 TS Based DVB Services over IP Based Networks", August 2009.
- [I-D.ietf-fecframe-interleaved-fec-scheme]
Begen, A., "RTP Payload Format for 1-D Interleaved Parity FEC", draft-ietf-fecframe-interleaved-fec-scheme-05 (work in progress), May 2009.
- [I-D.ietf-fecframe-raptor]
Watson, M., "Raptor FEC Schemes for FECFRAME", draft-ietf-fecframe-raptor-01 (work in progress), July 2009.
- [I-D.watson-fecframe-rtp-raptor]
Watson, M., "RTP Payload Format for Raptor FEC", draft-watson-fecframe-rtp-raptor-00 (work in progress), October 2008.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.
- [RFC5576] Lennox, J., Ott, J., and T. Schierl, "Source-Specific Media Attributes in the Session Description Protocol (SDP)", RFC 5576, June 2009.

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

[I-D.ietf-mmusic-rfc4756bis]
Begen, A., "Forward Error Correction Grouping Semantics in Session Description Protocol",
draft-ietf-mmusic-rfc4756bis-05 (work in progress),
October 2009.

7.2. Informative References

[DVB-A115]
Available at: http://www.dvb.org/technology/standards/al15.tm3783.AL-FEC_Evaluation.pdf, "DVB Application Layer FEC Evaluations (DVB Document A115)", May 2007.

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

Authors' Addresses

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

Email: abegen@cisco.com

Thomas Stockhammer
Nomor Research
Brecherspitzstrasse 8
Munich, 81541
Germany

Email: stockhammer@nomor.de

FEC Framework
Internet-Draft
Intended status: Informational
Expires: April 20, 2013

U. Kozat
DoCoMo USA Labs
A. Begen
Cisco
October 17, 2012

Pseudo Content Delivery Protocol (CDP) for Protecting Multiple Source
Flows in FEC Framework
draft-ietf-fecframe-pseudo-cdp-05

Abstract

This document provides a pseudo Content Delivery Protocol (CDP) to protect multiple source flows with one or more repair flows based on the FEC Framework and the Session Description Protocol (SDP) elements defined for the framework. The purpose of the document is not to provide a full-fledged protocol, but to show how the defined framework and SDP elements can be combined together to implement a CDP.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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

This Internet-Draft will expire on April 20, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction	4
2. Definitions/Abbreviations	4
3. Construction of a Repair Flow from Multiple Source Flows . . .	4
3.1. Example: Two Source Flows Protected by a Single Repair Flow	7
4. Reconstruction of Source Flows from Repair Flow(s)	11
4.1. Example: Multiple Source Flows Protected by a Single Repair Flow	11
5. Security Considerations	12
6. IANA Considerations	12
7. Acknowledgments	12
8. Normative References	12
Authors' Addresses	13

1. Introduction

The Forward Error Correction (FEC) Framework (described in [RFC6363]) and SDP Elements for FEC Framework (described in [RFC6364]) together define mechanisms sufficient enough to build an actual Content Delivery Protocol (CDP) with FEC protection. Methods to convey FEC Framework Configuration Information (described in [RFC6695]) on the other hand provides the signaling protocols that may be used as part of CDP to communicate FEC Scheme-Specific Information from FEC sender to a single as well as multiple FEC receivers. This document provides a guideline on how the mechanisms defined in [RFC6363] and [RFC6364] can be sufficiently used to design a CDP over a non-trivial scenario, namely protection of multiple source flows with one or more repair flows.

In particular, we provide clarifications and descriptions on how:

- o source and repair flows may be uniquely identified,
- o source blocks may be generated from one or more source flows,
- o repair flows may be paired with the source flows,
- o the receiver explicitly and implicitly identifies individual flows,
- o source blocks are regenerated at the receiver and the missing source symbols in a source block are recovered.

2. Definitions/Abbreviations

This document uses all the definitions and abbreviations from Section 2 of [RFC6363] minus the RFC 2119 requirements language.

3. Construction of a Repair Flow from Multiple Source Flows

At the sender side, CDP constructs the source blocks (SB) by multiplexing transport payloads from multiple flows (See Figure 1 and Figure 2). According to the FEC Framework, each source block is FEC-protected separately. Each source block is given to the specific FEC encoder used within the CDP as input and as the outputs Explicit Source FEC Payload ID, Repair FEC Payload ID, and Repair Payloads corresponding to that source block are generated. Note that Explicit Source FEC payload ID is optional and if CDP has implicit means of constructing the source block at the sender/receiver (e.g., by using any existing sequence numbers in the payload), the Explicit Source

$$\begin{array}{ccccc} \begin{array}{|c|} \hline \text{SB}_{(j+1)} \\ \hline \end{array} & \begin{array}{|c|} \hline \text{SB}_j \\ \hline \end{array} & \begin{array}{|c|} \hline \text{SB}_{(j-1)} \\ \hline \end{array} & \dots & \Rightarrow & \begin{array}{|c|} \hline \text{FEC} \\ \hline \text{Scheme} \\ \hline \end{array} & \begin{array}{l} \text{-----} \rightarrow r_1 \\ \text{Repair} \quad . \\ \text{Flows} \quad . \\ \text{-----} \rightarrow r_k \end{array} \end{array}$$

Source Block (SB)				
...	0
Payload_1	Payload_2	...	Payload_n	0
...
Symbol_1	Symbol_2	Symbol_3	...	Symbol_m
<----->	<----->	<----->	...	<----->

Symbol_1, ..., Symbol_m =>	FEC	=> Symbol_u, ..., Symbol_1
	Enc.	

FEC schemes typically expect a source block of certain size, say m symbols. Therefore, the FEC encoder divides each source block into m symbols (with some padding if the source block is shorter than the expected m symbols) and generates u repair symbols which are functions of the m symbols in the original source block. The repair symbols are grouped by the FEC scheme into repair payloads with each repair payload assigned a Repair FEC Payload ID in order to associate each repair payload with a particular source block at the receiver. If the payloads in a given source block have sequence numbers that can uniquely specify their location in the source block, an Explicit Source FEC Payload ID may not be generated for these payloads. Otherwise, Explicit Source FEC Payload IDs are generated for each payload and indicate the order the payloads appear in the source block.

3.1. Example: Two Source Flows Protected by a Single Repair Flow

In this section, we present an example of source flow and repair flow generation by the CDP. We have two source flows with flow IDs of 0 and 1 to be protected by a single repair flow (See Figure 5). The first source flow is multicast to 233.252.0.1 and the second source flow is multicast to 233.252.0.2. Both flows use the port number 30000.

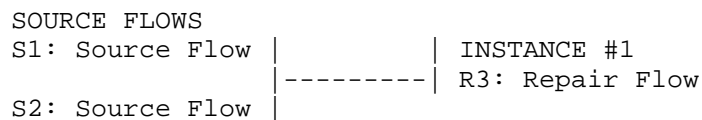


Figure 5: Example: Two source flows and one repair flow

The SDP description below states that the source flow defined by the tuple `{*,*,233.252.0.1,30000}` is identified with FID=0 and the source flow defined by the tuple `{*,*,233.252.0.2,30000}` is identified with FID=1 (via the 'id' parameter of the "fec-source-flow" attribute). The SDP description also states that the repair flow is to be received at the multicast address of 233.252.0.3 and at port 30000.

```
v=0
o=ali 1122334455 1122334466 IN IP4 fec.example.com
s=FEC Framework Examples
t=0 0
a=group:FEC-FR S1 S2 R3
m=video 30000 RTP/AVP 100
c=IN IP4 233.252.0.1/127
a=rtpmap:100 MP2T/90000
a=fec-source-flow: id=0
a=mid:S1
m=video 30000 RTP/AVP 101
c=IN IP4 233.252.0.2/127
a=rtpmap:101 MP2T/90000
a=fec-source-flow: id=1
a=mid:S2
m=application 30000 UDP/FEC
c=IN IP4 233.252.0.3/127
a=fec-repair-flow: encoding-id=0; ss-fssi=n:7,k:5
a=repair-window:150ms
a=mid:R3
```

Figure 6 shows the first and the second source blocks (SB_1 and SB_2) generated from these two source flows. In this example, SB_1 is of length 10000 bytes. Suppose that the FEC scheme uses a symbol length of 512 bytes. Then SB_1 can be divided into 20 symbols after padding the source block for 240 bytes. Assume that the FEC scheme is rate-2/3 erasure code, hence, it generates 10 repair symbols from 20 original symbols for SB_1. On the other hand, SB_2 is 7000-byte long and can be divided into 14 symbols after padding 168 bytes. Using the same encoder, suppose that 7 repair symbols are generated for SB_2.

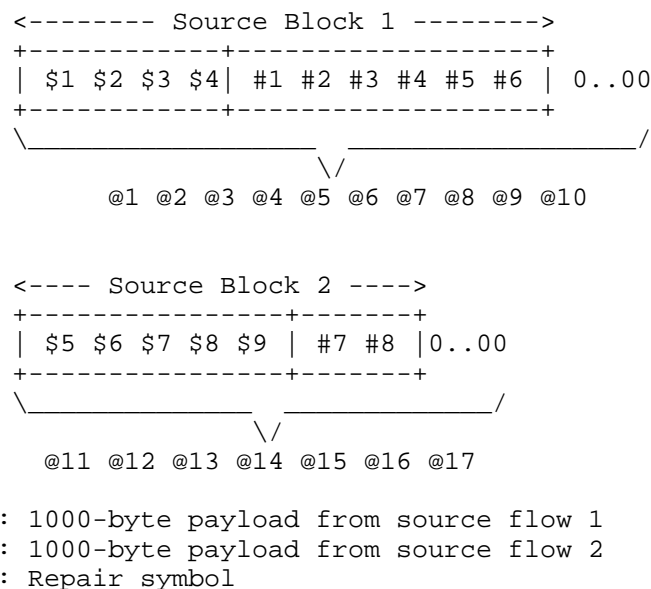


Figure 6: Source block with two source flows

The information on the unit of payload length, FEC scheme, symbol size, and coding rates can be specified in the FEC Scheme-Specific Information (FSSI) field of the SDP element. If the values of the payload lengths from each source flow and the order of appearance of source flows in every source block are fixed during the session, these values may be also provided in the FSSI field. To carry FSSI information to the FEC receivers, one may use the signaling methods described in [RFC6695]. In our example, we will consider the case where the ordering is fixed and known both at the sender and the receiver, but the payload lengths will be variable from one source block to another. We assume that the payload of a source flow with an FID smaller than another flow's FID precedes other payloads in a source block.

The FEC scheme gets the source blocks as input and generates the parity blocks for each source block to protect the whole source block. In the example, the repair payloads for SB_1 consist of 512-byte symbols, denoted by @1 to @10. Similarly @11 to @17 constitute the repair payloads for SB_2. The FEC scheme outputs the repair payloads along with the Repair FEC Payload IDs. In our example, Repair FEC Payload ID provides information on the source block sequence number and the order the repair symbols are generated. For instance @3 is the third FEC repair symbol for SB_1 and the three tuple {@3,SB_1,3} can uniquely deliver this information. In our example, the FEC scheme also provides Explicit Source FEC Payload IDs

that carry information to indicate which source symbols correspond to which source block sequence number and the relative position in the source block. For instance the two tuple {SB_2,2} can be attached to \$6 as the Explicit Source FEC Payload ID to indicate that \$6 is protected together with packets belonging to SB_2, and \$6 is the second payload in SB_2.

The source packets are generated from the source symbols by concatenating consecutive symbols in one packet. There should not be any fragmentation of a source symbol, e.g., symbols #7 and #8 can be concatenated in one transport payload of 2000-bytes (The implementation should make sure that the size of the resulting source packet - payload plus the overhead - is not larger than the path MTU), but one portion of symbol #7 should not be put in one source packet and the remaining portion in another source packet. The simplest implementation is to place each source symbol in a different source packet as shown in Figure 7.

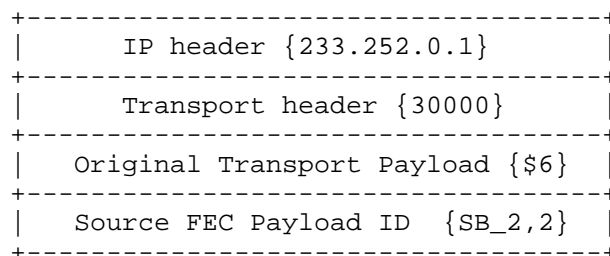


Figure 7: Example of a source packet for IPv4

The repair packets are generated from the repair symbols belonging to the same source block by grouping consecutive symbols in one packet. There should not be any fragmentation of a repair symbol, e.g., symbols @4, @5, and @6 can be concatenated in one transport payload of 1536-bytes, but @6 should not be divided into smaller sub-symbols and spread over multiple repair packets. The Repair FEC Payload ID must carry sufficient information for the decoding process and in our example indicating source block sequence number, length of each source payload, and the order that the first parity block in a repair packet is generated are sufficient. The exact header format of Repair FEC Payload ID may be specified in the FSSI field of the SDP element. In Figure 8 for instance, the repair symbols @4, @5, and @6 are concatenated together. The Payload ID {SB_1,4,4,6} states that the repair symbols protect SB_1, the first repair symbol in the payload is generated as the 4th symbol and the source block consists of two source flows carrying 4 and 6 packets from each.

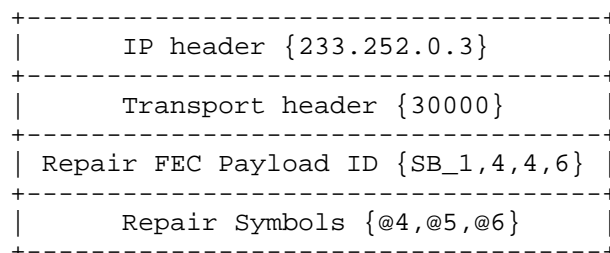


Figure 8: Example of a repair packet for IPv4

4. Reconstruction of Source Flows from Repair Flow(s)

Here we provide an example for reconstructing multiple source flows from a single repair flow.

4.1. Example: Multiple Source Flows Protected by a Single Repair Flow

At the receiver, source flows 1 and 2 are received at {233.252.0.1,30000} and {233.252.0.2,30000}, while the repair flow is received at {233.252.0.3,30000}. The CDP can map these tuples to the flow IDs using the SDP elements. Accordingly, the payloads received at {233.252.0.1,30000} and {233.252.0.2,30000} are mapped to flow IDs 0 and 1, respectively.

The CDP passes the flow IDs and received payloads along with the Explicit Source FEC Payload ID to the FEC scheme defined in the SDP description. The CDP also passes the received repair packet payloads and Repair FEC Payload ID to the FEC scheme. The FEC scheme can construct the original source block with missing packets by using the information given in the FEC Payload IDs. The FEC Repair Payload ID provides the information that SB_1 has packets from two flows with 4 packets from the first one and 6 packets from the second one. Flow IDs state that the packets from source flow 0 precedes the packets from source flow 1. Explicit Source FEC Payload IDs on the other hand provide the information about which source payload appears in what order. Therefore, the FEC scheme can depict an source block with exact locations of the missing packets. Figure 9 depicts the case for SB_1. Since the original source block with missing packets can be constructed at the decoder and the FEC scheme knows the coding rate (e.g., it might be carried in the FSSI field in the SDP description), a proper decoding operation can start as soon as the repair symbols are provided to the FEC scheme.

```

<----- Source Block 1 ----->
+-----+-----+
| $1 $2 X  X | #1 X  #3 #4 #5 #6 |
+-----+-----+

```

O: Symbols received from the source flow 1 for SB_1
 #: Symbols received from the source flow 2 for SB_1
 X: Lost source symbols

Figure 9: Source block regeneration

When the FEC scheme can recover any missing symbol while more repair symbols are arriving, it provides the recovered blocks along with the source flow IDs of the recovered blocks as outputs to the CDP. The receiver knows how long to wait to repair the remaining missing packets (e.g., specified by the 'repair-window' attribute in the SDP description). After the associated timer expires, the CDP hands over whatever could be recovered from the source flow to the application layer and continues with processing the next source block.

5. Security Considerations

For the general security considerations related to the FEC Framework, refer to [RFC6363]. For the security considerations related to the SDP elements in the FEC Framework, refer to [RFC6364]. There are no additional security considerations that apply to this document.

6. IANA Considerations

There are no IANA related issues considered in this document.

7. Acknowledgments

The authors would like to thank the FEC Framework Design Team for their inputs, suggestions and contributions.

8. Normative References

- [RFC6363] Watson, M., Begen, A., and V. Roca, "Forward Error Correction (FEC) Framework", RFC 6363, October 2011.
- [RFC6364] Begen, A., "Session Description Protocol Elements for the Forward Error Correction (FEC) Framework", RFC 6364, October 2011.

[RFC6695] Asati, R., "Methods to Convey Forward Error Correction (FEC) Framework Configuration Information", RFC 6695, August 2012.

Authors' Addresses

Ulas C. Kozat
DoCoMo USA Labs
3240 Hillview Avenue
Palo Alto, CA 94304-1201
USA

Phone: +1 650 496 4739
Email: kozat@docomolabs-usa.com

Ali Begen
Cisco
181 Bay Street
Toronto, ON M5J 2T3
Canada

Email: abegen@cisco.com

FEC Framework
Internet-Draft
Intended status: Standards Track
Expires: November 11, 2012

M. Watson
Netflix
T. Stockhammer
Nomor Research
M. Luby
Qualcomm Incorporated
May 10, 2012

Raptor FEC Schemes for FECFRAME
draft-ietf-fecframe-raptor-11

Abstract

This document describes Fully-Specified Forward Error Correction (FEC) Schemes for the Raptor and RaptorQ codes and their application to reliable delivery of media streams in the context of FEC Framework. The Raptor and RaptorQ codes are systematic codes, where a number of repair symbols are generated from a set of source symbols and sent in one or more repair flows in addition to the source symbols that are sent to the receiver(s) within a source flow. The Raptor and RaptorQ codes offer close to optimal protection against arbitrary packet losses at a low computational complexity. Six FEC Schemes are defined, two for protection of arbitrary packet flows, two that are optimised for small source blocks and another two for protection of a single flow that already contains a sequence number. Repair data may be sent over arbitrary datagram transport (e.g. UDP) or using RTP.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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

This Internet-Draft will expire on November 11, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction	5
2. Document Outline	6
3. Requirements Notation	6
4. Definitions and Abbreviations	6
4.1. Definitions	7
4.2. Abbreviations	7
5. General procedures for Raptor FEC Schemes	7
6. Raptor FEC Schemes for arbitrary packet flows	9
6.1. Introduction	9
6.2. Formats and Codes	9
6.2.1. FEC Framework Configuration Information	9
6.2.2. Source FEC Payload ID	10
6.2.3. Repair FEC Payload ID	11
6.3. Procedures	13
6.3.1. Source symbol construction	13
6.3.2. Repair packet construction	13
6.4. FEC Code Specification	13
7. Optimised Raptor FEC Scheme for arbitrary packet flows	13
7.1. Introduction	14
7.2. Formats and Codes	14
7.2.1. FEC Framework Configuration Information	14
7.2.2. Source FEC Payload ID	15
7.2.3. Repair FEC Payload ID	15
7.3. Procedures	15
7.3.1. Source symbol construction	15
7.3.2. Repair packet construction	15
7.4. FEC Code Specification	15
8. Raptor FEC Scheme for a single sequenced flow	16
8.1. Formats and codes	16
8.1.1. FEC Framework Configuration Information	16
8.1.2. Source FEC Payload ID	16
8.1.3. Repair FEC Payload ID	16
8.2. Procedures	18
8.2.1. Source symbol construction	18
8.2.2. Derivation of Source FEC Packet Identification Information	18
8.2.3. Repair packet construction	19
8.2.4. Procedures for RTP source flows	19
8.3. FEC Code Specification	20
9. Security Considerations	20
10. Session Description Protocol (SDP) Signaling	20
11. Congestion Control Considerations	21
12. IANA Considerations	21
12.1. Registration of FEC Scheme IDs	21
13. Acknowledgements	22
14. References	22

14.1. Normative References 22

14.2. Informative References 22

Authors' Addresses 23

1. Introduction

The Forward Error Correction (FEC) Framework [RFC6363] describes a general framework for the use of Forward Error Correction in association with arbitrary packet flows. Modeled after the FEC Building Block developed by the IETF Reliable Multicast Transport working group [RFC5052], the FEC Framework defines the concept of FEC Schemes which provide specific Forward Error Correction schemes. This document describes six FEC Schemes which make use of the Raptor and RaptorQ FEC codes as defined in [RFC5053] and [RFC6330].

The FEC protection mechanism is independent of the type of the source data, which can be an arbitrary sequence of packets, for example audio or video data. In general, the operation of the protection mechanism is as follows:

- o The sender determines a set of source packets (a source block) to be protected together based on the FEC Framework Configuration Information.
- o The sender arranges the source packets into a set of source symbols, each of which is the same size.
- o The sender applies the Raptor/RaptorQ protection operation on the source symbols to generate the required number of repair symbols.
- o The sender packetizes the repair symbols and sends the repair packet(s) and the source packets to the receiver(s). Per the FEC Framework requirements, the sender MUST transmit the source and repair packets in different source and repair flows, or in the case Real-time Transport Protocol (RTP) transport is used for repair packets, in different RTP streams.
- o At the receiver side, if all of the source packets are successfully received, there is no need for FEC recovery and the repair packets are discarded. However, if there are missing source packets, the repair packets can be used to recover the missing information.

The operation of the FEC mechanism requires that the receiver can identify the relationships between received source packets and repair packets and in particular which source packets are missing. In many cases, data already exists in the source packets which can be used to refer to source packets and to identify which packets are missing. In this case we assume it is possible to derive a "sequence number" directly or indirectly from the source packets and this sequence number can be used within the FEC Scheme. This case is referred to as a "single sequenced flow". In this case the FEC Source Payload ID

defined in [RFC6363] is empty and the source packets are not modified by the application of FEC, with obvious backwards compatibility advantages.

Otherwise, it is necessary to add data to the source packets for FEC purposes in the form of a non-empty FEC Source Payload ID. This case is referred to as the "arbitrary packet flow" case. Accordingly, this document defines six FEC Schemes, two for the case of a single sequenced flow and four for the case of arbitrary packet flows.

2. Document Outline

This document is organised as follows:

- o Section 5 defines general procedures applicable to the use of the Raptor and RaptorQ codes in the context of the FEC Framework.
- o Section 6 defines an FEC Scheme for the case of arbitrary source flows and follows the format defined for FEC Schemes in [RFC6363]. When used with Raptor codes, this scheme is equivalent to that defined in "3GPP TS 26.346: Multimedia Broadcast/Multicast Service (MBMS); Protocols and codecs" [MBMSTS].
- o Section 7 defines an FEC Scheme similar to that defined in Section 6 but with optimisations for the case where only limited source block sizes are required. When used with Raptor codes, this scheme is equivalent to that defined in "ETSI TS 102.034: Digital Video Broadcasting (DVB); Transport of MPEG-2 Based DVB Services over IP Based Networks" [dvbts] for arbitrary packet flows.
- o Section 8 defines an FEC Scheme for the case of a single flow which is already provided with a source packet sequence number. When used with Raptor codes, this scheme is equivalent to that defined in [dvbts] for the case of a single sequenced flow.

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 [RFC2119].

4. Definitions and Abbreviations

The definitions, notations and abbreviations commonly used in this

document are summarized in this section.

4.1. Definitions

The FEC-specific terminology used in this document is defined in [RFC6363]. In this document, as in [RFC6363], the first letter of each FEC-specific is capitalized along with the new terms defined here:

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

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

4.2. Abbreviations

This document uses abbreviations that apply to FEC Framework in general as defined in [RFC6363]. In addition, this document uses the following abbreviations

FSSI: FEC-Scheme-Specific Information.

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

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

ADU: Application Data Unit

ADUI: Application Data Unit Information.

SPI: Source Packet Information.

MSBL: Maximum Source Block Length

5. General procedures for Raptor FEC Schemes

This section specifies general procedures which apply to all Raptor and RaptorQ FEC Schemes, specifically the construction of source symbols from a set of source transport payloads.

For any field defined in this document, the octets are ordered in network byte order.

As described in [RFC6363] for each Application Data Unit (ADU) in a source block, the FEC Scheme is provided with:

- o A description of the source data flow with which the ADU is associated and an integer identifier associated with that flow.
- o The ADU itself.
- o The length of the ADU.

For each ADU, we define the Application Data Unit Information (ADUI) as follows:

Let

- o n be the number of ADUs in the source block.
- o T be the source symbol size in octets. Note: this information is provided by the FEC Scheme as defined below.
- o i the index to the $(i+1)$ -th ADU to be added to the source block, $0 \leq i < n$.
- o $f[i]$ denote the integer identifier associated with the source data flow from which the i -th ADU was taken.
- o $F[i]$ denote a single octet representing the value of $f[i]$.
- o $l[i]$ be a length indication associated with the i -th ADU - the nature of the length indication is defined by the FEC Scheme.
- o $L[i]$ denote two octets representing the value of $l[i]$ in network byte order (high order octet first) of the i -th ADU.
- o $R[i]$ denote the number of octets in the $(i+1)$ -th ADU.
- o $s[i]$ be the smallest integer such that $s[i]*T \geq (l[i]+3)$. Note $s[i]$ is the length of SPI[i] in units of symbols of size T octets.
- o $P[i]$ denote $s[i]*T - (l[i]+3)$ zero octets. Note: $P[i]$ are padding octets to align the start of each UDP packet with the start of a symbol.
- o ADUI[i] be the concatenation of $F[i]$, $L[i]$, $R[i]$ and $P[i]$.

Then, a source data block is constructed by concatenating ADUI[i] for $i = 0, 1, 2, \dots, n-1$. The source data block size, S , is then given by $\sum \{s[i]*T, i=0, \dots, n-1\}$. Symbols are allocated integer Encoding Symbol IDs consecutively starting from zero within the source block. Each ADU is associated with the Encoding Symbol ID of the first symbol containing SPI for that packet. Thus, the Encoding

Symbol ID value associated with the j -th source packet, $ESI[j]$, is given by $ESI[j] = 0$, for $j=0$ and $ESI[j] = \text{sum}\{s[i], i=0, \dots, (j-1)\}$, for $0 < j < n$.

Source blocks are identified by integer Source Block Numbers. This specification does not specify how Source Block Numbers are allocated to source blocks. The Source FEC Packet Identification Information consists of the identity of the source block and the Encoding Symbol ID associated with the packet.

6. Raptor FEC Schemes for arbitrary packet flows

6.1. Introduction

This section specifies an FEC Scheme for the application of the Raptor and RaptorQ codes to arbitrary packet flows. This scheme is recommended in scenarios where maximal generality is required.

When used with the Raptor codes specified in [RFC5053], this scheme is equivalent to that specified in [MBMSTS].

6.2. Formats and Codes

6.2.1. FEC Framework Configuration Information

6.2.1.1. FEC Scheme ID

The value of the FEC Scheme ID for the fully-specified FEC scheme defined in this section is XXX1 when [RFC5053] is used and XXX2 when [RFC6330] is used, as assigned by IANA.

NOTE: To the RFC Editor: please change these XXX notations once assigned, and remove this NOTE.

6.2.1.2. Scheme-Specific Elements

The scheme-specific elements of the FEC Framework Configuration information for this scheme are as follows:

MSBL Value range: An non-negative integer less than 8192 for FEC Scheme XXX1 and less than 56403 for FEC Scheme XXX2, in units of symbols. The field type is unsigned integer.

Encoding Symbol Size Name: "T", Value range: A non-negative integer less than 65536, in units of octets. The field type is unsigned integer.

Payload ID Format Name: "P", Value range: "A" or "B". The P bit shall be set to zero to indicate Payload ID Format A or to one to indicate Payload ID Format B.

An encoding format for the MSBL and Encoding Symbol Size is defined below.

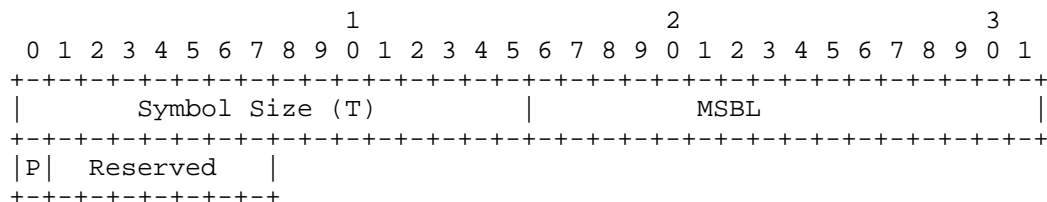


Figure 1: FEC Scheme Specific Information

The P bit shall be set to zero to indicate Payload ID Format A or to one to indicate Payload ID Format B. The last octet of FEC Scheme Specific Information SHOULD be omitted indicating that Payload ID Format A is in use. The Payload ID Format identifier defines which of the Source FEC Payload ID and Repair FEC Payload ID formats defined below shall be used. Payload ID Format B SHALL NOT be used for FEC Scheme XXX1. The two formats enable different use cases. Format A is appropriate in case the stream has many typically smaller source blocks and Format B is applicable if the stream has fewer large source blocks each with many encoding symbols.

6.2.2. Source FEC Payload ID

This scheme makes use of an Explicit Source FEC Payload ID, which is appended to the end of the source packets. Two formats are defined for the Source FEC Payload ID, format A and format B. The format that is used is signaled as part of the FEC Framework Configuration Information.

The Source FEC Payload ID for format A is provided in Figure 2.

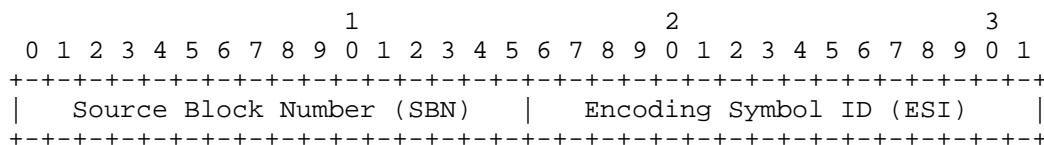


Figure 2: Source FEC Payload ID - Format A

Source Block Number (SBN), (16 bits): Identifier for the source block that the source data within the packet relates. The field type is unsigned integer.

Encoding Symbol ID (ESI), (16 bits): The starting symbol index of the source packet in the source block. The field type is unsigned integer.

The Source FEC Payload ID for format B is provided in Figure 3.

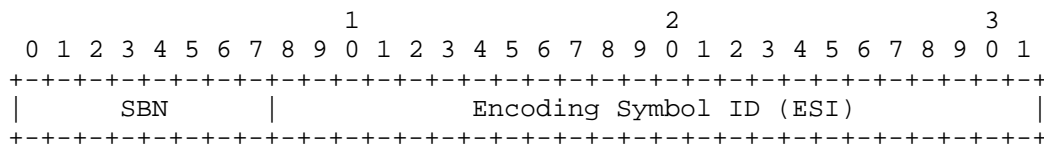


Figure 3: Source FEC Payload ID - Format B

Source Block Number (SBN), (8 bits): Identifier for the source block that the source data within the packet relates. The field type is unsigned integer.

Encoding Symbol ID (ESI), (24 bits): The starting symbol index of the source packet in the source block. The field type is unsigned integer.

6.2.3. Repair FEC Payload ID

Two formats for the Repair FEC Payload ID, Format A and Format B are defined below.

The Repair FEC Payload ID for format A is provided in Figure 4.

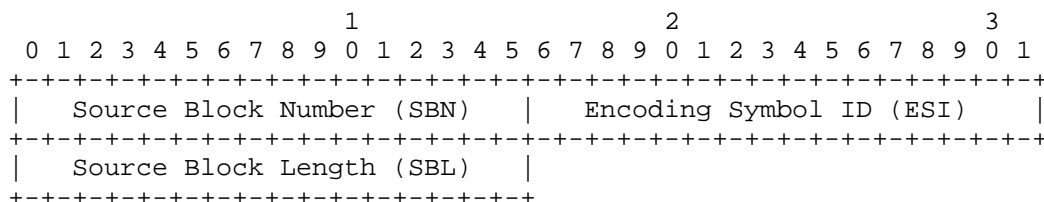


Figure 4: Repair FEC Payload ID - Format A

Source Block Number (SBN), (16 bits) Identifier for the source block that the repair symbols within the packet relate. For format A, it is of size 16 bits. The field type is unsigned integer.

Encoding Symbol ID (ESI), (16 bits) Identifier for the encoding symbols within the packet. The field type is unsigned integer.

Source Block Length (SBL), (16 bits) The number of source symbols in the source block. The field type is unsigned integer.

The Repair FEC Payload ID for format B is provided in Figure 5.

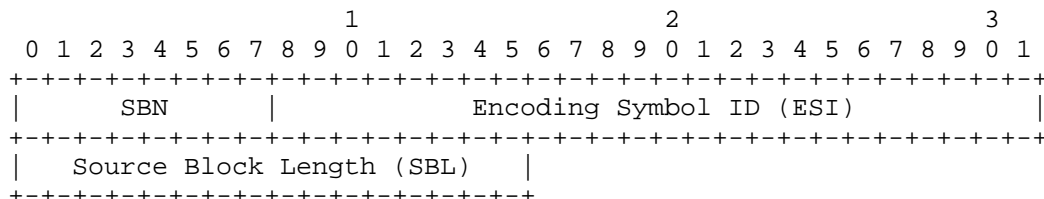


Figure 5: Repair FEC Payload ID - Format B

Source Block Number (SBN), (8 bits) Identifier for the source block that the repair symbols within the packet relate. For format B, it is of size 8 bits. The field type is unsigned integer.

Encoding Symbol ID (ESI), (24 bits) Identifier for the encoding symbols within the packet. The field type is unsigned integer.

Source Block Length (SBL), (16 bits) The number of source symbols in the source block. The field type is unsigned integer.

The interpretation of the Source Block Number, Encoding Symbol Identifier and Source Block Length is defined by the FEC Code Specification in [RFC5053] for FEC Scheme XXX1 and [RFC6330] for FEC Scheme XXX2.

6.3. Procedures

6.3.1. Source symbol construction

FEC Scheme XXX1 and FEC Scheme XXX2 use the procedures defined in Section 5 to construct a set of source symbols to which the FEC code can be applied. The sender MUST allocate Source Block Numbers to source blocks sequentially, wrapping around to zero after Source Block Number 65535 (Format A) or 255 (Format B).

During the construction of the source block:

- o the length indication, $l[i]$, included in the Source Packet Information for each packet shall be the transport payload length, i.e. the length of the ADU.
- o the value of $s[i]$ in the construction of the Source Packet Information for each packet shall be the smallest integer such that $s[i]*T \geq (l[i]+3)$.

6.3.2. Repair packet construction

For FEC Scheme XXX1, the ESI value placed into a repair packet is calculated as specified in Section 5.3.2 of [RFC5053].

For FEC Scheme XXX2 [RFC6330], the ESI value placed into a repair packet is calculated as specified in Section 4.4.2 of [RFC6330].

In both cases K is identical to SBL.

6.4. FEC Code Specification

The FEC encoder defined in [RFC5053] SHALL be used for FEC Scheme XXX1 and the FEC encoder defined in [RFC6330] SHALL be used for FEC Scheme XXX2. For both FEC Scheme XXX1 and FEC Scheme XXX2, the source symbols passed to the FEC encoder SHALL consist of the source symbols constructed according to Section 6.3.1. Thus the value of the parameter K used by the FEC encoder (equal to the Source Block Length) may vary amongst the blocks of the stream but SHALL NOT exceed the Maximum Source Block Length signaled in the FEC Scheme-specific information. The symbol size, T , to be used for source block construction and the repair symbol construction is equal to the Encoding Symbol Size signaled in the FEC Scheme Specific Information.

7. Optimised Raptor FEC Scheme for arbitrary packet flows

7.1. Introduction

This section specifies a slightly modified version of the FEC Scheme specified in Section 6 which is applicable to scenarios in which only relatively small block sizes will be used. These modifications admit substantial optimisations to both sender and receiver implementations.

In outline, the modifications are:

- o All source blocks within a stream are encoded using the same source block size. Code shortening is used to encode blocks of different sizes. This is achieved by padding every block to the required size using zero symbols before encoding. The zero symbols are then discarded after decoding. The source block size to be used for a stream is signaled in the Maximum Source Block Length (MSBL) field of the scheme-specific information. The extended source block is constructed by adding zero or more padding symbols such that the total number of symbols, MSBL, is one of the values listed in Section 7.4. Each padding symbol consists of T octets where the value of each octet is zero. MSBL MUST be selected as the smallest value of the possible values in Section 7.4 that is greater than or equal to K.
- o The possible choices of the MSBL for a stream is restricted to a small specified set. This allows explicit operation sequences for encoding and decoding the restricted set of source block lengths to be pre-calculated and embedded in software or hardware.

When used with the Raptor codes specified in [RFC5053], this scheme is equivalent to that specified in [dvbts] for arbitrary packet flows.

7.2. Formats and Codes

7.2.1. FEC Framework Configuration Information

7.2.1.1. FEC Scheme ID

The value of the FEC Scheme ID for the fully-specified FEC scheme defined in this section is XXX3 when [RFC5053] is used and XXX4 when [RFC6330] is used, as assigned by IANA.

NOTE: To the RFC Editor: please change these XXX notations once assigned, and remove this NOTE.

7.2.1.2. FEC Scheme specific information

The same as specified for FEC Scheme XXX1 for FEC Scheme XXX3, and the same as specified for FEC Scheme XXX2 for FEC Scheme XXX4, as specified in Section 6.2.1.2, except that the MSBL value is as defined in Section 7.4.

7.2.2. Source FEC Payload ID

The same as specified for FEC Scheme XXX1 for FEC Scheme XXX3, and the same as specified for FEC Scheme XXX2 for FEC Scheme XXX4, as specified in Section 6.2.2.

7.2.3. Repair FEC Payload ID

The same as specified for FEC Scheme XXX1 for FEC Scheme XXX3, and the same as specified for FEC Scheme XXX2 for FEC Scheme XXX4, as specified in Section 6.2.3.

7.3. Procedures

7.3.1. Source symbol construction

See Section 6.3.1.

7.3.2. Repair packet construction

The number of repair symbols contained within a repair packet is computed from the packet length. The ESI value placed into a repair packet is calculated as $X + \text{MSBL} - \text{SBL}$, where X would be the ESI value of the repair packet if the ESI were calculated as specified in Section 5.3.2 of [RFC5053] for FEC Scheme XXX3 and as specified in Section 4.4.2 of [RFC6330] for FEC Scheme XXX4, where $K = \text{SBL}$. The value of SBL SHALL be at most the value of MSBL.

7.4. FEC Code Specification

The FEC encoder defined in [RFC5053] SHALL be used for FEC Scheme XXX3 and the FEC encoder defined in [RFC6330] SHALL be used for FEC Scheme XXX4. The source symbols passed to the FEC encoder SHALL consist of the source symbols constructed according to Section 6.3.1 extended with zero or more padding symbols. The extension SHALL be such that the total number of symbols in the source block is equal to the MSBL signaled in the FEC Scheme Specific Information. Thus the value of the parameter K used by the FEC encoded is equal to the MSBL for all blocks of the stream. Padding symbols shall consist entirely of octets set to the value zero. The symbol size, T , to be used for source block construction and the repair symbol construction is equal

to the Encoding Symbol Size signaled in the FEC Scheme Specific Information.

For FEC Scheme XXX3, the parameter T SHALL be set such that the number of source symbols in any source block is at most 8192. The MSBL parameter, and hence the number of symbols used in the FEC Encoding and Decoding operations, SHALL be set to one of the following values:

101, 120, 148, 164, 212, 237, 297, 371, 450, 560, 680, 842, 1031, 1139, 1281

For FEC Scheme XXX4, the parameter T SHALL be set such that the number of source symbols in any source block is less than 56403. The MSBL parameter SHALL be set to one of the supported values for K' defined in Section 5.6 of [RFC6330].

8. Raptor FEC Scheme for a single sequenced flow

8.1. Formats and codes

8.1.1. FEC Framework Configuration Information

8.1.1.1. FEC Scheme ID

The value of the FEC Scheme ID for the fully-specified FEC scheme defined in this section is XXX5 when [RFC5053] is used and XXX6 when [RFC6330] is used, as assigned by IANA.

NOTE: To the RFC Editor: please change these XXX notations once assigned, and remove this NOTE.

8.1.1.2. Scheme-specific elements

The same as specified for FEC Scheme XXX1 for FEC Scheme XXX5, and the same as specified for FEC Scheme XXX2 for FEC Scheme XXX6, as specified in Section 6.2.1.2.

8.1.2. Source FEC Payload ID

The Source FEC Payload ID field is not used by this FEC Scheme. Source packets are not modified by this FEC Scheme.

8.1.3. Repair FEC Payload ID

Two formats for the Repair FEC Payload ID are defined, Format A and Format B.

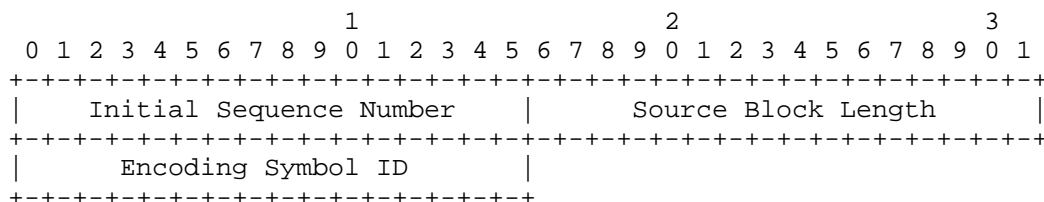


Figure 6: Repair FEC Payload ID - Format A

Initial Sequence Number (Flow i ISN) - 16 bits This field specifies the lowest 16 bits of the sequence number of the first packet to be included in this sub-block. If the sequence numbers are shorter than 16 bits then the received Sequence Number SHALL be logically padded with zero bits to become 16 bits in length respectively. The field type is unsigned integer.

Source Block Length (SBL) - 16 bits This field specifies the length of the source block in symbols. The field type is unsigned integer.

Encoding Symbol ID (ESI) - 16 bits This field indicates which repair symbols are contained within this repair packet. The ESI provided is the ESI of the first repair symbol in the packet. The field type is unsigned integer.

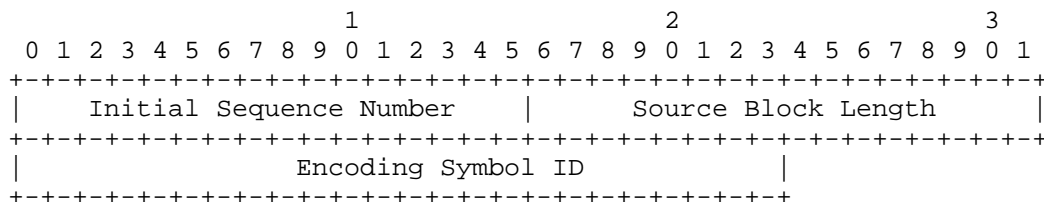


Figure 7: Repair FEC Payload ID - Format B

Initial Sequence Number (Flow i ISN) - 16 bits This field specifies the lowest 16 bits of the sequence number of the first packet to be included in this sub-block. If the sequence numbers are shorter than 16 bits then the received Sequence Number SHALL be logically padded with zero bits to become 16 bits in length respectively. The field type is unsigned integer.

Source Block Length (SBL) - 16 bits This field specifies the length of the source block in symbols. The field type is unsigned integer.

Encoding Symbol ID (ESI) - 24 bits This field indicates which repair symbols are contained within this repair packet. The ESI provided is the ESI of the first repair symbol in the packet. The field type is unsigned integer.

8.2. Procedures

8.2.1. Source symbol construction

FEC Scheme XXX5 and FEC Scheme XXX6 use the procedures defined in Section 5 to construct a set of source symbols to which the FEC code can be applied.

During the construction of the source block:

- o the length indication, $l[i]$, included in the Source Packet Information for each packet shall be dependent on the protocol carried within the transport payload. Rules for RTP are specified below.
- o the value of $s[i]$ in the construction of the Source Packet Information for each packet shall be the smallest integer such that $s[i]*T \geq (l[i]+3)$

8.2.2. Derivation of Source FEC Packet Identification Information

The Source FEC Packet Identification Information for a source packet is derived from the sequence number of the packet and information received in any repair FEC packet belonging to this Source Block. Source blocks are identified by the sequence number of the first source packet in the block. This information is signaled in all repair FEC packets associated with the source block in the Initial Sequence Number field.

The length of the Source Packet Information (in octets) for source packets within a source block is equal to length of the payload containing encoding symbols of the repair packets (i.e. not including the Repair FEC Payload ID) for that block, which MUST be the same for all repair packets. The Application Data Unit Information Length (ADUIL) in symbols is equal to this length divided by the Encoding Symbol Size (which is signaled in the FEC Framework Configuration Information). The set of source packets which are included in the source block is determined from the Initial Sequence Number (ISN) and Source Block Length (SBL) as follows:

Let,

- o I be the Initial Sequence Number of the source block
- o LP be the Source Packet Information Length in symbols
- o LB be the Source Block Length in symbols

Then, source packets with sequence numbers from I to I +(LB/LP)-1 inclusive are included in the source block. The Source Block Length LB MUST be chosen such that it is at least as large as the largest Source Packet Information Length LP.

Note that if no FEC repair packets are received then no FEC decoding is possible and it is unnecessary for the receiver to identify the Source FEC Packet Identification Information for the source packets.

The Encoding Symbol ID for a packet is derived from the following information:

- o The sequence number, Ns, of the packet
- o The Source Packet Information Length for the source block, LP
- o The Initial Sequence Number of the source block, I

Then the Encoding Symbol ID for packet with sequence number Ns is determined by the following formula:

$$ESI = (Ns - I) * LP$$

Note that all repair packet associated to a given Source Block MUST contain the same Source Block Length and Initial Sequence Number.

Note also that the source packet flow processed by the FEC encoder MUST have consecutive sequence numbers. In case the incoming source packet flow has a gap in the sequence numbers then implementors SHOULD insert an ADU in the source block that complies to the format of the source packet flow, but is ignored at the application with high probability. For additional guidelines refer to [RFC6363], Section 10.2, paragraph 5.

8.2.3. Repair packet construction

See Section 7.3.2

8.2.4. Procedures for RTP source flows

In the specific case of RTP source packet flows, then the RTP Sequence Number field SHALL be used as the sequence number in the

procedures described above. The length indication included in the Application Data Unit Information SHALL be the RTP payload length plus the length of the CSRCs, if any, the RTP Header Extension, if present, and the RTP padding octets, if any. Note that this length is always equal to the UDP payload length of the packet minus 12.

8.3. FEC Code Specification

The same as specified for FEC Scheme XXX3 for FEC Scheme XXX5, and the same as specified for FEC Scheme XXX4 for FEC Scheme XXX6, as specified in Section 7.4.

9. Security Considerations

For the general security considerations related to the use of FEC, refer to [RFC6363]. Also consider relevant security considerations in [RFC5053] and [RFC6330]. No security vulnerabilities specific to this document have been identified.

10. Session Description Protocol (SDP) Signaling

This section provides an SDP [RFC4566] example. The syntax follows the definition in [RFC6364]. Assume we have one source video stream (mid:S1) and one FEC repair stream (mid:R1). We form one FEC group with the "a=group:FEC-FR S1 R1" line. The source and repair streams are sent to the same port on different multicast groups. The repair window is set to 200 ms.

```
v=0
o=ali 1122334455 1122334466 IN IP4 fec.example.com
s=Raptor FEC Example
t=0 0
a=group:FEC-FR S1 R1
m=video 30000 RTP/AVP 100
c=IN IP4 233.252.0.1/127
a=rtpmap:100 MP2T/90000
a=fec-source-flow: id=0
a=mid:S1
m=application 30000 UDP/FEC
c=IN IP4 233.252.0.2/127
a=fec-repair-flow: encoding-id=6; fssi=Kmax:8192,T:128,P:A
a=repair-window:200ms
a=mid:R1
```


11. Congestion Control Considerations

For the general congestion control considerations related to the use of FEC, refer to [RFC6363].

12. IANA Considerations

12.1. Registration of FEC Scheme IDs

The value of FEC Scheme IDs is subject to IANA registration. For general guidelines on IANA considerations as they apply to this document, refer to [RFC6363].

This document registers six values in the FEC Framework (FECFRAME) FEC Encoding IDs registry (<http://www.iana.org/assignments/rmt-fec-parameters/rmt-fec-parameters.xml#fecframe-fec-encoding-ids>) as provided in Table 1. Each value refers to a fully-specified FEC scheme.

NOTE: To the RFC Editor: please change these XXX notations once assigned, and remove this NOTE.

FEC Encoding ID	FEC Scheme Description	Reference
XXX1	Raptor FEC Scheme for Arbitrary Packet Flows	Section 6 in this document using [RFC5053]
XXX2	RaptorQ FEC Scheme for Arbitrary Packet Flows	Section 6 in this document using [RFC6330].
XXX3	Raptor FEC Scheme Optimised for Arbitrary Packet Flows	Section 7 in this document using Raptor [RFC5053].
XXX4	RaptorQ FEC Scheme Optimised for Arbitrary Packet Flows	XXX4 for the Optimised RaptorQ FEC Scheme for Arbitrary Packet Flows (Section 7) using RaptorQ [RFC6330].

XXX5	Raptor FEC Scheme for a single sequence flow	XXX5 for the Raptor FEC Scheme for a single sequence flow (Section 8) using Raptor [RFC5053].
XXX6	RaptorQ FEC Scheme for a single sequence flow	XXX6 for the RaptorQ FEC Scheme for a single sequence flow (Section 8) using RaptorQ [RFC6330].

Table 1: FEC Framework (FECFRAME) FEC Encoding IDs

13. Acknowledgements

Thanks are due to Ali C. Begen and David Harrington for thorough review of earlier draft versions of this document.

14. References

14.1. Normative References

- [RFC6363] Watson, M., Begen, A., and V. Roca, "Forward Error Correction (FEC) Framework", RFC 6363, October 2011.
- [RFC5053] Luby, M., Shokrollahi, A., Watson, M., and T. Stockhammer, "Raptor Forward Error Correction Scheme for Object Delivery", RFC 5053, October 2007.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC6330] Luby, M., Shokrollahi, A., Watson, M., Stockhammer, T., and L. Minder, "RaptorQ Forward Error Correction Scheme for Object Delivery", RFC 6330, August 2011.

14.2. Informative References

- [RFC5052] Watson, M., Luby, M., and L. Vicisano, "Forward Error Correction (FEC) Building Block", RFC 5052, August 2007.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.
- [RFC6364] Begen, A., "Session Description Protocol Elements for the

Forward Error Correction (FEC) Framework", RFC 6364,
October 2011.

[dvbts] "ETSI TS 102 034 - Digital Video Broadcasting (DVB);
Transport of MPEG-2 Based DVB Services over IP Based
Networks", March 2005.

[MBMSTS] 3GPP, "Multimedia Broadcast/Multicast Service (MBMS);
Protocols and codecs", 3GPP TS 26.346, April 2005.

Authors' Addresses

Mark Watson
Netflix
100 Winchester Circle
Los Gatos, CA 95032
U.S.A.

Email: watsonm@netflix.com

Thomas Stockhammer
Nomor Research
Brecherspitzstrasse 8
Munich 81541
Germany

Email: stockhammer@nomor.de

Michael Luby
Qualcomm Incorporated
3165 Kifer Road
Santa Clara, CA 95051
U.S.A.

Email: luby@qualcomm.com

FEC Framework Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 27, 2012

M. Watson
Netflix
T. Stockhammer
Nomor Research
M. Luby
Qualcomm Incorporated
February 24, 2012

RTP Payload Format for Raptor FEC
draft-ietf-fecframe-rtp-raptor-07

Abstract

This document specifies an RTP Payload Format for Forward Error Correction repair data produced by the Raptor FEC Schemes. Raptor FEC Schemes are specified for use with the IETF FEC Framework which supports transport of repair data over both UDP and RTP. This document specifies the Payload Format which is required for the use of RTP to carry Raptor repair flows.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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

This Internet-Draft will expire on August 27, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Conventions, Definitions and Acronyms	4
3. Media Format Background	5
4. Payload Format	6
4.1. RTP Header Usage	6
4.2. Payload Header	6
4.3. Payload Data	6
5. Congestion Control Considerations	7
6. Media Types	8
6.1. Registration of the application/raptorfec media type	8
6.1.1. Media Type Definition	8
6.2. Registration of the video/raptorfec media type	9
6.2.1. Media Type Definition	9
6.3. Registration of the audio/raptorfec media type	11
6.3.1. Media Type Definition	11
6.4. Registration of the text/raptorfec media type	13
6.4.1. Media Type Definition	13
7. Mapping to SDP	15
8. Offer/Answer considerations	16
9. Declarative SDP Considerations	17
10. Repair Flow Generation and Recovery Procedures	18
10.1. Overview	18
10.2. Repair Packet Construction	18
10.3. Usage of RTCP	18
10.4. Source Packet Reconstruction	19
11. Session Description Protocol (SDP) Example	20
12. IANA Considerations	21
13. Security Considerations	22
14. References	23
14.1. Normative References	23
14.2. Informative References	24
Authors' Addresses	25

1. Introduction

The FEC Framework [RFC6363] defines a general framework for the use of Forward Error Correction in association with arbitrary packet flows, including flows over UDP and RTP [RFC3550]. Forward Error Correction operates by generating redundant data packets ("repair data") which can be sent independently from the original flow. At a receiver the original flow can be reconstructed provided a sufficient set of redundant data packets and possibly original data packets are received.

The FEC Framework provides for independence between application protocols and FEC codes. The use of a particular FEC code within the framework is defined by means of a FEC Scheme which may then be used with any application protocol compliant to the framework.

Repair data flows may be sent directly over a transport protocol such as UDP, or they may be encapsulated within specialized transports for multimedia, such as RTP.

This document defines the RTP Payload Format for the Raptor FEC Schemes defined in [I-D.ietf-fecframe-raptor].

2. Conventions, Definitions and Acronyms

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. Media Format Background

The Raptor and RaptorQ codes are efficient block-based fountain codes, meaning that from any group of source packets (or 'source block') one can generate an arbitrary number of repair packets. The Raptor and RaptorQ codes have the property that the original group of source symbols can be recovered with very high probability from any set of symbols (source and repair) only slightly greater in number than the original number of source symbols. The RaptorQ code additionally has the property that the probability that the original group of source symbols can be recovered from a set of symbols (source and repair) equal in number to the original number of source symbols is in many cases also very high.

[I-D.ietf-fecframe-raptor] defines six FEC Schemes for the use of the Raptor and RaptorQ codes with arbitrary packet flows: the first two schemes are fully applicable to arbitrary packet flows (using Raptor and RaptorQ respectively). The third and fourth schemes are slightly optimised versions of the first two schemes which are applicable in applications with relatively small block sizes. The fifth and sixth schemes are variants of the third and fourth schemes which are applicable to a single source flow which already has some kind of identifiable sequence number. The presence of a sequence number in the source flow allows for backwards-compatible operation (the source flows do not need to be modified in order to apply FEC). In this case, in the language of the FEC Framework, there is no need for an explicit FEC Source Payload Id and it is therefore not included in the packets.

This document specifies the payload format for RTP repair flows and can be used with any of the FEC Schemes defined in [I-D.ietf-fecframe-raptor].

4. Payload Format

4.1. RTP Header Usage

Header fields SHALL be set according to the rules of [RFC3550]. In addition, the following rules and definitions apply for the RTP header used with FEC repair packets:

- o Marker bit: The marker bit SHALL be set to 1 for the last protection RTP packet sent for each source block, and otherwise set to 0.
- o Payload Type (PT): The payload type codes SHALL be assigned dynamically through non-RTP means. If SDP is used for signaling, the the rules in Section 7 apply.
- o Timestamp: This field contains the time at which the packet is transmitted. The time SHOULD be as close as possible to the packet's actual time of transmission. The timestamp value has no use in the actual FEC protection process. However, implementations SHOULD supply a value that can be used for packet arrival timing or jitter calculations. The timestamp rate is specified using the "rate" media type parameter defined in Section 6. The operator SHALL select a 'rate' larger than 1000 Hz to provide sufficient resolution to RTCP operations and the operator SHOULD select the rate that matches the rate of the protected source RTP stream.
- o SSRC: The SSRC values MUST be set according to [RFC3550]. The SSRC value of the RTP repair flow MUST be different from the SSRC value of the protected source flow.

4.2. Payload Header

There is no Payload Header in this Payload Format.

4.3. Payload Data

Procedures and data formats for the use of Raptor Forward Error Correction in a FECFRAME context are fully defined in [RFC6363] and [I-D.ietf-fecframe-raptor] and are not duplicated here. The procedures of those documents apply in order to generate repair data streams to be carried by the payload formats defined in this document.

The RTP Payload SHALL contain a FEC Repair Payload as defined in [RFC6363] and [I-D.ietf-fecframe-raptor].

5. Congestion Control Considerations

See [RFC6363].

6. Media Types

6.1. Registration of the application/raptorfec media type

This RTP payload format is identified using the application/raptorfec media type which is registered in accordance with [RFC4855] and using the template of [RFC4288].

6.1.1. Media Type Definition

Type name: application

Subtype name: raptorfec

Required parameters:

- o rate: The RTP timestamp (clock) rate. The RTP timestamp (clock) rate is specified in Hz and the format is unsigned integer.
- o raptor-scheme-id: The value of this parameter is the FEC Scheme Id for the specific Raptor FEC Scheme that will be used as defined in [I-D.ietf-fecframe-raptor].
- o Kmax: The value of this parameter is the FEC Framework Configuration Information element "Maximum Source Block Length" as defined in [I-D.ietf-fecframe-raptor] encoded as a decimal integer. For specific requirements for this value refer to [I-D.ietf-fecframe-raptor].
- o T: The value of this parameter is the FEC Framework Configuration Information element "Encoding Symbol Size" as defined in [I-D.ietf-fecframe-raptor] encoded as a decimal integer. For specific requirements for this value refer to [I-D.ietf-fecframe-raptor].
- o repair-window: The maximum time that spans the source packets and the corresponding repair packets. The size of the repair window is specified in microseconds and the format is unsigned integer.

Optional parameters:

- o P: The value of this parameter is the FEC Framework Configuration Information element "Payload ID Format" as defined in [I-D.ietf-fecframe-raptor]. The default value of this parameter (when it does not appear explicitly) is 'A'.

Encoding considerations: This media type is framed and binary, see section 4.8 in [RFC4288]

Security considerations: Please see security consideration in [RFC6363]

Interoperability considerations:

Published specification: [I-D.ietf-fecframe-raptor]

Applications that use this media type: Real-time multimedia applications like video streaming, audio streaming, and video conferencing.

Additional information:

Magic number(s): <none defined>

File extension(s): <none defined>

Macintosh file type code(s): <none defined>

Person & email address to contact for further information: Thomas Stockhammer, stockhammer@nomor.de

Intended usage: COMMON

Restrictions on usage: This media type depends on RTP framing, and hence is only defined for transfer via RTP [[RFC3550]]. Transport within other framing protocols is not defined at this time.

Author: Thomas Stockhammer, Nomor Research

Change controller: IETF Audio/Video Transport working group delegated from the IESG.

6.2. Registration of the video/raptorfec media type

This RTP payload format is identified using the video/raptorfec media type which is registered in accordance with [RFC4855] and using the template of [RFC4288].

6.2.1. Media Type Definition

Type name: video

Subtype name: raptorfec

Required parameters:

- o rate: The RTP timestamp (clock) rate. The RTP timestamp (clock) rate is specified in Hz and the format is unsigned integer.
- o raptor-scheme-id: The value of this parameter is the FEC Scheme Id for the specific Raptor FEC Scheme that will be used as defined in [I-D.ietf-fecframe-raptor].
- o Kmax: The value of this parameter is the FEC Framework Configuration Information element "Maximum Source Block Length" as defined in [I-D.ietf-fecframe-raptor] encoded as a decimal integer. For specific requirements for this value refer to [I-D.ietf-fecframe-raptor].
- o T: The value of this parameter is the FEC Framework Configuration Information element "Encoding Symbol Size" as defined in [I-D.ietf-fecframe-raptor] encoded as a decimal integer. For specific requirements for this value refer to [I-D.ietf-fecframe-raptor].
- o repair-window: The maximum time that spans the source packets and the corresponding repair packets. The size of the repair window is specified in microseconds and the format is unsigned integer.

Optional parameters:

- o P: The value of this parameter is the FEC Framework Configuration Information element "Payload ID Format" as defined in [I-D.ietf-fecframe-raptor]. The default value of this parameter (when it does not appear explicitly) is 'A'.

Encoding considerations: This media type is framed and binary, see section 4.8 in [RFC4288]

Security considerations: Please see security consideration in [RFC6363]

Interoperability considerations:

Published specification: [I-D.ietf-fecframe-raptor]

Applications that use this media type: Real-time multimedia applications like video streaming, audio streaming, and video conferencing.

Additional information:

Magic number(s): <none defined>

File extension(s): <none defined>

Macintosh file type code(s): <none defined>

Person & email address to contact for further information: Thomas Stockhammer, stockhammer@nomor.de

Intended usage: COMMON

Restrictions on usage: This media type depends on RTP framing, and hence is only defined for transfer via RTP [[RFC3550]]. Transport within other framing protocols is not defined at this time.

Author: Thomas Stockhammer, Nomor Research.

Change controller: IETF Audio/Video Transport working group delegated from the IESG.

6.3. Registration of the audio/raptorfec media type

This RTP payload format is identified using the audio/raptorfec media type which is registered in accordance with [RFC4855] and using the template of [RFC4288].

6.3.1. Media Type Definition

Type name: audio

Subtype name: raptorfec

Required parameters:

- o rate: The RTP timestamp (clock) rate. The RTP timestamp (clock) rate is specified in Hz and the format is unsigned integer.
- o raptor-scheme-id: The value of this parameter is the FEC Scheme Id for the specific Raptor FEC Scheme that will be used as defined in [I-D.ietf-fecframe-raptor].
- o Kmax: The value of this parameter is the FEC Framework Configuration Information element "Maximum Source Block Length" as defined in [I-D.ietf-fecframe-raptor] encoded as a decimal integer. For specific requirements for this value refer to [I-D.ietf-fecframe-raptor].
- o T: The value of this parameter is the FEC Framework Configuration Information element "Encoding Symbol Size" as defined in [I-D.ietf-fecframe-raptor] encoded as a decimal integer. For

specific requirements for this value refer to [I-D.ietf-fecframe-raptor].

- o repair-window: The maximum time that spans the source packets and the corresponding repair packets. The size of the repair window is specified in microseconds and the format is unsigned integer.

Optional parameters:

- o P: The value of this parameter is the FEC Framework Configuration Information element "Payload ID Format" as defined in [I-D.ietf-fecframe-raptor]. The default value of this parameter (when it does not appear explicitly) is 'A'.

Encoding considerations: This media type is framed and binary, see section 4.8 in [RFC4288]

Security considerations: Please see security consideration in [RFC6363]

Interoperability considerations:

Published specification: [I-D.ietf-fecframe-raptor]

Applications that use this media type: Real-time multimedia applications like video streaming, audio streaming, and video conferencing.

Additional information:

Magic number(s): <none defined>

File extension(s): <none defined>

Macintosh file type code(s): <none defined>

Person & email address to contact for further information: Thomas Stockhammer, stockhammer@nomor.de

Intended usage: COMMON

Restrictions on usage: This media type depends on RTP framing, and hence is only defined for transfer via RTP [[RFC3550]]. Transport within other framing protocols is not defined at this time.

Author: Thomas Stockhammer, Nomor Research.

Change controller: IETF Audio/Video Transport working group delegated

from the IESG.

6.4. Registration of the text/raptorfec media type

This RTP payload format is identified using the text/raptorfec media type which is registered in accordance with [RFC4855] and using the template of [RFC4288].

6.4.1. Media Type Definition

Type name: text

Subtype name: raptorfec

Required parameters:

- o rate: The RTP timestamp (clock) rate. The RTP timestamp (clock) rate is specified in Hz and the format is unsigned integer.
- o raptor-scheme-id: The value of this parameter is the FEC Scheme Id for the specific Raptor FEC Scheme that will be used as defined in [I-D.ietf-fecframe-raptor].
- o Kmax: The value of this parameter is the FEC Framework Configuration Information element "Maximum Source Block Length" as defined in [I-D.ietf-fecframe-raptor] encoded as a decimal integer. For specific requirements for this value refer to [I-D.ietf-fecframe-raptor].
- o T: The value of this parameter is the FEC Framework Configuration Information element "Encoding Symbol Size" as defined in [I-D.ietf-fecframe-raptor] encoded as a decimal integer. For specific requirements for this value refer to [I-D.ietf-fecframe-raptor].
- o repair-window: The maximum time that spans the source packets and the corresponding repair packets. The size of the repair window is specified in microseconds and the format is unsigned integer.

Optional parameters:

- o P: The value of this parameter is the FEC Framework Configuration Information element "Payload ID Format" as defined in [I-D.ietf-fecframe-raptor]. The default value of this parameter (when it does not appear explicitly) is 'A'.

Encoding considerations: This media type is framed and binary, see section 4.8 in [RFC4288]

Security considerations: Please see security consideration in [RFC6363]

Interoperability considerations:

Published specification: [I-D.ietf-fecframe-raptor]

Applications that use this media type: Real-time multimedia applications like video streaming, audio streaming, and video conferencing.

Additional information:

Magic number(s): <none defined>

File extension(s): <none defined>

Macintosh file type code(s): <none defined>

Person & email address to contact for further information: Thomas Stockhammer, stockhammer@nomor.de

Intended usage: COMMON

Restrictions on usage: This media type depends on RTP framing, and hence is only defined for transfer via RTP [[RFC3550]]. Transport within other framing protocols is not defined at this time.

Author: Thomas Stockhammer, Nomor Research.

Change controller: IETF Audio/Video Transport working group delegated from the IESG.

7. Mapping to SDP

Applications that are using RTP transport commonly use Session Description Protocol (SDP) [RFC4566] to describe their RTP sessions. The information that is used to specify the media types in an RTP session has specific mappings to the fields in an SDP description. Note that if an application does not use SDP to describe the RTP sessions, an appropriate mapping must be defined and used to specify the media types and their parameters for the control/description protocol employed by the application.

The mapping of the above defined payload format media type and its parameters SHALL be done according to Section 3 of [RFC4855] following the suggestion therein regarding the mapping of payload-format-specific parameters into the "'a=fmtp'" field.

When the RTP Payload Formats defined in this document are used, the Media Type Parameters defined above MUST use the media types in this document and MUST NOT use those specified in [RFC6364].

8. Offer/Answer considerations

When offering Raptor FEC over RTP using SDP in an Offer/Answer model [RFC3264], the following considerations apply:

- o Each combination of the Kmax and T parameters produces different FEC data and is not compatible with any other combination. A sender application MAY desire to offer multiple offers with different sets of Kmax and T values as long as the parameter values are valid. The receiver SHOULD normally choose the offer with the largest value of the product of Kmax and T that it supports.
- o The size of the repair-window is related to the maximum delay between the transmission of a source packet and the associated repair packet. This directly impacts the buffering requirement on the receiver side and the receiver must consider this when choosing an offer.
- o When the P parameter is not present, the receiver MUST use FEC Payload ID Format A. In an answer which selects an offer in which the P parameter was omitted, the P parameter MUST either be omitted, or included with value "A".

9. Declarative SDP Considerations

In declarative usage, like SDP in the Real-time Streaming Protocol (RTSP) [RFC2326] or the Session Announcement Protocol (SAP) [RFC2974], the following considerations apply:

- o The payload format configuration parameters are all declarative and a participant **MUST** use the configuration that is provided for the session.
- o More than one configuration **MAY** be provided (if desired) by declaring multiple RTP payload types. In that case, the receivers should choose the repair session that is best for them.

10. Repair Flow Generation and Recovery Procedures

10.1. Overview

This document only specifies the repair flow construction when the repair packets are delivered with RTP. Source packet construction is covered in [I-D.ietf-fecframe-raptor]. This section provides an overview on how to generate a repair flow including the repair packets and on how to reconstruct missing source packets from a set of available source and repair packets. Detailed algorithms for the generation of Raptor and RaptorQ symbols are provided in [RFC5053] and [RFC6330], respectively.

As per the FEC Framework document [RFC6363] the FEC Framework Configuration Information includes among others the identification of the repair flow(s) and the source flow(s). Methods to convey FEC Framework Configuration Information are provided in [I-D.ietf-fecframe-config-signaling]. Specifically, the reader is referred to the SDP elements document [RFC6364], which describes the usage of 'SDP' encoding format as an example encoding format for FEC Framework Configuration Information.

For the generation of a repair flow

- o repair packets SHALL be constructed according to Section 10.2, and
- o RTCP SHALL be used according to Section 10.3.

For the reconstruction of a source packets of a source RTP session at the receiver based on the availability of a source RTP session and an repair RTP session the procedures in Section 10.4 may be used.

10.2. Repair Packet Construction

The construction of the repair packet is fully specified in Section 4. A repair packet is constructed by the concatenation of

- o an RTP header as specified in Section 4.1, and
- o payload data as defined in Section 4.3.

Repair Packet Construction may make use of the Sender Operation for RTP repair flows as specified in see [RFC6363], section 4.2.

10.3. Usage of RTCP

RTCP SHALL be used according to [RFC3550]. If the repair RTP session is sent in a separate RTP session the two sessions MUST be associated

using RTCP CNAME.

10.4. Source Packet Reconstruction

Source Packet Reconstruction may make use of the Receiver Operation for the case of RTP repair flows as specified in [RFC6363], section 4.3. Depending on the FEC scheme in use of the ones defined in [I-D.ietf-fecframe-raptor], the appropriate source blocks are formed. If enough data for decoding of any or all of the missing source payloads in the source block has been received, the respective FEC decoding procedures are applied.

In case the FEC scheme uses Raptor codes as defined in [RFC5053], then the Example FEC decoder as specified in [RFC5053], section 5.5, may be used.

In case the FEC scheme uses RaptorQ codes as defined in [RFC6330], then the Example FEC decoder as specified in [RFC6330], section 5.4, may be used.

11. Session Description Protocol (SDP) Example

This section provides an SDP [RFC4566] example. Assume we have one source video stream (mid:S1) and one FEC repair stream (mid:R1). The 'group' attribute and the FEC grouping semantics defined in [RFC5888] and [RFC5956], respectively, are used to associate source and repair flows. We form one FEC group with the "a=group:FEC S1 R1" line. The source and repair streams are sent to the same port on different multicast groups. The repair window is set to 200 ms.

```
v=0
o=ali 1122334455 1122334466 IN IP4 fec.example.com
s=Raptor RTP FEC Example
t=0 0
a=group:FEC-FR S1 R1
m=video 30000 RTP/AVP 100
c=IN IP4 233.252.0.1/127
a=rtpmap:100 MP2T/90000
a=fec-source-flow: id=0
a=mid:S1
m=application 30000 RTP/AVP 110
c=IN IP4 233.252.0.2/127
a=rtpmap:110 raptorfec/90000
a=fmtp:110 raptor-scheme-id=1; Kmax=8192; T=128; P=A; repair-window=200000
a=mid:R1
```


12. IANA Considerations

This memo requests that IANA registers `application/raptorfec` as specified in Section 6.1.1, `video/raptorfec` as specified in Section 6.2.1, `audio/raptorfec` as specified in Section 6.3.1 and `text/raptorfec` as specified in Section 6.4.1. The media type is also requested to be added to the IANA registry for "RTP Payload Format MIME types" (<http://www.iana.org/assignments/rtp-parameters>).

13. Security Considerations

Security Considerations related to the use of the FEC Framework are addressed in [RFC6363]. These consideration apply in full to users of the RTP Payload Formats defined in this document, since these are defined in terms of the FEC Framework.

No further security considerations related specifically to the Raptor FEC Schemes defined in [I-D.ietf-fecframe-raptor] have been identified.

RTP packets using the payload format defined in this specification are subject to the security considerations discussed in the RTP specification [RFC3550] and in any applicable RTP profile. The main security considerations for the RTP packet carrying the RTP payload format defined within this memo are confidentiality, integrity and source authenticity. Confidentiality is achieved by encrypting the RTP payload. Integrity of the RTP packets is achieved through a suitable cryptographic integrity protection mechanism. Such a cryptographic system can also allow the authentication of the source of the payload. A suitable security mechanism for this RTP payload format should provide confidentiality, integrity protection, and at least source authentication capable of determining if an RTP packet is from a member of the RTP session. Note that the appropriate mechanism to provide security to RTP and payloads following this memo MAY vary. It is dependent on the application, transport and signaling protocol employed. Therefore, a single mechanism is not sufficient, although if suitable, using the Secure Real-time Transport Protocol (SRTP) [RFC3711] is RECOMMENDED. Other mechanisms that may be used are IPsec [RFC4301] and Transport Layer Security (TLS) [RFC5246] (RTP over TCP); other alternatives exist.

14. References

14.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.
- [RFC4288] Freed, N. and J. Klensin, "Media Type Specifications and Registration Procedures", BCP 13, RFC 4288, December 2005.
- [RFC4855] Casner, S., "Media Type Registration of RTP Payload Formats", RFC 4855, February 2007.
- [RFC6363] Watson, M., Begen, A., and V. Roca, "Forward Error Correction (FEC) Framework", RFC 6363, October 2011.
- [RFC6364] Begen, A., "Session Description Protocol Elements for the Forward Error Correction (FEC) Framework", RFC 6364, October 2011.
- [I-D.ietf-fecframe-raptor]
Watson, M., Stockhammer, T., and M. Luby, "Raptor FEC Schemes for FECFRAME", draft-ietf-fecframe-raptor-06 (work in progress), November 2011.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.
- [RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.
- [RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.
- [RFC5053] Luby, M., Shokrollahi, A., Watson, M., and T. Stockhammer, "Raptor Forward Error Correction Scheme for Object

Delivery", RFC 5053, October 2007.

- [RFC6330] Luby, M., Shokrollahi, A., Watson, M., Stockhammer, T., and L. Minder, "RaptorQ Forward Error Correction Scheme for Object Delivery", RFC 6330, August 2011.

14.2. Informative References

- [RFC2326] Schulzrinne, H., Rao, A., and R. Lanphier, "Real Time Streaming Protocol (RTSP)", RFC 2326, April 1998.
- [RFC2974] Handley, M., Perkins, C., and E. Whelan, "Session Announcement Protocol", RFC 2974, October 2000.
- [RFC5888] Camarillo, G. and H. Schulzrinne, "The Session Description Protocol (SDP) Grouping Framework", RFC 5888, June 2010.
- [RFC5956] Begen, A., "Forward Error Correction Grouping Semantics in the Session Description Protocol", RFC 5956, September 2010.
- [I-D.ietf-fecframe-config-signaling]
Asati, R., "Methods to convey FEC Framework Configuration Information", draft-ietf-fecframe-config-signaling-06 (work in progress), September 2011.

Authors' Addresses

Mark Watson
Netflix
100 Winchester Circle
Los Gatos, CA 95032
U.S.A.

Email: watsonm@netflix.com

Thomas Stockhammer
Nomor Research
Brecherspitzstrasse 8
Munich 81541
Germany.

Email: stockhammer@nomor.de

Michael Luby
Qualcomm Incorporated
3165 Kifer Road
Santa Clara, CA 95051
U.S.A.

Email: luby@qualcomm.com

FEC Framework
Internet-Draft
Intended status: Standards Track
Expires: April 24, 2011

A. Begen
Cisco
October 21, 2010

Session Description Protocol Elements for FEC Framework
draft-ietf-fecframe-sdp-elements-11

Abstract

This document specifies the use of Session Description Protocol (SDP) to describe the parameters required to signal the Forward Error Correction (FEC) Framework Configuration Information between the sender(s) and receiver(s). This document also provides examples that show the semantics for grouping multiple source and repair flows together for the applications that simultaneously use multiple instances of the FEC Framework.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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

This Internet-Draft will expire on April 24, 2011.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction	4
2. Requirements Notation	4
3. Forward Error Correction (FEC) and FEC Framework	4
3.1. Forward Error Correction (FEC)	4
3.2. FEC Framework	5
3.3. FEC Framework Configuration Information	5
4. SDP Elements	6
4.1. Transport Protocol Identifiers	6
4.2. Media Stream Grouping	7
4.3. Source IP Addresses	7
4.4. Source Flows	7
4.5. Repair Flows	8
4.6. Repair Window	10
4.7. Bandwidth Specification	11
5. Scenarios and Examples	11
5.1. Declarative Considerations	11
5.2. Offer/Answer Model Considerations	12
6. SDP Examples	12
6.1. One Source Flow, One Repair Flow and One FEC Scheme	12
6.2. Two Source Flows, One Repair Flow and One FEC Scheme	13
6.3. Two Source Flows, Two Repair Flows and Two FEC Schemes	14
6.4. One Source Flow, Two Repair Flows and Two FEC Schemes	15
7. Security Considerations	16
8. IANA Considerations	16
8.1. Registration of Transport Protocols	16
8.2. Registration of SDP Attributes	17
9. Acknowledgments	18
10. References	18
10.1. Normative References	18
10.2. Informative References	19

Author's Address 19

1. Introduction

The Forward Error Correction (FEC) Framework, described in [I-D.ietf-fecframe-framework], outlines a general framework for using FEC-based error recovery in packet flows carrying media content. While a continuous signaling between the sender(s) and receiver(s) is not required for a Content Delivery Protocol (CDP) that uses the FEC Framework, a set of parameters pertaining to the FEC Framework has to be initially communicated between the sender(s) and receiver(s). A signaling protocol (such as the one described in [I-D.ietf-fecframe-config-signaling]) is required to enable such communication and the parameters need to be appropriately encoded so that they can be carried by the signaling protocol.

One format to encode the parameters is the Session Description Protocol (SDP) [RFC4566]. SDP provides a simple text-based format for announcements and invitations to describe multimedia sessions. These SDP announcements and invitations include sufficient information for the sender(s) and receiver(s) to participate in the multimedia sessions. SDP also provides a framework for capability negotiation, which can be used to negotiate all or a subset of the parameters pertaining to the individual sessions.

The purpose of this document is to introduce the SDP elements that are used by the CDPs using the FEC Framework that choose SDP [RFC4566] as their session description protocol.

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. Forward Error Correction (FEC) and FEC Framework

This section gives a brief overview of FEC and the FEC Framework.

3.1. Forward Error Correction (FEC)

Any application that needs a reliable transmission over an unreliable packet network has to cope with packet losses. FEC is an effective approach that provides reliable transmission particularly in multicast and broadcast applications where the feedback from the receiver(s) is either not available or quite limited.

In a nutshell, FEC groups source packets into blocks and applies

protection to generate a desired number of repair packets. These repair packets can be sent on demand or independently of any receiver feedback. The choice depends on the FEC scheme or the Content Delivery Protocol used by the application, the packet loss characteristics of the underlying network, the transport scheme (e.g., unicast, multicast and broadcast) and the application. At the receiver side, lost packets can be recovered by erasure decoding provided that a sufficient number of source and repair packets have been received.

3.2. FEC Framework

The FEC Framework [I-D.ietf-fecframe-framework] outlines a general framework for using FEC codes in multimedia applications that stream audio, video or other types of multimedia content. It defines the common components and aspects of Content Delivery Protocols (CDP). The FEC Framework also defines the requirements for the FEC schemes that need to be used within a CDP. However, the details of the FEC schemes are not specified within the FEC Framework. For example, the FEC Framework defines what configuration information has to be known at the sender and receiver(s) at minimum, but the FEC Framework neither specifies how the FEC repair packets are generated and used to recover missing source packets, nor dictates how the configuration information is communicated between the sender and receiver(s). These are rather specified by the individual FEC schemes or CDPs.

3.3. FEC Framework Configuration Information

The FEC Framework [I-D.ietf-fecframe-framework] defines a minimum set of information that has to be communicated between the sender and receiver(s) for a proper operation of an FEC scheme. This information is called the FEC Framework Configuration Information. This information includes unique identifiers for the source and repair flows that carry the source and repair packets, respectively. It also specifies how the sender applies protection to the source flow(s) and how the repair flow(s) can be used to recover lost data.

Multiple instances of the FEC Framework can simultaneously exist at the sender and the receiver(s) for different source flows, for the same source flow, or for various combinations of the source flows. Each instance of the FEC Framework provides the following FEC Framework Configuration Information:

1. Identification of the repair flows.

2. For each source flow protected by the repair flow(s):
 - a. Definition of the source flow.
 - b. An integer identifier for this flow definition (i.e., tuple). This identifier **MUST** be unique amongst all source flows that are protected by the same FEC repair flow. The identifiers **SHOULD** be allocated starting from zero and increasing by one for each flow. A source flow identifier need not be carried in source packets since source packets are directly associated with a flow by virtue of their packet headers.
3. The FEC Encoding ID, identifying the FEC scheme.
4. The length of the Explicit Source FEC Payload ID (in octets).
5. Zero or more FEC-Scheme-Specific Information (FSSI) elements, each consisting of a name and a value where the valid element names and value ranges are defined by the FEC scheme.

FSSI includes the information that is specific to the FEC scheme used by the CDP. FSSI is used to communicate the information that cannot be adequately represented otherwise and is essential for proper FEC encoding and decoding operations. The motivation behind separating the FSSI required only by the sender (which is carried in Sender-Side FEC-Scheme-Specific Information (SS-FSSI) container) from the rest of the FSSI is to provide the receiver or the third party entities a means of controlling the FEC operations at the sender. Any FSSI other than the one solely required by the sender **MUST** be communicated via the FSSI container.

The variable-length SS-FSSI and FSSI containers transmit the information in textual representation and contain zero or more distinct elements, whose descriptions are provided by the fully-specified FEC schemes.

4. SDP Elements

This section defines the SDP elements that **MUST** be used to describe the FEC Framework Configuration Information in multimedia sessions by the CDPs that choose SDP [RFC4566] as their session description protocol. Example SDP descriptions can be found in Section 6.

4.1. Transport Protocol Identifiers

This specification defines a new transport protocol identifier for the FEC schemes that take a UDP-formatted input stream and append an

Explicit Source FEC Payload ID as described in Section 5.3 of [I-D.ietf-fecframe-framework] to generate a source flow. This new protocol identifier is 'FEC/UDP'. To use input streams that are formatted according to another <proto> (as listed in the table for the 'proto' field in the Session Description Protocol (SDP) Parameters registry), the corresponding 'FEC/<proto>' transport protocol identifier MUST be registered with IANA by following the instructions specified in [RFC4566].

Note that if an FEC scheme does not use the Explicit Source FEC Payload ID as described in Section 4.1 of [I-D.ietf-fecframe-framework], then the original transport protocol identifier MUST be used to support backward compatibility with the receivers that do not support FEC at all.

This specification also defines another transport protocol identifier, 'UDP/FEC', to indicate the FEC Repair Packet format defined in Section 5.4 of [I-D.ietf-fecframe-framework]. For detailed registration information, refer to Section 8.1.

4.2. Media Stream Grouping

In FEC Framework, the 'group' attribute and the FEC grouping semantics defined in [RFC5888] and [RFC5956], respectively are used to associate source and repair flows together.

4.3. Source IP Addresses

The 'source-filter' attribute of SDP ("a=source-filter") as defined in [RFC4570] is used to express the source addresses or fully qualified domain names in the FEC Framework.

4.4. Source Flows

The FEC Framework allows that multiple source flows MAY be grouped and protected together by a single or multiple FEC Framework instances. For this reason, as described in Section 3.3, individual source flows MUST be identified with unique identifiers. For this purpose, we introduce the attribute 'fec-source-flow'.

The syntax for the new attribute in ABNF [RFC5234] is as follows:

```
fec-source-flow-line = "a=fec-source-flow:" SP source-id  
                        [";" SP tag-length] CRLF  
  
source-id = "id=" src-id  
src-id = 1*DIGIT ; Represented as 32-bit non-negative  
                ; integers and leading zeros are ignored  
  
tag-length = "tag-len=" tlen  
tlen = %x31-39 *DIGIT
```

The REQUIRED parameter 'id' is used to identify the source flow. Parameter 'id' MUST be an integer.

The 'tag-len' parameter is used to specify the length of the Explicit Source FEC Payload ID field (in octets). In the case that an Explicit Source FEC Payload ID is used, the 'tag-len' parameter MUST exist and indicate its length. Otherwise, the 'tag-len' parameter MUST NOT exist.

4.5. Repair Flows

A repair flow MUST contain only repair packets formatted as described in [I-D.ietf-fecframe-framework] for a single FEC Framework instance, i.e., packets belonging to source flows or other repair flows from a different FEC Framework instance cannot be sent within this flow. We introduce the attribute 'fec-repair-flow' to describe the repair flows.

The syntax for the new attribute in ABNF is as follows (CHAR and CTL are defined in [RFC5234]):

```

fec-repair-flow-line = "a=fec-repair-flow:" SP fec-encoding-id
                        [";" SP flow-preference]
                        [";" SP sender-side-scheme-specific]
                        [";" SP scheme-specific] CRLF

fec-encoding-id = "encoding-id=" enc-id
enc-id = 1*DIGIT ; FEC Encoding ID

flow-preference = "preference-lvl=" preference-level-of-the-flow
preference-level-of-the-flow = 1*DIGIT

sender-side-scheme-specific = "ss-fssi=" sender-info
sender-info = element *( "," element )
element      = name ":" value
name          = token
token         = 1*<any CHAR except CTLs or separators>
value         = *<any CHAR except CTLs or separators>
separator     = "(" / ")" / "<" / ">" / "@"
               / "," / ";" / ":" / "\" / DQUOTE
               / "/" / "[" / "]" / "?" / "="
               / "{" / "}" / SP / HTAB

scheme-specific = "fssi=" scheme-info
scheme-info = element *( "," element )

```

The REQUIRED parameter 'encoding-id' is used to identify the FEC scheme used to generate this repair flow. These identifiers (in the range of [0 - 255]) are registered by the FEC schemes that use the FEC Framework and are maintained by IANA.

The OPTIONAL parameter 'preference-lvl' is used to indicate the preferred order of using the repair flows. The exact usage of the parameter 'preference-lvl' and the pertaining rules MAY be defined by the FEC scheme or the CDP. If the parameter 'preference-lvl' does not exist, it means that the receiver(s) MAY receive and use the repair flows in any order. However, if a preference level is assigned to the repair flow(s), the receivers are encouraged to follow the specified order in receiving and using the repair flow(s).

The OPTIONAL parameters 'ss-fssi' and 'fssi' are containers to convey the FEC-Scheme-Specific Information (FSSI) that includes the information that is specific to the FEC scheme used by the CDP and is necessary for proper FEC encoding and decoding operations. The FSSI required only by the sender (called Sender-Side FSSI) MUST be communicated in the container specified by the parameter 'ss-fssi'. Any other FSSI MUST be communicated in the container specified by the parameter 'fssi'. In both containers, FSSI is transmitted in the

form of textual representation and MAY contain multiple distinct elements. If the FEC scheme does not require any specific information, the 'ss-fssi' and 'fssi' parameters MUST NOT exist.

4.6. Repair Window

Repair window is the time that spans an FEC block, which consists of the source block and the corresponding repair packets.

At the sender side, the FEC encoder processes a block of source packets and generates a number of repair packets. Then both the source and repair packets are transmitted within a certain duration not larger than the value of the repair window. The value of the repair window impacts the maximum number of source packets that can be included in an FEC block.

At the receiver side, the FEC decoder should wait at least for the duration of the repair window after getting the first packet in an FEC block to allow all the repair packets to arrive (The waiting time can be adjusted if there are missing packets at the beginning of the FEC block). The FEC decoder can start decoding the already received packets sooner, however, it SHOULD NOT register an FEC decoding failure until it waits at least for the repair-window duration.

This document specifies a new attribute to describe the size of the repair window in milliseconds and microseconds.

The syntax for the attribute in ABNF is as follows:

```
repair-window-line = "a=repair-window:" window-size unit CRLF
```

```
window-size = %x31-39 *DIGIT ; Represented as  
                                ; 32-bit non-negative integers
```

```
unit = "ms" / "us"
```

<unit> is the unit of time the repair window size is specified with. Two units are defined here: 'ms', which stands for milliseconds and 'us', which stands for microseconds.

The 'a=repair-window' attribute is a media-level attribute since each repair flow MAY have a different repair window size.

Specifying the repair window size in an absolute time value does not necessarily correspond to an integer number of packets or exactly match with the clock rate used in RTP (in case of RTP transport) causing mismatches among subsequent repair windows. However, in practice, this mismatch does not break anything in the FEC decoding

process.

4.7. Bandwidth Specification

The bandwidth specification as defined in [RFC4566] denotes the proposed bandwidth to be used by the session or media. The specification of bandwidth is OPTIONAL.

In the context of the FEC Framework, the bandwidth specification can be used to express the bandwidth of the repair flows or the bandwidth of the session. If included in the SDP, it SHALL adhere to the following rules:

The session-level bandwidth for an FEC Framework instance or the media-level bandwidth for the individual repair flows MAY be specified. In this case, it is RECOMMENDED to use the Transport Independent Application Specific (TIAS) bandwidth modifier [RFC3890] and the 'a=maxprate' attribute unless the Application Specific (AS) bandwidth modifier [RFC4566] is used. The use of AS bandwidth modifier is NOT RECOMMENDED since TIAS allows the calculation of the bitrate according to the IP version and transport protocol, whereas AS does not. Thus, in TIAS-based bitrate calculations, the packet size SHALL include all headers and payload, excluding the IP and UDP headers. In AS-based bitrate calculations, the packet size SHALL include all headers and payload, plus the IP and UDP headers.

For the ABNF syntax information of the TIAS and AS, refer to [RFC3890] and [RFC4566], respectively.

5. Scenarios and Examples

This section discusses the considerations for Session Announcement and Offer/Answer Models.

5.1. Declarative Considerations

In multicast-based applications, the FEC Framework Configuration Information pertaining to all FEC protection options available at the sender MAY be advertised to the receivers as a part of a session announcement. This way, the sender can let the receivers know all available options for FEC protection. Based on their needs, the receivers MAY choose protection provided by one or more FEC Framework instances and subscribe to the respective multicast session(s) to receive the repair flow(s). Unless explicitly required by the CDP, the receivers SHOULD NOT send an answer back to the sender specifying their choices since this can easily overwhelm the sender particularly in large-scale multicast applications.

5.2. Offer/Answer Model Considerations

In unicast-based applications, a sender and receiver MAY adopt the Offer/Answer Model [RFC3264] to set the FEC Framework Configuration Information. In this case, the sender offers the options available to this particular receiver and the receiver answers back to the sender with its choice(s).

Receivers supporting the SDP Capability Negotiation Framework [RFC5939] MAY also use this framework to negotiate all or a subset of the FEC Framework parameters.

The backward compatibility in Offer/Answer Model is handled as specified in [RFC5956].

6. SDP Examples

This section provides SDP examples that can be used by the FEC Framework.

[RFC5888] defines the media stream identification attribute ('mid') as a token in ABNF. In contrast, the identifiers for the source flows MUST be integers and SHOULD be allocated starting from zero and increasing by one for each flow. To avoid any ambiguity, using the same values for identifying the media streams and source flows is NOT RECOMMENDED, even when 'mid' values are integers.

In the examples below, random FEC Encoding IDs will be used for illustrative purposes. Artificial content for the SS-FSSI and FSSI will also be provided.

6.1. One Source Flow, One Repair Flow and One FEC Scheme

SOURCE FLOWS		INSTANCE #1
S1: Source Flow	-----	R1: Repair Flow

Figure 1: Scenario #1

In this example, we have one source video flow (mid:S1) and one FEC repair flow (mid:R1). We form one FEC group with the "a=group:FEC-FR S1 R1" line. The source and repair flows are sent to the same port on different multicast groups. The repair window is set to 150 ms.

```

v=0
o=ali 1122334455 1122334466 IN IP4 fec.example.com
s=FEC Framework Examples
t=0 0
a=group:FEC-FR S1 R1
m=video 30000 RTP/AVP 100
c=IN IP4 233.252.0.1/127
a=rtpmap:100 MP2T/90000
a=fec-source-flow: id=0
a=mid:S1
m=application 30000 UDP/FEC
c=IN IP4 233.252.0.2/127
a=fec-repair-flow: encoding-id=0; ss-fssi=n:7,k:5
a=repair-window:150ms
a=mid:R1

```

6.2. Two Source Flows, One Repair Flow and One FEC Scheme

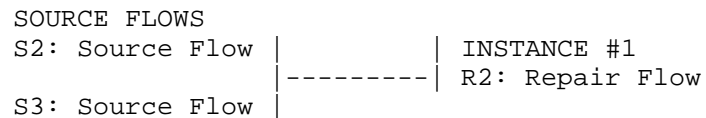


Figure 2: Scenario #2

In this example, we have two source video flows (mid:S2 and mid:S3) and one FEC repair flow (mid:R2), protecting both source flows. We form one FEC group with the "a=group:FEC-FR S2 S3 R2" line. The source and repair flows are sent to the same port on different multicast groups. The repair window is set to 150500 us.

```

v=0
o=ali 1122334455 1122334466 IN IP4 fec.example.com
s=FEC Framework Examples
t=0 0
a=group:FEC-FR S2 S3 R2
m=video 30000 RTP/AVP 100
c=IN IP4 233.252.0.1/127
a=rtpmap:100 MP2T/90000
a=fec-source-flow: id=0
a=mid:S2
m=video 30000 RTP/AVP 101
c=IN IP4 233.252.0.2/127
a=rtpmap:101 MP2T/90000
a=fec-source-flow: id=1
a=mid:S3
m=application 30000 UDP/FEC
c=IN IP4 233.252.0.3/127
a=fec-repair-flow: encoding-id=0; ss-fssi=n:7,k:5
a=repair-window:150500us
a=mid:R2

```

6.3. Two Source Flows, Two Repair Flows and Two FEC Schemes

SOURCE FLOWS		INSTANCE #1
S4: Source Flow -----		R3: Repair Flow
S5: Source Flow -----		INSTANCE #2
		R4: Repair Flow

Figure 3: Scenario #3

In this example, we have two source video flows (mid:S4 and mid:S5) and two FEC repair flows (mid:R3 and mid:R4). The source flows mid:S4 and mid:S5 are protected by the repair flows mid:R3 and mid:R4, respectively. We form two FEC groups with the "a=group:FEC-FR S4 R3" and "a=group:FEC-FR S5 R4" lines. The source and repair flows are sent to the same port on different multicast groups. The repair window is set to 200 ms and 400 ms for the first and second FEC group, respectively.

```

v=0
o=ali 1122334455 1122334466 IN IP4 fec.example.com
s=FEC Framework Examples
t=0 0
a=group:FEC-FR S4 R3
a=group:FEC-FR S5 R4
m=video 30000 RTP/AVP 100
c=IN IP4 233.252.0.1/127
a=rtpmap:100 MP2T/90000
a=fec-source-flow: id=0
a=mid:S4
m=video 30000 RTP/AVP 101
c=IN IP4 233.252.0.2/127
a=rtpmap:101 MP2T/90000
a=fec-source-flow: id=1
a=mid:S5
m=application 30000 UDP/FEC
c=IN IP4 233.252.0.3/127
a=fec-repair-flow: encoding-id=0; ss-fssi=n:7,k:5
a=repair-window:200ms
a=mid:R3
m=application 30000 UDP/FEC
c=IN IP4 233.252.0.4/127
a=fec-repair-flow: encoding-id=0; ss-fssi=n:14,k:10
a=repair-window:400ms
a=mid:R4

```

6.4. One Source Flow, Two Repair Flows and Two FEC Schemes

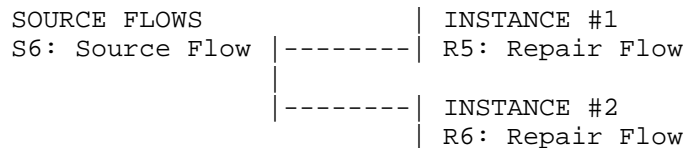


Figure 4: Scenario #4

In this example, we have one source video flow (mid:S6) and two FEC repair flows (mid:R5 and mid:R6) with different preference levels. The source flow mid:S6 is protected by both of the repair flows. We form two FEC groups with the "a=group:FEC-FR S6 R5" and "a=group:FEC-FR S6 R6" lines. The source and repair flows are sent to the same port on different multicast groups. The repair window is set to 200 ms for both FEC groups.

```
v=0
o=ali 1122334455 1122334466 IN IP4 fec.example.com
s=FEC Framework Examples
t=0 0
a=group:FEC-FR S6 R5
a=group:FEC-FR S6 R6
m=video 30000 RTP/AVP 100
c=IN IP4 233.252.0.1/127
a=rtpmap:100 MP2T/90000
a=fec-source-flow: id=0
a=mid:S6
m=application 30000 UDP/FEC
c=IN IP4 233.252.0.3/127
a=fec-repair-flow: encoding-id=0; preference-lvl=0; ss-fssi=n:7,k:5
a=repair-window:200ms
a=mid:R5
m=application 30000 UDP/FEC
c=IN IP4 233.252.0.4/127
a=fec-repair-flow: encoding-id=1; preference-lvl=1; ss-fssi=t:3
a=repair-window:200ms
a=mid:R6
```

7. Security Considerations

There is a weak threat if the SDP is modified in a way that it shows incorrect association and/or grouping of the source and repair flows. Such attacks can result in failure of FEC protection and/or mishandling of other media streams. It is RECOMMENDED that the receiver does integrity check on SDP to only trust SDP from trusted sources. The receiver MUST also follow the security considerations of SDP [RFC4566]. For other general security considerations related to SDP, refer to [RFC4566]. For the security considerations related to the use of source address filters in SDP, refer to [RFC4570].

The security considerations for the FEC Framework also apply. Refer to [I-D.ietf-fecframe-framework] for details.

8. IANA Considerations

Note to the RFC Editor: In the following, please replace "XXXX" with the number of this document prior to publication as an RFC.

8.1. Registration of Transport Protocols

This specification updates the Session Description Protocol (SDP) Parameters registry as defined in Section 8.2.2 of [RFC4566].

Specifically, it adds the following values to the table for the 'proto' field.

Type	SDP Name	Reference
-----	-----	-----
proto	FEC/UDP	[RFCXXXX]
proto	UDP/FEC	[RFCXXXX]

This specification also defines a class of new transport protocol identifiers. For all existing identifiers <proto> (listed in the table for the 'proto' field in the Session Description Protocol (SDP) Parameters registry), this specification defines the identifier 'FEC/<proto>'.

8.2. Registration of SDP Attributes

This document registers new attribute names in SDP.

SDP Attribute ("att-field"):

Attribute name: fec-source-flow
Long form: Pointer to FEC Source Flow
Type of name: att-field
Type of attribute: Media level
Subject to charset: No
Purpose: Provide parameters for an FEC source flow
Reference: [RFCXXXX]
Values: See [RFCXXXX]

SDP Attribute ("att-field"):

Attribute name: fec-repair-flow
Long form: Pointer to FEC Repair Flow
Type of name: att-field
Type of attribute: Media level
Subject to charset: No
Purpose: Provide parameters for an FEC repair flow
Reference: [RFCXXXX]
Values: See [RFCXXXX]

SDP Attribute ("att-field"):

Attribute name: repair-window
Long form: Pointer to FEC Repair Window
Type of name: att-field
Type of attribute: Media level
Subject to charset: No
Purpose: Indicate the size of the repair window
Reference: [RFCXXXX]
Values: See [RFCXXXX]

9. Acknowledgments

The author would like to thank the FEC Framework Design Team for their inputs, suggestions and contributions.

10. References

10.1. Normative References

- [I-D.ietf-fecframe-framework]
Watson, M., "Forward Error Correction (FEC) Framework",
draft-ietf-fecframe-framework-10 (work in progress),
September 2010.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session
Description Protocol", RFC 4566, July 2006.
- [RFC4570] Quinn, B. and R. Finlayson, "Session Description Protocol
(SDP) Source Filters", RFC 4570, July 2006.
- [RFC5888] Camarillo, G. and H. Schulzrinne, "The Session Description
Protocol (SDP) Grouping Framework", RFC 5888, June 2010.
- [RFC5956] Begen, A., "Forward Error Correction Grouping Semantics in
the Session Description Protocol", RFC 5956,
September 2010.
- [RFC3890] Westerlund, M., "A Transport Independent Bandwidth
Modifier for the Session Description Protocol (SDP)",
RFC 3890, September 2004.
- [RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax
Specifications: ABNF", STD 68, RFC 5234, January 2008.

- [RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.

10.2. Informative References

- [I-D.ietf-fecframe-config-signaling]
Asati, R., "Methods to convey FEC Framework Configuration Information", draft-ietf-fecframe-config-signaling-03 (work in progress), June 2010.
- [RFC5939] Andreasen, F., "Session Description Protocol (SDP) Capability Negotiation", RFC 5939, September 2010.

Author's Address

Ali Begen
Cisco
181 Bay Street
Toronto, ON M5J 2T3
Canada

Email: abegen@cisco.com

