

INTERNET-DRAFT
Intended Status: Standard Track

R. Huang
Huawei
October 19, 2015

RTP Payload Format for Interleaved Packets
draft-huang-payload-rtp-interleave-01

Abstract

This memo introduces a common RTP encapsulation for interleaved media. This method can be applied to any RTP payload formats for any applications when latency is not an issue.

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/lid-abstracts.html>

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

Copyright and License 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	3
2	Terminology	3
3	Interleaving	3
4	Interleaving RTP Payload Format	5
4.1	RTP Header	5
4.2	Payload Structure	6
4.3	Encapsulation frames	7
4.4	Example Packet	8
5	IANA Considerations.	10
5.1	Registration of audio/genitl	10
5.2	Registration of video/genitl	11
5.3	Usage of Interleaving	12
6	Security Considerations	12
7	Acknowledgments	12
8	References	13
8.1	Normative References	13
8.2	Informative References	13
	Authors' Addresses	13

1 Introduction

Interleaving is an effective method to disperse packet loss bursts into a series of isolated small losses which in general are easier to recover from and produce lower total distortion. It provides the advantages of requiring no increase in bit rate and can be combined with other types of error-resilience techniques like FEC, but at the cost of increasing latency. Interleaving is quite useful for applications which are not such sensitive of latency in network environments afflicted by fading an interference which may lead to burst losses, e.g., DSL and wireless network.

This memo introduces a common RTP encapsulation for interleaved media. This method can be applied to any RTP payload formats for any applications when latency is not an issue.

2 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].

This document uses the following terms:

Interleaving length: the packet number of an interleaving separation between packets or data originally adjacent.

Interleaving depth: the interleaving separation count of an interleaver output buffer. Usually an interleaver output buffer size is equal to interleaving length*interleaving depth.

3. Interleaving

An interleaver is simply used at the sender or at a middle box to interleave the RTP packets before transmission through the network. There are a lot of interleaving algorithms. A simple one could be that packets are firstly read into the interleaver in rows, with each row corresponding to a sequence block of n packets; and packets are transmitted by columns as soon as m rows of packets fill up. The interleaver permutes the packets so that the location of burst losses are converted into isolated ones. The effective ness of the interleaver depends on the interleaving length and interleaving depth, however, this is at the cost of latency. At the receiver, an interleaved packet cannot be used until all the packets it depends on are received.

Figure 1 shows the interleaving scheme with $n=4$ and $m=3$, where m indicates the interleaving depth and n indicates the interleaving

length. For this interleaver, the i -th packet in the original order has to be transmitted in the $((i-1) \bmod n) * m + ((i-1) \div n) + 1$ -th place. The highest decoding delay corresponding to this interleave is $(n-1) * (m-1)$.

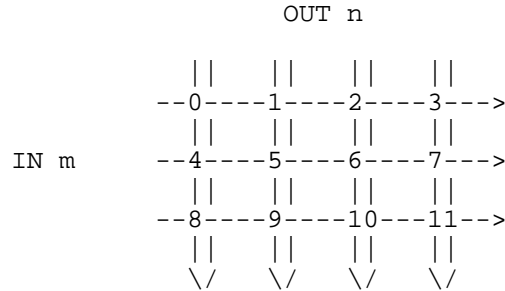
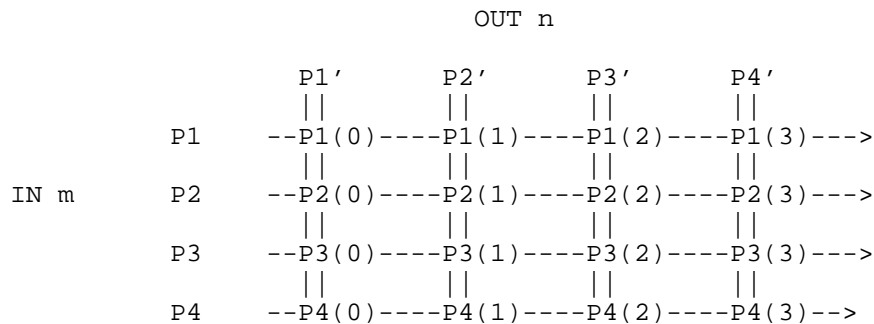


Figure 1 Packet interleaving with interleaving length $n=4$ and depth $m=3$

This kind of interleaving, called packet leaving in this document, is useful for audio applications that are non-interactive because it would turn a burst of consecutive lost packets into a series of isolated packet loss events. However, for video frames that are usually large enough to be fragmented into several packets, losing several non-consecutive packets from a large frame may not lead to perceptually less degradation. Thus, some implementations may use another interleaving to help loss recovery: The payload of each RTP packet is divided into n parts and the interleaver combines some part of one packet with some part of other packets to form a new RTP packet. As Figure 2 shows, each of the original RTP packets $P1$, $P2$, $P3$ and $P4$ are divided into 4 packets. The interleaver combines the first parts of $P1$, $P2$, $P3$ and $P4$ together to form a new packet $P1'$. $P2'$, $P3'$ and $P4'$ are formed in the same way. In this case, if $P1'$ is lost during the transmission, $P1$, $P2$, $P3$ and $P4$ can be easily recovered by FEC mechanism. It is specified as data interleaving in this document. In data interleaving, the number of separations of one packet has a fix value. To simplify the mechanism, it is required to be equal to the interleaving depth in the document.



\ / \ / \ / \ /
Figure 2 An illustration of data interleaving

However, data interleaving method introduces extra complexity when dividing and recovering packets. To reduce delay in some degree, some ways can be considered. For example, the interleaver has the ability to categorize the RTP packets into important ones and unimportant ones based on some application dependent rules, and then only applies this interleaving to the important packets, e.g., packets containing I frames, IDR frames or some other information frames, while leave unimportant ones intact. This method can achieve a compromise between reducing complexity caused delay and reducing burst losses. The detail discussion of how to reduce the interleaving delay is out of scope of this document.

Proposing a common interleaving RTP encapsulation allows any RTP payload format to use the interleaving scheme freely when needed.

4. Interleaving RTP Payload Format

This section introduces a new PTP payload format dedicated for interleaving. That means interleaved RTP packets could use this new RTP payload format when transmitting. It is allowed that interleaving RTP payload format is transmitted together with the uninterleaved payload format so that the de-interleaver can identify the interleaved packets to recover. In such a case, the specifications defined in [draft-ietf-avtcore-multi-media-rtp-session] SHOULD be considered. Interleaving can change the number of RTP packets, however the original RTP packets MUST be recovered at the de-interleaver.

4.1 RTP Header

The format of RTP header is specified in [RFC3550] and showed in Figure 2 for convenience.

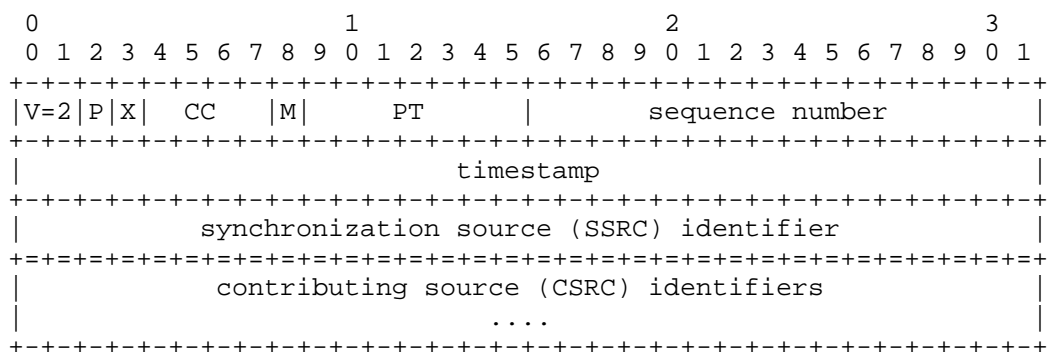


Figure 2 RTP header of RFC3550

The RTP header information to be set according to this RTP payload format is set as follows:

marker (M): 1 bit

The interpretation of the marker is determined by the interleaved frame encapsulated in this payload. It MUST be set to the value that the market bit of the frame would have been if it were transported in its own RTP packets. If multiple frames are packed into one RTP packet, the marker bit in the RTP header MUST be set to the value in accordance with the last frame.

payload type (PT): 7 bits

The assignment of the RTP payload type for this format is outside the scope of this memo and will not be specified here. The assignment of a payload type has to be performed either through the profile used or in a dynamic way. However, only the packets of the same one RTP stream are allowed to interleaved in one interleaving stream which is identified by the payload type.

sequence number: 16 bits

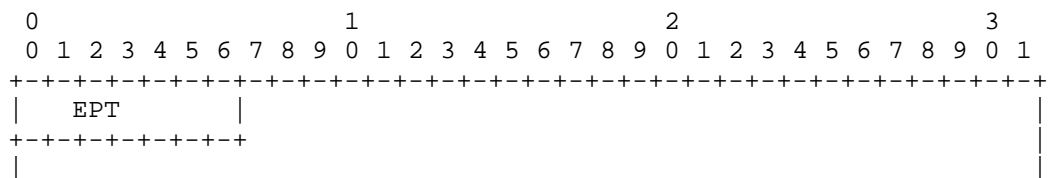
Set and used in accordance with RFC 3550.

timestamp: 32 bits

Set to the value that the timestamp of the frame would have been if it were transported in its own RTP packets. If multiple frames are packed into one RTP packet, the timestamp in the RTP header MUST be set to the timestamp of the last frame which would have been if it were transported in its own RTP packets.

4.2 Payload Structure

This section describes the interleaving payload structure. The interleaving payload structure is composed from a fixed interleaving header and one or multiple encapsulation frames, which is illustrated in the following figure.



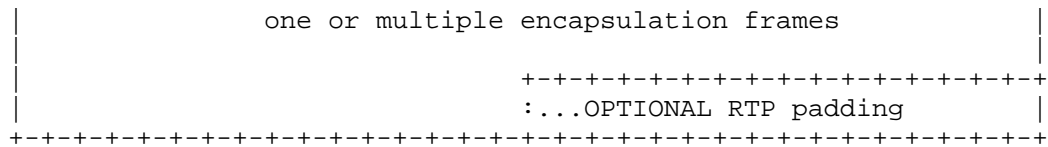


Figure 3: Format for the interleaving payload structure

encapsulation payload type (EPT): 7 bits

This field indicates the payload type of the encapsulation frames would have been if it were transported in their own RTP packets. Encapsulation frames with different payload type MUST NOT be packed into one RTP packet. The assignment of the encapsulation payload type has to be performed either through the profile used or in a dynamic way.

4.3 Encapsulation frames

The structure of encapsulation frames is showed in the figure 4.

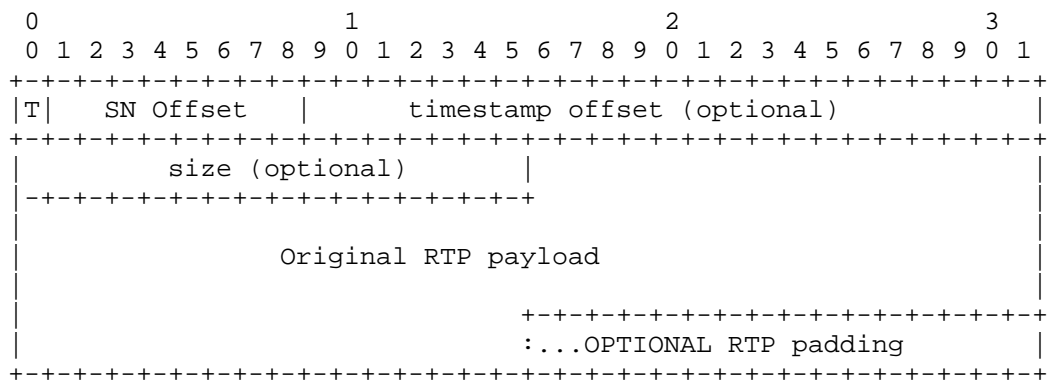


Figure 4: Format for the encapsulation frame structure

encapsulation frame type (T): 1 bit

This field indicates the encapsulation frame type.

T=0: Last encapsulation frame - This frame can be used alone or as the last encapsulation frame in an aggregation RTP packet which combines multiple encapsulation frames.

T=1: Aggregated encapsulation frame - This frame MUST be only used in an aggregation RTP packet combining multiple encapsulation frame. And it MUST NOT be used as the last

encapsulation frame when used in the aggregation RTP packet.

SN Offset: 8 bits

This field records the signed offset between the original sequence number that the encapsulation frame would have been if it were transported in its own RTP packets and the sequence number field in this RTP header. The original sequence number implying the decoding order can be calculated as following:

$$\text{original sequence number} = \text{sequence number} + \text{SN offset}$$

A negative SN offset indicates that the encapsulation frame is delayed for transmission, and a non-negative SN offset means the encapsulation frame is transmitted ahead of it should be.

When one RTP packet is divided into several parts to be interleaved, one part is encapsulated as one encapsulation frame and the different encapsulation frames belonging to the same RTP packet share the same SN offset. If multiple encapsulation frames combined in one RTP packet have the same SN offset, they SHOULD be handled according to their orders arranged in this packet.

timestamp offset: 23 bits

This field records the signed offset between the original timestamp that the encapsulation frame would have been if it were transported in its own RTP packets and the timestamp field in this RTP header. It is only applicable for aggregated encapsulation frame (T=1). The original timestamp can be calculated as following:

$$\text{original timestamp} = \text{timestamp} + \text{timestamp offset}$$

size: 16 bits

This field is an unsigned size information of the following encapsulation frame, which does not include encapsulation frame type, SN offset and timestamp offset. It is only applicable for aggregated encapsulation frame (T=1).

4.4 Example Packet

This section presents two example of an interleaving RTP packet. Figure 5 shows an example of an packet that only contain one encapsulation frame. In this example, the original timestamp is the timestamp in the RTP header.

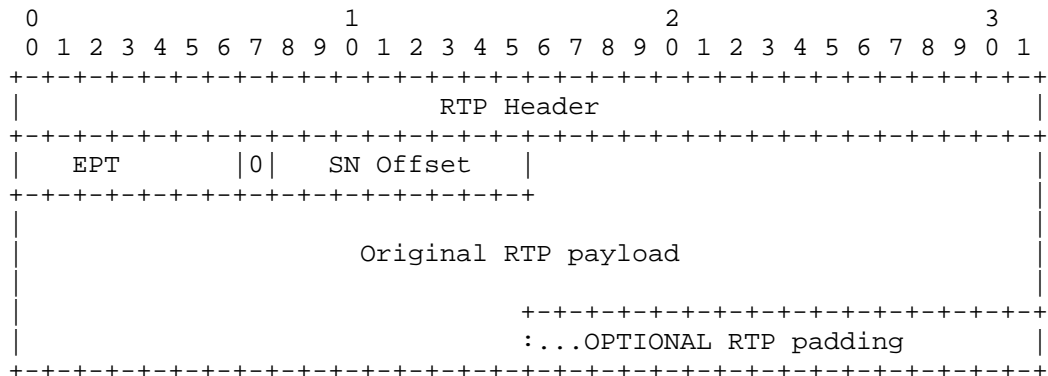


Figure 5 An interleaving RTP packet
including only one encapsulation frame

Figure 6 shows an example of a packet that contains 3 encapsulation frame, labeled as 1, 2 and 3 in the figure.

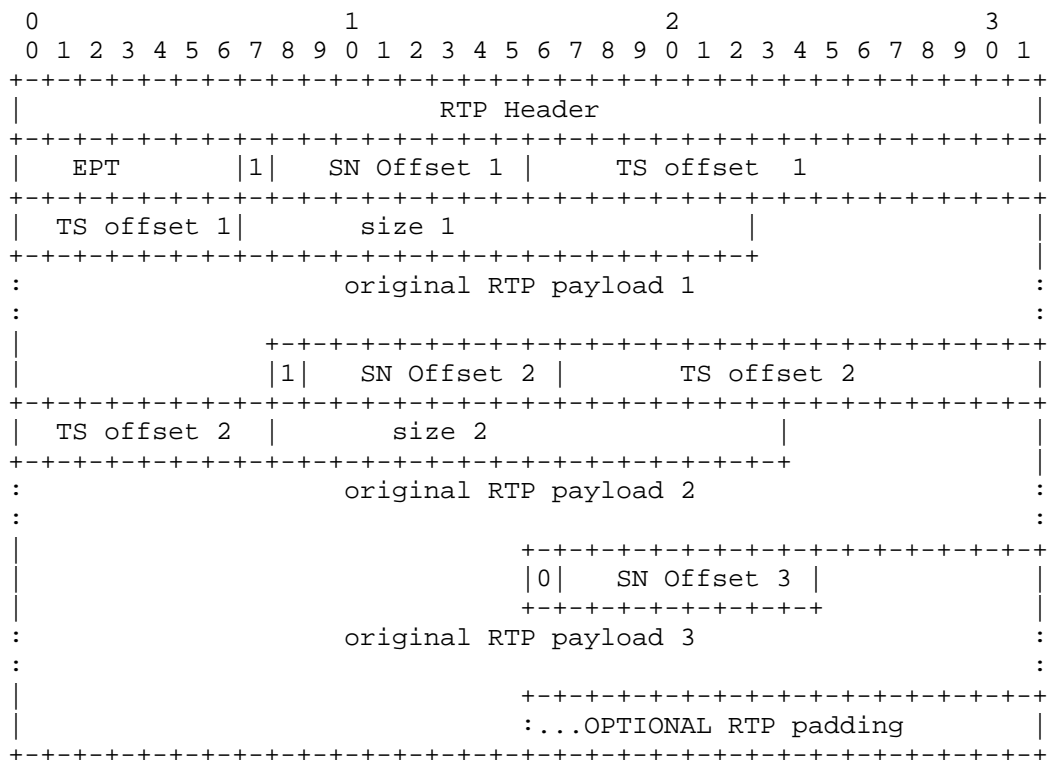


Figure 5 An interleaving RTP packet

including 3 encapsulation frames

5 IANA Considerations.

This section specifies the new media subtypes registered with IANA and the associated parameters that MUST be used to indicate the features of the RTP stream.

5.1 Registration of audio/genitl

The media subtype for audio interleaving payload is allocated from the IETF tree.

The receiver MUST ignore any unrecognized parameter.

Media Type name: audio

Subtype name: genitl

Required parameters:

codec: This parameter indicates the original payload type of the interleaved payload.

length: This parameter indicates the interleaving length of the interleaver.

depth: This parameter indicates the interleaving depth of the interleaver.

Optional parameters:

type: This parameter indicates the interleaving type described in section 3. The permissible values are 0 and 1, where 0 indicates packet interleaving and 1 indicates data interleaving. If omitted, it has the default value of 0.

Encoding considerations: This format is framed

Interoperability considerations: none

Published specification: this document.

Applications that uses this media type:

Additional information: none

Person & email address to contact for further information:

rachel.huang@huawei.com

Intended usage: COMMON

Restrictions on usage: This media type depends on RTP framing, and hence is only defined for transfer via RTP. Transport within other framing protocols SHALL NOT be defined as this is a robustness mechanism for RTP.

Author: Rachel Huang

Change controller: IETF Payload Working Group delegated from the IESG

5.2 Registration of video/genitl

The media subtype for video interleaving payload is allocated from the IETF tree.

The receiver MUST ignore any unrecognized parameter.

Media Type name: video

Subtype name: genitl

Required parameters:

length: This parameter indicates the interleaving length of the interleaver.

depth: This parameter indicates the interleaving depth of the interleaver.

Optional parameters:

type: This parameter indicates the interleaving type described in section 3. The permissible values are 0 and 1, where 0 indicates packet interleaving and 1 indicates data interleaving. If omitted, it has the default value of 0.

Encoding considerations: This format is framed

Interoperability considerations: none

Published specification: this document.

Applications that uses this media type:

Additional information: none

Person & email address to contact for further information:
rachel.huang@huawei.com

Intended usage: COMMON

Restrictions on usage: This media type depends on RTP framing, and hence is only defined for transfer via RTP. Transport within other framing protocols SHALL NOT be defined as this is a robustness mechanism for RTP.

Author: Rachel Huang

Change controller: IETF Payload Working Group delegated from the IESG

5.3 Usage of Interleaving

the interleaving stream can be sent instead of the original stream or multiplexed with the original stream. In the latter case, interleaving packets are sent alternatively with part of the original packets, and the de-interleaver can identify them by different payload types. The SDP example is illustrated as following:

```
m=audio 10000 RTP/AVP 100
a=rtpmap:96 G722/8000
a=rtpmap:100 genintl/8000
a=fmtp:100 96/4/3
```

This SDP indicates that an audio stream 100 is presented. The payload identifier 96 is the original payload type and 100 is the interleaved payload type. So this audio stream is an interleaved audio stream, whose original payload type is G.722 and the corresponding interleaving length and interleaving depth are 4 and 3.

```
m=video 49600 RTP/AVP 100 101
a=rtpmap:100 H264/90000
a=rtpmap:101 genintl/90000
a=fmtp:101 100/4/3
```

This SDP indicates that interleaved RTP packets are sent together with some of the uninterleaved original packets.

6 Security Considerations

TBD

7 Acknowledgments

TBD

8 References

8.1 Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3550] Schulzrinne, H., "RTP: A Transport Protocol for Real-Time Applications", RFC 3550, July 2003.
- [draft-ietf-avtcore-multi-media-rtp-session] Westerlund, M., Perkins, C., and J. Lennox, "Sending Multiple Types of Media in a single RTP Session", draft-ietf-avtcore-multi-media-rtp-session-07, March 2015.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

8.2 Informative References

Authors' Addresses

Rachel Huang
Huawei
101 Software Avenue, Yuhua District
Nanjing 210012
China

Email: rachel.huang@huawei.com

PAYLOAD
Internet-Draft
Intended status: Standards Track
Expires: April 21, 2016

V. Singh
Nemu Dialogue System Oy
A. Begen

M. Zanaty
Cisco
G. Mandyam
Qualcomm Innovation Center
October 19, 2015

RTP Payload Format for Flexible Forward Error Correction (FEC)
draft-ietf-payload-flexible-fec-scheme-01

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 April 21, 2016.

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	3
1.1. Use Cases for 1-D FEC Protection	6
1.2. Use Cases for 2-D Parity FEC Protection	7
1.3. Overhead Computation	9
2. Requirements Notation	9
3. Definitions and Notations	10
3.1. Definitions	10
3.2. Notations	10
4. Packet Formats	10
4.1. Source Packets	10
4.2. Repair Packets	10
5. Payload Format Parameters	14
5.1. Media Type Registration	14
5.1.1. Registration of audio/flexfec	15
5.1.2. Registration of video/flexfec	16
5.1.3. Registration of text/flexfec	17
5.1.4. Registration of application/flexfec	19
5.2. Mapping to SDP Parameters	20
5.2.1. Offer-Answer Model Considerations	21
5.2.2. Declarative Considerations	21
6. Protection and Recovery Procedures	22
6.1. Overview	22
6.2. Repair Packet Construction	22
6.3. Source Packet Reconstruction	24
6.3.1. Associating the Source and Repair Packets	24
6.3.2. Recovering the RTP Header	25
6.3.3. Recovering the RTP Payload	27
6.3.4. Iterative Decoding Algorithm for the 2-D Parity FEC Protection	27

7.	SDP Examples	29
7.1.	Example SDP for Flexible FEC Protection with in-band SSRC mapping	30
7.2.	Example SDP for Flex FEC Protection with explicit signalling in the SDP	30
8.	Congestion Control Considerations	30
9.	Security Considerations	31
10.	IANA Considerations	32
11.	Acknowledgments	32
12.	Change Log	32
12.1.	draft-ietf-payload-flexible-fec-scheme-01	32
12.2.	draft-ietf-payload-flexible-fec-scheme-00	32
12.3.	draft-singh-payload-ld2d-parity-scheme-00	32
12.4.	draft-ietf-fecframe-ld2d-parity-scheme-00	33
13.	References	33
13.1.	Normative References	33
13.2.	Informative References	34
	Authors' Addresses	35

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.

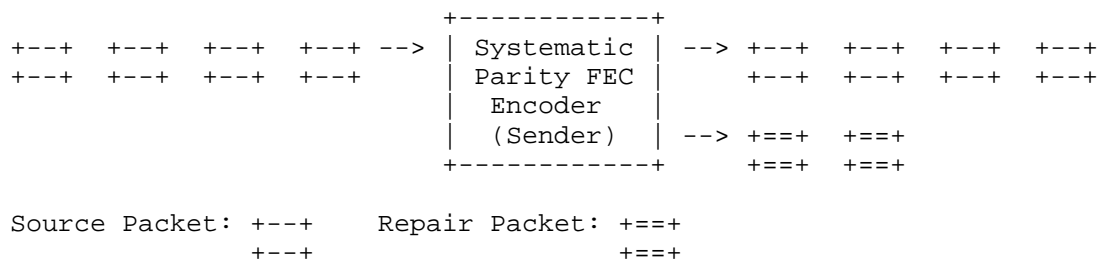


Figure 1: Block diagram for systematic parity FEC encoder

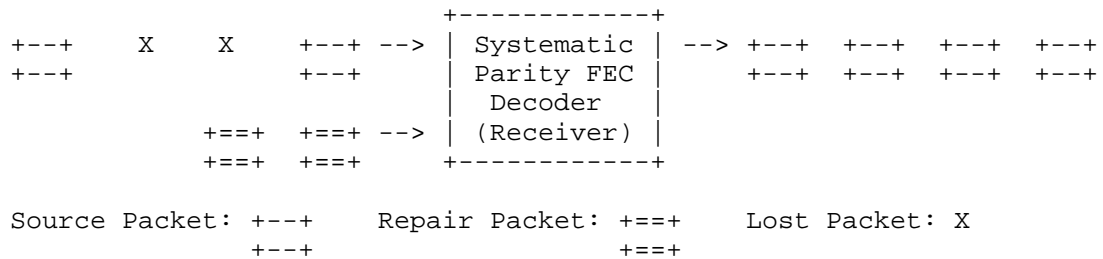


Figure 2: Block diagram for systematic parity FEC decoder

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.

$$\begin{array}{rcl}
 \begin{array}{ccccccccc}
 +-----+ \\
 | \text{S_1} & & \text{S_2} & & \text{S3} & & \dots & & \text{S_L} \\
 +-----+
 \end{array} & + \text{ XOR} & = \begin{array}{c} \text{R_1} \\ +====+ \end{array} \\
 \begin{array}{ccccccccc}
 +-----+ \\
 | \text{S_L+1} & & \text{S_L+2} & & \text{S_L+3} & & \dots & & \text{S_2xL} \\
 +-----+
 \end{array} & + \text{ XOR} & = \begin{array}{c} \text{R_2} \\ +====+ \end{array} \\
 \begin{array}{ccccccccc}
 \cdot & & \cdot & & \cdot & & & & \cdot \\
 \cdot & & \cdot & & \cdot & & & & \cdot \\
 \cdot & & \cdot & & \cdot & & & & \cdot
 \end{array} & & \\
 \begin{array}{ccccccccc}
 +-----+ \\
 | \text{S_}(D-1)\text{xL+1} & & \text{S_}(D-1)\text{xL+2} & & \text{S_}(D-1)\text{xL+3} & & \dots & & \text{S_DxL} \\
 +-----+
 \end{array} & + \text{ XOR} & = \begin{array}{c} \text{R_D} \\ +====+ \end{array}
 \end{array}$$

Figure 3: Generating non-interleaved (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.

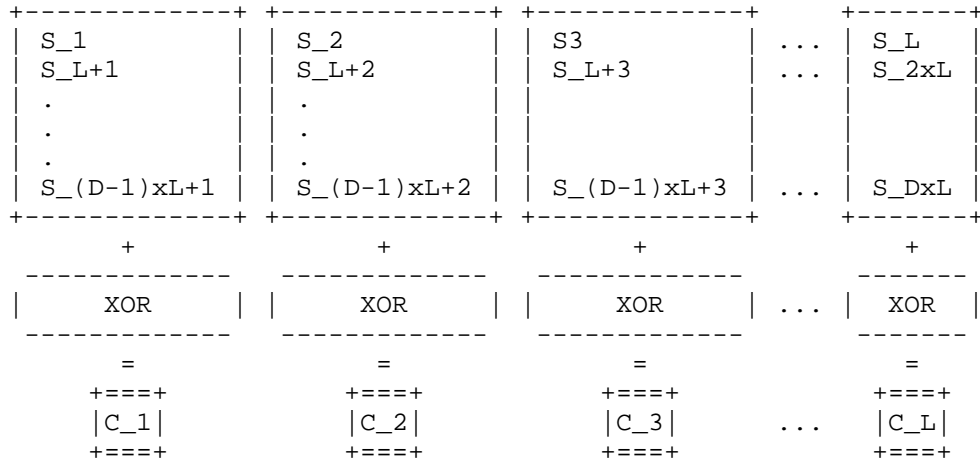


Figure 4: Generating interleaved (column) FEC packets

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.

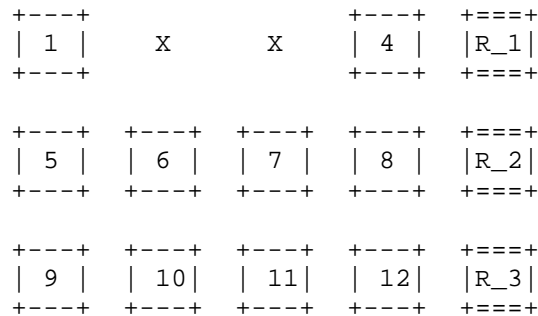


Figure 5: Example scenario where 1-D non-interleaved FEC protection fails error recovery (Burst Loss)

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.

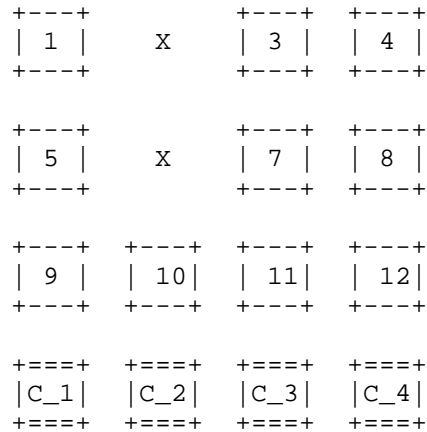


Figure 6: Example scenario where 1-D interleaved FEC protection fails error recovery (Periodic Loss)

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.

+----+			+----+	+====+
1	X	X	4	R_1
+----+			+----+	+====+
+----+	+----+	+----+	+----+	+====+
5	6	7	8	R_2
+----+	+----+	+----+	+----+	+====+
+----+			+----+	+====+
9	X	X	12	R_3
+----+			+----+	+====+
+====+	+====+	+====+	+====+	
C_1	C_2	C_3	C_4	
+====+	+====+	+====+	+====+	

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.

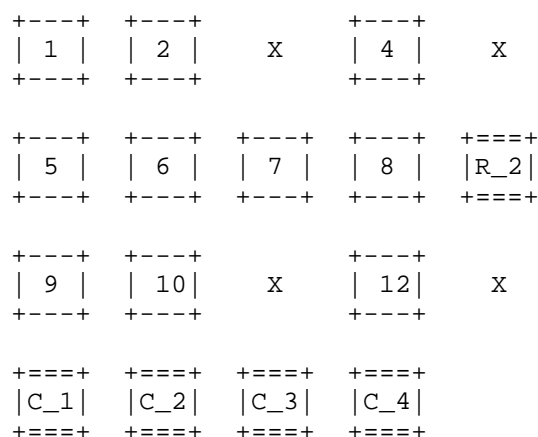


Figure 8: Example scenario #2 where 2-D parity FEC protection fails error recovery

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:

- o 1-D Non-interleaved FEC Protection: Overhead = $1/L$
- o 1-D Interleaved FEC Protection: Overhead = $1/D$
- o 2-D Parity FEC Protection: Overhead = $1/L + 1/D$

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

- o L: Number of columns of the source block.
- o D: Number of rows of the source block.
- o 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.

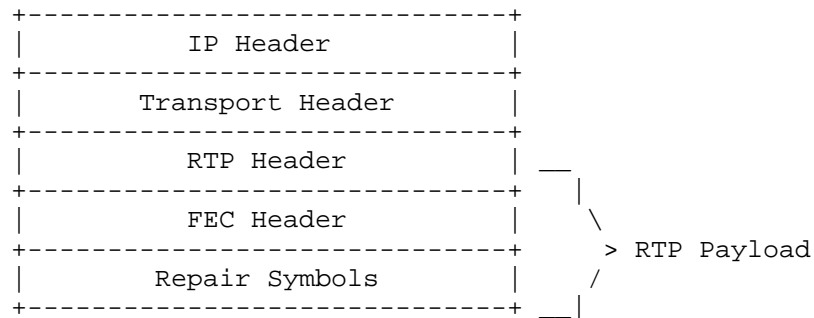


Figure 9: Format of repair packets

The RTP header is formatted according to [RFC3550] with some further clarifications listed below:

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

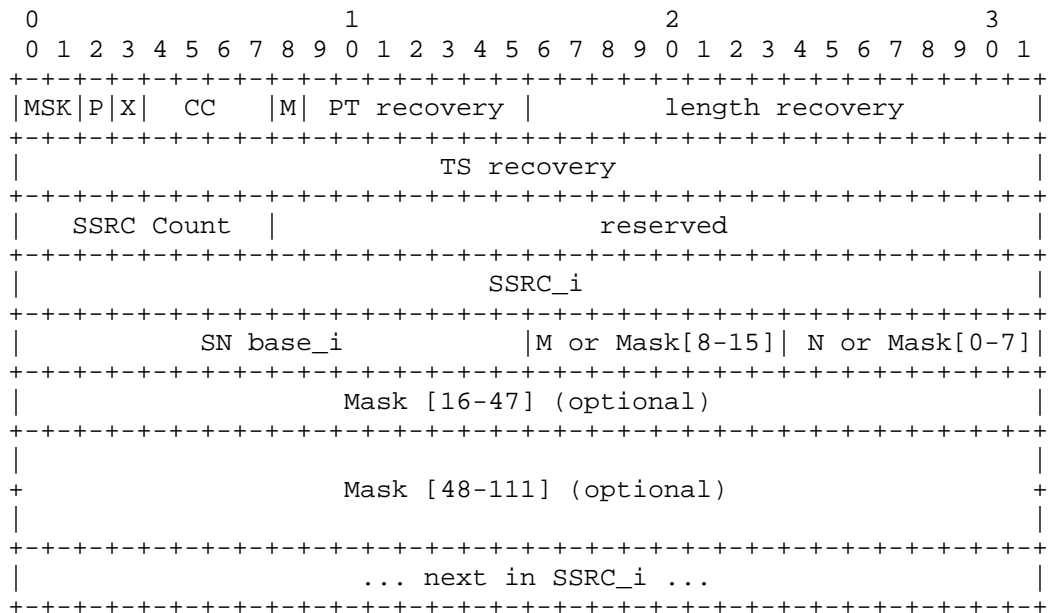


Figure 10: Format of the FEC header

The FEC header consists of the following fields:

- o The MSK field (2 bits) indicates the type of the mask. Namely:

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

- o The P, X, CC, M and PT recovery fields are used to determine the corresponding fields of the recovered packets.
- o The Length recovery (16 bits) field is used to determine the length of the recovered packets.
- o The TS recovery (32 bits) field is used to determine the timestamp of the recovered packets.
- o The SSRC count (8 bits) field describes the number of SSRCS protected by the FEC packet. 0 is not a valid value, and the packet MUST be ignored.
- o The Reserved (24 bits) field are reserved for future use. They MUST be set to zero by senders and ignored by receivers.
- o The SSRC_i (32 bits) field describes the SSRC of the packets protected by this particular FEC packet. If a FEC packet contains protects multiple SSRCS (indicated by the SSRC Count > 1), there will be multiple blocks of data containing the SSRC, SN base and Mask fields.
- o Editor's note: An alternate stream ID may replace SSRC.
- o The SN base_i (16 bits) field indicates the lowest sequence number, taking wrap around into account, of the source packets for a particular SSSRC (indicated in SSRC_i) protected by this repair packet.
- o Mask is a run-length encoding of packets for a particular SSRC_i protected by the FEC packet. Where a bit j set to 1 indicates that the source packet with sequence number (SN base_i + j) is protected by this FEC packet.
- o 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, $FEC = SN, SN+1, SN+2, \dots, SN+(M-1), SN+M$.
If $M>0$, $N>0$, indicates interleaved (column) FEC of every M packet in a group of N packets starting at SN base.
Hence, $FEC = SN+(M \times 0), SN+(M \times 1), \dots, SN+(M \times N)$.

Figure 12: Interpreting the M and N field values

By setting $SSRC$ count to 1, $M=1$, and $N \leq 1$, the FEC protects only one packet, i.e., the FEC payload carries just the packet indicated by SN_{Base_i} , which is effectively retransmitting the packet.

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.

5.1.1.1. Registration of audio/flexfec

Type name: audio

Subtype name: flexfec

Required parameters:

- o 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.
- 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:

- o L: indicates the number of columns of the source block that are protected by this FEC block and it applies to all the source SSRCs. L is a positive integer.
- o D: indicates the number of rows of the source block that are protected by this FEC block and it applies to all the source SSRCs. D is a positive integer.
- o ToP: indicates the type of 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.

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/flexfec

Type name: video

Subtype name: flexfec

Required parameters:

- o 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.
- 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:

- o L: indicates the number of columns of the source block that are protected by this FEC block and it applies to all the source SSRCs. L is a positive integer.
- o D: indicates the number of rows of the source block that are protected by this FEC block and it applies to all the source SSRCs. D is a positive integer.
- o ToP: indicates the type of 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.

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.

5.1.1.3. Registration of text/flexfec

Type name: text

Subtype name: flexfec

Required parameters:

- o 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.

- 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:

- o L: indicates the number of columns of the source block that are protected by this FEC block and it applies to all the source SSRCs. L is a positive integer.
- o D: indicates the number of rows of the source block that are protected by this FEC block and it applies to all the source SSRCs. D is a positive integer.
- o ToP: indicates the type of 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.

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/flexfec

Type name: application

Subtype name: flexfec

Required parameters:

- o 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.
- 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:

- o L: indicates the number of columns of the source block that are protected by this FEC block and it applies to all the source SSRCs. L is a positive integer.
- o D: indicates the number of rows of the source block that are protected by this FEC block and it applies to all the source SSRCs. D is a positive integer.
- o ToP: indicates the type of 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.

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:

- o The media type (e.g., "application") goes into the "m=" line as the media name.
- o 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.

- o 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:

- o 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.
- o 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.
- o 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.
- o 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:

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

- o The first 64 bits of the RTP header (64 bits).
- o 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:

- o 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.

- o The next bit in the FEC bit string is written into the P recovery bit in the FEC header.
- o The next bit in the FEC bit string is written into the X recovery bit in the FEC header.
- o The next 4 bits of the FEC bit string are written into the CC recovery field in the FEC header.
- o The next bit is written into the M recovery bit in the FEC header.
- o The next 7 bits of the FEC bit string are written into the PT recovery field in the FEC header.
- o The next 16 bits are skipped.
- o The next 32 bits of the FEC bit string are written into the TS recovery field in the FEC header.
- o The next 16 bits are written into the length recovery field in the FEC header.
- o 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:

$$p^*_{snb} + i * X_1 \text{ (modulo 65536)}$$

where p_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 marked 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 are as follows:

When $N = 0$:

$p_snb, p_snb+1, \dots, p_snb+(M-1), p_snb+M$

When $N > 0$:

$p_snb, p_snb+(M \times 1), p_snb+(M \times 2), \dots, p_snb+(M \times (N-1)), p_snb+(M \times N)$

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.

			+---+	+---+	+===+
X	X		3	4	R_1
			+---+	+---+	+===+
			+---+	+---+	+===+
5	6	7	8	R_2	
+---+	+---+	+---+	+---+	+---+	+===+
+---+			+---+	+===+	
9	X	X	12	R_3	
+---+			+---+	+===+	
+===+	+===+	+===+	+===+		
C_1	C_2	C_3	C_4		
+===+	+===+	+===+	+===+		

Figure 13: Example loss pattern for the iterative decoding algorithm

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.

+----+		+----+	+----+	+====+
1	X	3	4	R_1
+----+		+----+	+----+	+====+
+----+	+----+	+----+	+----+	+====+
5	6	7	8	R_2
+----+	+----+	+----+	+----+	+====+
+----+		+----+	+----+	+====+
9	X	11	12	R_3
+----+		+----+	+----+	+====+
+====+	+====+	+====+	+====+	
C_1	C_2	C_3	C_4	
+====+	+====+	+====+	+====+	

Figure 14: The resulting pattern after the first iteration

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.

+----+	+----+	+----+	+----+	+====+
1	2	3	4	R_1
+----+	+----+	+----+	+----+	+====+
+----+	+----+	+----+	+----+	+====+
5	6	7	8	R_2
+----+	+----+	+----+	+----+	+====+
+----+	+----+	+----+	+----+	+====+
9	10	11	12	R_3
+----+	+----+	+----+	+----+	+====+
+====+	+====+	+====+	+====+	
C_1	C_2	C_3	C_4	
+====+	+====+	+====+	+====+	

Figure 15: The resulting pattern after the second iteration

7. SDP Examples

This section provides two SDP [RFC4566] examples. The examples use the FEC grouping semantics defined in [RFC5956].

7.1. Example SDP for Flexible FEC Protection with in-band SSRC mapping

In this example, we have one source video stream and one FEC repair stream. The source and repair streams are multiplexed on different SSRCs. The repair window is set to 200 ms.

```
v=0
o=mo 1122334455 1122334466 IN IP4 fec.example.com
s=FlexFEC minimal SDP signalling Example
t=0 0
m=video 30000 RTP/AVP 96 98
c=IN IP4 143.163.151.157
a=rtpmap:96 VP8/90000
a=rtpmap:98 flexfec/90000
a=fmtp:98; repair-window=200ms
```

7.2. Example SDP for Flex FEC Protection with explicit signalling in the SDP

In this example, we have one source video stream (ssrc:1234) and one FEC repair streams (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=2-D Parity FEC with no in band signalling 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 flexfec/90000
a=fmtp:110 L:5; D:10; ToP:2; repair-window:200000
a=ssrc:1234
a=ssrc:2345
a=ssrc-group:FEC-FR 1234 2345
```

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 contributed 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-01

FEC packet format changed as per discussions in IETF93, Prague.

Replaced non-interleaved-parityfec and interleaved-parity-fec with flexfec.

SDP simplified for the case when association to RTP is made in the FEC header and not in the SDP.

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

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

12.3. 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:

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

12.4. draft-ietf-fecframe-ld2d-parity-scheme-00

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

13. References

13.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.
- [RFC3555] Casner, S. and P. Hoschka, "MIME Type Registration of RTP Payload Formats", RFC 3555, July 2003.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.
- [RFC5956] Begen, A., "Forward Error Correction Grouping Semantics in the Session Description Protocol", RFC 5956, September 2010.
- [RFC6363] Watson, M., Begen, A., and V. Roca, "Forward Error Correction (FEC) Framework", RFC 6363, October 2011.
- [RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, January 2013.
- [RFC7022] Begen, A., Perkins, C., Wing, D., and E. Rescorla, "Guidelines for Choosing RTP Control Protocol (RTCP) Canonical Names (CNAMEs)", RFC 7022, September 2013.

13.2. Informative References

- [Holmer13] Holmer, S., Shemer, M., and M. Paniconi, "Handling Packet Loss in WebRTC", Proc. of IEEE International Conference on Image Processing (ICIP 2013) , 9 2013.
- [I-D.singh-rmcat-adaptive-fec] Singh, V., Nagy, M., Ott, J., and L. Eggert, "Congestion Control Using FEC for Conversational Media", draft-singh-rmcat-adaptive-fec-01 (work in progress), October 2014.
- [Nagy14] Nagy, M., Singh, V., Ott, J., and L. Eggert, "Congestion Control using FEC for Conversational Multimedia Communication", Proc. of 5th ACM International Conference on Multimedia Systems (MMSys 2014) , 3 2014.
- [RFC2326] Schulzrinne, H., Rao, A., and R. Lanphier, "Real Time Streaming Protocol (RTSP)", RFC 2326, April 1998.
- [RFC2733] Rosenberg, J. and H. Schulzrinne, "An RTP Payload Format for Generic Forward Error Correction", RFC 2733, December 1999.
- [RFC2974] Handley, M., Perkins, C., and E. Whelan, "Session Announcement Protocol", RFC 2974, October 2000.
- [RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.
- [RFC5109] Li, A., "RTP Payload Format for Generic Forward Error Correction", RFC 5109, December 2007.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.
- [SMPTE2022-1] SMPTE 2022-1-2007, , "Forward Error Correction for Real-Time Video/Audio Transport over IP Networks", 2007.

Authors' Addresses

Varun Singh
Nemu Dialogue System Oy
Runeberginkatu 4c A 4
Helsinki, FIN 00100
Finland

Email: varun@callstats.io

Ali Begen

Email: acbegen@gmail.com

Mo Zanaty
Cisco
Raleigh, NC
USA

Email: mzanaty@cisco.com

Giridhar Mandyam
Qualcomm Innovation Center
5775 Morehouse Drive
San Diego, CA 92121
USA

Phone: +1 858 651 7200
Email: mandyam@quicinc.com

Payload Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 7, 2016

J. Uberti
S. Holmer
M. Flodman
Google
J. Lennox
D. Hong
Vidyo
July 6, 2015

RTP Payload Format for VP9 Video
draft-ietf-payload-vp9-00

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 January 7, 2016.

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
10.1. Normative References	18
10.2. Informative References	19
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 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.

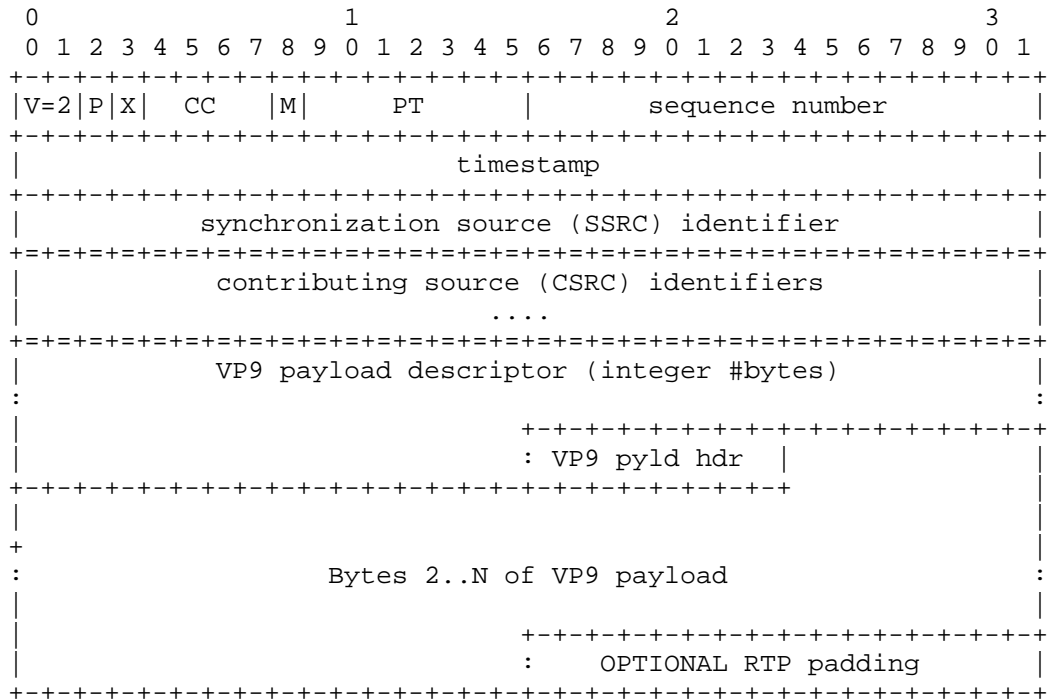
The SS data can also be used to specify the resolution of each spatial layer present in the VP9 stream.

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.



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.

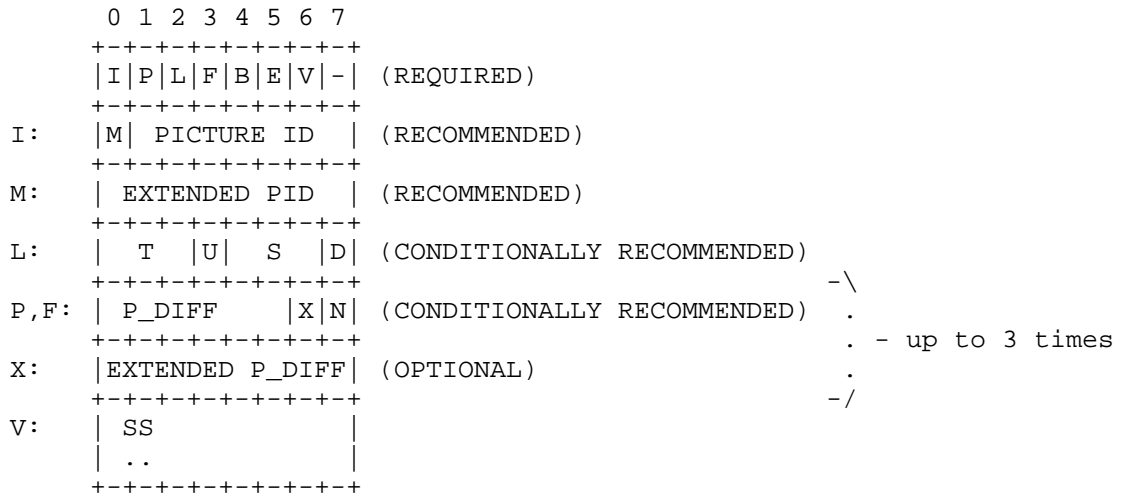


Figure 2

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.

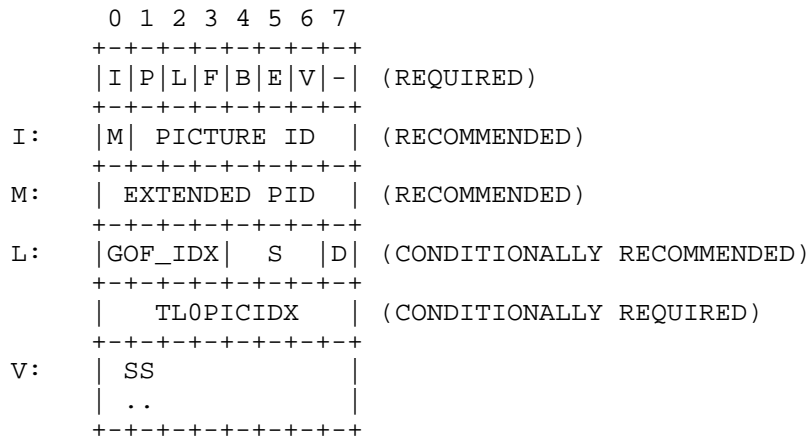


Figure 3

- 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, TLOPICIDX, 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:  |          | (OPTIONAL)  -\
    |   WIDTH   |           .
    +-----+           .
    |          | (OPTIONAL)  .
    +-----+           . - N_S + 1 times
    |   HEIGHT   | (OPTIONAL) .
    +-----+           .
    |          | (OPTIONAL)  .
    +-----+           -/
N_G: | T |U| R | - | - | (OPTIONAL)  -\
     +-----+           . - N_G + 1 times
     |   P_DIFF   | (OPTIONAL)  . - R times  .
     +-----+           -/

```

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]:

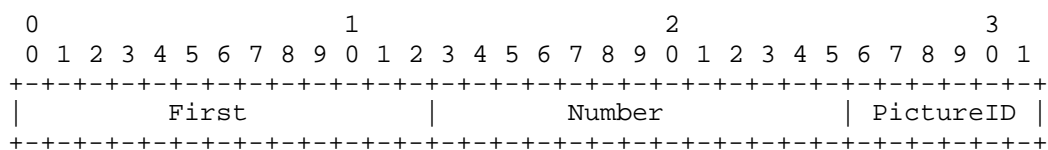


Figure 5

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.

Event	Sender	Receiver	Established reference
1000	Send golden frame PictureID = 0	Receive and decode golden frame	golden
1001		Send RPSI(0)	
1002	Receive RPSI(0)		
...	(sending regular frames)		
1100	Send altref frame PictureID = 100	Altref corrupted or lost	golden
1101		Send SLI(100)	golden
1102	Receive SLI(100)		
1103	Send frame with reference to golden	Receive and decode frame (decoder state restored)	golden
...	(sending regular frames)		
1200	Send altref frame PictureID = 200	Receive and decode altref frame	golden
1201		Send RPSI(200)	

1202	Receive RPSI(200)		altref
...	(sending regular frames)		
1300	Send golden frame PictureID = 300		
		Receive and decode golden frame	altref
1301		Send RPSI(300)	altref
1302	RPSI lost		
1400	Send golden frame PictureID = 400		
		Receive and decode golden frame	altref
1401		Send RPSI(400)	
1402	Receive RPSI(400)		golden

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 $\text{int}(\sqrt{\text{max-fs} * 8})$ - for instance, a max-fs of 1200 (capable of supporting 640x480 resolution) will support widths and heights up to 1552 pixels (97 macroblocks).

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.

Fragment identifier considerations: N/A.

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:

- o The media name in the "m=" line of SDP MUST be video.
- o The encoding name in the "a=rtpmap" line of SDP MUST be VP9 (the media subtype).
- o The clock rate in the "a=rtpmap" line MUST be 90000.
- o The parameters "max-fs", and "max-fr", MUST be included in the "a=fmtp" line of SDP if SDP is used to declare receiver capabilities. 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

10.1. Normative 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]

Lennox, J., Hong, D., Uberti, J., Holmer, S., and M. Flodman, "The Layer Refresh Request (LRR) RTCP Feedback Message", draft-lennox-avtext-lrr-00 (work in progress), March 2015.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

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

[RFC4585] Ott, J., Wenger, S., Sato, N., Burmeister, C., and J. Rey, "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", RFC 4585, July 2006.

[RFC4855] Casner, S., "Media Type Registration of RTP Payload Formats", RFC 4855, February 2007.

[RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, January 2013.

10.2. Informative References

[RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, July 2003.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

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

Email: holmer@google.com

Magnus Flodman
Google, Inc.
Kungsbron 2
Stockholm 111 22
Sweden

Email: mflodman@google.com

Jonathan Lennox
Vidyo, Inc.
433 Hackensack Avenue
Seventh Floor
Hackensack, NJ 07601
US

Email: jonathan@vidyo.com

Danny Hong
Vidyo, Inc.
433 Hackensack Avenue
Seventh Floor
Hackensack, NJ 07601
US

Email: danny@vidyo.com