Internet-Draft Cisco Systems
Intended status: Standards Track November 3, 2008
Expires: May 7, 2009

SDP Elements for FEC Framework
draft-ietf-fecframe-sdp-elements-02

Status of this Memo

Copyright Notice

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.

Table of Contents

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 MUST 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 must 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 MAY 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 MUST be 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 potentially limited.

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

protection to generate a desired number of repair packets. These repair packets may 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.

For a proper operation, the information required by the FEC Framework and the details of an FEC scheme have to be communicated between the sender and receiver(s). One way to provide this information is to use the Session Description Protocol (SDP) [RFC4566]. SDP provides a commonly used text-based format for announcements and invitations that describe multimedia sessions. These SDP announcements and invitations include sufficient information for clients to participate in multimedia sessions. By using the SDP capability negotiation framework, all or a subset of the parameters pertaining to the FEC Framework MAY also be negotiated between the sender and receiver(s).

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

Note that there are many similarities between the FEC Framework [I-D.ietf-fecframe-framework] and the FEC Building Block [RFC5052], which describes a framework that uses FEC codes to provide reliability to bulk data transfer applications running over IP multicast or broadcast. See [I-D.ietf-fecframe-framework] for further details.

3.3. FEC Framework Configuration Information

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

The FEC Framework Configuration Information includes identifiers for unique identification of the source and repair flows that carry the source and repair packets, respectively. For example, a packet flow that is transmitted over UDP is uniquely identified by the tuple of {Source IP Address, Destination IP Address, Source UDP Port, Destination UDP Port}. However, an integer identifier MAY be used internally within the FEC scheme as a shorthand to identify this flow.

Multiple instances of the FEC Framework MAY simultaneously exist at the sender and the receiver(s) for different source flows, for the same source flow, or for various combinations of source flows. Each instance of the FEC Framework MUST provide 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. Note that an application MAY wildcard some of the fields if only a subset of the fields of the tuple (e.g., {Destination IP Address, Destination UDP Port}) is sufficient.

3. The FEC Encoding ID that identifies the FEC scheme.

4. The length of the Explicit Source FEC Payload ID (in bytes).

   This value MAY be zero indicating that no Explicit Source FEC Payload ID is used by the FEC scheme. If it is nonzero, however, it means that the Explicit Source FEC Payload ID is used. In this case, only one FEC scheme MUST be used for this source flow, unless the generic tag (defined in [I-D.ietf-fecframe-framework]) is used by all of the FEC schemes protecting this source flow.

5. An opaque container for the FEC-Scheme-Specific Information (FSSI) that is required by only the receiver or by both the receiver and sender.

6. Another opaque container for the FSSI that is only required by the sender. This is referred to as the Sender-Side FEC-Scheme-Specific Information (SS-FSSI).

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 from the rest of the FSSI is to provide the receiver or the 3rd 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 opaque SS-FSSI and FSSI containers transmit the information in the form of an octet string. The FEC schemes define the structure of this octet string, which MAY contain multiple distinct elements. If the FEC scheme does not require any specific information, the FSSI MAY be null. For the fully-specified FEC schemes, a full description of the encoded information in both containers MUST be provided. See [I-D.ietf-fecframe-framework] for details.

4. SDP Descriptors for FEC Framework

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 configurations can be found in Section 5.

4.1. Transport Protocol Identifiers

This specification defines a class of new transport protocol identifiers for SDP media descriptions. For all existing identifiers <proto>, this specification defines the identifier 'FEC/<proto>'.

This identifier MAY be used as the transport protocol identifier in the media descriptions for the source data to indicate that the FEC Source Packet format defined in Section 6.3 of [I-D.ietf-fecframe-framework] is used, where the original transport payload field is formatted according to <proto>. However, if the FEC scheme does not use the Explicit Source FEC Payload ID as described in Section 6.3 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 6.4 of [I-D.ietf-fecframe-framework].

4.2. Media Stream Grouping

The FEC Framework [I-D.ietf-fecframe-framework] states that multiple instances of the FEC Framework MAY exist at the sender and the receiver(s), and a source flow MAY be protected by multiple FEC Framework instances. Furthermore, within a single FEC Framework instance, multiple source flows MAY be protected by multiple repair flows. However, each repair flow MUST provide protection for a single FEC Framework instance. An example scenario is shown in Figure 1. Here, source flows 0 and 1 are grouped together and protected by repair flow 3; source flow 0 is also protected by repair flow 4; source flows 1 and 2 are grouped together and protected by repair flows 5, 6 and 7.

The motivation behind grouping source flows before applying FEC protection is that a better coding performance may be achieved by doing so and many receivers may benefit from this grouping. For example, consider a layered video source that consists of one base layer (e.g., source flow 0) and one enhancement layer (e.g., source flow 1), where each layer is carried in a separate flow. Repair flow 3 protects the combination of the base and enhancement layers for the receivers who receive both layers, and repair flow 4 protects the base layer only, for the receivers who want the base layer only, or who receive both layers but prefer FEC protection for the base layer only due to a bandwidth and/or processing-power limitation.

Using multiple FEC Framework instances for a single source flow provides flexibility to the receivers. Different instances may offer repair flows that are generated by different FEC schemes, and receivers choose receiving the appropriate repair flow(s) that they can support and decode. Alternatively, different instances (whether they use the same FEC scheme or not) may use larger and smaller source block sizes, which accommodate the receivers that have looser and tighter latency requirements, respectively. In addition,

different instances may also provide FEC protection at different redundancy levels. This is particularly useful in multicast scenarios where different receivers might experience different packet loss rates and each receiver can choose the repair flow that is tailored to its needs.

```
                                          ____| FEC FRAMEWORK
                                         /    | INSTANCE
                                        /     | 3: Repair Flow
                                       /
                  SOURCE FLOWS        /       | FEC FRAMEWORK
                  0: Source Flow |___/ |------| INSTANCE
              __| 1: Source Flow |            | 4: Repair Flow
             |  | 2: Source Flow
             |                          | FEC FRAMEWORK
             |__________________________| INSTANCE
                                        | 5: Repair Flow
                                        | 6: Repair Flow
                                        | 7: Repair Flow
```

Figure 1: Example scenario with multiple FEC Framework instances

The 'group' attribute and the FEC grouping semantics defined in [RFC4756] are used to associate source and repair flows together with the following additional requirement:

In the case that the Explicit Source FEC Payload ID is used, then only one FEC scheme MUST be used for this source flow, unless the generic tag is used by all of the FEC schemes for the Source FEC Payload ID, as defined in [I-D.ietf-fecframe-framework].

The 'group' attribute MAY be used to group multiple repair flows with one or more source flows. Note that [RFC3388] prohibits an "m" line identified by its 'mid' attribute from appearing in more than one "a=group:FEC" line. Thus, [RFC3388] mandates us to write

```
a=group:FEC 0 1 2 3 4 5 6 7
```

for the scenario sketched in Figure 1. This limitation prevents us from indicating particular associations between the source and repair flows by using an "a=group:FEC" line per FEC Framework instance [RFC4756].

Editor's note: The FEC grouping and flow association issues are currently under discussion in FECFRAME and MMUSIC WGs (See [I-D.begen-mmusic-rfc4756bis]). This section will be updated once a decision is made.

The FEC Framework also supports additivity among the repair flows,

meaning that multiple repair flows MAY be decoded jointly to improve the recovery chances of the missing packets. In addition, the sender MAY assign different levels of priority to each repair flow. See Section 4.5 for details.

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.

Editor's note: Additional requirements or exceptions regarding source filters are TBD.

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:" source-id
     [";" SP tag-length] CRLF

source-id = "id=" src-id
src-id = 1*DIGIT

tag-length = "tag-len=" tlen
tlen = *DIGIT
```

The MANDATORY parameter 'id' is used to identify the source flow. Note that the parameter 'id' MUST be an integer.

The OPTIONAL 'tag-len' parameter is used to specify the length of the Explicit Source FEC Payload ID field (in bytes) and MUST be used according to the requirements listed in Section 4.2. If no value is specified for the 'tag-len' parameter, it indicates a value of zero.

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. In other words, packets belonging to source flows or other repair flows from a different FEC Framework instance MUST NOT 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:

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

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

flow-priority = "priority=" priority-of-the-flow
priority-of-the-flow = *DIGIT

sender-side-scheme-specific = "ss-fssi=" sender-info
sender-info = *CHAR

scheme-specific = "fssi=" scheme-info
scheme-info = *CHAR
```

The MANDATORY parameter 'encoding-id' is used to identify the FEC scheme used to generate this repair flow. These identifiers MUST be registered with IANA by the FEC schemes that use the FEC Framework.

The OPTIONAL parameter 'priority' is used to indicate the priorities of the repair flows. The exact usage of the parameter 'priority' and the pertaining rules MAY be defined by the FEC scheme or the CDP. If no value is specified for the parameter 'priority', it means that the receiver(s) MAY receive and use the repair flows in any order. However, if a priority is assigned to the repair flow(s), the receivers MUST follow the specified order in receiving and using the repair flow(s).

The OPTIONAL parameters 'ss-fssi' and 'fssi' are opaque 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 an octet string. The FEC schemes define the structure of this octet string, which MAY contain multiple distinct elements. If the FEC scheme does not require any specific information, the 'ss-fssi' and 'fssi' parameters MAY be null and ignored.

4.6. Repair Window

An FEC encoder processes a block of source packets and generates a number of repair packets, which are then transmitted within a certain duration. At the receiver side, the FEC decoder tries to decode all the packets received within the repair window to recover the missing packets, if there are any. Repair window stands for the time that spans the source packets and the corresponding repair packets. Assuming that there is no issue of delay variation, the FEC decoder SHOULD NOT wait longer than the repair window since additional waiting would not help the recovery process.

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
     [SP unit] CRLF

window-size = 1*DIGIT

unit = ms / us
```

<unit> is the unit of time the repair window size is specified with. Currently, two units are defined: "ms", which stands for milliseconds and "us", which stands for microseconds. The default unit is "ms". Alternative units MAY be defined in the future by registering them with IANA.

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

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 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 for the session.

The media-level bandwidth for the individual repair flows MAY also be specified. In this case, it is RECOMMENDED to use the TIAS bandwidth modifier [RFC3890].

The Application Specific (AS) bandwidth modifier [RFC4566] MAY be used instead of TIAS, however, this 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. SDP examples that can be used by the FEC Framework are also provided.

5.1. Session Announcement 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 group(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.

5.2. Offer/Answer 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 all available options to the receiver and the receiver answers back to the sender with its choice(s). Note that some FEC protection options MAY be offered to only a particular set of (e.g., premium) receivers.

Receivers supporting the SDP Capability Negotiation Framework [I-D.ietf-mmusic-sdp-capability-negotiation] 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 [RFC3388]. If a receiver receives an offer containing FEC grouping and it does not understand the FEC grouping semantics, it MAY respond with an answer that ignores the grouping attribute or MAY refuse the request. In the first case, the offerer MUST establish the connection without FEC. In the second case, if the offerer still wishes to establish the session, it SHOULD retry the request with an offer without FEC.

5.3. Examples

Editor's note: More examples showing the usage of multiple FEC Framework instances, additivity of the repair flows and prioritization of the repair flows will be provided once the issues related to FEC grouping and flow association are resolved.

Editor's note: As of now, no FEC Encoding ID has been registered with IANA. In the examples below, an FEC Encoding ID of zero will be used for illustrative purposes. Artificial content for the SS-FSSI and FSSI will also be provided.

[RFC3388] 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.

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

```
SOURCE FLOWS             | INSTANCE #1
0: Source Flow |---------| 1: Repair Flow
```

Figure 6: 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 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 S1 R1
m=video 30000 RTP/AVP 100
c=IN IP4 224.1.1.1/127
a=rtpmap:100 MP2T/90000
a=fec-source-flow: id=0
a=mid:S1
m=application 30000 udp/fec
c=IN IP4 224.1.2.1/127
a=fec-repair-flow: encoding-id=0; ss-fssi=1Q2A3Z; fssi=4W5S6X
a=repair-window: 150
a=mid:R1
```

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

```
SOURCE FLOWS             | INSTANCE #1
0: Source Flow |_________| 2: Repair Flow
1: Source Flow |
```

Figure 8: Scenario #2

In this example, we have two source video flows (mid:S1 and mid:S2) and one FEC repair flow (mid:R1), protecting both source flows. We form one FEC group with the "a=group:FEC S1 S2 R1" 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 S1 S2 R1
m=video 30000 RTP/AVP 100
c=IN IP4 224.1.1.1/127
a=rtpmap:100 MP2T/90000
a=fec-source-flow: id=0
a=mid:S1
m=video 30000 RTP/AVP 101
c=IN IP4 224.1.1.2/127
a=rtpmap:101 MP2T/90000
a=fec-source-flow: id=1
a=mid:S2
m=application 30000 udp/fec
c=IN IP4 224.1.2.1/127
a=fec-repair-flow: encoding-id=0; ss-fssi=1Q2A3Z; fssi=4W5S6X
a=repair-window: 150500 us
a=mid:R1
```

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

```
SOURCE FLOWS              | INSTANCE #1
0: Source Flow |---------| 2: Repair Flow
1: Source Flow |_
                 \-------| INSTANCE #2
                         | 3: Repair Flow
```

Figure 10: Scenario #3

In this example, we have two source video flows (mid:S1 and mid:S2) and two FEC repair flows (mid:R1 and mid:R2). The source flows mid:S1 and mid:S2 are protected by the repair flows mid:R1 and mid:R2, respectively. We form two FEC groups with the "a=group:FEC S1 R1" and "a=group:FEC S2 R2" 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 S1 R1
a=group:FEC S2 R2
m=video 30000 RTP/AVP 100
c=IN IP4 224.1.1.1/127
a=rtpmap:100 MP2T/90000
a=fec-source-flow: id=0
a=mid:S1
m=video 30000 RTP/AVP 101
c=IN IP4 224.1.1.2/127
a=rtpmap:101 MP2T/90000
a=fec-source-flow: id=1
a=mid:S2
m=application 30000 udp/fec
c=IN IP4 224.1.2.1/127
a=fec-repair-flow: encoding-id=0; ss-fssi=1Q2A3Z; fssi=4W5S6X
a=repair-window: 200 ms
a=mid:R1
m=application 30000 udp/fec
c=IN IP4 224.1.2.2/127
a=fec-repair-flow: encoding-id=0; ss-fssi=123QAZ; fssi=456WSX
a=repair-window: 400 ms
a=mid:R2
```

6. Security Considerations

For the general security considerations related to SDP, refer to [RFC4566]. For the security considerations related to source/FEC media stream grouping in SDP and use of source address filters in SDP, refer to [RFC4756] and [RFC4570], respectively.

7. IANA Considerations

7.1. Transport Protocols

The 'proto' sub-field of the media description line ("m=") describes the transport protocol used. This document registers the following values:

UDP/FEC

Editor's note: Additional transport protocols to be registered are TBD.

7.2. Attribute Names

As recommended by [RFC4566], the following attribute names should be registered with IANA.

The contact information for the registrations is:

Ali Begen
abegen@cisco.com

```
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:            See this document
     Reference:          This document
     Values:             See this document

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:            See this document
     Reference:          This document
     Values:             See this document

SDP Attribute ("att-field"):
     Attribute name:     repair-window
     Long form:          Repair Window Size
     Type of name:       att-field
     Type of attribute:  Media level
     Subject to charset: No
     Purpose:            See this document
     Reference:          This document
     Values:             See this document
```

8. Acknowledgments

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

9. Change Log

9.1. draft-ietf-fecframe-sdp-elements-02

The following are the major changes compared to version 01:

- o Clarified the definitions for the FSSI fields.
- o Hostnames in the SDP examples are fixed.

9.2. draft-ietf-fecframe-sdp-elements-01

The following are the major changes compared to version 00:

- o Additive repair flows can now be from different instances. The sender may also assign different levels of priorities to each repair flow regardless of whether the repair flows are additive or not.
- o SDP examples are fixed.
- o Comments posted in the mailing list are incorporated.

9.3. draft-ietf-fecframe-sdp-elements-00

This is the initial version, which is based on an earlier individual submission. The following are the major changes compared to that document:

- o The opaque container in the FEC Framework Configuration Information (FEC-Scheme-Specific Information) is now divided into two parts: information needed only by the sender and information needed by the receiver. The repair flow descriptors are also updated accordingly.
- o "Minimum Buffer Size" is now called "Repair Window." Its size can also be specified in microseconds in addition to milliseconds.
- o Simple examples with complete SDPs are included.
- o "Scheme ID" is changed to "Encoding ID" to be consistent with the framework draft.

o Some other editorial changes.

10. References

10.1. Normative References

[I-D.ietf-fecframe-framework]
Watson, M., "Forward Error Correction (FEC) Framework", draft-ietf-fecframe-framework-03 (work in progress), October 2008.

[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.

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

[RFC3388] Camarillo, G., Eriksson, G., Holler, J., and H. Schulzrinne, "Grouping of Media Lines in the Session Description Protocol (SDP)", RFC 3388, December 2002.

[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

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

[I-D.ietf-fecframe-config-signaling]
Asati, R., "Methods to convey FEC Framework Configuration Information", draft-ietf-fecframe-config-signaling-01 (work in progress), November 2008.

[I-D.begen-mmusic-rfc4756bis]
Begen, A., "Forward Error Correction Grouping Semantics in Session Description Protocol", draft-begen-mmusic-rfc4756bis-00 (work in progress), October 2008.

[I-D.ietf-mmusic-sdp-capability-negotiation]
Andreasen, F., "SDP Capability Negotiation", draft-ietf-mmusic-sdp-capability-negotiation-09 (work in progress), July 2008.

Author's Address

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

Email: abegen@cisco.com

Full Copyright Statement

Intellectual Property

Acknowledgment

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