MMUSIC A. Begen
Internet-Draft Cisco Systems
Obsoletes: 4756 September 23, 2009
(if approved)
Intended status: Standards Track
Expires: March 27, 2010

Forward Error Correction Grouping Semantics in Session Description Protocol
draft-ietf-mmusic-rfc4756bis-03

Status of this Memo

Copyright Notice

Abstract

The Session Description Protocol (SDP) supports grouping media lines. SDP also has semantics defined for grouping the associated source and Forward Error Correction (FEC)-based repair flows. However, the semantics that was defined in RFC 4756 generally fail to provide the specific grouping relationships between the source and repair flows when there are more than one source and/or repair flows in the same group. Furthermore, the existing semantics does not support describing additive repair flows. This document addresses these issues by introducing new FEC grouping semantics. SSRC-level grouping semantics is also introduced in this document for Real-time Transport Protocol (RTP) streams using SSRC multiplexing.

Table of Contents

1. Introduction

Any application that needs a reliable transmission over an unreliable packet network has to cope with packet losses. Forward Error Correction (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, 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.

For example, one of the most basic FEC schemes is the parity codes, where an exclusive OR (XOR) operation is applied to a group of packets (i.e., source block) to generate a single repair packet. At the receiver side, this scheme provides a full recovery if only one packet is lost within the source block and the repair packet is received. There are various other ways of generating repair packets, possibly with different loss-recovery capabilities.

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. The FEC Framework specification states that source and repair packets MUST be carried in different streams, which are referred to as the source and repair flows, respectively. At the receiver side, the receivers should know which flows are the source flows and which flows are the repair flows. The receivers should also know the exact association of the source and repair flows so that they can use the correct data to repair the original content in case there is a packet loss. Currently, SDP [RFC4566] uses [RFC3388] and [RFC4756] for this purpose.

In order to provide applications more flexibility, the FEC Framework [I-D.ietf-fecframe-framework] allows a source flow to be protected by multiple FEC schemes, each of which requires an instance of the FEC Framework. Thus, multiple instances of the FEC Framework MAY exist at the sender and the receiver(s). Furthermore, within a single FEC Framework instance, multiple source flows MAY be grouped and protected by one or more repair flows.

It should be noted that the FEC Framework requires the source and

repair packets to be carried in different streams. When Real-time Transport Protocol (RTP) [RFC3550] is used to carry the source and repair streams, the FEC Framework recommends that each stream is carried in its own RTP session. This provides flexibility in using FEC in a backward-compatible manner. However, in some scenarios, a single RTP session may be desired to carry multiple RTP streams via SSRC multiplexing in order to reduce the port usage. For such scenarios, an appropriate grouping semantics is also required.

A basic example scenario is shown in Figure 1. Here, source flow S1 is protected by repair flow R1. Also, source flows S1 and S2 are grouped and protected together by repair flow R2.

```
            SOURCE FLOWS             | FEC FRAMEWORK INSTANCE #1
          | S1: Source Flow |--------| R1: Repair Flow
      +---|
      |   | S2: Source Flow
      |
      +______________________________| FEC FRAMEWORK INSTANCE #2
                                     | R2: Repair Flow
```

Figure 1: Example scenario with two FEC Framework instances where R1 protects S1, and R2 protects the group of S1 and S2

Grouping source flows before applying FEC protection may allow us to achieve a better coding performance. As a typical scenario, suppose that source flows S1 and S2 in Figure 1 correspond to the base and enhancement layers in a layered video content, respectively. Repair flow R2 protects the combination of the base and enhancement layers for the receivers who receive both layers, and repair flow R1 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 any other limitation.

It should be noted that the grouping semantics defined in this document offers flexibility about which source streams can be grouped together prior to FEC protection. However, not all FEC schemes support the full range of the possible scenarios (e.g., when the source streams carry different top-level media types such as audio and video).

Using multiple FEC Framework instances for a single source flow provides flexibility to the receivers. An example scenario is sketched in Figure 2. 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 may experience different packet loss rates and each receiver can choose the repair flow that is tailored to its needs.

```
        SOURCE FLOWS              | FEC FRAMEWORK INSTANCE #1
        S3: Source Flow |---------| R3: Repair Flow
                        |
                        |---------| FEC FRAMEWORK INSTANCE #2
                                  | R4: Repair Flow
```

Figure 2: Example scenario with two FEC Framework instances, each with a single repair flow protecting the same source flow S3

To summarize, the FEC Framework supports the following:

1. A source flow MAY be protected by multiple different FEC schemes.

2. An FEC scheme MAY generate multiple repair flows.

3. Source flows MAY be grouped prior to FEC protection. That is, one or more repair flows MAY protect a group of source flows.

To fully benefit from the flexibility provided by the FEC Framework, the grouping semantics for FEC MUST support these features.

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. Requirements and Issues with RFC 4756

3.1. Source and Repair Flow Association

Currently, the 'group' attribute and the FEC grouping semantics defined in [RFC3388] and [RFC4756], respectively, are used to associate source and repair flows together.

The 'group' attribute is used to group multiple repair flows with one or more source flows. However, [RFC3388] prohibits an "m" line identified by its 'mid' attribute from appearing in more than one "a=group" line using the same semantics. This limitation prevents us

from indicating specific associations between the source and repair flows by using an "a=group:FEC" line per FEC Framework instance. For example, for the scenario sketched in Figure 1, [RFC3388] mandates us to write

```
     a=group:FEC S1 S2 R1 R2
```

Clearly, this "a=group:FEC" line does not say anything specific about which repair flows are protecting which source flows.

A new work ([I-D.ietf-mmusic-rfc3388bis]) is currently in progress in the MMUSIC WG to remove this limitation in [RFC3388]. However, [RFC4756] also needs to be updated according to the FEC Framework requirements.

3.2. Support for Additivity

The FEC Framework also supports additive repair flows. Additivity among the repair flows means that multiple repair flows may be decoded jointly to improve the recovery chances of the missing packets in a single or the same set of source flows. Additive repair flows can be generated by the same FEC scheme or different FEC schemes.

For example, in Figure 3, repair flows R5 and R6 may be additive within the FEC Framework instance #1. Alternatively, all three repair flows R5, R6 and R7 could be additive, too.

```
        SOURCE FLOWS              | FEC FRAMEWORK INSTANCE #1
        S4: Source Flow |---------| R5: Repair Flow
                        |         | R6: Repair Flow
                        |
                        |---------| FEC FRAMEWORK INSTANCE #2
                                  | R7: Repair Flow
```

Figure 3: Example scenario with two FEC Framework instances, where two repair flows in the first instance and a single repair flow in the second instance protect the same source flow S4

4. FEC Grouping

4.1. New Grouping Semantics

Each "a=group" line is used to indicate an association relationship between the source and repair flows. The flows included in one "a=group" line are called an FEC Group. If there are more than one repair flows included in an FEC group, they are considered to be

additive. Repair flows that are in different FEC groups are non-additive.

By extending [I-D.ietf-mmusic-rfc3388bis] we define "FEC-XR" as the new grouping semantics that can support the features of the FEC Framework.

The "a=group:FEC-XR" semantics MUST always be used to associate the source and repair flows except when the source and repair flows are specified in the same media description, i.e., in the same "m" line.

4.2. SDP Example

For the scenario sketched in Figure 1, we can write the following SDP:

```
v=0
o=ali 1122334455 1122334466 IN IP4 fec.example.com
s=New FEC Grouping Semantics
t=0 0
a=group:FEC-XR S1 R1
a=group:FEC-XR S1 S2 R2
m=video 30000 RTP/AVP 100
c=IN IP4 233.252.0.1/127
a=rtpmap:100 MP2T/90000
a=mid:S1
m=video 30000 RTP/AVP 101
c=IN IP4 233.252.0.2/127
a=rtpmap:101 MP2T/90000
a=mid:S2
m=application 30000 RTP/AVP 110
c=IN IP4 233.252.0.3/127
a=rtpmap:110 1d-interleaved-parityfec/90000
a=fmtp:110 L=5; D=10; repair-window=200000
a=mid:R1
m=application 30000 RTP/AVP 111
c=IN IP4 233.252.0.4/127
a=rtpmap:111 1d-interleaved-parityfec/90000
a=fmtp:111 L=10; D=10; repair-window=400000
a=mid:R2
```

In this example, the source and repair flows are carried in their own RTP sessions and the grouping is achieved through the "a=group: FEC-XR" lines.

For the additivity issues, let us consider the scenario sketched in Figure 3. Suppose that repair flows R5 and R6 are additive but repair flow R7 is not additive with any of the other repair flows.

In this case, we MUST write

```
a=group:FEC-XR S4 R5 R6
a=group:FEC-XR S4 R7
```

If none of the repair flows are additive, we MUST write

```
a=group:FEC-XR S4 R5
a=group:FEC-XR S4 R6
a=group:FEC-XR S4 R7
```

Note that additivity is not necessarily a transitive relation. Thus, each set of additive repair flows MUST be stated explicitly.

4.3. Grouping for SSRC-Multiplexed RTP Streams

[RFC5576] defines a grouping attribute, called 'ssrc-group', for the RTP streams that are SSRC multiplexed and carried in the same RTP session. The grouping is based on the Synchronization Source (SSRC) identifiers. Since SSRC-multiplexed RTP streams are defined in the same "m" line, the 'group' attribute cannot be used. Instead, the 'ssrc-group' attribute MUST be used.

Per [RFC3550], the SSRC identifiers for the RTP streams that are carried in the same RTP session MUST be unique. However, the SSRC identifiers are not guaranteed to be unique among different RTP sessions. Thus, the 'ssrc-group' attribute MUST only be used at the media level [RFC5576]. The semantics of "FEC-XR" for the 'ssrc-group' attribute is exactly the same as the one defined for the 'group' attribute.

Let us consider the following scenario where there are two source flows (e.g., one video and one audio) and a single repair flow that protects only one of the source flows (e.g., video). Suppose that all these flows are separate RTP streams that are SSRC multiplexed in the same RTP session.

```
SOURCE FLOWS             | FEC FRAMEWORK INSTANCE #1
S5: Source Flow |--------| R8: Repair Flow
S6: Source Flow
```

Figure 4: Example scenario with one FEC Framework instance, where a single repair flow protects only one of the source flows

The following SDP describes the scenario sketched in Figure 4.

```
v=0
o=ali 1122334455 1122334466 IN IP4 fec.example.com
s=New FEC Grouping Semantics for SSRC Multiplexing
t=0 0
m=video 30000 RTP/AVP 100 101 110
c=IN IP4 233.252.0.1/127
a=rtpmap:100 JPEG/90000
a=rtpmap:101 L16/32000/2
a=rtpmap:110 1d-interleaved-parityfec/90000
a=fmtp:110 L=5; D=10; repair-window=200000
a=ssrc:1000 cname:fec@example.com
a=ssrc:1010 cname:fec@example.com
a=ssrc:2110 cname:fec@example.com
a=ssrc-group:FEC-XR 1000 2110
a=mid:Group1
```

Note that in actual use, SSRC values, which are random 32-bit numbers, may be much larger than the ones shown in this example. Also note that before receiving an RTP packet for each stream, the receiver cannot know which SSRC identifier is associated with which payload type.

The additivity of the repair flows is handled in the same way as described in Section 4.2. In other words, the repair flows that are included in an "a=ssrc-group" line are additive. Repair flows that are in different "a=ssrc-group" lines are non-additive.

4.4. Offer-Answer Model Considerations

When offering FEC grouping using SDP in an Offer/Answer model [RFC3264], the following considerations apply.

A node that is receiving an offer from a sender may or may not understand line grouping. It is also possible that the node understands line grouping but it does not understand the "FEC-XR" semantics. From the viewpoint of the sender of the offer, these cases are indistinguishable.

When a node is offered a session with the "FEC-XR" grouping semantics but it does not support line grouping or the FEC grouping semantics, the node SHOULD respond to the offer either:

o With an answer that ignores the grouping attribute.

  In this case, the original sender of the offer MUST first check whether using the FEC grouping semantics of [RFC4756] will create any ambiguity or not, while keeping in mind the limitations explained in Section 3.1. If using the "FEC" semantics rather

than the "FEC-XR" semantics still provides an exact association among the source and repair flows, the sender of the offer MUST send a new offer using the "FEC" semantics. However, if an exact association cannot be described, the sender MUST send a new offer without FEC.

o With a refusal to the request (e.g., 488 Not Acceptable Here or 606 Not Acceptable in SIP).

In this case, if the sender of the offer still wishes to establish the session, it MUST first check whether using the FEC grouping semantics of [RFC4756] will create any ambiguity or not, while keeping in mind the limitations explained in Section 3.1. If using the "FEC" semantics rather than the "FEC-XR" semantics still provides an exact association among the source and repair flows, the sender of the offer SHOULD send a new offer using the "FEC" semantics. However, if an exact association cannot be described, the sender SHOULD send a new offer without FEC.

Note that in both cases described above, when the sender of the offer sends a new offer with the "FEC" semantics, and the node understands it, the session will be established and the rules pertaining to [RFC4756] will be valid.

However, if the node does not understand the "FEC" semantics, it SHOULD respond to the offer either (1) with an answer that ignores the grouping attribute, or (2) with a refusal to the request. In the first case, the sender MUST send a new offer without FEC. In the second case, if the sender of the offer still wishes to establish the session, it SHOULD retry the request with an offer without FEC.

5. Security Considerations

There is a weak threat for the receiver that the FEC grouping can be modified to indicate FEC relationships that do not exist. Such attacks may result in failure of FEC to protect, and/or mishandling of other media payload streams. It is RECOMMENDED that the receiver SHOULD do integrity check on SDP and follow the security considerations of SDP [RFC4566] to only trust SDP from trusted sources.

6. IANA Considerations

This document registers the following semantics with IANA in Semantics for the 'group' SDP Attribute under SDP Parameters:

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

Semantics	Token	Reference
Forward Error Correction XR	FEC-XR	[RFCXXXX]

This document also registers the following semantics with IANA in Semantics for the 'ssrc-group' SDP Attribute under SDP Parameters:

Semantics	Token	Reference
Forward Error Correction XR	FEC-XR	[RFCXXXX]

7. Acknowledgments

Some parts of this document are based on [RFC4756]. Thus, the author would like to thank those who contributed to [RFC4756]. Also, thanks to Jonathan Lennox who has contributed to Section 4.3.

8. References

8.1. Normative References

[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.

[I-D.ietf-mmusic-rfc3388bis]
Camarillo, G., "The SDP (Session Description Protocol) Grouping Framework", draft-ietf-mmusic-rfc3388bis-03 (work in progress), July 2009.

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

[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.

8.2. Informative References

[I-D.ietf-fecframe-framework]
Watson, M., "Forward Error Correction (FEC) Framework", draft-ietf-fecframe-framework-05 (work in progress), July 2009.

[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.

Author's Address

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

Email: abegen@cisco.com