Abstract

This document describes the RTP payload format for the Mixed Excitation Linear Prediction Enhanced (MELPe) speech coder algorithm developed by Atlanta Signal Processing (ASPI), Texas Instruments (TI), SignalCom (now Microsoft) and Thales Communications with noise preprocessor contributions from AT&T under contract with NSA/DOD as international NATO Standard STANAG 4591. All three different speech encoding rates and sample frames sizes are included. Comfort noise procedures and packet loss concealment are detailed. Also, within the document there are included necessary details for the use of MELP with SDP.

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/1id-abstracts.html

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

Copyright and License Notice
# Table of Contents

1  Introduction ............................................. 3  
1.1  Terminology ........................................... 3  
2  Background ............................................... 3  
3  RTP Payload Format ....................................... 4  
   3.1  MELPe Bitstream Definition ............................. 5  
      3.1.1  2400 bps Bitstream Structure ...................... 5  
      3.1.2  1200 bps Bitstream Structure ...................... 8  
      3.1.3  600 bps Bitstream Structure ...................... 11  
   3.2  MELPe Comfort Noise Bitstream Definition .......... 14  
   3.3  Multiple MELPe frames in a RTP packet ............ 16  
4  MAPPING TO SDP PARAMETERS ............................ 18  
5  DISCONTINUOUS TRANSMISSION ............................ 20  
6  PACKET LOSS CONCEALMENT .............................. 20  
7  Security Considerations ................................ 21  
8  IANA Considerations ..................................... 21  
9  References ................................................. 21  
   9.1  Normative References .............................. 21  
   9.2  Informative References ............................ 22  
Authors’ Addresses .......................................... 22
1 Introduction

This document describes how compressed MELPe speech as produced by the MELPe codec may be formatted for use as an RTP payload. Details are provided to packetize the three different codec data frames (2400, 1200, and 600) into RTP packets. The sender may send one or more codec data frames per packet, depending on the application scenario or based on the transport network condition, bandwidth restriction, delay requirements and packet-loss tolerance.

1.1 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2 Background

The MELP speech coder was developed by the US military as an upgrade from LPC-based CELP standard vocoder for low bit-rate communications [MELP]. MELP was further enhanced and subsequently adopted by NATO as MELPe for use by its members and Partnership for Peace countries for military and other governmental communications [MELPE]. Commercial/civilian applications have arisen because of the low bit-rate property of MELPe with its (relatively) high intelligibility. As such MELPe is being used in a variety of wired and radio communications systems. VoIP/SIP systems need to transport MELPe without decoding and re-encoding in order to preserve its intelligibility. Hence it is desirable and necessary to define the proper payload formatting and use conventions of MELPe in RTP payloads.

The MELPe codec [MELPE] supports three different vocoder rates; 2400, 1200, and 600 bps. The basic 2400 bps rate vocoder uses a 22.5 ms frame of speech consisting of 180 8000 Hz, 16-bit speech samples. The 1200 and 600 bps rate vocoders uses respectively three and four 22.5 ms frames of speech each. These reduced rate vocoders internally use multiple 2400 bps parameter sets with further processing to strategically remove redundancy. The payload sizes for each of the rates are 54, 81, and 54 bits respectively for the 2400, 1200, and 600 bps frames. Dynamic rate switching is permitted but only if supported by both endpoints.

The MELPe algorithm distinguishes between voiced and un-voiced speech and encodes each differently. Unvoiced speech can be coded with
fewer information bits for the same quality. Forward error correction (FEC) is applied to the 2400 bps codec unvoiced speech for better protection of the subtle differences in signal reconstruction. The lower bit rate coders do not allocate any bits for FEC and rely on strong error protection and correction in the communications channel.

Comfort noise handling for MELPe is recommended to follow SCIP-210 Appendix B [SCIP210]. After VAD no longer indicates the presence of speech/voice, a grace period of a minimum of two comfort noise vocoder frames are to be transmitted. The contents of the comfort noise frames is described in the next section.

Packet loss concealment (PLC) exploits the FEC (and more precisely, double bits errors of the pitch/voicing parameter) of the 2400 bps speech coder. The pitch/voicing parameter has a sparse set of permitted values. A value of zero indicates a non-voiced frame. At least three bits are set for all valid pitch parameters. The PLC erasure indication utilizes any of the errored encodings of a non-voiced frame as will be described infra.

3 RTP Payload Format

The MELPe codec uses 22.5, 67.5 or 90 ms frames with a sampling rate clock of 8 kHz, so the RTP timestamp MUST be in units of 1/8000 of a second.

The RTP payload for MELPe has the format shown in the figure below. No addition header specific to this payload format is required.

This format is intended for the situations where the sender and the receiver send one or more codec data frames per packet. The RTP packet looks as follows:

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        RTP Header                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| one or more frames of MELPe |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1 - Packet format diagram
The RTP header of the packetized encoded MELPe speech has the expected values as described in [RFC3550]. The usage of M bit SHOULD be as specified in the applicable RTP profile, for example, RFC 3551 [RFC3551], where [RFC3551] specifies that if the sender does not suppress silence (i.e., sends a frame on every frame interval), the M bit will always be zero. When more then one codec data frame is present in a single RTP packet, the timestamp is, as always, that of the oldest data frame represented in the RTP packet.

The assignment of an RTP payload type for this new packet format is outside the scope of this document, and will not be specified here. It is expected that the RTP profile for a particular class of applications will assign a payload type for this encoding, or if that is not done, then a payload type in the dynamic range shall be chosen by the sender.

3.1 MELPe Bitstream Definition

The total number of bits used to describe one frame of 2400 bps speech is 54, which fits in 7 octets (with two unused bits). For the 1200 bps speech the total number of bits used is 81, which fits in 11 octets (with seven unused bits). For the 600 bps speech the total number of bits used is 54, which fits in 7 octets (with two unused bits). Unused bits are coded as described in 3.3 in support of dynamic rate switching.

In the MELPe bitstream definition, the most significant bits are considered priority bits. The intention was that these bits receive greater protection in the underlying communications channel. For IP networks, such additional protection is irrelevant. However, for convenience of interoperable gateway devices, the bitstreams will be presented identically in IP networks.

3.1.1 2400 bps Bitstream Structure

According to Table 3 of [MELPE], the 2400 bit/s MELPe bit transmission order (bit priority is not shown for clarity) is the following:

+--------+-------------+-------------+
| Bit    |    Voiced   |   Unvoiced  |
+--------+-------------+-------------+
    |    |       g20   |       g20   |
    |    |       BP0   |     FEC10   |
    |    |        P0   |        P0   |

Demjanenko, Satterlee Expires June 6, 2015
| B_04 | LSF20 | LSF20 |
| B_05 | LSF30 | LSF30 |
| B_06 | g23   | g23   |
| B_07 | g24   | g24   |
| B_08 | LSF35 | LSF35 |
| B_09 | g21   | g21   |
| B_10 | g22   | g22   |
| B_11 | P4    | P4    |
| B_12 | LSF34 | LSF34 |
| B_13 | P5    | P5    |
| B_14 | P1    | P1    |
| B_15 | P2    | P2    |
| B_16 | LSF40 | LSF40 |
| B_17 | P6    | P6    |
| B_18 | LSF10 | LSF10 |
| B_19 | LSF16 | LSF16 |
| B_20 | LSF45 | LSF45 |
| B_21 | P3    | P3    |
| B_22 | LSF15 | LSF15 |
| B_23 | LSF14 | LSF14 |
| B_24 | LSF25 | LSF25 |
| B_25 | BP3   | FEC13 |
| B_26 | LSF13 | LSF13 |
| B_27 | LSF12 | LSF12 |
| B_28 | LSF24 | LSF24 |
| B_29 | LSF44 | LSF44 |
| B_30 | FM0   | FEC40 |
| B_31 | LSF11 | LSF11 |
| B_32 | LSF23 | LSF23 |
| B_33 | FM7   | FEC22 |
| B_34 | FM6   | FEC21 |
| B_35 | FM5   | FEC20 |
| B_36 | g11   | g11   |
| B_37 | g10   | g10   |
| B_38 | BP2   | FEC12 |
| B_39 | BP1   | FEC11 |
| B_40 | LSF21 | LSF21 |
| B_41 | LSF33 | LSF33 |
| B_42 | LSF22 | LSF22 |
| B_43 | LSF32 | LSF32 |
| B_44 | LSF31 | LSF31 |
| B_45 | LSF43 | LSF43 |
| B_46 | LSF42 | LSF42 |
Table 3.1 - The bitstream definition for MELPe 2400 bps.

The 2400 bps MELPe RTP payload is constructed as per Figure 2. Note that bit B_01 is placed in the LSB of the first byte with all other bits in sequence. When filling octets, the least significant bits of the seventh octet are filled with bits B_49 to B_54 respectively.

<table>
<thead>
<tr>
<th>MSB</th>
<th>LSB</th>
</tr>
</thead>
<tbody>
<tr>
<td>B_08</td>
<td>B_07</td>
</tr>
<tr>
<td>B_16</td>
<td>B_15</td>
</tr>
<tr>
<td>B_24</td>
<td>B_23</td>
</tr>
<tr>
<td>B_32</td>
<td>B_31</td>
</tr>
<tr>
<td>B_40</td>
<td>B_39</td>
</tr>
<tr>
<td>B_48</td>
<td>B_47</td>
</tr>
<tr>
<td>B_54</td>
<td>B_53</td>
</tr>
<tr>
<td>B_52</td>
<td>B_51</td>
</tr>
<tr>
<td>B_50</td>
<td>B_49</td>
</tr>
</tbody>
</table>

NOTES:
g = Gain
BP = Bandpass Voicing
P = Pitch/Voicing
LSF = Line Spectral Frequencies
FEC = Forward Error Correction Parity Bits
FM = Fourier Magnitudes
AF = Aperiodic Flag

B_01 = least significant bit of data set
3.1.2 1200 bps Bitstream Structure

According to Tables D9a and D9b of [MELPE], the 1200 bit/s MELPe bit transmission order is the following:

<table>
<thead>
<tr>
<th>Bit</th>
<th>Modes 1-4 (Voiced)</th>
<th>Mode 5 (Unvoiced)</th>
</tr>
</thead>
<tbody>
<tr>
<td>B_01</td>
<td>Syn</td>
<td>Syn</td>
</tr>
<tr>
<td>B_02</td>
<td>Pitch&amp;UV0</td>
<td>Pitch&amp;UV0</td>
</tr>
<tr>
<td>B_03</td>
<td>Pitch&amp;UV1</td>
<td>Pitch&amp;UV1</td>
</tr>
<tr>
<td>B_04</td>
<td>Pitch&amp;UV2</td>
<td>Pitch&amp;UV2</td>
</tr>
<tr>
<td>B_05</td>
<td>Pitch&amp;UV3</td>
<td>Pitch&amp;UV3</td>
</tr>
<tr>
<td>B_06</td>
<td>Pitch&amp;UV4</td>
<td>Pitch&amp;UV4</td>
</tr>
<tr>
<td>B_07</td>
<td>Pitch&amp;UV5</td>
<td>Pitch&amp;UV5</td>
</tr>
<tr>
<td>B_08</td>
<td>Pitch&amp;UV6</td>
<td>Pitch&amp;UV6</td>
</tr>
<tr>
<td>B_09</td>
<td>Pitch&amp;UV7</td>
<td>Pitch&amp;UV7</td>
</tr>
<tr>
<td>B_10</td>
<td>Pitch&amp;UV8</td>
<td>Pitch&amp;UV8</td>
</tr>
<tr>
<td>B_11</td>
<td>Pitch&amp;UV9</td>
<td>Pitch&amp;UV9</td>
</tr>
<tr>
<td>B_12</td>
<td>Pitch&amp;UV10</td>
<td>Pitch&amp;UV10</td>
</tr>
<tr>
<td>B_13</td>
<td>Pitch&amp;UV11</td>
<td>Pitch&amp;UV11</td>
</tr>
<tr>
<td>B_14</td>
<td>LSP0</td>
<td>LSP0</td>
</tr>
<tr>
<td>B_15</td>
<td>LSP1</td>
<td>LSP1</td>
</tr>
<tr>
<td>B_16</td>
<td>LSP2</td>
<td>LSP2</td>
</tr>
<tr>
<td>B_17</td>
<td>LSP3</td>
<td>LSP3</td>
</tr>
<tr>
<td>B_18</td>
<td>LSP4</td>
<td>LSP4</td>
</tr>
<tr>
<td>B_19</td>
<td>LSP5</td>
<td>LSP5</td>
</tr>
<tr>
<td>B_20</td>
<td>LSP6</td>
<td>LSP6</td>
</tr>
<tr>
<td>B_21</td>
<td>LSP7</td>
<td>LSP7</td>
</tr>
<tr>
<td>B_22</td>
<td>LSP8</td>
<td>LSP8</td>
</tr>
<tr>
<td>B_23</td>
<td>LSP9</td>
<td>LSP9</td>
</tr>
<tr>
<td>B_24</td>
<td>LSP10</td>
<td>LSP10</td>
</tr>
<tr>
<td>B_25</td>
<td>LSP11</td>
<td>LSP11</td>
</tr>
<tr>
<td>B_26</td>
<td>LSP12</td>
<td>LSP12</td>
</tr>
<tr>
<td>B_27</td>
<td>LSP13</td>
<td>LSP13</td>
</tr>
<tr>
<td>B_28</td>
<td>LSP14</td>
<td>LSP14</td>
</tr>
<tr>
<td>B_29</td>
<td>LSP15</td>
<td>LSP15</td>
</tr>
<tr>
<td>B_30</td>
<td>LSP16</td>
<td>LSP16</td>
</tr>
<tr>
<td>B_31</td>
<td>LSP17</td>
<td>LSP17</td>
</tr>
<tr>
<td>B_32</td>
<td>LSP18</td>
<td>LSP18</td>
</tr>
<tr>
<td>B_33</td>
<td>LSP19</td>
<td>LSP19</td>
</tr>
<tr>
<td>B_34</td>
<td>LSP20</td>
<td>LSP20</td>
</tr>
<tr>
<td>B_35</td>
<td>LSP21</td>
<td>LSP21</td>
</tr>
<tr>
<td>B_36</td>
<td>LSP22</td>
<td>LSP22</td>
</tr>
<tr>
<td>B_37</td>
<td>LSP23</td>
<td>LSP23</td>
</tr>
<tr>
<td>B_38</td>
<td>LSP24</td>
<td>LSP24</td>
</tr>
<tr>
<td>B_39</td>
<td>LSP25</td>
<td>LSP25</td>
</tr>
<tr>
<td>B_40</td>
<td>LSP26</td>
<td>LSP26</td>
</tr>
<tr>
<td>B_41</td>
<td>LSP27</td>
<td>GAIN0</td>
</tr>
<tr>
<td>B_42</td>
<td>LSP28</td>
<td>GAIN1</td>
</tr>
<tr>
<td>B_43</td>
<td>LSP29</td>
<td>GAIN2</td>
</tr>
<tr>
<td>B_44</td>
<td>LSP30</td>
<td>GAIN3</td>
</tr>
<tr>
<td>B_45</td>
<td>LSP31</td>
<td>GAIN4</td>
</tr>
<tr>
<td>B_46</td>
<td>LSP32</td>
<td>GAIN5</td>
</tr>
<tr>
<td>B_47</td>
<td>LSP33</td>
<td>GAIN6</td>
</tr>
<tr>
<td>B_48</td>
<td>LSP34</td>
<td>GAIN7</td>
</tr>
<tr>
<td>B_49</td>
<td>LSP35</td>
<td>GAIN8</td>
</tr>
<tr>
<td>B_50</td>
<td>LSP36</td>
<td>GAIN9</td>
</tr>
<tr>
<td>B_51</td>
<td>LSP37</td>
<td></td>
</tr>
<tr>
<td>B_52</td>
<td>LSP38</td>
<td></td>
</tr>
<tr>
<td>B_53</td>
<td>LSP39</td>
<td></td>
</tr>
<tr>
<td>B_54</td>
<td>LSP40</td>
<td></td>
</tr>
<tr>
<td>B_55</td>
<td>LSP41</td>
<td></td>
</tr>
<tr>
<td>B_56</td>
<td>LSP42</td>
<td></td>
</tr>
<tr>
<td>B_57</td>
<td>GAIN0</td>
<td></td>
</tr>
<tr>
<td>B_58</td>
<td>GAIN1</td>
<td></td>
</tr>
<tr>
<td>B_59</td>
<td>GAIN2</td>
<td></td>
</tr>
<tr>
<td>B_60</td>
<td>GAIN3</td>
<td></td>
</tr>
<tr>
<td>B_61</td>
<td>GAIN4</td>
<td></td>
</tr>
<tr>
<td>B_62</td>
<td>GAIN5</td>
<td></td>
</tr>
<tr>
<td>B_63</td>
<td>GAIN6</td>
<td></td>
</tr>
<tr>
<td>B_64</td>
<td>GAIN7</td>
<td></td>
</tr>
<tr>
<td>B_65</td>
<td>GAIN8</td>
<td></td>
</tr>
<tr>
<td>B_66</td>
<td>GAIN9</td>
<td></td>
</tr>
<tr>
<td>B_67</td>
<td>BP0</td>
<td></td>
</tr>
<tr>
<td>B_68</td>
<td>BP1</td>
<td></td>
</tr>
<tr>
<td>B_69</td>
<td>BP2</td>
<td></td>
</tr>
<tr>
<td>B_70</td>
<td>BP3</td>
<td></td>
</tr>
<tr>
<td>B_71</td>
<td>BP4</td>
<td></td>
</tr>
<tr>
<td>B_72</td>
<td>BP5</td>
<td></td>
</tr>
<tr>
<td>B_73</td>
<td>JITTER</td>
<td></td>
</tr>
<tr>
<td>B_74</td>
<td>FS0</td>
<td></td>
</tr>
</tbody>
</table>
NOTES:
BP = Band pass voicing
FS = Fourier magnitudes

Table 3.2 - The bitstream definition for MELPe 1200 bps.

The 1200 bps MELPe RTP payload is constructed as per Figure 3. Note that bit B_01 is placed in the LSB of the first byte with all other bits in sequence. When filling octets, the least significant bit of the eleventh octet is filled with bit B_81.
3.1.3 600 bps Bitstream Structure

According to Tables M-11 to M-16 of [MELPE], the 600 bit/s MELPe bit transmission order (bit priority is not shown for clarity) is the following:

<table>
<thead>
<tr>
<th>Bit</th>
<th>Mode 1 (Voiced)</th>
<th>Mode 2 (voiced)</th>
<th>Mode 3 (voiced)</th>
</tr>
</thead>
<tbody>
<tr>
<td>B_01</td>
<td>Voicing (4)</td>
<td>Voicing (4)</td>
<td>Voicing (4)</td>
</tr>
<tr>
<td>B_02</td>
<td>Voicing (3)</td>
<td>Voicing (3)</td>
<td>Voicing (3)</td>
</tr>
<tr>
<td>B_03</td>
<td>Voicing (2)</td>
<td>Voicing (2)</td>
<td>Voicing (2)</td>
</tr>
<tr>
<td>B_04</td>
<td>Voicing (1)</td>
<td>Voicing (1)</td>
<td>Voicing (1)</td>
</tr>
<tr>
<td>B_05</td>
<td>Voicing (0)</td>
<td>Voicing (0)</td>
<td>Voicing (0)</td>
</tr>
<tr>
<td>B_06</td>
<td>LSF1,4 (3)</td>
<td>Pitch (5)</td>
<td>Pitch (7)</td>
</tr>
<tr>
<td>B_07</td>
<td>LSF1,4 (2)</td>
<td>Pitch (4)</td>
<td>Pitch (6)</td>
</tr>
<tr>
<td>B_08</td>
<td>LSF1,4 (1)</td>
<td>Pitch (3)</td>
<td>Pitch (5)</td>
</tr>
<tr>
<td>B_09</td>
<td>LSF1,4 (0)</td>
<td>Pitch (2)</td>
<td>Pitch (4)</td>
</tr>
<tr>
<td>B_10</td>
<td>LSF1,3 (3)</td>
<td>Pitch (1)</td>
<td>Pitch (3)</td>
</tr>
<tr>
<td>B_11</td>
<td>LSF1,3 (2)</td>
<td>Pitch (0)</td>
<td>Pitch (2)</td>
</tr>
<tr>
<td>B_12</td>
<td>LSF1,3 (1)</td>
<td>LSF1,3 (3)</td>
<td>Pitch (1)</td>
</tr>
<tr>
<td>B_13</td>
<td>LSF1,3 (0)</td>
<td>LSF1,3 (2)</td>
<td>Pitch (0)</td>
</tr>
<tr>
<td>B_14</td>
<td>LSF1,2 (3)</td>
<td>LSF1,3 (1)</td>
<td>LSF1,3 (3)</td>
</tr>
<tr>
<td>B_15</td>
<td>LSF1,2 (2)</td>
<td>LSF1,3 (0)</td>
<td>LSF1,3 (2)</td>
</tr>
<tr>
<td>B_16</td>
<td>LSF1,2 (1)</td>
<td>LSF1,2 (3)</td>
<td>LSF1,3 (1)</td>
</tr>
<tr>
<td>B_17</td>
<td>LSF1,2 (0)</td>
<td>LSF1,2 (2)</td>
<td>LSF1,3 (0)</td>
</tr>
<tr>
<td>B_18</td>
<td>LSF1,1 (5)</td>
<td>LSF1,2 (1)</td>
<td>LSF1,2 (4)</td>
</tr>
<tr>
<td>B_19</td>
<td>LSF1,1 (4)</td>
<td>LSF1,2 (0)</td>
<td>LSF1,2 (3)</td>
</tr>
<tr>
<td>B_20</td>
<td>LSF1,1 (3)</td>
<td>LSF1,1 (5)</td>
<td>LSF1,2 (2)</td>
</tr>
<tr>
<td>B_21</td>
<td>LSF1,1 (2)</td>
<td>LSF1,1 (4)</td>
<td>LSF1,2 (1)</td>
</tr>
<tr>
<td>B_22</td>
<td>LSF1,1 (1)</td>
<td>LSF1,1 (3)</td>
<td>LSF1,2 (0)</td>
</tr>
<tr>
<td>B_23</td>
<td>LSF1,1 (0)</td>
<td>LSF1,1 (2)</td>
<td>LSF1,1 (5)</td>
</tr>
<tr>
<td>B_24</td>
<td>LSF2,4 (3)</td>
<td>LSF1,1 (1)</td>
<td>LSF1,1 (4)</td>
</tr>
<tr>
<td>B_25</td>
<td>LSF2,4 (2)</td>
<td>LSF1,1 (0)</td>
<td>LSF1,1 (3)</td>
</tr>
<tr>
<td>B_26</td>
<td>LSF2,4 (1)</td>
<td>LSF2,3 (3)</td>
<td>LSF1,1 (2)</td>
</tr>
<tr>
<td>B_27</td>
<td>LSF2,4 (0)</td>
<td>LSF2,3 (2)</td>
<td>LSF1,1 (1)</td>
</tr>
<tr>
<td>B_28</td>
<td>LSF2,3 (3)</td>
<td>LSF2,3 (1)</td>
<td>LSF1,1 (0)</td>
</tr>
<tr>
<td>B_29</td>
<td>LSF2,3 (2)</td>
<td>LSF2,3 (0)</td>
<td>LSF2,3 (3)</td>
</tr>
<tr>
<td>B_30</td>
<td>LSF2,3 (1)</td>
<td>LSF2,2 (4)</td>
<td>LSF2,3 (2)</td>
</tr>
<tr>
<td>B_31</td>
<td>LSF2,3 (0)</td>
<td>LSF2,2 (3)</td>
<td>LSF2,3 (1)</td>
</tr>
</tbody>
</table>
### Table 3.3a - The bitstream definition for MELPe 600 bps (part 1 of 2).

<table>
<thead>
<tr>
<th>Bit</th>
<th>Mode 4 (voiced)</th>
<th>Mode 5 (voiced)</th>
<th>Mode 6 (voiced)</th>
</tr>
</thead>
<tbody>
<tr>
<td>B_00</td>
<td>Voicing (4)</td>
<td>Voicing (4)</td>
<td>Voicing (4)</td>
</tr>
<tr>
<td>B_01</td>
<td>Voicing (3)</td>
<td>Voicing (3)</td>
<td>Voicing (3)</td>
</tr>
<tr>
<td>B_02</td>
<td>Voicing (2)</td>
<td>Voicing (2)</td>
<td>Voicing (2)</td>
</tr>
<tr>
<td>B_03</td>
<td>Voicing (1)</td>
<td>Voicing (1)</td>
<td>Voicing (1)</td>
</tr>
<tr>
<td>B_04</td>
<td>Voicing (0)</td>
<td>Voicing (0)</td>
<td>Voicing (0)</td>
</tr>
<tr>
<td>B_05</td>
<td>Pitch (7)</td>
<td>Pitch (7)</td>
<td>Pitch (7)</td>
</tr>
<tr>
<td>B_06</td>
<td>Pitch (6)</td>
<td>Pitch (6)</td>
<td>Pitch (6)</td>
</tr>
<tr>
<td>B_07</td>
<td>Pitch (5)</td>
<td>Pitch (5)</td>
<td>Pitch (5)</td>
</tr>
<tr>
<td>B_08</td>
<td>Pitch (4)</td>
<td>Pitch (4)</td>
<td>Pitch (4)</td>
</tr>
<tr>
<td>B_09</td>
<td>Pitch (3)</td>
<td>Pitch (3)</td>
<td>Pitch (3)</td>
</tr>
<tr>
<td>B_10</td>
<td>Pitch (2)</td>
<td>Pitch (2)</td>
<td>Pitch (2)</td>
</tr>
<tr>
<td>B_11</td>
<td>Pitch (1)</td>
<td>Pitch (1)</td>
<td>Pitch (1)</td>
</tr>
<tr>
<td>B_13</td>
<td>Pitch (0)</td>
<td>Pitch (0)</td>
<td>Pitch (0)</td>
</tr>
<tr>
<td>------</td>
<td>-----------</td>
<td>-----------</td>
<td>-----------</td>
</tr>
<tr>
<td>B_14</td>
<td>LSF1,3 (3)</td>
<td>LSF1,3 (3)</td>
<td>LSF1,3 (3)</td>
</tr>
<tr>
<td>B_15</td>
<td>LSF1,3 (2)</td>
<td>LSF1,3 (2)</td>
<td>LSF1,3 (2)</td>
</tr>
<tr>
<td>B_16</td>
<td>LSF1,3 (1)</td>
<td>LSF1,3 (1)</td>
<td>LSF1,3 (1)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B_17</th>
<th>LSF1,3 (0)</th>
<th>LSF1,3 (0)</th>
<th>LSF1,3 (0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>B_18</td>
<td>LSF1,2 (3)</td>
<td>LSF1,2 (4)</td>
<td>LSF1,2 (4)</td>
</tr>
<tr>
<td>B_19</td>
<td>LSF1,2 (2)</td>
<td>LSF1,2 (3)</td>
<td>LSF1,2 (3)</td>
</tr>
<tr>
<td>B_20</td>
<td>LSF1,2 (1)</td>
<td>LSF1,2 (2)</td>
<td>LSF1,2 (2)</td>
</tr>
<tr>
<td>B_21</td>
<td>LSF1,2 (0)</td>
<td>LSF1,2 (1)</td>
<td>LSF1,2 (1)</td>
</tr>
<tr>
<td>B_22</td>
<td>LSF1,1 (5)</td>
<td>LSF1,2 (0)</td>
<td>LSF1,2 (0)</td>
</tr>
<tr>
<td>B_23</td>
<td>LSF1,1 (4)</td>
<td>LSF1,1 (5)</td>
<td>LSF1,1 (6)</td>
</tr>
<tr>
<td>B_24</td>
<td>LSF1,1 (3)</td>
<td>LSF1,1 (4)</td>
<td>LSF1,1 (5)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B_25</th>
<th>LSF1,1 (2)</th>
<th>LSF1,1 (3)</th>
<th>LSF1,1 (4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>B_26</td>
<td>LSF1,1 (1)</td>
<td>LSF1,1 (2)</td>
<td>LSF1,1 (3)</td>
</tr>
<tr>
<td>B_27</td>
<td>LSF1,1 (0)</td>
<td>LSF1,1 (1)</td>
<td>LSF1,1 (2)</td>
</tr>
<tr>
<td>B_28</td>
<td>LSF2,3 (3)</td>
<td>LSF1,1 (0)</td>
<td>LSF1,1 (1)</td>
</tr>
<tr>
<td>B_29</td>
<td>LSF2,3 (2)</td>
<td>LSF2,3 (3)</td>
<td>LSF2,3 (3)</td>
</tr>
<tr>
<td>B_30</td>
<td>LSF2,3 (1)</td>
<td>LSF2,3 (2)</td>
<td>LSF2,3 (3)</td>
</tr>
<tr>
<td>B_31</td>
<td>LSF2,3 (0)</td>
<td>LSF2,3 (1)</td>
<td>LSF2,3 (2)</td>
</tr>
<tr>
<td>B_32</td>
<td>LSF2,2 (4)</td>
<td>LSF2,3 (0)</td>
<td>LSF2,3 (1)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B_33</th>
<th>LSF2,2 (3)</th>
<th>LSF2,2 (4)</th>
<th>LSF2,3 (0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>B_34</td>
<td>LSF2,2 (2)</td>
<td>LSF2,2 (3)</td>
<td>LSF2,2 (4)</td>
</tr>
<tr>
<td>B_35</td>
<td>LSF2,2 (1)</td>
<td>LSF2,2 (2)</td>
<td>LSF2,2 (3)</td>
</tr>
<tr>
<td>B_36</td>
<td>LSF2,2 (0)</td>
<td>LSF2,2 (1)</td>
<td>LSF2,2 (2)</td>
</tr>
<tr>
<td>B_37</td>
<td>LSF2,1 (6)</td>
<td>LSF2,2 (0)</td>
<td>LSF2,2 (1)</td>
</tr>
<tr>
<td>B_38</td>
<td>LSF2,1 (5)</td>
<td>LSF2,1 (5)</td>
<td>LSF2,2 (0)</td>
</tr>
<tr>
<td>B_39</td>
<td>LSF2,1 (4)</td>
<td>LSF2,1 (4)</td>
<td>LSF2,1 (6)</td>
</tr>
<tr>
<td>B_40</td>
<td>LSF2,1 (3)</td>
<td>LSF2,1 (3)</td>
<td>LSF2,1 (5)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B_41</th>
<th>LSF2,1 (2)</th>
<th>LSF2,1 (2)</th>
<th>LSF2,1 (4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>B_42</td>
<td>LSF2,1 (1)</td>
<td>LSF2,1 (1)</td>
<td>LSF2,1 (3)</td>
</tr>
<tr>
<td>B_43</td>
<td>LSF2,1 (0)</td>
<td>LSF2,1 (0)</td>
<td>LSF2,1 (2)</td>
</tr>
<tr>
<td>B_44</td>
<td>GAIN2 (4)</td>
<td>GAIN2 (4)</td>
<td>LSF2,1 (1)</td>
</tr>
<tr>
<td>B_45</td>
<td>GAIN2 (3)</td>
<td>GAIN2 (3)</td>
<td>LSF2,1 (0)</td>
</tr>
<tr>
<td>B_46</td>
<td>GAIN2 (2)</td>
<td>GAIN2 (2)</td>
<td>GAIN1 (8)</td>
</tr>
<tr>
<td>B_47</td>
<td>GAIN2 (1)</td>
<td>GAIN2 (1)</td>
<td>GAIN1 (7)</td>
</tr>
<tr>
<td>B_48</td>
<td>GAIN2 (0)</td>
<td>GAIN2 (0)</td>
<td>GAIN1 (6)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B_49</th>
<th>GAIN1 (5)</th>
<th>GAIN1 (5)</th>
<th>GAIN1 (5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>B_50</td>
<td>GAIN1 (4)</td>
<td>GAIN1 (4)</td>
<td>GAIN1 (4)</td>
</tr>
<tr>
<td>B_51</td>
<td>GAIN1 (3)</td>
<td>GAIN1 (3)</td>
<td>GAIN1 (3)</td>
</tr>
<tr>
<td>B_52</td>
<td>GAIN1 (2)</td>
<td>GAIN1 (2)</td>
<td>GAIN1 (2)</td>
</tr>
<tr>
<td>B_53</td>
<td>GAIN1 (1)</td>
<td>GAIN1 (1)</td>
<td>GAIN1 (1)</td>
</tr>
<tr>
<td>B_54</td>
<td>GAIN1 (0)</td>
<td>GAIN1 (0)</td>
<td>GAIN1 (0)</td>
</tr>
</tbody>
</table>
Notes:
xxxx (0) = LSB
xxxx (nbits-1) = MSB
LSF1,p = MSVQ indice of the pth stage of the two first frames
LSF2,p = MSVQ indice of the pth stage of the two last frames
GAIN1 = VQ/MSVQ indice of the 1st stage
GAIN2 = MSVQ indice of the 2nd stage

Table 3.3b - The bitstream definition for MELPe 600 bps (part 2 of 2).

The 600 bps MELPe RTP payload is constructed as per Figure 4. Note that bit B_01 is placed in the LSB of the first byte with all other bits in sequence. When filling octets, the least significant bits of the seventh octet are filled with bits B_49 to B_54 respectively.

<table>
<thead>
<tr>
<th>MSB</th>
<th>LSB</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>B_08</td>
<td>B_07</td>
</tr>
<tr>
<td>B_06</td>
<td>B_05</td>
</tr>
<tr>
<td>B_04</td>
<td>B_03</td>
</tr>
<tr>
<td>B_02</td>
<td>B_01</td>
</tr>
<tr>
<td>B_16</td>
<td>B_15</td>
</tr>
<tr>
<td>B_14</td>
<td>B_13</td>
</tr>
<tr>
<td>B_12</td>
<td>B_11</td>
</tr>
<tr>
<td>B_10</td>
<td>B_09</td>
</tr>
<tr>
<td>B_24</td>
<td>B_23</td>
</tr>
<tr>
<td>B_22</td>
<td>B_21</td>
</tr>
<tr>
<td>B_20</td>
<td>B_19</td>
</tr>
<tr>
<td>B_18</td>
<td>B_17</td>
</tr>
<tr>
<td>B_32</td>
<td>B_31</td>
</tr>
<tr>
<td>B_30</td>
<td>B_29</td>
</tr>
<tr>
<td>B_28</td>
<td>B_27</td>
</tr>
<tr>
<td>B_26</td>
<td>B_25</td>
</tr>
<tr>
<td>B_40</td>
<td>B_39</td>
</tr>
<tr>
<td>B_38</td>
<td>B_37</td>
</tr>
<tr>
<td>B_36</td>
<td>B_35</td>
</tr>
<tr>
<td>B_34</td>
<td>B_33</td>
</tr>
<tr>
<td>B_48</td>
<td>B_47</td>
</tr>
<tr>
<td>B_46</td>
<td>B_45</td>
</tr>
<tr>
<td>B_44</td>
<td>B_43</td>
</tr>
<tr>
<td>B_42</td>
<td>B_41</td>
</tr>
<tr>
<td>RSV</td>
<td>RSV</td>
</tr>
<tr>
<td>B_54</td>
<td>B_53</td>
</tr>
<tr>
<td>B_52</td>
<td>B_51</td>
</tr>
<tr>
<td>B_50</td>
<td>B_49</td>
</tr>
</tbody>
</table>

Figure 4 - Packed MELPe 600 bps payload octets.

3.2 MELPe Comfort Noise Bitstream Definition

Table B.3-1 of [SCIP210] identifies the usage of MELPe 2400 bps parameters for conveying comfort noise.

+-------------------------------+-------------+
| MELPe Parameter | Value |
+-------------------------------+-------------+
<table>
<thead>
<tr>
<th>msvq[0] (line spectral frequencies)</th>
<th>* See Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>msvq[1] (line spectral frequencies)</td>
<td>Set to 0</td>
</tr>
<tr>
<td>msvq[2] (line spectral frequencies)</td>
<td>Set to 0</td>
</tr>
<tr>
<td>msvq[3] (line spectral frequencies)</td>
<td>Set to 0</td>
</tr>
<tr>
<td>fsvq (Fourier magnitudes)</td>
<td>Set to 0</td>
</tr>
<tr>
<td>gain[0] (gain)</td>
<td>Set to 0</td>
</tr>
<tr>
<td>gain[1] (gain)</td>
<td>* See Note</td>
</tr>
<tr>
<td>pitch (pitch - overall voicing)</td>
<td>Set to 0</td>
</tr>
<tr>
<td>bp (bandpass voicing)</td>
<td>Set to 0</td>
</tr>
<tr>
<td>af (aperiodic flag/jitter index)</td>
<td>Set to 0</td>
</tr>
<tr>
<td>sync (sync bit)</td>
<td>Alternations</td>
</tr>
</tbody>
</table>

Note: The default value shall be the respective parameters from the vocoder frame. It is recommended that msvq[0] and gain[1] values be derived by averaging the respective parameter from some number of previous vocoder frames.

Table 3.4 - MELPe Comfort Noise Parameters

Since only msvq[0] (also known as LSF1x or the first LSP) and gain[1] (also known as g2x or the second gain) are required, the following bit order is used for comfort noise frames.

<table>
<thead>
<tr>
<th>Bit</th>
<th>Comfort Noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>B_01</td>
<td>LSF10</td>
</tr>
<tr>
<td>B_02</td>
<td>LSF11</td>
</tr>
<tr>
<td>B_03</td>
<td>LSF12</td>
</tr>
<tr>
<td>B_04</td>
<td>LSF13</td>
</tr>
<tr>
<td>B_05</td>
<td>LSF14</td>
</tr>
<tr>
<td>B_06</td>
<td>LSF15</td>
</tr>
<tr>
<td>B_07</td>
<td>LSF16</td>
</tr>
<tr>
<td>B_08</td>
<td>g20</td>
</tr>
</tbody>
</table>
NOTES:
g = Gain
LSF = Line Spectral Frequencies

Table 3.5 - The bitstream definition for MELPe Comfort Noise.

The Comfort Noise MELPe RTP payload is constructed as per Figure 5. Note that bit B_01 is placed in the LSB of the first byte with all other bits in sequence. When filling octets, the least significant bits of the second octet are filled with bits B_09 to B_13 respectively.

Figure 5 - Packed MELPe Comfort Noise payload octets.

3.3 Multiple MELPe frames in a RTP packet

A MELPe RTP packet may consist of zero or more MELPe coder frames, followed by zero or one MELPe Comfort Noise frames. The presence of a comfort noise frame can be deduced from the length of the RTP payload. The default packetization interval is one coder frame (22.5, 67.5 or 90 ms) according to the coder rate (2400, 1200 or 600 bps). For some applications, a longer packetization interval may be required to reduce the packet rate.

All MELPe frames in a single RTP packet MUST be of the same coder rate. Dynamic switching between frame rates within an RTP stream may be permitted (if supported by both ends) provided that reserved bits, RSVA, RSVB, and RSVC are filled in as per Table 3.6. If rate switching is not used, all reserved bits are encoded as 0 by the

Demjanenko, Satterlee Expires June 6, 2015 [Page 16]
sender and ignored by the receiver. (RSV0 is always coded as 0).

<table>
<thead>
<tr>
<th>Coder Rate</th>
<th>RSVA</th>
<th>RSVB</th>
<th>RSVC</th>
</tr>
</thead>
<tbody>
<tr>
<td>2400 bps</td>
<td>0</td>
<td>0</td>
<td>N/A</td>
</tr>
<tr>
<td>1200 bps</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>600 bps</td>
<td>0</td>
<td>1</td>
<td>N/A</td>
</tr>
<tr>
<td>Comfort Noise</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>(reserved)</td>
<td>1</td>
<td>1</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Table 3.6 - MELPe Frame Rate Indicators.

It is important to observe that senders have the following additional restrictions:

SHOULD NOT include more MELPe frames in a single RTP packet than will fit in the MTU of the RTP transport protocol.

Frames MUST NOT be split between RTP packets.

It is RECOMMENDED that the number of frames contained within an RTP packet is consistent with the application. For example, in a telephony and other real time applications where delay is important, then the fewer frames per packet the lower the delay, whereas for a bandwidth constrained links or delay insensitive streaming messaging application, more than one or many frames per packet would be acceptable.

Information describing the number of frames contained in an RTP packet is not transmitted as part of the RTP payload. The way to determine the number of MELPe frames is to count the total number of octets within the RTP packet, and divide the octet count by the number of expected octets per frame (7/11/7 per frame). Keep in mind the last frame may be a 2 octet comfort noise frame.

When dynamic rate switching is used and more than one frame is contained in a RTP packet, it is recommended to inspect the coder rate bits contained in the last octet. If the coder rate indicates a Comfort Noise frame, then inspect the third last octet for the coder rate. All MELPe speech frames in the RTP packet will be of this same coder rate.
The information carried in the MIME media type specification has a specific mapping to fields in the Session Description Protocol (SDP) [RFC2327], which is commonly used to describe RTP sessions. When SDP is used to specify sessions employing the MELPe codec, the mapping is as follows:

- The MIME type ("audio") goes in SDP "m=" as the media name.
- The MIME subtype (payload format name) goes in SDP "a=rtpmap" as the encoding name.
- The parameter "rate" goes in the SDP "a=fmtp" attribute by copying it directly from the MIME media type string as "rate=value" or "rate=value1,value2" or "rate=value1,value2,value3".

When conveying information by SDP, the encoding name SHALL be "MELP" (the same as the MIME subtype). Alternative encoding name types, "MELP2400", "MELP1200", and "MELP600", may be used in SDP to convey fixed rate configurations. These names have been observed in systems that do not support dynamic frame rate switching as specified by the parameter, "rate".

An example of the media representation in SDP for describing MELPe might be:

```
m=audio 49120 RTP/AVP 97
a=rtpmap:97 MELP/8000
```

An alternative example of SDP for fixed rate configurations might be:

```
m=audio 49120 RTP/AVP 97 100 101 102
a=rtpmap:97 MELP/8000
a=rtpmap:100 MELP2400/8000
a=rtpmap:101 MELP1200/8000
a=rtpmap:102 MELP600/8000
```

If the encoding name "MELP" is received without a "rate" parameter, the fixed coder rate of 2400 MUST be used. The alternate encoding names, "MELP2400", "MELP1200", and "MELP600" directly specify the MELPe coder rate of 2400, 1200, and 600 respectively and MUST not specify a "rate" parameter.

A remote MELPe encoder SHALL receive "rate" parameter in the SDP "a=fmtp" attribute by copying them directly from the MIME media type string as a semicolon separated with parameter=value, where parameter is "rate", and value can be one or more of 2400, 1200, and 600 separated by commas (where each rate value indicates the corresponding MELPe coder). An example of the media representation in
SDP for describing MELPe when all three coder rates are supported might be:

m=audio 49120 RTP/AVP 97
a=rtpmap:97 MELP/8000
a=fmtp:97 rate=2400,600,1200

For streaming media, the "rate" parameter specifies the possible rates used by the sender. In an Offer/Answer mode [RFC3264], "rate" is a bi-directional parameter. Both sides MUST use a common "rate" value or values.

The offer contains the rates supported by the offerer listed in its preferred order. The answerer may agree to any rate by listing the rate first in the answerer response. Additionally the answerer may indicate any secondary rate or rates that it supports. The initial rate used by both parties SHALL be the first bandwidth rate specified in the answerer response.

For example if offerer rates are "2400,600", and answer rates are "600,2400", the initial rate is 600. If other rates are provided by the answerer, any common rate between offer and answer may be used at any time in the future, Activation of these other common rates is beyond the scope of this document.

The use of a lower rate is often important for a case such as when one end point utilizes a bandwidth constrained link (e.g. 1200 bps radio link or slower), where only the lower coder rate will work.

Parameter ptime can not be used for the purpose of specifying MELPe operating mode, due to fact that for the certain values it will be impossible to distinguish which mode is about to be used (e.g. when ptime=67, it would be impossible to distinguish if packet is carrying 1 frames of 67.5 ms or 3 frames of 22.5 ms etc.).

When SDP is used for broadcast MELPe audio, multiple MELPe rtpmap values (such as 97, 98, and 99 as used below) may be used to convey MELPe coded voice at different rates. The receiver can then select an appropriate MELPe codec by using 97, 98, or 99.

m=audio 49120 RTP/AVP 97 98 99
a=rtpmap:97 MELP/8000
a=fmtp:97 rate=2400
a=rtpmap:98 MELP/8000
a=fmtp:98 rate=1200
a=rtpmap:99 MELP/8000
a=fmtp:99 rate=600
Note that the payload format (encoding) names are commonly shown in upper case. MIME subtypes are commonly shown in lower case. These names are case-insensitive in both places. Similarly, parameter names are case-insensitive both in MIME types and in the default mapping to the SDP a=fmtp attribute.

5 DISCONTINUOUS TRANSMISSION

A primary application of MELPe is for radio communications of voice conversations and discontinuous transmissions are normal. When MELPe is used in an IP network, MELPe RTP packet transmissions may cease and resume frequently. RTP SSRC sequence number gaps indicate lost packets to be filled by PLC while abrupt loss of RTP packets indicate intended discontinuous transmission.

If a MELPe coder so desires, it may send a comfort noise frame as per SCIP-210 Appendix B [SCIP210] prior to ceasing transmission. A receiver may optionally use comfort noise during its silence periods. No SDP negotiations are required.

6 PACKET LOSS CONCEALMENT

MELPe packet loss concealment (PLC) uses the special properties and coding for the pitch/voicing parameter of the MELPe 2400 bps coder. The PLC erasure indication may utilize any of the errored encodings of a non-voiced frame as identified in Table 1 of [MELPE]. For the sake of simplicity it is recommended to use a code value of 3 for the pitch/voicing parameter (represented by the bits P6 to P0 in Table 3.1). Hence, set bits P0 and P1 to one and bits P2, P3, P4, P5, and P6 to zero.

When using PLC in a 1200 bps or 600 bps mode, the MELPe 2400 bps decoder is called three or four times respectively to cover the loss of a MELPe frame.
7 Security Considerations

RTP packets using the payload format defined in this specification are subject to the general security considerations discussed in [RFC3550] and any appropriate profile (e.g. [RFC2736]).

As this format transports encoded speech, the main security issues include confidentiality and authentication of the speech itself. The payload format itself does not have any built-in security mechanisms. Confidentiality of the media streams is achieved by encryption, therefore external mechanisms, such as SRTP [RFC3711], MAY be used for that purpose. The data compression used with this payload format is applied end-to-end; hence encryption may be performed after compression with no conflict between the two operations.

A potential denial-of-service threat exists for data encoding using compression techniques that have non-uniform receiver-end computational load. The attacker can inject pathological datagrams into the stream which are complex to decode and cause the receiver to become overloaded. However, the encodings covered in this document do not exhibit any significant non-uniformity.

8 IANA Considerations

9 References

9.1 Normative References


the Session Description Protocol (SDP)" IETF RFC 3264, June 2002.


9.2 Informative References


Authors' Addresses

Victor Demjanenko, Ph.D.
VOCAL Technologies, Ltd.
520 Lee Entrance, Suite 202
Buffalo, NY 14228
USA
Phone: +1 716 688 4675
Email: victor.demjanenko@vocal.com

David Satterlee
VOCAL Technologies, Ltd.
520 Lee Entrance, Suite 202
Buffalo, NY 14228
USA
Phone: +1 716 688 4675
Email: david.satterlee@vocal.com
Abstract

This memo describes an RTP Payload format for SMPTE Ancillary data, as defined by SMPTE ST 291-1. SMPTE Ancillary data is generally used along with professional video formats to carry a range of ancillary data types, including time code, KLV metadata, Closed Captioning, and the Active Format Description (AFD).

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 July 31, 2015.

Copyright Notice

Copyright (c) 2015 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.
1. Introduction

This memo describes an RTP Payload format for Society of Motion Picture and Television Engineers (SMPTE) Ancillary data (ANC), as defined by SMPTE ST 291-1 [ST291]. ANC can carry a range of data types, including time code, KLV metadata, Closed Captioning, and the Active Format Description (AFD).

ANC is generally associated with the carriage of metadata within the bit stream multiplex of a Serial Digital Interface (SDI) such as SMPTE ST 259 [ST259], the standard definition (SD) Serial Digital Interface (with ANC data inserted as per SMPTE ST 125 [ST125]), or SMPTE ST 292-1 [ST292], the 1.5 Gb/s Serial Digital Interface for high definition (HD) television applications.

ANC data packet payload definitions for a specific application are specified by a SMPTE Standard, Recommended Practice, or Registered Disclosure Document, or by a document generated by another organization, a company, or an individual (an Entity). When a payload format is registered with SMPTE, an application document describing the payload format is required, and the registered ancillary data packet is identified by a registered data identification word.

This RTP payload supports ANC data packets regardless of whether they originate from an SD or HD interface, or if the ANC data packet is from the vertical ancillary space (VANC) or the horizontal ancillary space (HANC), or if the ANC packet is located in the luma (Y) or color-difference (C) channel. Sufficient information is provided to enable the ANC packets at the output of the decoder to be restored to their "original" locations in the serial digital video signal raster.
(if that is desired). This payload could be used by itself, or used along with a range of RTP video formats. In particular, it has been specifically designed so that it could be used along with RFC 4175 [RFC4175] "RTP Payload Format for Uncompressed Video" or RFC 5371 [RFC5371] "RTP Payload Format for JPEG 2000 Video Streams."

The data model in this document for the ANC data RTP payload is based on the data model of SMPTE ST 2038 [ST2038], which standardizes the carriage of ANC data packets in an MPEG-2 Transport Stream.

1.1. Requirements 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 RFC 2119 [RFC2119].

2. RTP Payload Format for SMPTE ST 291 Ancillary Data

The format of an RTP packet containing SMPTE ST 291 Ancillary Data is shown below:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|X| CC    |M|    PT       |        sequence number        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           timestamp                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           synchronization source (SSRC) identifier           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Extended Sequence Number    |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| ANC_Count     |C|   Line_Number       |   Horizontal_Offset   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        DID        |        SDID       |   Data_Count      | R |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    User_Data_Words...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Checksum_Word   |octet_align|    (next ANC data packet)...+
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: SMPTE Ancillary Data RTP Packet Format

RTP packet header fields SHALL be interpreted as per RFC 3550 [RFC3550], with the following specifics:

Timestamp: 32 bits
The timestamp field is interpreted in a similar fashion to RFC 4175 [RFC4175]:

For progressive scan video, the timestamp SHALL denote the sampling instant of the frame to which the ancillary data in the RTP packet belongs. Packets MUST NOT include ANC data from multiple frames, and all packets with ANC data belonging to the same frame MUST have the same timestamp.

For interlaced video, the timestamp SHALL denote the sampling instant of the field to which the ancillary data in the RTP packet belongs. Packets MUST NOT include ANC data from multiple fields, and all packets belonging to the same field MUST have the same timestamp.

A 90-kHz timestamp SHOULD be used in both cases. If the sampling instant does not correspond to an integer value of the clock, the value SHALL be truncated to the next lowest integer, with no ambiguity.

Marker bit (M): 1 bit

The marker bit set to "1" SHALL indicate the last RTP packet containing ANC data for a frame (for progressive scan video) or the last RTP packet containing ANC data for a field (for interlaced video).

2.1. Payload Header Definitions

The ANC RTP payload header fields are defined as:

Extended Sequence Number: 16 bits

The high order bits of the extended 32-bit sequence number, in network byte order. This is the same as the Extended Sequence Number field in RFC 4175 [RFC4175].

Length: 16 bits

Number of octets of the ANC RTP payload, beginning with the "C" bit of the first ANC packet data.

ANC_Count: 8 bits

This field is the count of the total number of ANC data packets carried in the RTP payload. A single ANC RTP packet payload SHALL NOT carry more than 255 ANC data packets.

And for each ANC data packet in the payload, the following header fields MUST be present:

C: 1 bit
For HD signals, this flag, when set to "1", indicates that the ANC data corresponds to the color-difference channel (C). When set to "0", this flag indicates that the ANC data corresponds to the luma (Y) channel. For SD signals, this flag SHALL be set to "0".

Line_Number: 11 bits
This field contains the line number (as defined in ITU-R BT.1700 [BT1700] for SD video or ITU-R BT.1120 [BT1120] for HD video) that corresponds to the location of the ANC data packet. The lines that are available to convey ANC data are as defined in the applicable sample structure specification (e.g., SMPTE 274M [ST274], SMPTE ST 296 [ST296], ITU-R BT.656 [BT656]) and may be further restricted per SMPTE RP 168 [RP168]. A value of 0x7FF (all bits in the field are '1') SHALL indicate that the ANC data is carried without any specific location within the frame.

Horizontal_Offset: 12 bits
This field defines the location of the ANC packet relative to the start of active video (SAV). 0 means that the Ancillary Data Flag (ADF) of the ANC packet begins immediately following SAV. For HD, this shall be in units of luma sample numbers as specified by the defining document of the particular image (e.g., SMPTE 274M [ST274] for 1920 x 1080 active images, or SMPTE ST 296 [ST296] for 1280 x 720 progressive active images). For SD, this is in units of (27MHz) multiplexed word numbers, as specified in SMPTE ST 125 [ST125]. It should be noted that HANC space in the digital blanking area will generally have higher luma sample numbers than any samples in the active digital line.

The fields DID, SDID, Data_Count, User_Data_Words, and Checksum_Word represent the 10-bit words carried in the ANC data packet, as per SMPTE ST 291 [ST291]:

DID: 10 bits
Data Identification Word

SDID: 10 bits
Secondary Data Identification Word. Used only for a "Type 2" ANC data packet. Note that in a "Type 1" ANC data packet, this word will actually carry the Data Block Number (DBN).

Data_Count: 10 bits
The lower 8 bits of Data_Count, corresponding to bits b7 (MSB) through b0 (LSB) of the 10-bit Data_Count word, contain the actual count of 10-bit words in User_Data_Words. Bit b8
is the even parity for bits b7 through b0, and bit b9 is the inverse (logical NOT) of bit b8.

R: 2 reserved bits
R is a field of two reserved bits that MUST be set to zero.

User_Data_Words: integer number of 10 bit words
User_Data_Words (UDW) are used to convey information of a type as identified by the DID word or the DID and SDID words. The number of 10-bit words in the UDW is defined by the Data_Count field.

Checksum_Word: 10 bits
The Checksum_Word can be used to determine the validity of the ANC data packet from the DID word through the UDW. The lower 8 bits of Checksum_Word, corresponding to bits b8 (MSB) through b0 (LSB) of the 10-bit data count word, contain the actual checksum value. Bit b9 is the inverse (logical NOT) of bit b8. The checksum value is equal to the nine least significant bits of the sum of the nine least significant bits of the DID word, the SDID word, the Data_Count word, and all User_Data_Words in the ANC data packet. The checksum is initialized to zero before calculation, and any end carry resulting from the checksum calculation is ignored.

octet_align: 0-7 bits as needed to complete octet
Octet align contains enough "0" bits as needed to complete the last octet of an ANC packet’s data in the RTP payload. This ensures that the next ANC packet’s data in the RTP payload begins octet-aligned despite ANC packets being made up of 10-bit words. If an ANC data packet in the RTP payload ends aligned with an octet, there is no need to add any octet alignment bits.

3. Payload Format Parameters

This RTP payload format is identified using the video/smpte291 media type, which is registered in accordance with RFC 4855 [RFC4855], and using the template of RFC 4288 [RFC4288].

Note that the Media Type Definition is in the "video" tree due to the expected use of SMPTE ST 291 Ancillary Data with video formats.

3.1. Media Type Definition

Type name: video
Subtype name: smpte291
Required parameters:

Rate: RTP timestamp clock rate.

Optional parameters:

DID: Data identification word

SDID: Secondary data identification word

The presence of the DID and SDID parameters signal that all ancillary data packets of this stream are of a particular type, i.e., labeled with a particular DID and SDID. DID and SDID values are registered with SMPTE as per SMPTE ST 291-1 [ST291]. DID and SDID values should be specified in hexadecimal with a "0x" prefix (such as "0x61"). For example, EIA 608 Closed Caption data would be DID=0x61 and SDID=0x02. If DID and SDID are not specified, then the ancillary data stream may potentially contain ancillary data packets of any type.

DID and SDID values can be found on the SMPTE Registry for Data Identification Word Assignments for Registered DIDs at:


Encoding considerations: This media type is framed and binary; see Section 4.8 of RFC 4288 [RFC4288].

Security considerations: See Section 5 of [this RFC]

Interoperability considerations: Data items in smpte291 can be very diverse. Receivers might only be capable of interpreting a subset of the possible data items. Some implementations may care about the location of the ANC data packets in the SDI raster, but other implementations may not care.

Published specification: [this RFC]

Applications that use this media type: Devices that stream real-time professional video, especially those that must interoperate with legacy serial digital interfaces (SDI).

Additional Information: none

Person & email address to contact for further information: T. Edwards <thomas.edwards@fox.com>, IETF Payload Working Group <payload@ietf.org>
Intended usage: COMMON

Restrictions on usage: This media type depends on RTP framing, and hence is only defined for transfer via RTP RFC 3550 [RFC3550]. Transport within other framing protocols is not defined at this time.

Author: T. Edwards <thomas.edwards@fox.com>

Change controller: IETF Payload working group delegated from the IESG.

3.2. Mapping to SDP

The mapping of the above defined payload format media type and its parameters SHALL be done according to Section 3 of RFC 4855 [RFC4855].

- The type name ("video") goes in SDP "m=" as the media name.
- The subtype name ("smpte291") goes in SDP "a=rtpmap" as the encoding name.
- The optional DID and SDID parameters go in the SDP "a=fmtp" attribute as a semicolon-separated list of parameter=value pairs.

A sample SDP mapping for ancillary data is as follows:

```plaintext
m=video 30000 RTP/AVP 112
a=rtpmap:112 smpte291/90000
a=fmtp:112 DID=0x61; SDID=0x02;
```

In this example, a dynamic payload type 112 is used for ancillary data. The 90 kHz RTP timestamp rate is specified in the "a=rtpmap" line after the subtype. The RTP sampling clock is 90 kHz. In the "a=fmtp:" line, DID 0x61 and SDID 0x02 are specified (which are registered to EIA 608 Closed Caption Data by SMPTE).

3.3. Offer/Answer Model and Declarative Considerations

When offering SMPTE ST 291 Ancillary data over RTP using the Session Description Protocol (SDP) in an Offer/Answer model [RFC3264] or in a declarative manner (e.g., SDP in the Real-Time Streaming Protocol (RTSP) [RFC2326] or the Session Announcement Protocol (SAP) [RFC2974]), the offerer could provide a list of streams available with specific DID & SDIDs, and the answerer could specify which streams with specific DID & SDIDs it would like to accept.
4. IANA Considerations

One media type (video/smpte291) has been defined and needs registration in the media types registry. See Section 3.1

5. Security Considerations

RTP packets using the payload format defined in this specification are subject to the security considerations discussed in the RTP specification [RFC3550] and any applicable RTP profile, e.g., AVP [RFC3551].

To avoid potential buffer overflow attacks, receivers should take care to validate that the ANC packets in the RTP payload are of the appropriate length (using the Data_Count field) for the ANC data type specified by DID & SDID. Also the Checksum_Word should be checked against the ANC data packet to ensure that its data has not been damaged in transit.

Some receivers will simply move the ANC data packet bits from the RTP payload into a serial digital interface (SDI). It may still be a good idea for these "re-embedders" to perform the above mentioned validity tests to avoid downstream SDI systems from becoming confused by bad ANC packets, which could be used for a denial of service attack.

"Re-embedders" into SDI should also double check that the Line_Number and Horizontal_Offset leads to the ANC data packet being inserted into a legal area to carry ancillary data in the SDI video bit stream of the output video format.

6. References

6.1. Normative References


6.2. Informative References


Author’s Address

Thomas G. Edwards
FOX
10201 W. Pico Blvd.
Los Angeles, CA  90035
USA

Phone: +1 310 369 6696
Email: thomas.edwards@fox.com
RTP Payload Format for Non-Interleaved and Interleaved Parity Forward Error Correction (FEC)  
draft-ietf-payload-flexible-fec-scheme-00

Abstract

This document defines new RTP payload formats for the Forward Error Correction (FEC) packets that are generated by the non-interleaved and interleaved parity codes from a source media encapsulated in RTP. These parity codes are systematic codes, where a number of repair symbols are generated from a set of source symbols. These repair symbols are sent in a repair flow separate from the source flow that carries the source symbols. The non-interleaved and interleaved parity codes offer a good protection against random and bursty packet losses, respectively, at a cost of decent complexity. The RTP payload formats that are defined in this document address the scalability issues experienced with the earlier specifications including RFC 2733, RFC 5109 and SMPTE 2022-1, and offer several improvements. Due to these changes, the new payload formats are not backward compatible with the earlier specifications, but endpoints that do not implement the scheme can still work by simply ignoring the FEC packets.

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 18, 2015.

1. Introduction

This document defines new RTP payload formats for the Forward Error Correction (FEC) that is generated by the non-interleaved and interleaved parity codes from a source media encapsulated in RTP [RFC3550]. The type of the source media protected by these parity codes can be audio, video, text or application. The FEC data are generated according to the media type parameters, which are communicated out-of-band (e.g., in SDP). Furthermore, the associations or relationships between the source and repair flows may be communicated in-band or out-of-band. Situations where adaptivity of FEC parameters is desired, the endpoint can use the in-band mechanism, whereas when the FEC parameters are fixed, the endpoint may prefer to negotiate them out-of-band.

Both the non-interleaved and interleaved parity codes use the eXclusive OR (XOR) operation to generate the repair symbols. In a nutshell, the following steps take place:

1. The sender determines a set of source packets to be protected by FEC based on the media type parameters.
2. The sender applies the XOR operation on the source symbols to generate the required number of repair symbols.
3. The sender packetizes the repair symbols and sends the repair packet(s) along with the source packets to the receiver(s) (in different flows). The repair packets may be sent proactively or on-demand.
Note that the source and repair packets belong to different source and repair flows, and the sender must provide a way for the receivers to demultiplex them, even in the case they are sent in the same 5-tuple (i.e., same source/destination address/port with UDP). This is required to offer backward compatibility for endpoints that do not understand the FEC packets (See Section 4). 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. Figure 1 and Figure 2 describe example block diagrams for the systematic parity FEC encoder and decoder, respectively.

![Block diagram for systematic parity FEC encoder](image1)

**Figure 1: Block diagram for systematic parity FEC encoder**

![Block diagram for systematic parity FEC decoder](image2)

**Figure 2: Block diagram for systematic parity FEC decoder**

In Figure 2, it is clear that the FEC packets have to be received by the endpoint within a certain amount of time for the FEC recovery process to be useful. In this document, we refer to the time that spans a FEC block, which consists of the source packets and the corresponding repair packets, as the repair window. At the receiver side, the FEC decoder should wait at least for the duration of the repair window after getting the first packet in a 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 a FEC decoding failure until it waits at least for the duration of the repair window.

Suppose that we have a group of $D \times L$ source packets that have sequence numbers starting from 1 running to $D \times L$, and a repair packet is generated by applying the XOR operation to every $L$ consecutive packets as sketched in Figure 3. This process is referred to as 1-D non-interleaved FEC protection. As a result of this process, $D$ repair packets are generated, which we refer to as non-interleaved (or row) FEC packets.

If we apply the XOR operation to the group of the source packets whose sequence numbers are $L$ apart from each other, as sketched in Figure 4. In this case the endpoint generates $L$ repair packets. This process is referred to as 1-D interleaved FEC protection, and the resulting $L$ repair packets are referred to as interleaved (or column) FEC packets.

Figure 3: Generating non-interleaved (row) FEC packets

Figure 4: Generating interleaved (column) FEC packets
1.1. Use Cases for 1-D FEC Protection

We generate one non-interleaved repair packet out of \(L\) consecutive source packets or one interleaved repair packet out of \(D\) non-consecutive source packets. Regardless of whether the repair packet is a non-interleaved or an interleaved one, it can provide a full recovery of the missing information if there is only one packet missing among the corresponding source packets. This implies that 1-D non-interleaved FEC protection performs better when the source packets are randomly lost. However, if the packet losses occur in bursts, 1-D interleaved FEC protection performs better provided that \(L\) is chosen large enough, i.e., \(L\)-packet duration is not shorter than the observed burst duration. If the sender generates non-interleaved FEC packets and a burst loss hits the source packets, the repair operation fails. This is illustrated in Figure 5.
The sender may generate interleaved FEC packets to combat with the bursty packet losses. However, two or more random packet losses may hit the source and repair packets in the same column. In that case, the repair operation fails as well. This is illustrated in Figure 6. Note that it is possible that two burst losses may occur back-to-back, in which case interleaved FEC packets may still fail to recover the lost data.

1.2. Use Cases for 2-D Parity FEC Protection

In networks where the source packets are lost both randomly and in bursts, the sender ought to generate both non-interleaved and interleaved FEC packets. This type of FEC protection is known as 2-D parity FEC protection. At the expense of generating more FEC...
packets, thus increasing the FEC overhead, 2-D FEC provides superior protection against mixed loss patterns. However, it is still possible for 2-D parity FEC protection to fail to recover all of the lost source packets if a particular loss pattern occurs. An example scenario is illustrated in Figure 7.

Figure 7: Example scenario #1 where 2-D parity FEC protection fails error recovery

2-D parity FEC protection also fails when at least two rows are missing a source and the FEC packet and the missing source packets (in at least two rows) are aligned in the same column. An example loss pattern is sketched in Figure 8. Similarly, 2-D parity FEC protection cannot repair all missing source packets when at least two columns are missing a source and the FEC packet and the missing source packets (in at least two columns) are aligned in the same row.
1.3. Overhead Computation

The overhead is defined as the ratio of the number of bytes belonging to the repair packets to the number of bytes belonging to the protected source packets.

Generally, repair packets are larger in size compared to the source packets. Also, not all the source packets are necessarily equal in size. However, if we assume that each repair packet carries an equal number of bytes carried by a source packet, we can compute the overhead for different FEC protection methods as follows:

- **1-D Non-interleaved FEC Protection:** Overhead = \( \frac{1}{L} \)
- **1-D Interleaved FEC Protection:** Overhead = \( \frac{1}{D} \)
- **2-D Parity FEC Protection:** Overhead = \( \frac{1}{L} + \frac{1}{D} \)

where \( L \) and \( D \) are the number of columns and rows in the source block, respectively.

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. Definitions and Notations

3.1. Definitions

This document uses a number of definitions from [RFC6363].

3.2. Notations

- \( L \): Number of columns of the source block.
- \( D \): Number of rows of the source block.
- bitmask: Run-length encoding of packets protected by a FEC packet. If the bit \( i \) in the mask is set to 1, the source packet number \( N + i \) is protected by this FEC packet. Here, \( N \) is the sequence number base, which is indicated in the FEC packet as well.

4. Packet Formats

This section defines the formats of the source and repair packets.

4.1. Source Packets

The source packets MUST contain the information that identifies the source block and the position within the source block occupied by the packet. Since the source packets that are carried within an RTP stream already contain unique sequence numbers in their RTP headers [RFC3550], we can identify the source packets in a straightforward manner and there is no need to append additional field(s). The primary advantage of not modifying the source packets in any way is that it provides backward compatibility for the receivers that do not support FEC at all. In multicast scenarios, this backward compatibility becomes quite useful as it allows the non-FEC-capable and FEC-capable receivers to receive and interpret the same source packets sent in the same multicast session.

4.2. Repair Packets

The repair packets MUST contain information that identifies the source block they pertain to and the relationship between the contained repair symbols and the original source block. For this purpose, we use the RTP header of the repair packets as well as another header within the RTP payload, which we refer to as the FEC header, as shown in Figure 9.
The RTP header is formatted according to [RFC3550] with some further clarifications listed below:

- **Marker (M) Bit**: This bit is not used for this payload type, and SHALL be set to 0.

- **Payload Type**: The (dynamic) payload type for the repair packets is determined through out-of-band means. Note that this document registers new payload formats for the repair packets (Refer to Section 5 for details). According to [RFC3550], an RTP receiver that cannot recognize a payload type must discard it. This provides backward compatibility. If a non-FEC-capable receiver receives a repair packet, it will not recognize the payload type, and hence, will discard the repair packet.

- **Sequence Number (SN)**: The sequence number has the standard definition. It MUST be one higher than the sequence number in the previously transmitted repair packet. The initial value of the sequence number SHOULD be random (unpredictable, based on [RFC3550]).

- **Timestamp (TS)**: The timestamp SHALL be set to a time corresponding to the repair packet’s transmission time. Note that the timestamp value has no use in the actual FEC protection process and is usually useful for jitter calculations.

- **Synchronization Source (SSRC)**: The SSRC value SHALL be randomly assigned as suggested by [RFC3550]. This allows the sender to multiplex the source and repair flows on the same port, or multiplex multiple repair flows on a single port. The repair flows SHOULD use the RTCP CNAME field to associate themselves with the source flow.
In some networks, the RTP Source, which produces the source packets and the FEC Source, which generates the repair packets from the source packets may not be the same host. In such scenarios, using the same CNAME for the source and repair flows means that the RTP Source and the FEC Source MUST share the same CNAME (for this specific source-repair flow association). A common CNAME may be produced based on an algorithm that is known both to the RTP and FEC Source [RFC7022]. This usage is compliant with [RFC3550].

Note that due to the randomness of the SSRC assignments, there is a possibility of SSRC collision. In such cases, the collisions MUST be resolved as described in [RFC3550].

The format of the FEC header is shown in Figure 10.

```
+---------------+-------------------------------------+
|    MSK bits   | Use                                 |
+---------------+-------------------------------------+
|      00       | 16-bit mask                         |
|      01       | 48-bit mask                         |
|      10       | 112-bit mask                        |
|      11       | packets indicated by offset M and N |
+---------------+-------------------------------------+
```

Figure 11: MSK bit values
The P, X, CC, M and PT recovery fields are used to determine the corresponding fields of the recovered packets.

The SN base field is used to indicate the lowest sequence number, taking wrap around into account, of those source packets protected by this repair packet.

The TS recovery field is used to determine the timestamp of the recovered packets.

The Length recovery field is used to determine the length of the recovered packets.

Mask is a run-length encoding of packets protected by the FEC packet. Where a bit i set to 1 indicates that the source packet with sequence number (SN base + i) is protected by this FEC packet.

If the the MSK field is set to 11, it indicates the offset of packets protected by this FEC packet. Consequently, the following conditions may occur:

If M=0, N=0, regular protection pattern code with the values of L and D are indicated in the SDP description.
If M>0, N=0, indicates a non-interleaved (row) FEC of M packets starting at SN base.
   Hence, FEC = SN, SN+1, SN+2, ..., SN+(M-1), SN+M.
If M>0, N>0, indicates interleaved (column) FEC of every M packet in a group of N packets starting at SN base.
   Hence, FEC = SN+(Mx0), SN+(Mx1), ..., SN+(MxN).

Figure 12: Interpreting the M and N field values

The details on setting the fields in the FEC header are provided in Section 6.2.

It should be noted that a mask-based approach (similar to the ones specified in [RFC2733] and [RFC5109]) may not be very efficient to indicate which source packets in the current source block are associated with a given repair packet. In particular, for the applications that would like to use large source block sizes, the size of the mask that is required to describe the source-repair packet associations may be prohibitively large. The 8-bit fields proposed in [SMPTE2022-1] indicate a systematized approach. Instead the approach in this document uses the 8-bit fields to indicate packet offsets protected by the FEC packet. The approach in [SMPTE2022-1] is inherently more efficient for regular patterns, it
does not provide flexibility to represent other protection patterns (e.g., staircase).

5. Payload Format Parameters

This section provides the media subtype registration for the non-interleaved and interleaved parity FEC. The parameters that are required to configure the FEC encoding and decoding operations are also defined in this section.

5.1. Media Type Registration

This registration is done using the template defined in [RFC6838] and following the guidance provided in [RFC3555].

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

5.1.1. Registration of audio/non-interleaved-parityfec

Type name: audio

Subtype name: non-interleaved-parityfec

Required parameters:

- rate: The RTP timestamp (clock) rate. The rate SHALL be larger than 1000 Hz to provide sufficient resolution to RTCP operations. However, it is RECOMMENDED to select the rate that matches the rate of the protected source RTP stream.

- L: Number of columns of the source block. L is a positive integer.

- D: Number of rows of the source block. D is a positive integer.

- ToP: Type of the protection applied by the sender: 0 for 1-D interleaved FEC protection, 1 for 1-D non-interleaved FEC protection, and 2 for 2-D parity FEC protection. The ToP value of 3 is reserved for future uses.

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

Optional parameters: None.
Encoding considerations: This media type is framed (See Section 4.8 in the template document [RFC6838]) and contains binary data.

Security considerations: See Section 9 of [RFCXXXX].

Interoperability considerations: None.

Published specification: [RFCXXXX].

Applications that use this media type: Multimedia applications that want to improve resiliency against packet loss by sending redundant data in addition to the source media.

Fragment identifier considerations: None.

Additional information: None.

Person & email address to contact for further information: Varun Singh <varun.singh@iki.fi> and IETF Audio/Video Transport Payloads Working Group.

Intended usage: COMMON.

Restriction on usage: This media type depends on RTP framing, and hence, is only defined for transport via RTP [RFC3550].

Author: Varun Singh <varun.singh@iki.fi>.

Change controller: IETF Audio/Video Transport Working Group delegated from the IESG.

Provisional registration? (standards tree only): Yes.

5.1.2.  Registration of video/non-interleaved-parityfec

Type name: video

Subtype name: non-interleaved-parityfec

Required parameters:

- rate: The RTP timestamp (clock) rate. The rate SHALL be larger than 1000 Hz to provide sufficient resolution to RTCP operations. However, it is RECOMMENDED to select the rate that matches the rate of the protected source RTP stream.

- L: Number of columns of the source block. L is a positive integer.
o D: Number of rows of the source block. D is a positive integer.

o ToP: Type of the protection applied by the sender: 0 for 1-D interleaved FEC protection, 1 for 1-D non-interleaved FEC protection, and 2 for 2-D parity FEC protection. The ToP value of 3 is reserved for future uses.

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

Optional parameters: None.

Encoding considerations: This media type is framed (See Section 4.8 in the template document [RFC6838]) and contains binary data.

Security considerations: See Section 9 of [RFCXXXX].

Interoperability considerations: None.

Published specification: [RFCXXXX].

Applications that use this media type: Multimedia applications that want to improve resiliency against packet loss by sending redundant data in addition to the source media.

Fragment identifier considerations: None.

Additional information: None.

Person & email address to contact for further information: Varun Singh <varun.singh@iki.fi> and IETF Audio/Video Transport Payloads Working Group.

Intended usage: COMMON.

Restriction on usage: This media type depends on RTP framing, and hence, is only defined for transport via RTP [RFC3550].

Author: Varun Singh <varun.singh@iki.fi>.

Change controller: IETF Audio/Video Transport Working Group delegated from the IESG.

Provisional registration? (standards tree only): Yes.
5.1.3. Registration of text/non-interleaved-parityfec

Type name: text

Subtype name: non-interleaved-parityfec

Required parameters:

- **rate**: The RTP timestamp (clock) rate. The rate SHALL be larger than 1000 Hz to provide sufficient resolution to RTCP operations. However, it is RECOMMENDED to select the rate that matches the rate of the protected source RTP stream.
- **L**: Number of columns of the source block. L is a positive integer.
- **D**: Number of rows of the source block. D is a positive integer.
- **ToP**: Type of the protection applied by the sender: 0 for 1-D interleaved FEC protection, 1 for 1-D non-interleaved FEC protection, and 2 for 2-D parity FEC protection. The ToP value of 3 is reserved for future uses.
- **repair-window**: The time that spans the source packets and the corresponding repair packets. The size of the repair window is specified in microseconds.

Optional parameters: None.

Encoding considerations: This media type is framed (See Section 4.8 in the template document [RFC6838]) and contains binary data.

Security considerations: See Section 9 of [RFCXXXX].

Interoperability considerations: None.

Published specification: [RFCXXXX].

Applications that use this media type: Multimedia applications that want to improve resiliency against packet loss by sending redundant data in addition to the source media.

Fragment identifier considerations: None.

Additional information: None.
Person & email address to contact for further information: Varun Singh <varun.singh@iki.fi> and IETF Audio/Video Transport Payloads Working Group.

Intended usage: COMMON.

Restriction on usage: This media type depends on RTP framing, and hence, is only defined for transport via RTP [RFC3550].

Author: Varun Singh <varun.singh@iki.fi>.

Change controller: IETF Audio/Video Transport Working Group delegated from the IESG.

Provisional registration? (standards tree only): Yes.

5.1.4. Registration of application/non-interleaved-parityfec

Type name: application

Subtype name: non-interleaved-parityfec

Required parameters:

<table>
<thead>
<tr>
<th>o rate:</th>
<th>The RTP timestamp (clock) rate. The rate SHALL be larger than 1000 Hz to provide sufficient resolution to RTCP operations. However, it is RECOMMENDED to select the rate that matches the rate of the protected source RTP stream.</th>
</tr>
</thead>
<tbody>
<tr>
<td>o L:</td>
<td>Number of columns of the source block. L is a positive integer.</td>
</tr>
<tr>
<td>o D:</td>
<td>Number of rows of the source block. D is a positive integer.</td>
</tr>
<tr>
<td>o ToP:</td>
<td>Type of the protection applied by the sender: 0 for 1-D interleaved FEC protection, 1 for 1-D non-interleaved FEC protection, and 2 for 2-D parity FEC protection. The ToP value of 3 is reserved for future uses.</td>
</tr>
<tr>
<td>o repair-window:</td>
<td>The time that spans the source packets and the corresponding repair packets. The size of the repair window is specified in microseconds.</td>
</tr>
</tbody>
</table>

Optional parameters: None.

Encoding considerations: This media type is framed (See Section 4.8 in the template document [RFC6838]) and contains binary data.
Security considerations: See Section 9 of [RFCXXXX].

Interoperability considerations: None.

Published specification: [RFCXXXX].

Applications that use this media type: Multimedia applications that want to improve resiliency against packet loss by sending redundant data in addition to the source media.

Fragment identifier considerations: None.

Additional information: None.

Person & email address to contact for further information: Varun Singh <varun.singh@iki.fi> and IETF Audio/Video Transport Payloads Working Group.

Intended usage: COMMON.

Restriction on usage: This media type depends on RTP framing, and hence, is only defined for transport via RTP [RFC3550].

Author: Varun Singh <varun.singh@iki.fi>.

Change controller: IETF Audio/Video Transport Working Group delegated from the IESG.

Provisional registration? (standards tree only): Yes.

5.1.5. Registration of audio/interleaved-parityfec

Type name: audio

Subtype name: interleaved-parityfec

Required parameters:

- rate: The RTP timestamp (clock) rate. The rate SHALL be larger than 1000 Hz to provide sufficient resolution to RTCP operations. However, it is RECOMMENDED to select the rate that matches the rate of the protected source RTP stream.

- L: Number of columns of the source block. L is a positive integer.

- D: Number of rows of the source block. D is a positive integer.
ToP: Type of the protection applied by the sender: 0 for 1-D interleaved FEC protection, 1 for 1-D non-interleaved FEC protection, and 2 for 2-D parity FEC protection. The ToP value of 3 is reserved for future uses.

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

Optional parameters: None.

Encoding considerations: This media type is framed (See Section 4.8 in the template document [RFC6838]) and contains binary data.

Security considerations: See Section 9 of [RFCXXXX].

Interoperability considerations: None.

Published specification: [RFCXXXX].

Applications that use this media type: Multimedia applications that want to improve resiliency against packet loss by sending redundant data in addition to the source media.

Fragment identifier considerations: None.

Additional information: None.

Person & email address to contact for further information: Varun Singh <varun.singh@iki.fi> and IETF Audio/Video Transport Payloads Working Group.

Intended usage: COMMON.

Restriction on usage: This media type depends on RTP framing, and hence, is only defined for transport via RTP [RFC3550].

Author: Varun Singh <varun.singh@iki.fi>.

Change controller: IETF Audio/Video Transport Working Group delegated from the IESG.

Provisional registration? (standards tree only): Yes.
5.1.6. Registration of video/interleaved-parityfec

Type name: video

Subtype name: interleaved-parityfec

Required parameters:

- **rate**: The RTP timestamp (clock) rate. The rate SHALL be larger than 1000 Hz to provide sufficient resolution to RTCP operations. However, it is RECOMMENDED to select the rate that matches the rate of the protected source RTP stream.

- **L**: Number of columns of the source block. L is a positive integer.

- **D**: Number of rows of the source block. D is a positive integer.

- **ToP**: Type of the protection applied by the sender: 0 for 1-D interleaved FEC protection, 1 for 1-D non-interleaved FEC protection, and 2 for 2-D parity FEC protection. The ToP value of 3 is reserved for future uses.

- **repair-window**: The time that spans the source packets and the corresponding repair packets. The size of the repair window is specified in microseconds.

Optional parameters: None.

Encoding considerations: This media type is framed (See Section 4.8 in the template document [RFC6838]) and contains binary data.

Security considerations: See Section 9 of [RFCXXXX].

Interoperability considerations: None.

Published specification: [RFCXXXX].

Applications that use this media type: Multimedia applications that want to improve resiliency against packet loss by sending redundant data in addition to the source media.

Fragment identifier considerations: None.

Additional information: None.
Person & email address to contact for further information: Varun Singh <varun.singh@iki.fi> and IETF Audio/Video Transport Payloads Working Group.

Intended usage: COMMON.

Restriction on usage: This media type depends on RTP framing, and hence, is only defined for transport via RTP [RFC3550].

Author: Varun Singh <varun.singh@iki.fi>.

Change controller: IETF Audio/Video Transport Working Group delegated from the IESG.

Provisional registration? (standards tree only): Yes.

5.1.7. Registration of text/interleaved-parityfec

Type name: text

Subtype name: interleaved-parityfec

Required parameters:

- rate: The RTP timestamp (clock) rate. The rate SHALL be larger than 1000 Hz to provide sufficient resolution to RTCP operations. However, it is RECOMMENDED to select the rate that matches the rate of the protected source RTP stream.

- L: Number of columns of the source block. L is a positive integer.

- D: Number of rows of the source block. D is a positive integer.

- ToP: Type of the protection applied by the sender: 0 for 1-D interleaved FEC protection, 1 for 1-D non-interleaved FEC protection, and 2 for 2-D parity FEC protection. The ToP value of 3 is reserved for future uses.

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

Optional parameters: None.

Encoding considerations: This media type is framed (See Section 4.8 in the template document [RFC6838]) and contains binary data.
Security considerations: See Section 9 of [RFCXXXX].

Interoperability considerations: None.

Published specification: [RFCXXXX].

Applications that use this media type: Multimedia applications that want to improve resiliency against packet loss by sending redundant data in addition to the source media.

Fragment identifier considerations: None.

Additional information: None.

Person & email address to contact for further information: Varun Singh <varun.singh@iki.fi> and IETF Audio/Video Transport Payloads Working Group.

Intended usage: COMMON.

Restriction on usage: This media type depends on RTP framing, and hence, is only defined for transport via RTP [RFC3550].

Author: Varun Singh <varun.singh@iki.fi>.

Change controller: IETF Audio/Video Transport Working Group delegated from the IESG.

Provisional registration? (standards tree only): Yes.

5.1.8. Registration of application/interleaved-parityfec

Type name: application

Subtype name: interleaved-parityfec

Required parameters:

- rate: The RTP timestamp (clock) rate. The rate SHALL be larger than 1000 Hz to provide sufficient resolution to RTCP operations. However, it is RECOMMENDED to select the rate that matches the rate of the protected source RTP stream.

- L: Number of columns of the source block. L is a positive integer.

- D: Number of rows of the source block. D is a positive integer.
o ToP: Type of the protection applied by the sender: 0 for 1-D interleaved FEC protection, 1 for 1-D non-interleaved FEC protection, and 2 for 2-D parity FEC protection. The ToP value of 3 is reserved for future uses.

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

Optional parameters: None.

Encoding considerations: This media type is framed (See Section 4.8 in the template document [RFC6838]) and contains binary data.

Security considerations: See Section 9 of [RFCXXXX].

Interoperability considerations: None.

Published specification: [RFCXXXX].

Applications that use this media type: Multimedia applications that want to improve resiliency against packet loss by sending redundant data in addition to the source media.

Fragment identifier considerations: None.

Additional information: None.

Person & email address to contact for further information: Varun Singh <varun.singh@iki.fi> and IETF Audio/Video Transport Payloads Working Group.

Intended usage: COMMON.

Restriction on usage: This media type depends on RTP framing, and hence, is only defined for transport via RTP [RFC3550].

Author: Varun Singh <varun.singh@iki.fi>.

Change controller: IETF Audio/Video Transport Working Group delegated from the IESG.

Provisional registration? (standards tree only): Yes.
5.2. Mapping to SDP Parameters

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. In this section, we provide these mappings for the media subtypes registered by this document. 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 media type specification for "non-interleaved-parityfec" and "interleaved-parityfec" and their parameters in SDP is as follows:

- The media type (e.g., "application") goes into the "m=" line as the media name.
- The media subtype goes into the "a=rtpmap" line as the encoding name. The RTP clock rate parameter ("rate") also goes into the "a=rtpmap" line as the clock rate.
- The remaining required payload-format-specific parameters go into the "a=fmtp" line by copying them directly from the media type string as a semicolon-separated list of parameter=value pairs.

SDP examples are provided in Section 7.

5.2.1. Offer-Answer Model Considerations

When offering 1-D interleaved parity FEC over RTP using SDP in an Offer/Answer model [RFC3264], the following considerations apply:

- Each combination of the L and D parameters produces a different FEC data and is not compatible with any other combination. A sender application may desire to offer multiple offers with different sets of L and D values as long as the parameter values are valid. The receiver SHOULD normally choose the offer that has a sufficient amount of interleaving. If multiple such offers exist, the receiver may choose the offer that has the lowest overhead or the one that requires the smallest amount of buffering. The selection depends on the application requirements.
- The value for the repair-window parameter depends on the L and D values and cannot be chosen arbitrarily. More specifically, L and D values determine the lower limit for the repair-window size.
The upper limit of the repair-window size does not depend on the L and D values.

- Although combinations with the same L and D values but with different repair-window sizes produce the same FEC data, such combinations are still considered different offers. 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.

- There are no optional format parameters defined for this payload. Any unknown option in the offer MUST be ignored and deleted from the answer. If FEC is not desired by the receiver, it can be deleted from the answer.

5.2.2. Declarative 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:

- The payload format configuration parameters are all declarative and a participant MUST use the configuration that is provided for the session.

- More than one configuration may be provided (if desired) by declaring multiple RTP payload types. In that case, the receivers should choose the repair flow that is best for them.

6. Protection and Recovery Procedures

This section provides a complete specification of the 1-D and 2-D parity codes and their RTP payload formats.

6.1. Overview

The following sections specify the steps involved in generating the repair packets and reconstructing the missing source packets from the repair packets.

6.2. Repair Packet Construction

The RTP header of a repair packet is formed based on the guidelines given in Section 4.2.

The FEC header includes 12 octets (or up to 28 octets when the longer optional masks are used). It is constructed by applying the XOR
operation on the bit strings that are generated from the individual source packets protected by this particular repair packet. The set of the source packets that are associated with a given repair packet can be computed by the formula given in Section 6.3.1.

The bit string is formed for each source packet by concatenating the following fields together in the order specified:

- The first 64 bits of the RTP header (64 bits).
- Unsigned network-ordered 16-bit representation of the source packet length in bytes minus 12 (for the fixed RTP header), i.e., the sum of the lengths of all the following if present: the CSRC list, extension header, RTP payload and RTP padding (16 bits).

By applying the parity operation on the bit strings produced from the source packets, we generate the FEC bit string. The FEC header is generated from the FEC bit string as follows:

- The first (most significant) 2 bits in the FEC bit string are skipped. The MSK bits in the FEC header are set to the appropriate value, i.e., it depends on the chosen bitmask length.
- The next bit in the FEC bit string is written into the P recovery bit in the FEC header.
- The next bit in the FEC bit string is written into the X recovery bit in the FEC header.
- The next 4 bits of the FEC bit string are written into the CC recovery field in the FEC header.
- The next bit is written into the M recovery bit in the FEC header.
- The next 7 bits of the FEC bit string are written into the PT recovery field in the FEC header.
- The next 16 bits are skipped.
- The next 32 bits of the FEC bit string are written into the TS recovery field in the FEC header.
- The next 16 bits are written into the length recovery field in the FEC header.
- Depending on the chosen MSK value, the bit mask of appropriate length will be set to the appropriate values.
As described in Section 4.2, the SN base field of the FEC header MUST be set to the lowest sequence number of the source packets protected by this repair packet. When MSK represents a bitmask (MSK=00,01,10), the SN base field corresponds to the lowest sequence number indicated in the bitmask. When MSK=11, the following considerations apply: 1) for the interleaved FEC packets, this corresponds to the lowest sequence number of the source packets that forms the column, 2) for the non-interleaved FEC packets, the SN base field MUST be set to the lowest sequence number of the source packets that forms the row.

The repair packet payload consists of the bits that are generated by applying the XOR operation on the payloads of the source RTP packets. If the payload lengths of the source packets are not equal, each shorter packet MUST be padded to the length of the longest packet by adding octet 0’s at the end.

Due to this possible padding and mandatory FEC header, a repair packet has a larger size than the source packets it protects. This may cause problems if the resulting repair packet size exceeds the Maximum Transmission Unit (MTU) size of the path over which the repair flow is sent.

6.3. Source Packet Reconstruction

This section describes the recovery procedures that are required to reconstruct the missing source packets. The recovery process has two steps. In the first step, the FEC decoder determines which source and repair packets should be used in order to recover a missing packet. In the second step, the decoder recovers the missing packet, which consists of an RTP header and RTP payload.

In the following, we describe the RECOMMENDED algorithms for the first and second steps. Based on the implementation, different algorithms MAY be adopted. However, the end result MUST be identical to the one produced by the algorithms described below.

Note that the same algorithms are used by the 1-D parity codes, regardless of whether the FEC protection is applied over a column or a row. The 2-D parity codes, on the other hand, usually require multiple iterations of the procedures described here. This iterative decoding algorithm is further explained in Section 6.3.4.

6.3.1. Associating the Source and Repair Packets

We denote the set of the source packets associated with repair packet \( p^* \) by set \( T(p^*) \). Note that in a source block whose size is \( L \) columns by \( D \) rows, set \( T \) includes \( D \) source packets plus one repair packet for the FEC protection applied over a column, and \( L \) source packets plus
one repair packet for the FEC protection applied over a row. Recall that 1-D interleaved and non-interleaved FEC protection can fully recover the missing information if there is only one source packet missing in set T. If there are more than one source packets missing in set T, 1-D FEC protection will not work.

6.3.1.1. Signaled in SDP

The first step is associating the source and repair packets. If the endpoint relies entirely on out-of-band signaling (MSK=11, and M=N=0), then this information may be inferred from the media type parameters specified in the SDP description. Furthermore, the payload type field in the RTP header, assists the receiver distinguish an interleaved or non-interleaved FEC packet.

Mathematically, for any received repair packet, p*, we can determine the sequence numbers of the source packets that are protected by this repair packet as follows:

\[ p*_\text{snb} + i \times X_1 \text{ (modulo 65536)} \]

where \( p*_\text{snb} \) denotes the value in the SN base field of p*'s FEC header, \( X_1 \) is set to L and 1 for the interleaved and non-interleaved FEC packets, respectively, and

\[ 0 \leq i < X_2 \]

where \( X_2 \) is set to D and L for the interleaved and non-interleaved FEC packets, respectively.

6.3.1.2. Using bitmasks

When using fixed size bitmasks (16-, 48-, 112-bits), the SN base field in the FEC header indicates the lowest sequence number of the source packets that forms the FEC packet. Finally, the bits made by "1" in the bitmask are offsets from the SN base and make up the rest of the packets protected by the FEC packet. The bitmasks are able to represent arbitrary protection patterns, for example, 1-D interleaved, 1-D non-interleaved, 2-D, staircase.

6.3.1.3. Using M and N Offsets

When value of M is non-zero, the 8-bit fields indicate the offset of packets protected by an interleaved (N>0) or non-interleaved (N=0) FEC packet. Using a combination of interleaved and non-interleaved FEC packets can form 2-D protection patterns.
Mathematically, for any received repair packet, p*, we can determine the sequence numbers of the source packets that are protected by this repair packet as follows:

When N = 0:
\[ p*_\text{snb}, p*_\text{snb}+1, \ldots, p*_\text{snb}+(M-1), p*_\text{snb}+M \]

When N > 0:
\[ p*_\text{snb}, p*_\text{snb}+(Mx1), p*_\text{snb}+(Mx2), \ldots, p*_\text{snb}+(Mx(N-1)), p*_\text{snb}+(MxN) \]

6.3.2. Recovering the RTP Header

For a given set T, the procedure for the recovery of the RTP header of the missing packet, whose sequence number is denoted by SEQNUM, is as follows:

1. For each of the source packets that are successfully received in T, compute the 80-bit string by concatenating the first 64 bits of their RTP header and the unsigned network-ordered 16-bit representation of their length in bytes minus 12.

2. For the repair packet in T, compute the FEC bit string from the first 80 bits of the FEC header.

3. Calculate the recovered bit string as the XOR of the bit strings generated from all source packets in T and the FEC bit string generated from the repair packet in T.

4. Create a new packet with the standard 12-byte RTP header and no payload.

5. Set the version of the new packet to 2. Skip the first 2 bits in the recovered bit string.

6. Set the Padding bit in the new packet to the next bit in the recovered bit string.

7. Set the Extension bit in the new packet to the next bit in the recovered bit string.

8. Set the CC field to the next 4 bits in the recovered bit string.

9. Set the Marker bit in the new packet to the next bit in the recovered bit string.

10. Set the Payload type in the new packet to the next 7 bits in the recovered bit string.
11. Set the SN field in the new packet to SEQNUM. Skip the next 16 bits in the recovered bit string.

12. Set the TS field in the new packet to the next 32 bits in the recovered bit string.

13. Take the next 16 bits of the recovered bit string and set the new variable Y to whatever unsigned integer this represents (assuming network order). Convert Y to host order. Y represents the length of the new packet in bytes minus 12 (for the fixed RTP header), i.e., the sum of the lengths of all the following if present: the CSRC list, header extension, RTP payload and RTP padding.

14. Set the SSRC of the new packet to the SSRC of the source RTP stream.

This procedure recovers the header of an RTP packet up to (and including) the SSRC field.

6.3.3. Recovering the RTP Payload

Following the recovery of the RTP header, the procedure for the recovery of the RTP payload is as follows:

1. Append Y bytes to the new packet.

2. For each of the source packets that are successfully received in T, compute the bit string from the Y octets of data starting with the 13th octet of the packet. If any of the bit strings generated from the source packets has a length shorter than Y, pad them to that length. The padding of octet 0 MUST be added at the end of the bit string. Note that the information of the first 8 octets are protected by the FEC header.

3. For the repair packet in T, compute the FEC bit string from the repair packet payload, i.e., the Y octets of data following the FEC header. Note that the FEC header may be 12, 16, 32 octets depending on the length of the bitmask.

4. Calculate the recovered bit string as the XOR of the bit strings generated from all source packets in T and the FEC bit string generated from the repair packet in T.

5. Append the recovered bit string (Y octets) to the new packet generated in Section 6.3.2.
6.3.4. Iterative Decoding Algorithm for the 2-D Parity FEC Protection

In 2-D parity FEC protection, the sender generates both non-interleaved and interleaved FEC packets to combat with the mixed loss patterns (random and bursty). At the receiver side, these FEC packets are used iteratively to overcome the shortcomings of the 1-D non-interleaved/interleaved FEC protection and improve the chances of full error recovery.

The iterative decoding algorithm runs as follows:

1. Set num_recovered_until_this_iteration to zero
2. Set num_recovered_so_far to zero
3. Recover as many source packets as possible by using the non-interleaved FEC packets as outlined in Section 6.3.2 and Section 6.3.3, and increase the value of num_recovered_so_far by the number of recovered source packets.
4. Recover as many source packets as possible by using the interleaved FEC packets as outlined in Section 6.3.2 and Section 6.3.3, and increase the value of num_recovered_so_far by the number of recovered source packets.
5. If num_recovered_so_far > num_recovered_until_this_iteration
   ---num_recovered_until_this_iteration = num_recovered_so_far
   ---Go to step 3
Else
   ---Terminate

The algorithm terminates either when all missing source packets are fully recovered or when there are still remaining missing source packets but the FEC packets are not able to recover any more source packets. For the example scenarios when the 2-D parity FEC protection fails full recovery, refer to Section 1.2. Upon termination, variable num_recovered_so_far has a value equal to the total number of recovered source packets.

Example:

Suppose that the receiver experienced the loss pattern sketched in Figure 13.
The receiver executes the iterative decoding algorithm and recovers source packets #1 and #11 in the first iteration. The resulting pattern is sketched in Figure 14.

Since the if condition holds true, the receiver runs a new iteration. In the second iteration, source packets #2 and #10 are recovered, resulting in a full recovery as sketched in Figure 15.
7. SDP Examples

This section provides two SDP [RFC4566] examples. The examples use the FEC grouping semantics defined in [RFC4756].

7.1. Example SDP for 1-D Parity FEC Protection

In this example, we have one source video stream (ssrc:1234) and one FEC repair stream (ssrc:2345). We form one FEC group with the "a=ssrc-group:FEC-FR 1234 2345" line. The source and repair streams are multiplexed on different SSRCs. The repair window is set to 200 ms.

v=0
o=ali 1122334455 1122334466 IN IP4 fec.example.com
s=1-D Interleaved Parity FEC Example
t=0 0
m=video 30000 RTP/AVP 100 110
c=IN IP4 233.252.0.1/127
a=rtpmap:100 MP2T/90000
a=rtpmap:110 interleaved-parityfec/90000
a=fmtp:110 L:5; D:10; ToP:0; repair-window:200000
a=ssrc:1234
a=ssrc:2345
a=ssrc-group:FEC-FR 1234 2345
7.2. Example SDP for 2-D Parity FEC Protection

In this example, we have one source video stream (ssrc:1234) and two FEC repair streams (ssrc:2345 and ssrc:3456). We form one FEC group with the "a=ssrc-group:FEC-FR 1234 2345 3456" line. The source and repair streams are multiplexed on different SSRCs. The repair window is set to 200 ms.

Note that the sender might be generating two repair flows carrying non-interleaved and interleaved FEC packets, however the receiver might be interested only in the interleaved FEC packets. The receiver can identify the repair flow carrying the desired repair data by checking the payload types associated with each repair flow described in the SDP description.

8. Congestion Control Considerations

FEC is an effective approach to provide applications resiliency against packet losses. However, in networks where the congestion is a major contributor to the packet loss, the potential impacts of using FEC SHOULD be considered carefully before injecting the repair flows into the network. In particular, in bandwidth-limited networks, FEC repair flows may consume most or all of the available bandwidth and consequently may congest the network. In such cases, the applications MUST NOT arbitrarily increase the amount of FEC protection since doing so may lead to a congestion collapse. If desired, stronger FEC protection MAY be applied only after the source rate has been reduced [I-D.singh-rmcat-adaptive-fec].

In a network-friendly implementation, an application SHOULD NOT send/receive FEC repair flows if it knows that sending/receiving those FEC repair flows would not help at all in recovering the missing packets.
However, it MAY still continue to use FEC if considered for bandwidth estimation instead of speculatively probe for additional capacity [Holmer13][Nagy14]. It is RECOMMENDED that the amount of FEC protection is adjusted dynamically based on the packet loss rate observed by the applications.

In multicast scenarios, it may be difficult to optimize the FEC protection per receiver. If there is a large variation among the levels of FEC protection needed by different receivers, it is RECOMMENDED that the sender offers multiple repair flows with different levels of FEC protection and the receivers join the corresponding multicast sessions to receive the repair flow(s) that is best for them.

Editor’s note: Additional congestion control considerations regarding the use of 2-D parity codes should be added here.

9. Security Considerations

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 may 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 may exist.

10. IANA Considerations

New media subtypes are subject to IANA registration. For the registration of the payload formats and their parameters introduced in this document, refer to Section 5.
11. Acknowledgments

Some parts of this document are borrowed from [RFC5109]. Thus, the
author would like to thank the editor of [RFC5109] and those who
ccontributed to [RFC5109].

12. Change Log

Note to the RFC-Editor: please remove this section prior to
publication as an RFC.

12.1. draft-ietf-payload-flexible-fec-scheme-00

Initial WG version, based on draft-singh-payload-1d2d-parity-scheme-
00.

12.2. draft-singh-payload-1d2d-parity-scheme-00

This is the initial version, which is based on draft-ietf-fecframe-
1d2d-parity-scheme-00. The following are the major changes compared
to that document:

- Updated packet format with 16-, 48-, 112- bitmask.
- Updated the sections on: repair packet construction, source packet
  construction.
- Updated the media type registration and aligned to RFC6838.

12.3. draft-ietf-fecframe-1d2d-parity-scheme-00

- Some details were added regarding the use of CNAME field.
- Offer-Answer and Declarative Considerations sections have been
  completed.
- Security Considerations section has been completed.
- The timestamp field definition has changed.

13. References

13.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
13.2. Informative References

[Holmer13]

[I-D.singh-rmcat-adaptive-fec]

[Nagy14]

[RFC2326]


Authors’ Addresses

Varun Singh  
Aalto University  
Espoo, FIN  
Finland  

Email: varun@comnet.tkk.fi

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

Email: abegen@cisco.com
Abstract

This memo describes an RTP payload format for the VP9 video codec. The payload format has wide applicability, as it supports applications from low bit-rate peer-to-peer usage, to high bit-rate video conferences. It includes provisions for temporal and spatial scalability.

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 September 10, 2015.

Copyright Notice

Copyright (c) 2015 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 ............................................. 2
2. Conventions, Definitions and Acronyms ..................... 2
3. Media Format Description .................................. 3
4. Payload Format ........................................... 4
   4.1. RTP Header Usage ................................... 4
   4.2. VP9 Payload Description ............................... 6
       4.2.1. Scalability Structure (SS): ....................... 10
   4.3. VP9 Payload Header ................................... 12
   4.4. Frame Fragmentation ................................... 12
   4.5. Examples of VP9 RTP Stream ........................... 12
5. Using VP9 with RPSI and SLI Feedback ....................... 12
   5.1. RPSI ................................................. 12
   5.2. SLI .................................................. 13
   5.3. Example ............................................... 13
6. Payload Format Parameters .................................. 15
   6.1. Media Type Definition ................................ 15
   6.2. SDP Parameters ....................................... 17
       6.2.1. Mapping of Media Subtype Parameters to SDP .... 17
       6.2.2. Offer/Answer Considerations ...................... 17
7. Security Considerations .................................... 17
8. Congestion Control ........................................ 18
9. IANA Considerations ....................................... 18
10. References ............................................... 18
Authors’ Addresses .......................................... 19

1. Introduction

This memo describes an RTP payload specification applicable to the transmission of video streams encoded using the VP9 video codec [I-D.grange-vp9-bitstream]. The format described in this document can be used both in peer-to-peer and video conferencing applications.

TODO: VP9 description. Please see [I-D.grange-vp9-bitstream].

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 Description

The VP9 codec can maintain up to eight reference frames, of which up to three can be referenced or updated by any new frame.

VP9 also allows a reference frame to be resampled and used as a reference for another frame of a different resolution. This allows internal resolution changes without requiring the use of key frames.

These features together enable an encoder to implement various forms of coarse-grained scalability, including temporal, spatial and quality scalability modes, as well as combinations of these, without the need for explicit scalable coding tools.

Temporal layers define different frame rates of video; spatial and quality layers define different and possibly dependent representations of a single input frame. Spatial layers allow a frame to be encoded at different resolutions, whereas quality layers allow a frame to be encoded at the same resolution but at different qualities (and thus with different amounts of coding error). VP9 supports quality layers as spatial layers without any resolution changes; hereinafter, the term "spatial layer" is used to represent both spatial and quality layers.

This payload format specification defines how such temporal and spatial scalability layers can be described and communicated.

Layers are designed (and MUST be encoded) such that if any layer, and all higher layers, are removed from the bitstream along any of the two dimensions, the remaining bitstream is still correctly decodable.

For terminology, this document uses the term "layer frame" to refer to a single encoded VP9 frame for a particular resolution/quality, and "super frame" to refer to all the representations (layer frames) at a single instant in time. A super frame thus consists of one or more layer frames, encoding different spatial layers.

Within a super frame, a layer frame with spatial layer ID equal to S, where S > 0, can depend on a frame with a lower spatial layer ID. This "inter-layer" dependency results in additional coding gain to the traditional "inter-picture" dependency, where a frame depends on previously coded frame in time. For simplicity, this payload format assumes that, within a super frame if inter-layer dependency is used, a spatial layer S frame can only depend on spatial layer S-1 frame when S > 0. Additionally, if inter-picture dependency is used, spatial layer S frame is assumed to only depend on previously coded spatial layer S-1 frame.
TODO: Describe how simulcast can be supported?

Given above simplifications for inter-layer and inter-picture dependencies, a flag (the D bit described below) is used to indicate whether a spatial layer S frame depends on spatial layer S-1 frame. Then a receiver only needs to know the inter-picture dependency structure for a given spatial layer frame in order to determine its decodability. Two modes of describing the inter-picture dependency structure are possible: "flexible mode" and "non-flexible mode". An encoder can only switch between the two on the very first packet of a key frame with temporal layer ID equal to 0.

In flexible mode, each packet can contain up to 3 reference indices, which identifies all frames referenced by the frame transmitted in the current packet for inter-picture prediction. This (along with the D bit) enables a receiver to identify if a frame is decodable or not and helps it understand the temporal layer structure so that it can drop packets as it sees fit. Since this is signaled in each packet it makes it possible to have very flexible temporal layer hierarchies and patterns which are changing dynamically.

In non-flexible mode, the inter-picture dependency (the reference indices) of a group of frames (GOF) MUST be pre-specified as part of the scalability structure (SS) data. In this mode, each packet will have an index to refer to one of the described frames, from which the frames referenced by the frame transmitted in the current packet for inter-picture prediction can be identified.

The SS data can also be used to specify the resolution of each spatial layer present in the VP9 stream.

4. Payload Format

This section describes how the encoded VP9 bitstream is encapsulated in RTP. To handle network losses usage of RTP/AVPF [RFC4585] is RECOMMENDED. All integer fields in the specifications are encoded as unsigned integers in network octet order.

4.1. RTP Header Usage
The general RTP payload format for VP9 is depicted below.

```
+---+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|X|  CC   |M|     PT      |       sequence number         |
+---+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           timestamp                            |
+---+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           synchronization source (SSRC) identifier            |
|                                                      ...
|                                   VP9 payload descriptor (integer #bytes) |
|                                                      :
|                                                      : VP9 pyld hdr |
|                                                      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                      : Bytes 2..N of VP9 payload |
|                                                      : OPTIONAL RTP padding |
```

The VP9 payload descriptor and VP9 payload header will be described in the next section. OPTIONAL RTP padding MUST NOT be included unless the P bit is set.

**Figure 1**

**Marker bit (M):** MUST be set to 1 for the final packet of the highest spatial layer frame (the final packet of the super frame), and 0 otherwise. Unless spatial scalability is in use for this super frame, this will have the same value as the E bit described below. Note that a MANE MUST set this value to 1 for the target spatial layer frame when shaping out higher spatial layers.

**Timestamp:** The RTP timestamp indicates the time when the input frame was sampled, at a clock rate of 90 kHz. If the input frame is encoded with multiple layer frames, all of the layer frames of the super frame MUST have the same timestamp.

**Sequence number:** The sequence numbers are monotonically increasing in order of the encoded bitstream.
The remaining RTP header fields are used as specified in [RFC3550].

4.2. VP9 Payload Description

In flexible mode (with the F bit below set to 1), the first octets after the RTP header are the VP9 payload descriptor, with the following structure.

```
0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
 |I|P|L|F|B|E|V|-| (REQUIRED)
 +-+-+-+-+-+-+-+-+
 I: |M| PICTURE ID | (RECOMMENDED)
    +----------------
 M: | EXTENDED PID | (RECOMMENDED)
    +----------------
 L: | T |U| S |D | (CONDITIONALLY RECOMMENDED)
    +-----------------
 P,F: | P_DIFF |X|N | (CONDITIONALLY RECOMMENDED) .
       +----------------- . - up to 3 times
 X: |EXTENDED P_DIFF| (OPTIONAL)
    +----------------
 V: | SS            |
    +----------------
```

Figure 2
In non-flexible mode (with the F bit below set to 0), the first octets after the RTP header are the VP9 payload descriptor, with the following structure.

```
0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|I|P|L|F|B|E|V|-| (REQUIRED)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
I: |M| PICTURE ID | (RECOMMENDED)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
M: | EXTENDED PID | (RECOMMENDED)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
L: |GOF_IDX| S |D| (CONDITIONALLY RECOMMENDED)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| TL0PICIDX | (CONDITIONALLY REQUIRED)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
V: | SS            |
| ..            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Figure 3

**I:** Picture ID (PID) present. When set to one, the OPTIONAL PID MUST be present after the mandatory first octet and specified as below. Otherwise, PID MUST NOT be present.

**P:** Inter-picture predicted layer frame. When set to zero, the layer frame does not utilize inter-picture prediction. In this case, up-switching to current spatial layer’s frame is possible from directly lower spatial layer frame. P SHOULD also be set to zero when encoding a layer synchronization frame in response to an LRR [I-D.lennox-avtext-lrr].

**L:** Layer indices present. When set to one, the one or two octets following the mandatory first octet and the PID (if present) is as described by "Layer indices" below. If the F bit (described below) is set to 1 (indicating flexible mode), then only one octet is present for the layer indices. Otherwise if the F bit is set to 0 (indicating non-flexible mode), then two octets are present for the layer indices.

**F:** Flexible mode. F set to one indicates flexible mode and if the P bit is also set to one, then the octets following the mandatory first octet, the PID, and layer indices (if present) are as described by "Reference indices" below. This MUST only be set to one if the I bit is also set to one; if the I bit is set to zero, then this MUST also be set to zero and ignored by receivers. The
value of this F bit CAN ONLY CHANGE on the very first packet of a
key picture. This is a packet with the P bit equal to zero, S or
D bit (described below) equal to zero, B bit (described below)
equal to 1, and temporal layer ID equal to 0.

B: Start of a layer frame. MUST be set to 1 if the first payload
octet of the RTP packet is the beginning of a new VP9 layer frame,
and MUST NOT be 1 otherwise. Note that this layer frame might not
be the very first layer frame of a super frame.

E: End of a layer frame. MUST be set to 1 for the final RTP packet
of a VP9 layer frame, and 0 otherwise. This enables a decoder to
finish decoding the layer frame, where it otherwise may need to
wait for the next packet to explicitly know that the layer frame
is complete. Note that, if spatial scalability is in use, more
layer frames from the same super frame may follow; see the
description of the M bit above.

V: Scalability structure (SS) data present. When set to one, the
OPTIONAL SS data MUST be present in the payload descriptor.
Otherwise, the SS data MUST NOT be present.

-: Bit reserved for future use. MUST be set to zero and MUST be
ignored by the receiver.

The mandatory first octet is followed by the extension data fields
that are enabled:

M: The most significant bit of the first octet is an extension flag.
The field MUST be present if the I bit is equal to one. If set,
the PID field MUST contain 15 bits; otherwise, it MUST contain 7
bits. See PID below.

Picture ID (PID): Picture ID represented in 7 or 15 bits, depending
on the M bit. This is a running index of the pictures. The field
MUST be present if the I bit is equal to one. If M is set to
zero, 7 bits carry the PID; else if M is set to one, 15 bits carry
the PID. The sender may choose between 7 or 15 bits index. The
PID SHOULD start on a random number, and MUST wrap after reaching
the maximum ID. The receiver MUST NOT assume that the number of
bits in PID stay the same through the session.

Layer indices: This information is optional but recommended whenever
encoding with layers. In the flexible mode (when the F bit is set
to 1), one octet is used to specify a layer frame’s temporal layer
ID (T) and spatial layer ID (S) as shown in Figure 2.
Additionally, a bit (U) is used to indicate that the current frame
is a "switching up point" frame. Another bit (D) is used to
indicate whether inter-layer prediction is used for the current layer frame.

In the non-flexible mode (when the F bit is set to 0), two octets are used as depicted in Figure 3. Like the flexible mode, the first byte contains the spatial layer ID and the D bit. Unlike the flexible mode, instead of the T and U fields, a group of frames index (GOF_IDX) is specified, which can be used to obtain the values of T and U fields from the scalable structure (SS) data described below. An additional octet to represent the temporal layer 0 index, TL0PICIDX, is present so that all minimally required frames can be tracked.

The T and S fields, whether obtained directly or indirectly from the SS data, indicate the temporal and spatial layers and can help MCUs measure bitrates per layer and can help them make a quick decision on whether to relay a packet or not. They can also help receivers determine what layers they are currently decoding.

T: The temporal layer ID of current frame. This field is only present in the flexible mode (F = 1).

U: Switching up point. This bit is only present in the flexible mode (F = 1). If this bit is set to 1 for the current frame with temporal layer ID equal to T, then "switch up" to a higher frame rate is possible as subsequent higher temporal layer frames will not depend on any frame before the current frame (in coding time) with temporal layer ID greater than T.

S: The spatial layer ID of current frame. Note that frames with spatial layer S > 0 may be dependent on decoded spatial layer S-1 frame within the same super frame.

D: Inter-layer dependency used. MUST be set to one if current spatial layer S frame depends on spatial layer S-1 frame of the same super frame. MUST only be set to zero if current spatial layer S frame does not depend on spatial layer S-1 frame of the same super frame. For the base layer frame with S equal to 0, this D bit MUST be set to zero.

GOF_IDX: An index to a frame in the group of frames (GOF) described by the SS data. This field is only present in the non-flexible mode (F = 0). In this mode, the SS data SHOULD have been received and the temporal characteristics of each frame must have been specified as group of frames in the SS data (see the description of "Scalability structure" below). Here, the values of the T and the U fields are derived from the SS data. Additionally, the frame’s inter-picture dependency can
also be obtained from the SS data. In the case no SS data has been received or the received SS data does not specify GOF (N_G is set to 0), then GOF_IDX MUST be ignored and the stream is assumed to have no temporal hierarchy with both T and U equal to 0.

TL0PICIDX: 8 bits temporal layer zero index. TL0PICIDX is only present in the non-flexible mode (F = 0). This is a running index for the temporal base layer frames, i.e., the frames with temporal layer ID (TID) set to 0. If TID is larger than 0, TL0PICIDX indicates which temporal base layer frame the current frame depends on. TL0PICIDX MUST be incremented when TID is 0. The index SHOULD start on a random number, and MUST restart at 0 after reaching the maximum number 255.

Reference indices: These bytes are optional, but recommended when encoding with temporal layers in the flexible mode. When P and F are both set to one, then at least one reference index has to be specified as below. Additional reference indices (total of up to 3 reference indices are allowed) may be specified using the N bit below. When either P or F is set to zero, then no reference index is specified.

P_DIFF: The reference index specified as the relative PID from the current frame. For example, when P_DIFF=3 on a packet containing the frame with PID 112 means that the frame refers back to the frame with PID 109. This calculation is done modulo the size of the PID field, i.e., either 7 or 15 bits. For most layer structures a 6-bit relative PID will be enough; however, the X bit can be used to refer to older frames.

X: 1 if this layer index has an extended P_DIFF.

N: 1 if there is additional P_DIFF following the current P_DIFF.

4.2.1. Scalability Structure (SS):

The scalability structure (SS) data describes the resolution of each layer frame within a super frame as well as the inter-picture dependencies for a group of frames (GOF). If the VP9 payload descriptor’s "V" bit is set, the SS data is present in the position indicated in Figure 2 and Figure 3.
V: | N_S | Y | N_G |
---|---|---|---|
Y: | WIDTH | (OPTIONAL) |
---|---|---|---|
N_G: | T | U | R | (-/-) | (OPTIONAL) |
---|---|---|---|---|---|---|---|---|

Figure 4

N_S: N_S + 1 indicates the number of spatial layers present in the VP9 stream.

Y: Each spatial layer’s frame resolution present. When set to one, the OPTIONAL WIDTH (2 octets) and HEIGHT (2 octets) MUST be present for each layer frame. Otherwise, the resolution MUST NOT be present.

N_G: N_G + 1 indicates the number of frames in a GOF. If N_G is greater than 0, then the SS data allows the inter-picture dependency structure of the VP9 stream to be pre-declared, rather than indicating it on the fly with every packet. If N_G is greater than 0, then for N_G + 1 pictures in the GOF, each frame’s temporal layer ID (T), switch up point (U), and the R reference indices (P_DIFFs) are specified.

N_G=0 indicates that either there is only one temporal layer or no fixed inter-picture dependency information is present going forward in the bitstream.

Note that for a given super frame, all layer frames follow the same inter-picture dependency structure. However, the frame rate of each spatial layer can be different from each other and this can be controlled with the use of the D bit described above. The specified dependency structure in the SS data MUST be for the highest frame rate layer.

In a scalable stream sent with a fixed pattern, the SS data SHOULD be included in the first packet of every key frame. This is a packet with P bit equal to zero, S or D bit equal to zero, B bit equal to 1,
and temporal layer ID (TID) equal to 0. The SS data SHOULD also be included in the first packet of the first frame in which the SS changes. If the SS data is included in a frame with TID not equal to 0, it MUST also be repeated in the first packet of the first frame with a lower TID, until TID equals to 0.

4.3. VP9 Payload Header

TODO: need to describe VP9 payload header.

4.4. Frame Fragmentation

VP9 frames are fragmented into packets, in RTP sequence number order, beginning with a packet with the B bit set, and ending with a packet with the RTP marker bit set. There is no mechanism for finer-grained access to parts of a VP9 frame.

4.5. Examples of VP9 RTP Stream

TODO

5. Using VP9 with RPSI and SLI Feedback

The VP9 payload descriptor defined in Section 4.2 above contains an optional PictureID parameter. One use of this parameter is included to enable use of reference picture selection index (RPSI) and slice loss indication (SLI), both defined in [RFC4585].

5.1. RPSI

TODO: Update to indicate which frame within the picture.

The reference picture selection index is a payload-specific feedback message defined within the RTCP-based feedback format. The RPSI message is generated by a receiver and can be used in two ways. Either it can signal a preferred reference picture when a loss has been detected by the decoder -- preferably then a reference that the decoder knows is perfect -- or, it can be used as positive feedback information to acknowledge correct decoding of certain reference pictures. The positive feedback method is useful for VP9 used as unicast. The use of RPSI for VP9 is preferably combined with a special update pattern of the codec’s two special reference frames -- the golden frame and the altref frame -- in which they are updated in an alternating leapfrog fashion. When a receiver has received and correctly decoded a golden or altref frame, and that frame had a PictureID in the payload descriptor, the receiver can acknowledge this simply by sending an RPSI message back to the sender. The
message body (i.e., the "native RPSI bit string" in [RFC4585]) is simply the PictureID of the received frame.

5.2. SLI

TODO: Update to indicate which frame within the picture.

The slice loss indication is another payload-specific feedback message defined within the RTCP-based feedback format. The SLI message is generated by the receiver when a loss or corruption is detected in a frame. The format of the SLI message is as follows [RFC4585]:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         First           |        Number           | PictureID |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5

Here, First is the macroblock address (in scan order) of the first lost block and Number is the number of lost blocks. PictureID is the six least significant bits of the codec-specific picture identifier in which the loss or corruption has occurred. For VP9, this codec-specific identifier is naturally the PictureID of the current frame, as read from the payload descriptor. If the payload descriptor of the current frame does not have a PictureID, the receiver MAY send the last received PictureID+1 in the SLI message. The receiver MAY set the First parameter to 0, and the Number parameter to the total number of macroblocks per frame, even though only parts of the frame is corrupted. When the sender receives an SLI message, it can make use of the knowledge from the latest received RPSI message. Knowing that the last golden or altref frame was successfully received, it can encode the next frame with reference to that established reference.

5.3. Example

TODO: this example is copied from the VP8 payload format specification, and has not been updated for VP9. It may be incorrect.

The use of RPSI and SLI is best illustrated in an example. In this example, the encoder may not update the altref frame until the last sent golden frame has been acknowledged with an RPSI message. If an update is not received within some time, a new golden frame update is
sent instead. Once the new golden frame is established and acknowledged, the same rule applies when updating the altref frame.

<table>
<thead>
<tr>
<th>Event</th>
<th>Sender</th>
<th>Receiver</th>
<th>Established reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>Send golden frame PictureID = 0</td>
<td>Receive and decode golden frame</td>
<td></td>
</tr>
<tr>
<td>1001</td>
<td></td>
<td></td>
<td>Send RPSI(0)</td>
</tr>
<tr>
<td>1002</td>
<td>Receive RPSI(0)</td>
<td></td>
<td>golden</td>
</tr>
<tr>
<td></td>
<td>(sending regular frames)</td>
<td></td>
<td>golden</td>
</tr>
<tr>
<td>1100</td>
<td>Send altref frame PictureID = 100</td>
<td>Altref corrupted or lost</td>
<td>golden</td>
</tr>
<tr>
<td>1101</td>
<td></td>
<td></td>
<td>Send SLI(100)</td>
</tr>
<tr>
<td>1102</td>
<td>Receive SLI(100)</td>
<td></td>
<td>golden</td>
</tr>
<tr>
<td>1103</td>
<td>Send frame with reference to golden</td>
<td>Receive and decode golden frame (decoder state restored)</td>
<td>golden</td>
</tr>
<tr>
<td></td>
<td>(sending regular frames)</td>
<td></td>
<td>golden</td>
</tr>
<tr>
<td>1200</td>
<td>Send altref frame PictureID = 200</td>
<td>Receive and decode altref frame</td>
<td>golden</td>
</tr>
<tr>
<td>1201</td>
<td></td>
<td></td>
<td>Send RPSI(200)</td>
</tr>
</tbody>
</table>
Table 1: Example signaling between sender and receiver

Note that the scheme is robust to loss of the feedback messages. If the RPSI is lost, the sender will try to update the golden (or altref) again after a while, without releasing the established reference. Also, if an SLI is lost, the receiver can keep sending SLI messages at any interval allowed by the RTCP sending timing restrictions as specified in [RFC4585], as long as the picture is corrupted.

6. Payload Format Parameters

This payload format has two required parameters.

6.1. Media Type Definition

This registration is done using the template defined in [RFC6838] and following [RFC4855].

Type name: video
Subtype name: VP9

Required parameters:
These parameters MUST be used to signal the capabilities of a receiver implementation. These parameters MUST NOT be used for any other purpose.

max-fr: The value of max-fr is an integer indicating the maximum frame rate in units of frames per second that the decoder is capable of decoding.

max-fs: The value of max-fs is an integer indicating the maximum frame size in units of macroblocks that the decoder is capable of decoding.

The decoder is capable of decoding this frame size as long as the width and height of the frame in macroblocks are less than \( \text{int}(\sqrt{\text{max-fs} \times 8}) \) - for instance, a max-fs of 1200 (capable of supporting 640x480 resolution) will support widths and heights up to 1552 pixels (97 macroblocks).

Optional parameters: none

Encoding considerations:
This media type is framed in RTP and contains binary data; see Section 4.8 of [RFC6838].

Security considerations: See Section 7 of RFC xxxx.
[RFC Editor: Upon publication as an RFC, please replace "XXXX" with the number assigned to this document and remove this note.]

Interoperability considerations: None.

Published specification: VP9 bitstream format
[I-D.grange-vp9-bitstream] and RFC XXXX.
[RFC Editor: Upon publication as an RFC, please replace "XXXX" with the number assigned to this document and remove this note.]

Applications which use this media type:
For example: Video over IP, video conferencing.

Additional information: None.

Person & email address to contact for further information:
TODO [Pick a contact]

Intended usage: COMMON
Restrictions on usage:
This media type depends on RTP framing, and hence is only defined for transfer via RTP [RFC3550].

Author: TODO [Pick a contact]

Change controller:
IETF Payload Working Group delegated from the IESG.

6.2. SDP Parameters

The receiver MUST ignore any fmtp parameter unspecified in this memo.

6.2.1. Mapping of Media Subtype Parameters to SDP

The media type video/VP9 string is mapped to fields in the Session Description Protocol (SDP) [RFC4566] as follows:

- The media name in the "m=" line of SDP MUST be video.
- The encoding name in the "a=rtpmap" line of SDP MUST be VP9 (the media subtype).
- The clock rate in the "a=rtpmap" line MUST be 90000.
- The parameters "max-fs", and "max-fr", MUST be included in the "a=fmtp" line of SDP. These parameters are expressed as a media subtype string, in the form of a semicolon separated list of parameter=value pairs.

6.2.1.1. Example

An example of media representation in SDP is as follows:

```
m=video 49170 RTP/AVPF 98
a=rtpmap:98 VP9/90000
a=fmtp:98 max-fr=30; max-fs=3600;
```

6.2.2. Offer/Answer Considerations

TODO: Update this for VP9

7. Security Considerations

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 encryption of the RTP payload. Integrity of the RTP packets through suitable cryptographic integrity protection mechanism. Cryptographic system may 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 or not. Note that the appropriate mechanism to provide security to RTP and payloads following this memo may vary. It is dependent on the application, the transport, and the signaling protocol employed. Therefore a single mechanism is not sufficient, although if suitable the usage of SRTP [RFC3711] is recommended. This RTP payload format and its media decoder do not exhibit any significant non-uniformity in the receiver-side computational complexity for packet processing, and thus are unlikely to pose a denial-of-service threat due to the receipt of pathological data. Nor does the RTP payload format contain any active content.

8. Congestion Control

Congestion control for RTP SHALL be used in accordance with RFC 3550 [RFC3550], and with any applicable RTP profile; e.g., RFC 3551 [RFC3551]. The congestion control mechanism can, in a real-time encoding scenario, adapt the transmission rate by instructing the encoder to encode at a certain target rate. Media aware network elements MAY use the information in the VP9 payload descriptor in Section 4.2 to identify non-reference frames and discard them in order to reduce network congestion. Note that discarding of non-reference frames cannot be done if the stream is encrypted (because the non-reference marker is encrypted).

9. IANA Considerations

The IANA is requested to register the following values:
- Media type registration as described in Section 6.1.

10. References

[I-D.grange-vp9-bitstream]
Grange, A. and H. Alvestrand, "A VP9 Bitstream Overview",
draft-grange-vp9-bitstream-00 (work in progress), February 2013.

[I-D.lennox-avtext-lrr]
draft-lennox-avtext-lrr-00 (work in progress), March 2015.


Authors’ Addresses

Justin Uberti
Google, Inc.
747 6th Street South
Kirkland, WA  98033
USA

Email: justin@uberti.name

Stefan Holmer
Google, Inc.
Kungsbron 2
Stockholm  111 22
Sweden