Payload Working Group
Internet-Draft
Intended Status: Standards Track

Expires: April 16, 2018

Victor Demjanenko
John Punaro
David Satterlee
VOCAL Technologies, Ltd.

October 13, 2017

RTP Payload Format for TSVCIS Codec
draft-demjanenko-payload-tsvcis-00

Status of This Memo

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

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

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.

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."

Abstract

This document describes the RTP payload format for the Tactical Secure Voice Cryptographic Interoperability Specification (TSVCIS) speech coder. TSVCIS is a scalable narrowband voice coder supporting varying encoder data rates and fallbacks. It is implemented as an augmentation to the Mixed Excitation Linear Prediction Enhanced (MELPe) speech coder by conveying additional speech coder parameters for enhancing voice quality. TSVCIS augmented speech data is

processed in conjunction with its temporal matched MELP 2400 speech data. The RTP packetization of TSVCIS and MELPe speech coder data is described in detail.

Table of Contents

1. Introduction 2
1.1. Conventions 3
2. Background 3
3. Payload Format 4
3.1. MELPe Bitstream Definitions 5
3.1.1. 2400 bps Bitstream Structure 6
3.1.2. 1200 bps Bitstream Structure 6
3.1.3. 600 bps Bitstream Structure 7
3.1.4. Comfort Noise Bitstream Definition 8
3.2. TSVCIS Bitstream Definition 8
3.3. Multiple TSVCIS Frames in an RTP Packet 10
3.4. Congestion Control Considerations 11
4. Payload Format Parameters 11
4.1. Media Type Definitions 11
4.2. Mapping to SDP 13
4.3. Declarative SDP Considerations 14
4.4. Offer/Answer SDP Considerations 15
5. Discontinuous Transmissions 15
6. Packet Loss Concealment 16
7. IANA Considerations 16
8. Security Considerations 16
9. RFC Editor Considerations 17
10. References 17
10.1. Normative References 17
10.2. Informative References 19
Authors' Addresses 19

1. Introduction

This document describes how compressed Tactical Secure Voice Cryptographic Interoperability Specification (TSVCIS) speech as produced by the TSVCIS codec may be formatted for use as an RTP payload. The TSVCIS speech coder (or TSVCIS speech aware communications equipment on any intervening transport link) may adjust to restricted bandwidth conditions by reducing the amount of augmented speech data and relying on the underlying MELPe speech coder for the most constrained bandwidth links.

Details are provided for packetizing the TSVCIS augmented speech data along with MELPe 2400 bps speech parameters in a RTP packet. The sender may send one or more codec data frames per packet, depending

on the application scenario or based on transport network conditions, bandwidth restrictions, delay requirements, and packet loss tolerance.

1.1. Conventions

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].

Best current practices for writing an RTP payload format specification were followed [RFC2736].

2. Background

The MELP speech coder was developed by the US military as an upgrade from the LPC-based CELP standard vocoder for low-bitrate communications [MELP]. ("LPC" stands for "Linear-Predictive Coding", and "CELP" stands for "Code-Excited Linear Prediction".) MELP was further enhanced and subsequently adopted by NATO as MELPe for use by its members and Partnership for Peace countries for military and other governmental communications as international NATO Standard STANAG 4591 [MELPE].

The Tactical Secure Voice Cryptographic Interoperability Specification (TSVCIS) is a specification written by the Tactical Secure Voice Working Group (TSVWG) for enabling all modern tactical secure voice devices to be interoperable across the Department of Defense [TSVCIS]. One of the most important aspects is that the voice modes defined in TSVCIS are based on a fixed rate variant of Naval Research Lab’s (NRL’s) Variable Date Rate (VDR) Vocoder which uses the MELPe standard as its base [NRLVDR]. A complete TSVCIS speech frame consists of MELPe speech parameters and corresponding TSVCIS augmented speech data.

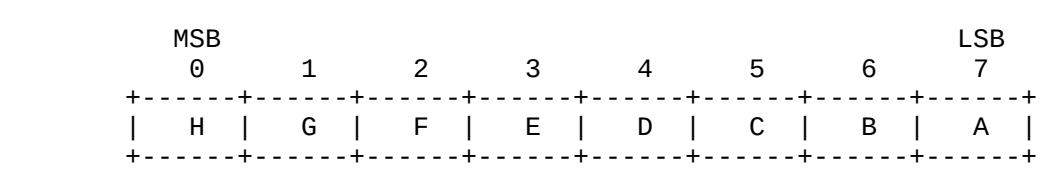
In addition to the augmented speech data, the TSVCIS specification identifies which speech coder and framing bits are to be encrypted, and how they are protected by forward error correction (FEC) techniques (using block codes). At the RTP transport layer, only the speech coder related bits need to be considered and are conveyed in unencrypted form. In most IP-based network deployments, standard link encryption methods (SRTP, VPNs, FIPS 140 link encryptors or Type 1 Ethernet encryptors) would be used to secure the RTP speech contents. Further, it is desirable to support the highest voice quality between endpoint which is only possible without the overhead of FEC.

TSVCIS augmented speech data is derived from the signal processing

and data already performed by the MELPe speech coder. For the purposes of this specification, only the general parameter nature of TSVCIS will be characterized. Depending on the bandwidth available (and FEC requirements), a varying number of TSVCIS specific speech coder parameters need to be transported. These are first byte-packed and then conveyed from encoder to decoder.

Byte packing of TSVCIS speech data into packed parameters is processed as per the following example:

Two-bit field: bits A and B (A is MSB, B is LSB)
Six-bit field: bits C, D, E, F, G, and H (C is MSB, H is LSB)



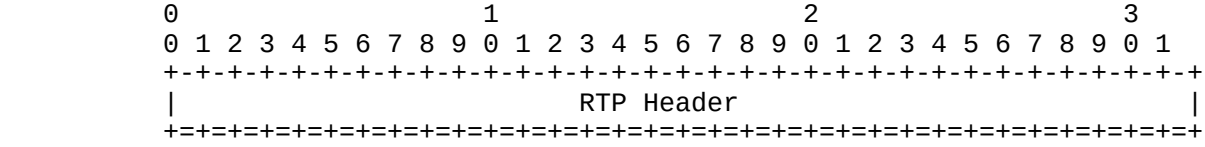
This packing method places the two-bit field "first" in the lowest bits followed by the next six-bit field. Parameters may be split between octets with the most significant bits in the earlier octet. Any unfilled bits in the last octet SHOULD be filled with zero.

In order to accommodate a varying amount of TSVCIS augmented speech data, it is only necessary to specify the number of octets containing the packed TSVCIS parameters. The encoding to do so is presented in Section 3.2. The preferred sets of TSVCIS parameters is specified in the speech coder specification [TSVCIS] and is beyond the scope of this RFC to describe or limit.

3. Payload Format

The TSVCIS codec augments the standard MELP 2400, 1200 and 600 bitrates and hence uses 22.5, 67.5, or 90 ms frames with a sampling rate clock of 8 kHz, so the RTP timestamp MUST be in units of 1/8000 of a second.

The RTP payload for TSVCIS has the format shown in Figure 1. No additional header specific to this payload format is needed. This format is intended for situations where the sender and the receiver send one or more codec data frames per packet.



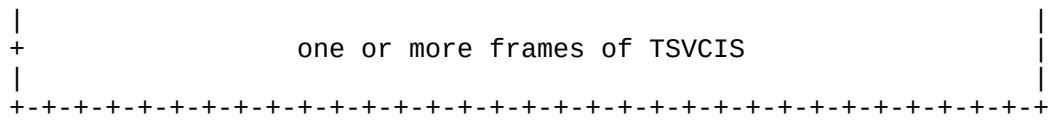


Figure 1: Packet Format Diagram

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

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

3.1. MELPe Bitstream Definitions

The TCVCIS speech coder includes all three MELPe coder rates used as base speech parameters or as speech coders for bandwidth restricted links. RTP packetization of MELPe follows RFC 8130 and is repeated here for all three MELPe rates [RFC8130] which with promoted suggestions or recommendations now regarded as requirements. The bits previously labeled as RSVA, RSVB, and RSVC in RFC 8130 SHOULD be filled with rate coding, CODA, CODB, and CODC, as shown in Table 1 (compatible with Table 7 in Section 3.3 of [RFC8130]).

Coder Bitrate	CODA	CODB	CODC	Length
2400 bps	0	0	N/A	7
1200 bps	1	0	0	11
600 bps	0	1	N/A	7
Comfort Noise	1	0	1	2
TSVCIS data	1	1	N/A	var.

Table 1: TSVCIS/MELPe Frame Bitrate Indicators and Frame Length

The total number of bits used to describe one MELPe frame of 2400 bps speech is 54, which fits in 7 octets (with two rate code bits). For MELPe 1200 bps speech, the total number of bits used is 81, which fits in 11 octets (with three rate code bits and four unused bits). For MELPe 600 bps speech, the total number of bits used is 54, which fits in 7 octets (with two rate code bits). The comfort nosie frame consists of 13 bits, which fits in 2 octets (with three rate code bits). TSVCIS packed parameters will use the last code combination in a trailing byte as discussed in Section 3.2.

It should be noted that CODB for both the 2400 and 600 bps modes MAY deviate from the values in Table 1 when bit 55 is used as an end-to-end framing bit. Frame decoding would remain distinct as CODA being zero on its own would indicate a 7-byte frame for either rate and the use of 600 bps speech coding could be deduced from the RTP timestamp (and anticipated by the SDP negotiations).

3.1.1. 2400 bps Bitstream Structure

The 2400 bps MELPe RTP payload is constructed as per Figure 2. Note that CODA must be filled with 0 and CODB SHOULD be filled with 0 as per Section 3.1. CODB MAY contain an end-to-end framing bit if required by the endpoints.

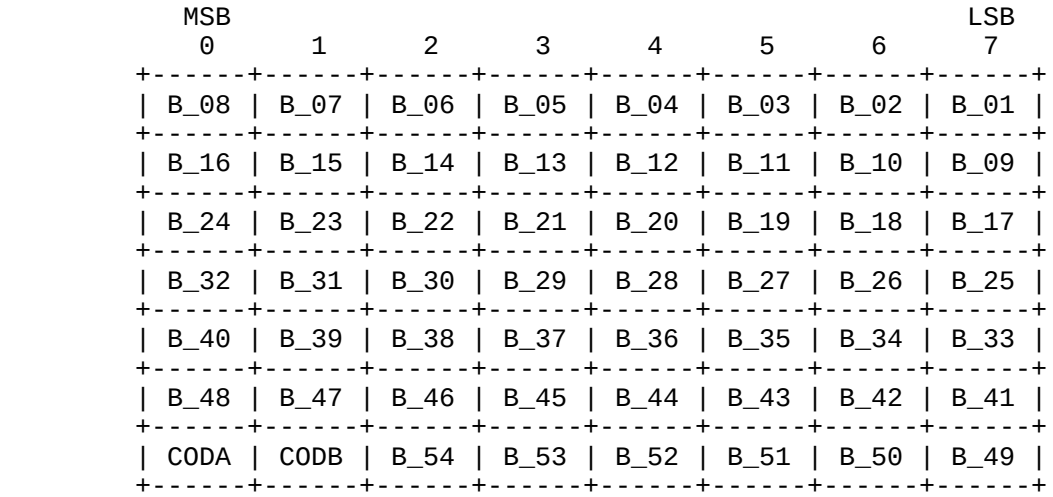


Figure 2: Packed MELPe 2400 bps Payload Octets

3.1.2. 1200 bps Bitstream Structure

The 1200 bps MELPe RTP payload is constructed as per Figure 3. Note that CODA, CODB, and CODC MUST be filled with 1, 0, and 0 respectively as per Section 3.1. RSV0 SHOULD be coded as 0.

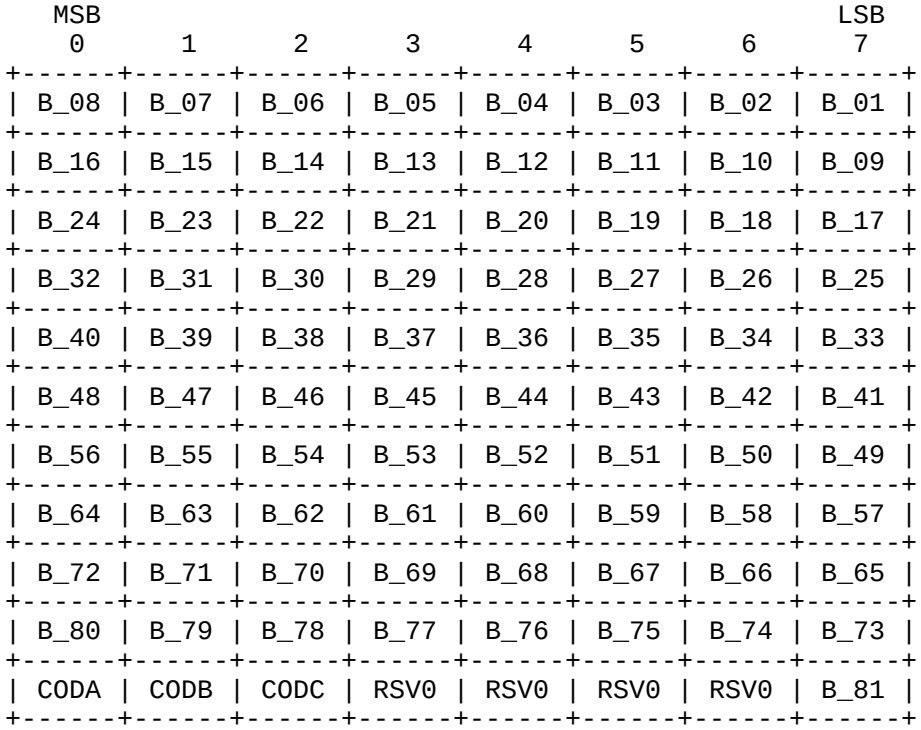
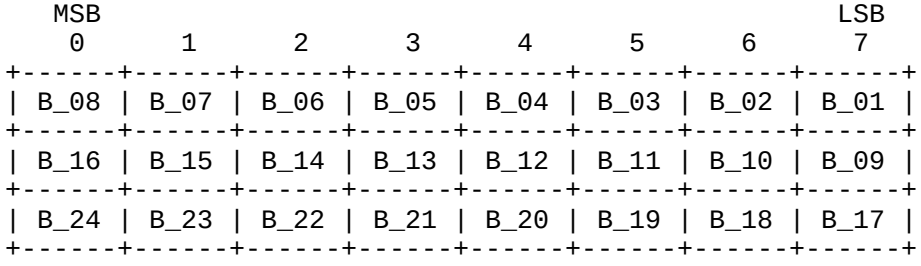


Figure 3: Packed MELPe 1200 bps Payload Octets

3.1.3. 600 bps Bitstream Structure

The 600 bps MELPe RTP payload is constructed as per Figure 4. Note CODA must be filled with 0 and CODB SHOULD be filled with 1 as per Section 3.1. CODB MAY contain an end-to-end framing bit if required by the endpoints.



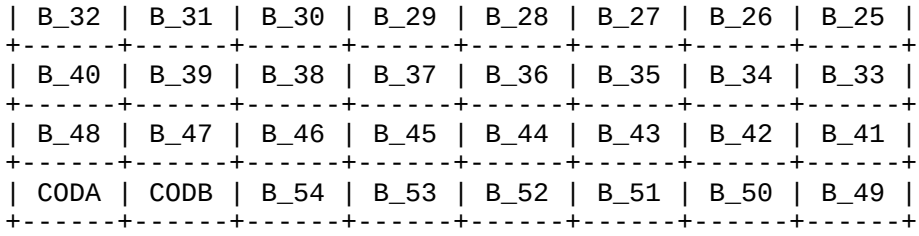


Figure 4: Packed MELPe 600 bps Payload Octets

3.1.4. Comfort Noise Bitstream Definition

The comfort noise MELPe RTP payload is constructed as per Figure 5. Note that CODA, CODB, and CODC MUST be filled with 1, 0, and 1 respectively as per Section 3.1.

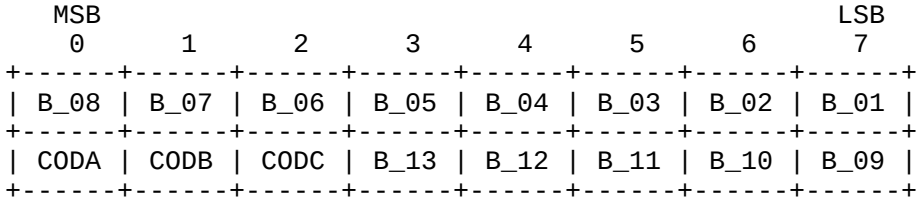
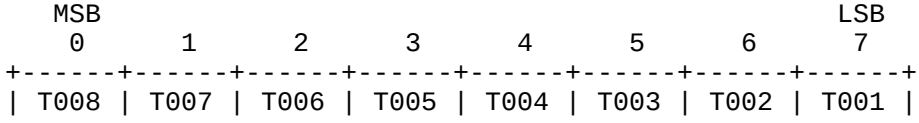


Figure 5: Packed MELPe Comfort Noise Payload Octets

3.2. TSVCIS Bitstream Definition

The TSVCIS augmented speech data as packed parameters MUST be placed immediately after a corresponding MELPe 2400 bps payload. The packed parameters are counted in octets (TC). In the preferred placement, shown in Figure 6, a single trailing octet SHALL be appended to include a two-bit rate code, CODA and CODB, (both bits set to one) and a six-bit modified count (MTC). The special modified count value of all ones (representing a MTC value of 63) SHALL NOT be used for this format as it is used as the indicator for the alternate packing format shown next. In a standard implementation, the TSVCIS speech coder uses a minimum of 15 octets for parameters in octet packed form. The modified count (MTC) MUST be reduced by 15 from the full octet count (TC). Computed MTC = TC-15. This accommodates a maximum of 77 parameter octets (maximum value of MTC is 62, 77 is the sum of 62+15).



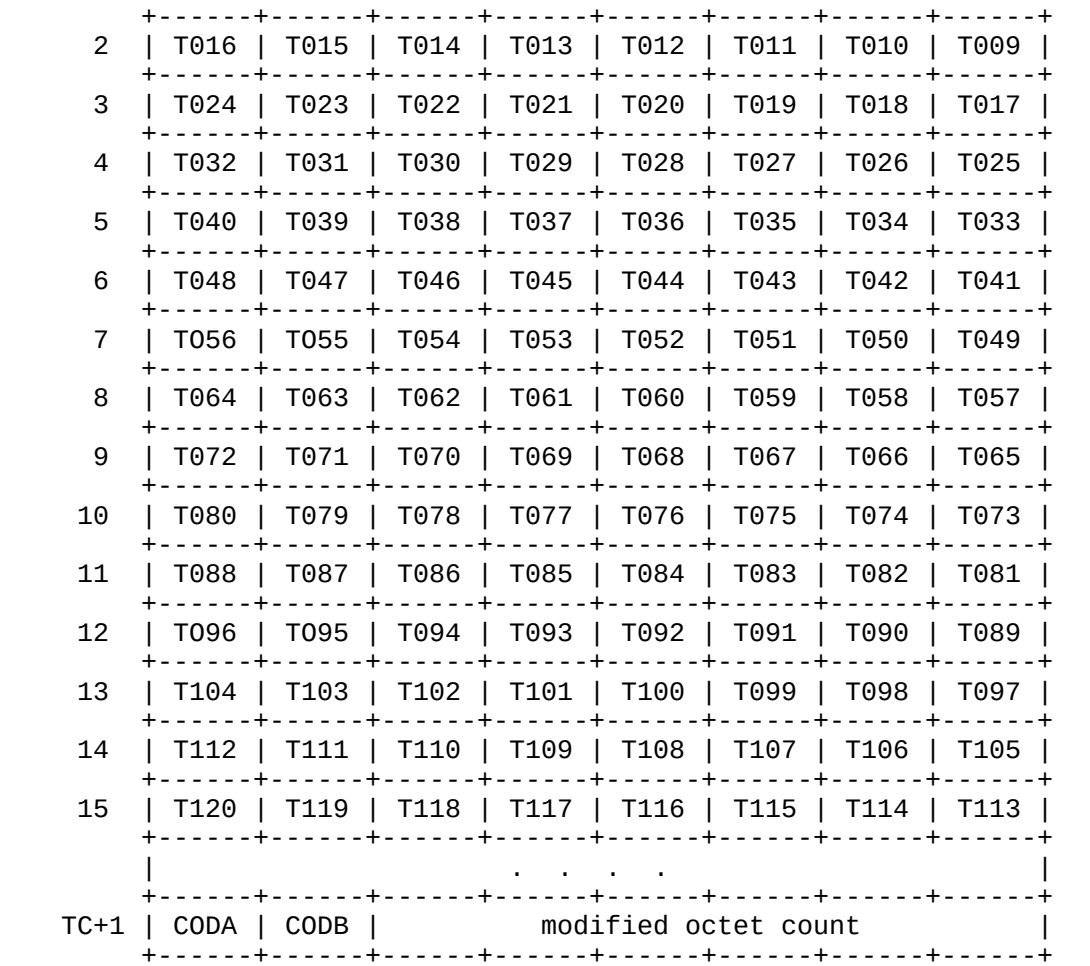
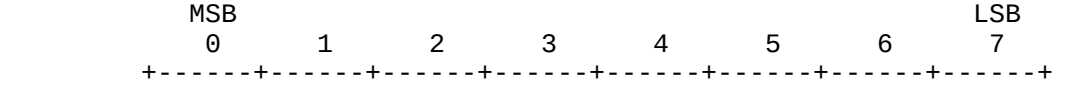


Figure 6: Preferred Packed TSVCIS Payload Octets

In order to accommodate all other NRL VDR configurations for TSVCIS, an alternate parameter placement MUST use two trailing bytes as shown in Figure 7. The last trailing byte MUST be filled with a two-bit rate code, CODA and CODB, (both bits set to one) and its six-bit count field MUST be filled with ones. The second to last trailing byte MUST contain the parameter count (TC) in octets and MAY represent any value from one to 255. The value of zero SHALL be considered as reserved.



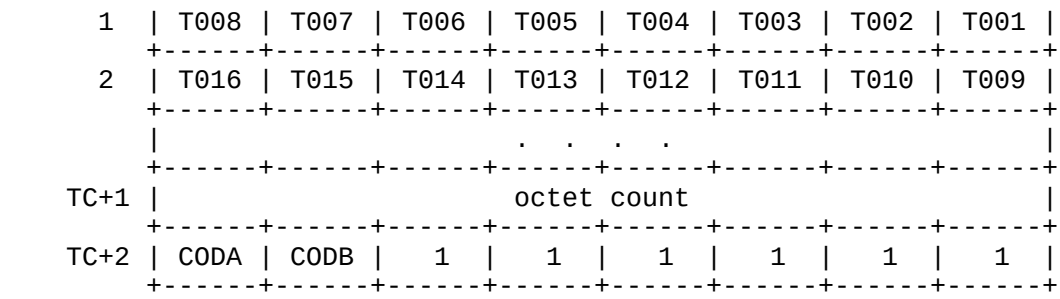


Figure 7: Length Unrestricted Packed TSVCIS Payload Octets

3.3. Multiple TSVCIS Frames in an RTP Packet

A TSVCIS RTP packet MAY consist of zero or more TSVCIS coder frames (each consisting of MELPe and TSVCIS coder data) followed by zero or one MELPe comfort noise frame. The presence of a comfort noise frame can be determined by its rate code bits in its last octet.

The default packetization interval is one coder frame (22.5, 67.5, or 90 ms) according to the coder bitrate (2400, 1200, or 600 bps). For some applications, a longer packetization interval is used to reduce the packet rate.

A TSVCIS RTP packet comprised of no coder frame and no comfort noise frame MAY be used periodically by an endpoint to indicate connectivity by an otherwise idle receiver.

TSVCIS coder frames in a single RTP packet MAY be of different coder bitrates. With the exception for the variable length TSVCIS parameter frames, the coder rate bits in the trailing byte identify the contents and length as per Table 1.

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

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

Frames MUST NOT be split between RTP packets.

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

packet would be acceptable.

Information describing the number of frames contained in an RTP packet is not transmitted as part of the RTP payload. The way to determine the number of TSVCIS/MELPe frames is to identify each frame type and length thereby counting the total number of octets within the RTP packet.

3.4. Congestion Control Considerations

The target bitrate of TSVCIS can be adjusted at any point in time, thus allowing congestion management. Furthermore, the amount of encoded speech or audio data encoded in a single packet can be used for congestion control, since the packet rate is inversely proportional to the packet duration. A lower packet transmission rate reduces the amount of header overhead but at the same time increases latency and loss sensitivity, so it ought to be used with care.

Since UDP does not provide congestion control, applications that use RTP over UDP SHOULD implement their own congestion control above the UDP layer [RFC8085] and MAY also implement a transport circuit breaker [RFC8083]. Work in the RMCAT working group [RMCAT] describes the interactions and conceptual interfaces necessary between the application components that relate to congestion control, including the RTP layer, the higher-level media codec control layer, and the lower-level transport interface, as well as components dedicated to congestion control functions.

4. Payload Format Parameters

This RTP payload format is identified using the TSVCIS media subtype, which is registered in accordance with RFC 4855 [RFC4855] and per the media type registration template from RFC 6838 [RFC6838].

4.1. Media Type Definitions

Type name: audio

Subtype names: TSVCIS

Required parameters: N/A

Optional parameters:

ptime: the recommended length of time (in milliseconds) represented by the media in a packet. It SHALL use the nearest rounded-up ms integer packet duration. For TSVCIS, this

corresponds to the following values: 23, 45, 68, 90, 112, 135, 156, and 180. Larger values can be used as long as they are properly rounded. See Section 6 of RFC 4566 [RFC4566].

maxptime: the maximum length of time (in milliseconds) that can be encapsulated in a packet. It SHALL use the nearest rounded-up ms integer packet duration. For TSVCIS, this corresponds to the following values: 23, 45, 68, 90, 112, 135, 156, and 180. Larger values can be used as long as they are properly rounded. See Section 6 of RFC 4566 [RFC4566].

bitrate: specifies the MELPe coder bitrates supported. Possible values are a comma-separated list of rates from the following set: 2400, 1200, 600. The modes are listed in order of preference; first is preferred. If "bitrate" is not present, the fixed coder bitrate of 2400 MUST be used.

tcmax: specifies the TSVCIS maximum value for TC supported or desired ranging from 1 to 255. If "tcmax" is not present, a default value of TBD is used.

[EDITOR NOTE - the value for TBD is to be discussed and stated. A value of 35 is suggested.]

Encoding considerations: This media subtype is framed and binary; see Section 4.8 of RFC 6838 [RFC6838].

Security considerations: Please see Section 8 of RFCxxxx (this RFC).

Interoperability considerations: N/A

Published specification: N/A

Applications that use this media type: N/A

Additional information: N/A

Deprecated alias names for this type: N/A

Magic number(s): N/A

File extension(s): N/A

Macintosh file type code(s): N/A

Person & email address to contact for further information:

Victor Demjanenko, Ph.D.

VOCAL Technologies, Ltd.
520 Lee Entrance, Suite 202
Buffalo, NY 14228
United States of America
Phone: +1 716 688 4675
Email: victor.demjanenko@vocal.com

Intended usage: COMMON

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

Author: Victor Demjanenko

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

Provisional registration? (standards tree only): No

4.2. Mapping to SDP

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

The information carried in the media type specification has a specific mapping to fields in the Session Description Protocol (SDP) [RFC4566], which is commonly used to describe RTP sessions. When SDP is used to specify sessions employing the TSVCIS codec, the mapping is as follows:

- The media type ("audio") goes in SDP "m=" as the media name.
- The media subtype (payload format name) goes in SDP "a=rtpmap" as the encoding name.
- The parameter "bitrate" goes in the SDP "a=fmtp" attribute by copying it as a "bitrate=<value>" string.
- The parameter "tcmax" goes in the SDP "a=fmtp" attribute by copying it as a "tcmax=<value>" string.
- The parameters "ptime" and "maxptime" go in the SDP "a=ptime" and "a=maxptime" attributes, respectively.

When conveying information via SDP, the encoding name SHALL be "TSVCIS" (the same as the media subtype).

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

```
m=audio 49120 RTP/AVP 96
a=rtpmap:96 TSVCIS/8000
```

The optional media type parameter "bitrate", when present, MUST be included in the "a=fmtp" attribute in the SDP, expressed as a media type string in the form of a semicolon-separated list of parameter=value pairs. The string "value" can be one or more of 2400, 1200, and 600, separated by commas (where each bitrate value indicates the corresponding MELPe coder). An example of the media representation in SDP for describing TSVCIS when all three coder bitrates are supported might be:

```
m=audio 49120 RTP/AVP 96
a=rtpmap:96 TSVCIS/8000
a=fmtp:96 bitrate=2400,600,1200
```

The optional media type parameter "tcmax", when present, MUST be included in the "a=fmtp" attribute in the SDP, expressed as a media type string in the form of a semicolon-separated list of parameter=value pairs. The string "value" is an integer number in the range of 1 to 255 representing the maximum number of TSVCIS parameter octets supported. An example of the media representation in SDP for describing TSVCIS with a maximum of 101 octets supported is as follows:

```
m=audio 49120 RTP/AVP 96
a=rtpmap:96 TSVCIS/8000
a=fmtp:96 tcmax=101
```

Parameter "ptime" cannot be used for the purpose of specifying the TSVCIS operating mode, due to the fact that for certain values it will be impossible to distinguish which mode is about to be used (e.g., when ptime=68, it would be impossible to distinguish if the packet is carrying one frame of 67.5 ms or three frames of 22.5 ms).

Note that the payload format (encoding) names are commonly shown in upper case. Media subtypes are commonly shown in lower case. These names are case insensitive in both places. Similarly, parameter names are case insensitive in both the media subtype name and the default mapping to the SDP a=fmtp attribute.

4.3. Declarative SDP Considerations

For declarative media, the "bitrate" parameter specifies the possible bitrates used by the sender. Multiple TSVCIS rtpmap values (such as

97, 98, and 99, as used below) MAY be used to convey TSVCIS-coded voice at different bitrates. The receiver can then select an appropriate TSVCIS codec by using 97, 98, or 99.

```
m=audio 49120 RTP/AVP 97 98 99
a=rtpmap:97 TSVCIS/8000
a=fmtp:97 bitrate=2400
a=rtpmap:98 TSVCIS/8000
a=fmtp:98 bitrate=1200
a=rtpmap:99 TSVCIS/8000
a=fmtp:99 bitrate=600
```

For declarative media, the "tcmax" parameter specifies the maximum number of TSVCIS packed parameter octets used by the sender or the sender’s communications channel.

4.4. Offer/Answer SDP Considerations

In the Offer/Answer model [RFC3264], "bitrate" is a bidirectional parameter. Both sides MUST use a common "bitrate" value or values. The offer contains the bitrates supported by the offerer, listed in its preferred order. The answerer MAY agree to any bitrate by listing the bitrate first in the answerer response. Additionally, the answerer MAY indicate any secondary bitrate or bitrates that it supports. The initial bitrate used by both parties SHALL be the first bitrate specified in the answerer response.

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

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

In the Offer/Answer model [RFC3264], "tcmax" is a bidirectional parameter. Both sides SHOULD use a common "tcmax" value. The offer contains the tcmax supported by the offerer. The answerer MAY agree to any tcmax equal or less than this value by stating the desired tcmax in the answerer response. The answerer alternatively MAY identify its own tcmax and rely on TSVCIS ignoring any augmented data it cannot use.

5. Discontinuous Transmissions

A primary application of TSVCIS is for radio communications of voice

conversations, and discontinuous transmissions are normal. When TSVCIS is used in an IP network, TSVCIS RTP packet transmissions may cease and resume frequently. RTP synchronization source (SSRC) sequence number gaps indicate lost packets to be filled by PLC, while abrupt loss of RTP packets indicates intended discontinuous transmissions.

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

6. Packet Loss Concealment

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

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

7. IANA Considerations

This memo requests that IANA registers TSVCIS as specified in Section 4.1. The media type is also requested to be added to the IANA registry for "RTP Payload Format MIME types" (http://www.iana.org/assignments/rtp-parameters).

8. 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 such as RTP/AVP [RFC3551], RTP/AVPF [RFC4585], RTP/SAVP [RFC3711], or RTP/SAVPF [RFC5124]. However, as discussed in [RFC7202], it is not an RTP payload format’s responsibility to discuss or mandate what solutions are used to meet such basic security goals as confidentiality, integrity, and source authenticity for RTP in general. This responsibility lies with anyone using RTP in an application. They can find guidance on available security mechanisms and important considerations in [RFC7201]. Applications SHOULD use one or more appropriate strong security mechanisms. The rest of this section discusses the security-impacting properties of the payload

format itself.

This RTP payload format and the TSVCIS 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. Additionally, the RTP payload format does not contain any active content.

Please see the security considerations discussed in [RFC6562] regarding VAD and its effect on bitrates.

9. RFC Editor Considerations

Note to RFC Editor: This section may be removed after carrying out all the instructions of this section.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC2736] Handley, M. and C. Perkins, "Guidelines for Writers of RTP Payload Format Specifications", BCP 36, RFC 2736, DOI 10.17487/RFC2736, December 1999, <http://www.rfc-editor.org/info/rfc2736>.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, DOI 10.17487/RFC3264, June 2002, <http://www.rfc-editor.org/info/rfc3264>.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <http://www.rfc-editor.org/info/rfc3550>.

[RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, DOI 10.17487/RFC3551, July 2003, <http://www.rfc-editor.org/info/rfc3551>.

[RFC8130] Demjanenko, V., and D. Satterlee, "RTP Payload Format for the Mixed Excitation Linear Prediction Enhanced (MELPe)

Codec", RFC 8130, DOI 10.tbd/RFC8130, March 2017, <http://www.rfc-editor.org/info/rfc8130>.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, DOI 10.17487/RFC3711, March 2004, <http://www.rfc-editor.org/info/rfc3711>.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, DOI 10.17487/RFC4566, July 2006, <http://www.rfc-editor.org/info/rfc4566>.

[RFC4855] Casner, S., "Media Type Registration of RTP Payload Formats", RFC 4855, DOI 10.17487/RFC4855, February 2007, <http://www.rfc-editor.org/info/rfc4855>.

[RFC5124] Ott, J. and E. Carrara, "Extended Secure RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/SAVPF)", RFC 5124, DOI 10.17487/RFC5124, February 2008, <http://www.rfc-editor.org/info/rfc5124>.

[RFC6562] Perkins, C. and JM. Valin, "Guidelines for the Use of Variable Bit Rate Audio with Secure RTP", RFC 6562, DOI 10.17487/RFC6562, March 2012, <http://www.rfc-editor.org/info/rfc6562>.

[RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <http://www.rfc-editor.org/info/rfc6838>.

[RFC8083] Perkins, C. and V. Singh, "Multimedia Congestion Control: Circuit Breakers for Unicast RTP Sessions", RFC 8083, DOI 10.17487/RFC8083, March 2017, <http://www.rfc-editor.org/info/rfc8083>.

[RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", RFC 8085, DOI 10.17487/RFC8085, March 2017, <http://www.rfc-editor.org/info/rfc8085>.

[NRLVDR] Heide, D., Cohen, A., Lee, Y., and T. Moran, "Universal Vocoder Using Variable Data Rate Vocoding", Naval Research Lab, NRL/FR/5555-13-10,239, June 2013.

[MELP] Department of Defense Telecommunications Standard, "Analog-to-Digital Conversion of Voice by 2,400 Bit/Second Mixed Excitation Linear Prediction (MELP)", MIL-STD-3005, December 1999.

[MELPE] North Atlantic Treaty Organization (NATO), "The 600 Bit/S, 1200 Bit/S and 2400 Bit/S NATO Interoperable Narrow Band Voice Coder", STANAG No. 4591, January 2006.

[SCIP210] National Security Agency, "SCIP Signaling Plan", SCIP-210, December 2007.

10.2. Informative References

[TSVCIS] National Security Agency, "Tactical Secure Voice Cryptographic Interoperability Specification (TSVCIS) Version 2.1", NSA 09-01A, July 2012.

[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, DOI 10.17487/RFC4585, July 2006, <http://www.rfc-editor.org/info/rfc4585>.

[RFC7201] Westerlund, M. and C. Perkins, "Options for Securing RTP Sessions", RFC 7201, DOI 10.17487/RFC7201, April 2014, <http://www.rfc-editor.org/info/rfc7201>.

[RFC7202] Perkins, C. and M. Westerlund, "Securing the RTP Framework: Why RTP Does Not Mandate a Single Media Security Solution", RFC 7202, DOI 10.17487/RFC7202, April 2014, <http://www.rfc-editor.org/info/rfc7202>.

[RMCAT] IETF, RTP Media Congestion Avoidance Techniques (rmcat) Working Group, <https://datatracker.ietf.org/wg/rmcat/about/>.

Authors' Addresses

Victor Demjanenko, Ph.D.
VOCAL Technologies, Ltd.
520 Lee Entrance, Suite 202
Buffalo, NY 14228
United States of America

Phone: +1 716 688 4675
Email: victor.demjanenko@vocal.com

John Punaro
VOCAL Technologies, Ltd.
520 Lee Entrance, Suite 202
Buffalo, NY 14228

United States of America

Phone: +1 716 688 4675
Email: john.punaro@vocal.com

David Satterlee
VOCAL Technologies, Ltd.
520 Lee Entrance, Suite 202
Buffalo, NY 14228
United States of America

Phone: +1 716 688 4675
Email: david.satterlee@vocal.com

payload
Internet-Draft
Intended status: Standards Track
Expires: July 13, 2018

Reisenbauer
Frequentis
Brandhuber
eurofunk
Hagedorn
Hagedorn
Hoehnsch
T-Systems
Wenk
Frequentis

January 9, 2018

RTP Payload Format for the TETRA Audio Codec
draft-df-stecker-expertenforum-payload-tetra-00

Abstract

This document specifies a Real-time Transport Protocol (RTP) payload format to be used for TETRA encoded speech signals. The payload format is designed to be able to interoperate with existing TETRA transport formats on non-IP networks. This version of the document does not specify a file format for transport of TETRA speech data in storage mode applications such as email as would be required by the IETF. A media type registration is included, specifying the use of the RTP payload format and the storage format.

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 https://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on July 13, 2018.

Copyright Notice

Copyright (c) 2018 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 (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 3
2. Conventions Used In This Document 3
3. Media Format Background 3
4. Payload format 4
4.1. RTP Header Usage 4
4.2. Payload Header 4
4.2.1. I bit: Frame Indicator 4
4.2.2. F bit: Frame Type 5
4.2.3. CTRL: Control bit(5 bits) 5
4.2.4. C bit: Failed Crypto operation indication 5
4.2.5. FRAME_NR: FN (5 bits) 6
4.2.6. R: Audio Signal Relevance (3 bits) 6
4.2.7. S: Spare (7 bits) 6
4.3. Payload Data 6
4.4. Payload layout 7
5. Payload example 7
6. Congestion Control Considerations 8
7. Payload Format Parameters 8
7.1. Media Type Definition 8
8. Mapping to SDP 9
8.1. Offer/Answer Considerations 10
8.2. Declarative SDP Considerations 10
9. IANA Considerations 10
10. Security Considerations 10
11. References 11
11.1. Normative References 11
11.2. Informative References 12
Authors' Addresses 13

1. Introduction

This document specifies the payload format for packetization of TErrestial Trunked Radio (TETRA) encoded speech signals into the Real-time Transport Protocol (RTP) [RFC3550]. The payload format supports transmission of multiple channels, multiple frames per payload, robustness against packet loss, and interoperation with existing TETRA transport formats on non-IP networks, as described in Section Section 3.

The payload format itself is specified in Section Section 4.

2. Conventions Used In This Document

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] when they appear in ALL CAPS. These words may also appear in this document in lower case as plain English words, absent their normative meanings.

The following acronyms are used in this document:

- ETSI: European Telecommunications Standards Institute
- TETRA: TErrestial Trunked Radio

The byte order used in this document is network byte order, i.e., the most significant byte first. The bit order is also the most significant bit first. This is presented in all figures as having the most significant bit leftmost on a line and with the lowest number. Some bit fields may wrap over multiple lines in which cases the bits on the first line are more significant than the bits on the next line.

Best current practices for writing an RTP payload format specification were followed [RFC2736] updated with [RFC8088].

3. Media Format Background

The TETRA codec is used as vocoder for TETRA systems. The TETRA codec is designed for compressing 30ms of audio speech data into 137 bits. The TETRA codec is designed in such a way that on the air interface two of theses 30ms samples are transported together (sub-block 1 and sub-block 2). The codec allows that data of the first 30ms voice frame can be stolen and used for other purposes, e.g. for the exchange of dynamically updated key-material in end-to-end encrypted voice sessions. For E1 lines there are two optional formats defined [3], the first format is called FSTE (First Speech Transport Encoding Format), the other format is called OSTE

(Optimized Speech Transport Encoding Format). These two formats defer mainly insofar that the OSTE format transports an additional 5 bit frame number, which provides timing information from the air interface to the receiving side in order to save the need for buffering due to different transports speed on air and in 64 kbit/s circuit switched networks. The RTP payload format is defined such that the value of this frame number can be transported.

4. Payload format

The RTP payload format is designed in such a way that it can carry the information needed to map the FSTE and OSTE format from [ETSI-TETRA-ISI]. The RTP format is defined such that both of the independent sub-blocks can be transferred separately or together within one RTP frame. Both of them contain the same information in terms of control bits - the information is propagated redundantly. This redundancy is driven by on one hand to simplify the encoding process in direction from E1 to RTP on the other to provide the option to go for either 30ms or 60ms packet size. The redundant information SHALL be propagated consistently equal - otherwise the behavior of the receiver is unspecified. The payload format is chosen such that the TETRA data bits are octet aligned.

4.1. RTP Header Usage

The format of the RTP header is specified in [RFC3550]. The use of the fields of the RTP header by the TETRA payload format is consistent with that specification.

The payload length of TETRA is an integer number of octets; therefore, no padding is necessary.

The timestamp, sequence number, and marker bit (M) of the RTP header are used in accordance with Section 4.1 of [RFC3551].

The RTP payload type for Tetra is to be assigned dynamically.

4.2. Payload Header

4.2.1. I bit: Frame Indicator

1: The following frame contains a first block of two sub-blocks

0: The following frame contains a separated sub-block. A sub-block marked as such could either be a second sub-block, or an independent block, which does not have a relation with any first block. To distinguish between the one and the other the information of the Control bits has to be evaluated.

4.2.2. F bit: Frame Type

Value	Frame contains
0	FSTE encoded data
1	OSTE encoded data

4.2.3. CTRL: Control bit(5 bits)

Ctrl 1..3 according table 2 of [ETSI-TETRA-ISI].

Value	Sub block 1	Sub block 2
000	normal	normal
001	C stolen	normal
010	U stolen	normal
011	C stolen	C stolen
100	C stolen	U stolen
101	U stolen	C stolen
110	U stolen	U stolen
111	O&M ISI block	

Ctrl 4..5 according table 3 of [ETSI-TETRA-ISI].

Value	Sub block 1	Sub block 2
00	BFI no errors	BFI no errors
01	BFI no errors	BFI with error(s)
10	BFI with error(s)	BFI no error(s)
11	BFI with error(s)	BFI with error(s)

NOTE: The meaning of C4 and C5 is outside the scope of the present

4.2.4. C bit: Failed Crypto operation indication

This bit may be set to "1" if an encryption or a decryption operation could not be performed successfully for the specific half-block. Consequently, the encryption status of the half-block audio data is unknown. If a receiver decides to forward the TETRA audio data to OSTE or FSTE or to directly hand over the TETRA audio data to a TETRA audio decoder, the contained audio might be scrambled - depending if

the audio originally was generated as a plain-override half-block or as an encrypted half-block.

4.2.5. FRAME_NR: FN (5 bits)

Those bits contain an uplink frame number as defined in table 8 of [ETSI-TETRA-ISI]. If no frame number is available the FRAME_NR value SHALL be set to 00000.

4.2.6. R: Audio Signal Relevance (3 bits)

The Audio Signal Relevance bits contain information about the Relevance of the voice packet contained here.

R 1

0: no audio signal relevance propagated (R2 and R3 do not contain any valid information)

1: audio signal relevance propagated in R2 and R3

R 2..3 According to table 1 of [BDBOS-BIP20]

value	relevance
00	no audio signal relevance (level ? -72 dBm0)
01	low audio signal relevance (-52dBm0 ? level > -72dBm0)
10	medium audio signal relevance (-32dBm0 ? level > -52dBm0)
11	high audio signal relevance (0dBm0 ? level > -32dBm0)

4.2.7. S: Spare (7 bits)

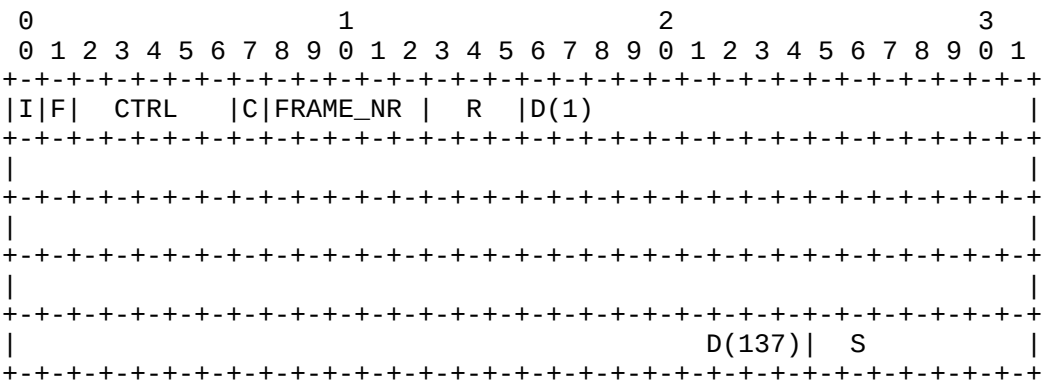
Those bits are reserved for future use and set to "0" currently.

4.3. Payload Data

Reference [ETSI-TETRA-ISI] contains the definition for the generation of the codec data. Data bits D1..D137 in chapter 8 correspond to the "Bit number in speech frame" row of table 4 of [ETSI-TETRA-ISI].

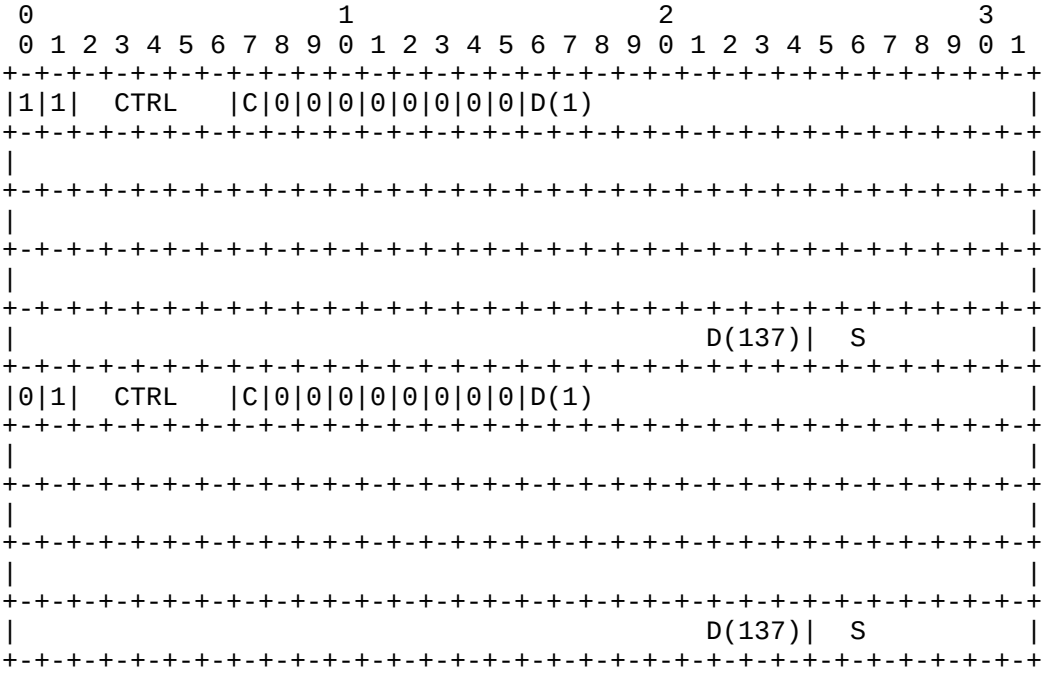
The payload itself contains TETRA ACELP coded speech information encoded according to table 4 of [ETSI-TETRA-Codec].

4.4. Payload layout



5. Payload example

The following example shows how a first and a consecutive 30 ms frame is combined into a single 60ms RTP packet. Note: This example shows of usage of OSTE mapping.



Both halves of information contain exact the same CTRL bits

6. Congestion Control Considerations

Tetra uses a fixed bitrate which cannot be adjusted at all.

Congestion control for RTP SHALL be used in accordance with RFC 3550 [RFC3550], and with any applicable RTP profile; e.g., RFC 3551 [RFC3551]. An additional requirement if best-effort service is being used is: users of this payload format MUST monitor packet loss to ensure that the packet loss rate is within acceptable parameters.

7. Payload Format Parameters

This RTP payload format is identified using one media subtype (audio/TETRA) which is registered in accordance with RFC 4855 [RFC4855] and using the template of RFC 4288 [RFC4288].

7.1. Media Type Definition

The media type for the TETRA codec is expected to be allocated from the IETF tree once this draft turns into an RFC. This media type registration covers both real-time transfer via RTP and non-real-time transfers via stored files.

Media Type name:
   audio
Media Subtype name:
   TETRA
Required parameters:
   none

Optional parameters:

These parameters apply to RTP transfer only.

maxptime:
   The maximum amount of media which can be encapsulated in a payload packet, expressed as time in milliseconds. The time is calculated as the sum of the time that the media present in the packet represents. The time SHOULD be an integer multiple of the frame size. If this parameter is not present, the sender MAY encapsulate any number of speech frames into one RTP packet.
ptime:
   see RFC 4566 [RFC4566].
drgw-fe:
   As long as there is no official RTP payload definition from IETF this proprietary parameter ("digital radio gateway forum of experts") is marked with the only possible value 1. It marks the session to be established according to this specification.

Security considerations: See Section 7 of RFC 4867 [RFC4867].

Interoperability considerations:

Published specification:

Applications that use this media type:

This media type is used in applications needing transport or storage of encoded voice. Some examples include; Voice over IP, streaming media, voice messaging, and voice recording on recording systems.

Intended usage:
COMMON

8. Mapping to SDP

The information carried in the media type specification has a specific mapping to fields in the Session Description Protocol (SDP)[4], which is commonly used to describe RTP sessions. When SDP is used to specify sessions employing the TETRA codec, the mapping is as follows:

Media Type name:
audio
Media subtype name:
TETRA Required parameters:none Optional parameters:none
Mapping MIME Parameters into SDP
The information carried in the MIME media type specification has a specific mapping to fields in the Session Description Protocol [RFC4566], which is commonly used to describe RTP sessions. When SDP is used to specify sessions employing the TETRA codec, the mapping is as follows:

* The MIME type ("audio") goes in SDP "m=" as the media name.
* The MIME subtype (payload format name) goes in SDP "a=rtpmap" as the encoding name. The RTP clock rate in "a=rtpmap" MUST be 8000.
* The parameters "ptime" and "maxptime" go in the SDP "a=ptime" and "a=maxptime" attributes, respectively.
* Any remaining parameters go in the SDP "a=fmtp" attribute by copying them directly from the media type parameter string as a semicolon-separated list of parameter=value pairs.

Here is an example SDP session of usage of TETRA:

```
m=audio 49120 RTP/AVP 99
a=rtpmap:99 TETRA/8000
a=maxptime:60
a=ptime:60
a=fmtp:99
```

8.1. Offer/Answer Considerations

The following considerations apply when using SDP Offer-Answer procedures to negotiate the use of TETRA payload in RTP:

- o In most cases, the parameters "maxptime" and "ptime" will not affect interoperability; however, the setting of the parameters can affect the performance of the application. The SDP offer-answer handling of the "ptime" parameter is described in RFC3264 [RFC3264]. The "maxptime" parameter MUST be handled in the same way.
- o Any unknown parameter in an offer SHALL be removed in the answer.

8.2. Declarative SDP Considerations

For declarative media, the "ptime" and "maxptime" parameter specifies the possible variants used by the sender. Multiple TETRA rtpmap values MAY be used to convey TETRA-coded voice at different packet rates. The receiver can then select an appropriate MELPe codec by using one of the rtpmap values.

9. IANA Considerations

This memo requests that IANA registers [audio/TETRA]. The media type is also requested to be added to the IANA registry for "RTP Payload Format MIME types" (<http://www.iana.org/assignments/rtp-parameters>).

10. 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 systems 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. Other mechanism that may be used are IPsec [RFC4301] and TLS [RFC4346] (RTP over TCP), but also other alternatives may exist.

11. References

11.1. Normative References

[BDBOS-BIP20]
Bundesanstalt fuer den Digitalfunk der Behoerden und Organisationen mit Sicherheitsaufgaben, "BIP 20 QOS Dienstguete-Parameter BOS-Interoperabilitaetsprofil fuer Endgeraete zur Nutzung im Digitalfunk BOS; Version 2014-04 - Revision 2", 2014.

[ETSI-TETRA-Codec]
European Telecommunications Standards Institute, "EN 300 395-2; Terrestrial Trunked Radio (TETRA); Speech codec for full-rate traffic channel; Part 2: TETRA codec V1.3.1", 2005, <http://www.etsi.org/deliver/etsi_en/300300_300399/30039502/01.03.01_60/en_30039502v010301p.pdf>.

[ETSI-TETRA-ISI]
European Telecommunications Standards Institute, "TS 100 392-3-6; Terrestrial Trunked Radio (TETRA); Voice plus Data (V+D); Part 3: Interworking at the Inter-System Interface (ISI); Sub-part 6: Speech format implementation for circuit mode transmission V1.1.1", 2003, <http://www.etsi.org/deliver/etsi_ts/100300_100399/1003920306/01.01.01_60/ts_1003920306v010101p.pdf>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <https://www.rfc-editor.org/info/rfc3550>.

[RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, DOI 10.17487/RFC3551, July 2003, <https://www.rfc-editor.org/info/rfc3551>.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, DOI 10.17487/RFC4566, July 2006, <https://www.rfc-editor.org/info/rfc4566>.

11.2. Informative References

[RFC2736] Handley, M. and C. Perkins, "Guidelines for Writers of RTP Payload Format Specifications", BCP 36, RFC 2736, DOI 10.17487/RFC2736, December 1999, <https://www.rfc-editor.org/info/rfc2736>.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, DOI 10.17487/RFC3264, June 2002, <https://www.rfc-editor.org/info/rfc3264>.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, DOI 10.17487/RFC3711, March 2004, <https://www.rfc-editor.org/info/rfc3711>.

[RFC4288] Freed, N. and J. Klensin, "Media Type Specifications and Registration Procedures", RFC 4288, DOI 10.17487/RFC4288, December 2005, <https://www.rfc-editor.org/info/rfc4288>.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <https://www.rfc-editor.org/info/rfc4301>.

[RFC4346] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1", RFC 4346, DOI 10.17487/RFC4346, April 2006, <https://www.rfc-editor.org/info/rfc4346>.

[RFC4855] Casner, S., "Media Type Registration of RTP Payload Formats", RFC 4855, DOI 10.17487/RFC4855, February 2007, <https://www.rfc-editor.org/info/rfc4855>.

[RFC4867] Sjoberg, J., Westerlund, M., Lakaniemi, A., and Q. Xie, "RTP Payload Format and File Storage Format for the Adaptive Multi-Rate (AMR) and Adaptive Multi-Rate Wideband (AMR-WB) Audio Codecs", RFC 4867, DOI 10.17487/RFC4867, April 2007, <https://www.rfc-editor.org/info/rfc4867>.

[RFC8088] Westerlund, M., "How to Write an RTP Payload Format", RFC 8088, DOI 10.17487/RFC8088, May 2017, <https://www.rfc-editor.org/info/rfc8088>.

Authors' Addresses

Andreas Reisenbauer
Frequentis AG
Innovationsstr. 1
Vienna 1100
Austria

Email: andreas.reisenbauer@frequentis.com

Udo Brandhuber
eurofunk Kappacher GmbH
Germany

Email: ubrandhuber@eurofunk.com

Joachim Hagedorn
Hagedorn Informationssysteme GmbH
Germany

Email: joachim@hagedorn-infosysteme.de

Klaus-Peter Hoehnsch
T-Systems International GmbH
Germany

Email: klaus-peter.hoehnsch@t-systems.com

Stefan Wenk
Frequentis AG
Innovationsstr. 1
Vienna 1100
Austria

Email: stefan.wenk@frequentis.com

PAYLOAD
Internet-Draft
Intended status: Standards Track
Expires: September 6, 2018

M. Zanaty
Cisco
V. Singh
callstats.io
A. Begen
Networked Media
G. Mandyam
Qualcomm Innovation Center

March 5, 2018

RTP Payload Format for Flexible Forward Error Correction (FEC) draft-ietf-payload-flexible-fec-scheme-06

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 FEC repair packets are generated from a set of source packets. These repair packets are sent in a redundancy RTP stream separate from the source RTP stream that carries the source packets. RTP source packets that were lost in transmission can be reconstructed using the source and repair packets that were received. The non-interleaved and interleaved parity codes which are defined in this specification 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 this specification can still work by simply ignoring the FEC repair 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 https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any

time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 6, 2018.

Copyright Notice

Copyright (c) 2018 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 (https://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. Parity Codes 4
1.1.1. 1-D Non-interleaved (Row) FEC Protection 5
1.1.2. 1-D Interleaved (Column) FEC Protection 5
1.1.3. Use Cases for 1-D FEC Protection 6
1.1.4. 2-D (Row and Column) FEC Protection 8
1.1.5. 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 16
5.1. Media Type Registration - Parity Codes 16
5.1.1. Registration of audio/flexfec 16
5.1.2. Registration of video/flexfec 18
5.1.3. Registration of text/flexfec 19
5.1.4. Registration of application/flexfec 20
5.2. Mapping to SDP Parameters 22
5.2.1. Offer-Answer Model Considerations 22
5.2.2. Declarative Considerations 23
6. Protection and Recovery Procedures - Parity Codes 23
6.1. Overview 23
6.2. Repair Packet Construction 24

6.3. Source Packet Reconstruction 25
6.3.1. Associating the Source and Repair Packets 26
6.3.2. Recovering the RTP Header 27
6.3.3. Recovering the RTP Payload 28
6.3.4. Iterative Decoding Algorithm for the 2-D Parity FEC Protection 29
7. SDP Examples 31
7.1. Example SDP for Flexible FEC Protection with in-band SSRC mapping 31
7.2. Example SDP for Flex FEC Protection with explicit signalling in the SDP 31
8. Congestion Control Considerations 32
9. Security Considerations 33
10. IANA Considerations 33
11. Acknowledgments 33
12. Change Log 33
12.1. draft-ietf-payload-flexible-fec-scheme-05 34
12.2. draft-ietf-payload-flexible-fec-scheme-03 34
12.3. draft-ietf-payload-flexible-fec-scheme-02 34
12.4. draft-ietf-payload-flexible-fec-scheme-01 34
12.5. draft-ietf-payload-flexible-fec-scheme-00 34
12.6. draft-singh-payload-1d2d-parity-scheme-00 34
12.7. draft-ietf-fecframe-1d2d-parity-scheme-00 35
13. References 35
13.1. Normative References 35
13.2. Informative References 36
Authors' Addresses 37

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 RTP streams may be communicated in-band or out-of-band. For situations where adaptivitiy 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.

The Redunadncy RTP Stream [RFC7656] repair packets proposed in this document protect the Source RTP Stream packets that belong to the same RTP session.

1.1. Parity Codes

Both the non-interleaved and interleaved parity codes use the eXclusive OR (XOR) operation to generate the repair packets. 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 packets to generate the required number of repair packets.

3. The sender sends the repair packet(s) along with the source packets, in different RTP streams, to the receiver(s). The repair packets may be sent proactively or on-demand based on RTCP feedback messages such as NACK [RFC4585].

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.

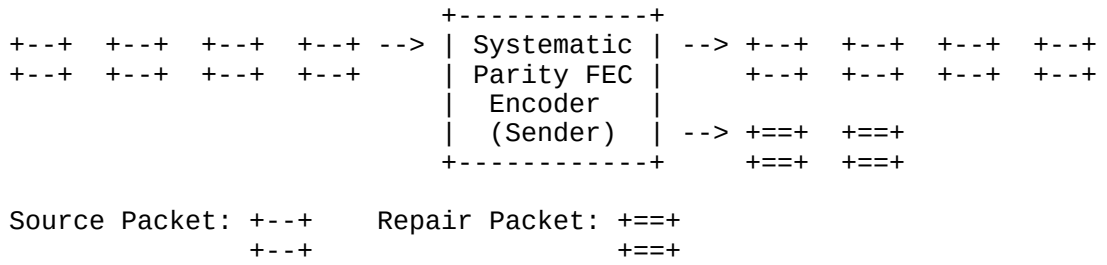


Figure 1: Block diagram for systematic parity FEC encoder

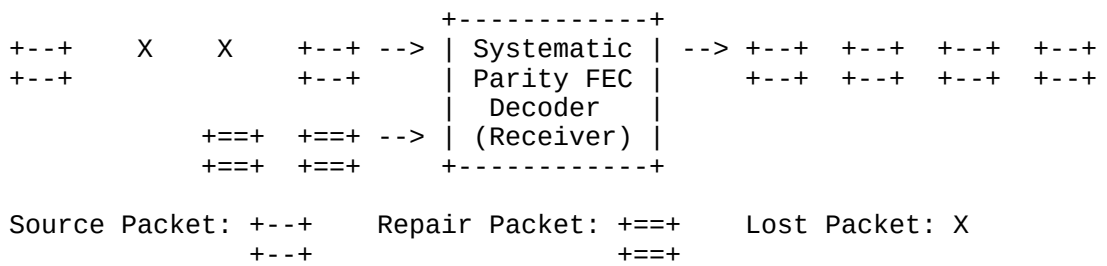


Figure 2: Block diagram for systematic parity FEC decoder

In Figure 2, it is clear that the FEC repair 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 buffer source and repair packets at least for the duration of the repair window, to allow all the repair packets to arrive. 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.

1.1.1. 1-D Non-interleaved (Row) FEC Protection

Suppose that we have a group of D x L source packets that have sequence numbers starting from 1 running to D x 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 repair packets.

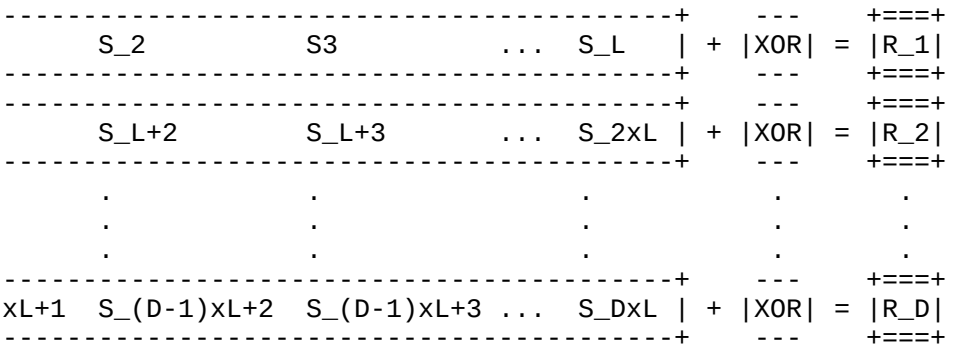


Figure 3: Generating non-interleaved (row) FEC repair packets

1.1.2. 1-D Interleaved (Column) FEC Protection

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 repair packets.

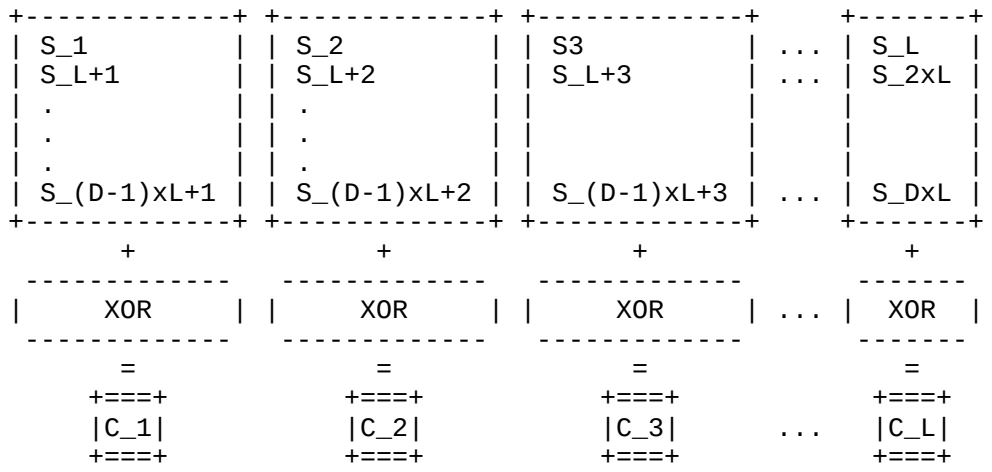


Figure 4: Generating interleaved (column) FEC repair packets

1.1.3. Use Cases for 1-D FEC Protection

A sender may 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 repair packets and a burst loss hits the source packets, the repair operation fails. This is illustrated in Figure 5.

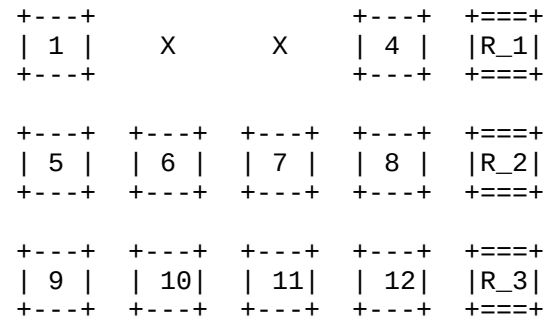


Figure 5: Example scenario where 1-D non-interleaved FEC protection fails error recovery (Burst Loss)

The sender may generate interleaved FEC repair 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 repair packets may still fail to recover the lost data.

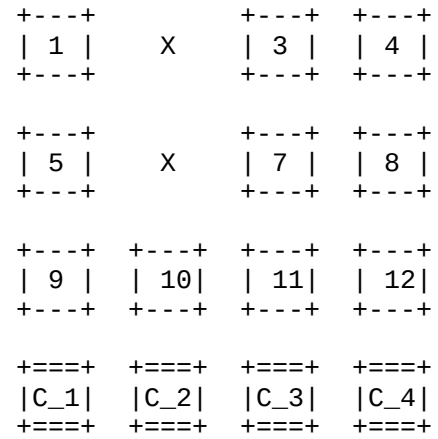


Figure 6: Example scenario where 1-D interleaved FEC protection fails error recovery (Periodic Loss)

1.1.4. 2-D (Row and Column) 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 repair packets. This type of FEC protection is known as 2-D parity FEC protection. At the expense of generating more FEC repair packets, thus increasing the FEC overhead, 2-D FEC provides superior protection against mixed loss patterns. However, it is still possible for 2-D parity FEC protection to fail to recover all of the lost source packets if a particular loss pattern occurs. An example scenario is illustrated in Figure 7.

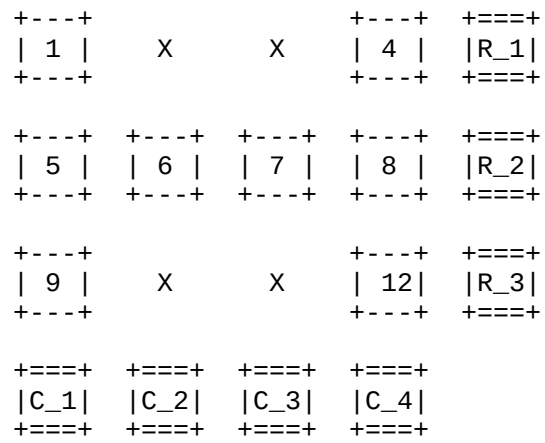


Figure 7: Example scenario #1 where 2-D parity FEC protection fails error recovery

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

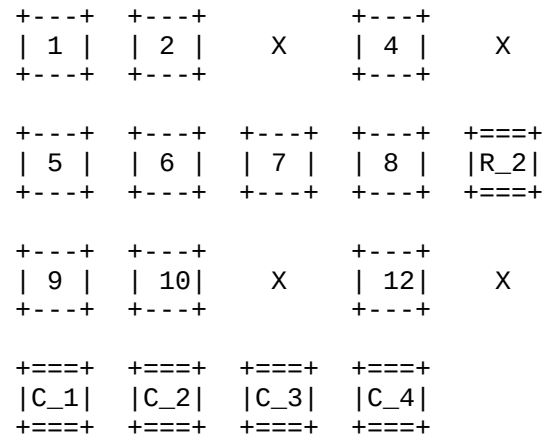


Figure 8: Example scenario #2 where 2-D parity FEC protection fails error recovery

1.1.5. Overhead Computation

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

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

- 1-D Non-interleaved FEC Protection: Overhead = 1/L
- 1-D Interleaved FEC Protection: Overhead = 1/D
- 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

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

4. Packet Formats

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

4.1. Source Packets

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

4.2. Repair Packets

The repair packets MUST contain information that identifies the source block they pertain to and the relationship between the contained repair packets 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.

Note that all the source stream packets that are protected by a particular FEC packet need to be in the same RTP session.

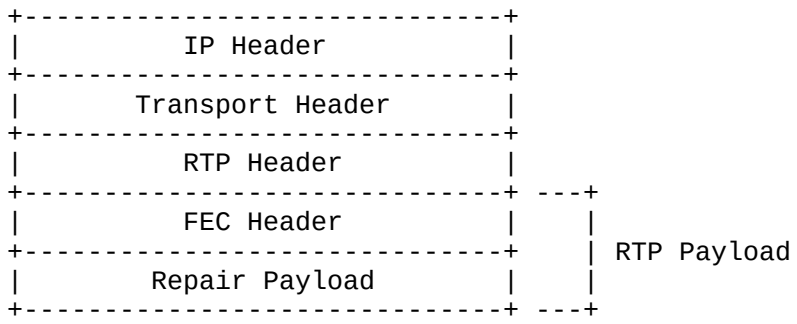


Figure 9: Format of repair packets

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

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

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

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

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

- Synchronization Source (SSRC): The SSRC value for each repair stream SHALL be randomly assigned as suggested by [RFC3550]. This allows the sender to multiplex the source and repair RTP streams on the same port, or multiplex multiple repair streams on a single port. The repair streams SHOULD use the RTCP CNAME field to associate themselves with the source stream.

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 RTP streams means that the RTP Source and the FEC Source MUST share the same CNAME (for this specific source-repair stream 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.

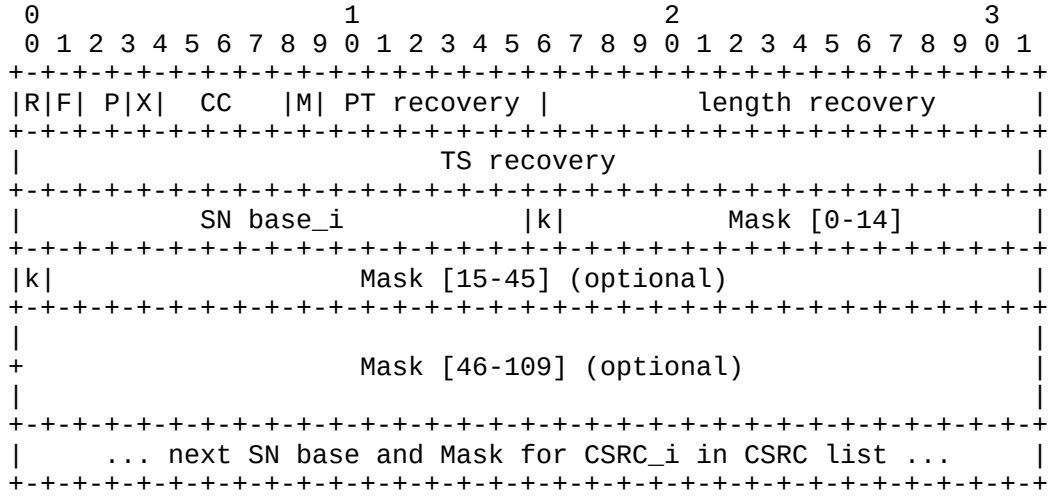


Figure 10: Format of the FEC header

The FEC header consists of the following fields:

- The R bit MUST be set to 1 to indicate a retransmission packet, and MUST be set to 0 for repair packets.

- The F field (1 bit) indicates the type of the mask. Namely:

F bit	Use
0	flexible mask
1	packets indicated by offset M and N

Figure 11: F-bit values

- The P, X, CC, M and PT recovery fields are used to determine the corresponding fields of the recovered packets.
- The Length recovery (16 bits) field is used to determine the length of the recovered packets.
- The TS recovery (32 bits) field is used to determine the timestamp of the recovered packets.
- The CSRC_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 CSRC Count > 1), there will be multiple blocks of data containing the SN base and Mask fields.
- 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 CSRC_i) protected by this repair packet.
- If the F-bit is set to 0, it represents that the source packets of all the SSRCs protected by this particular repair packet are indicated by using a flexible bitmask. Mask is a run-length encoding of packets for a particular CSRC_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 + 1) is protected by this FEC packet.
- The k-bit in the bitmasks indicates if it is 15-, 46-, or a 110-bitmask. k=1 denotes that another mask follows, and k=0 denotes that it is the last block of bit mask.
-

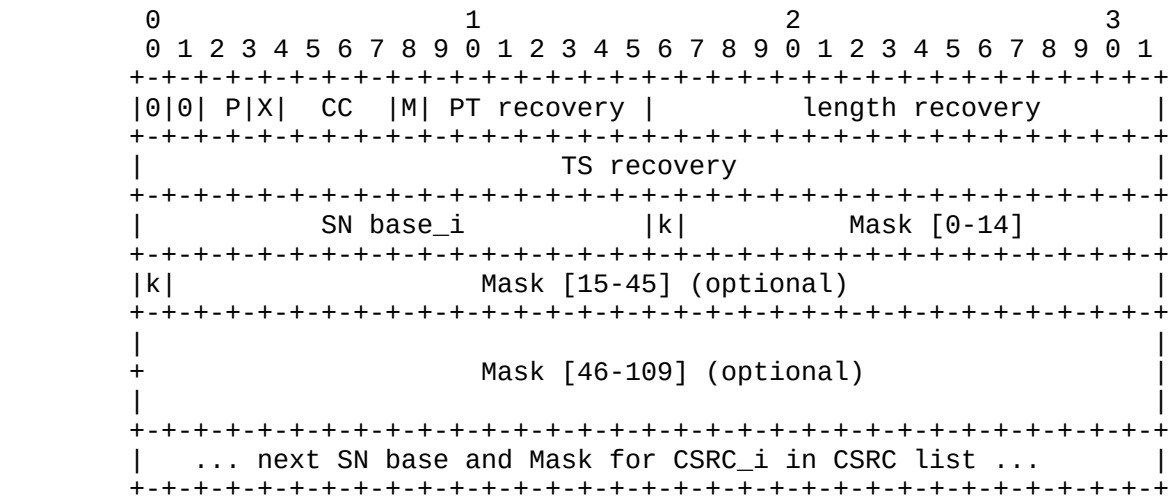


Figure 12: Protocol format for F=0

o If the F-bit is set to 1, it represents that the source packets of all the SSRCs protected by this particular repair packet are indicated by using fixed offsets.

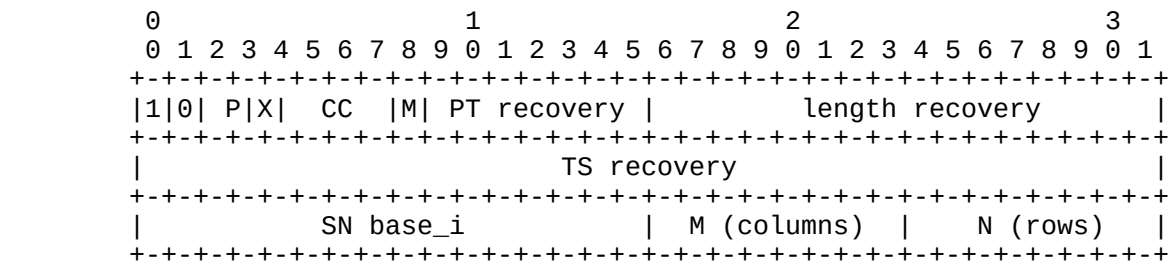


Figure 13: Protocol format for F=1

Consequently, the following conditions occur for M and N values:

```
If M>0, N=0,  is Row FEC, and no column FEC will follow
            Hence, FEC = SN, SN+1, SN+2, ... , SN+(M-1), SN+M.

If M>0, N=1,  is Row FEC, and column FEC will follow.
              Hence, FEC = SN, SN+1, SN+2, ... , SN+(M-1), SN+M.
         and more to come

If M>0, N>1,  indicates column FEC of every M packet
                 in a group of N packets starting at SN base.
              Hence, FEC = SN+(Mx0), SN+(Mx1), ... , SN+(MxN).
```

Figure 14: Interpreting the M and N field values

By setting R to 1, F to 1, this 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.

Note that the parsing of this packet is different. The sequence number (SN base_i) replaces the length recovery in the FEC packet. The CSRC Count (CC) which would be 1, M and N would be set to 0, and the reserved bits from the FEC header are removed. By doing this, we save 64 bits.

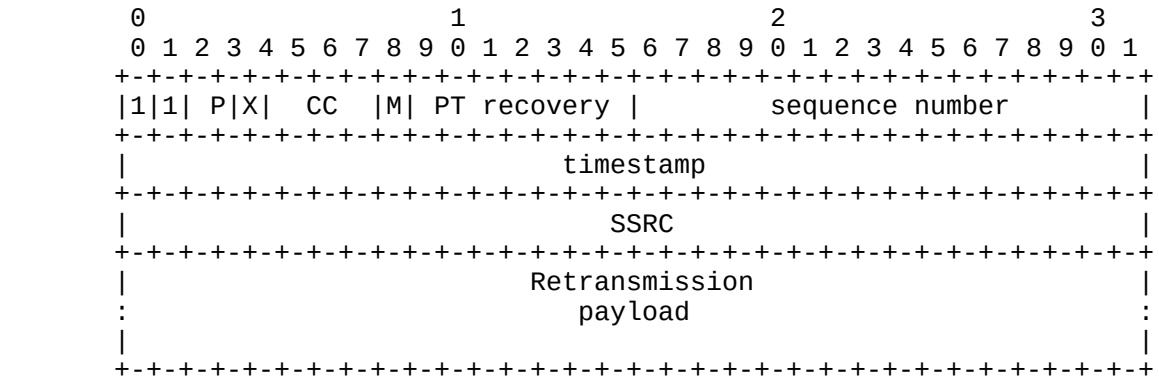


Figure 15: Protocol format for Retransmission

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. If no specific FEC code is specified in the subtype, then the FEC code defaults to the parity code defined in this specification.

5.1. Media Type Registration - Parity Codes

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

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

5.1.1. Registration of audio/flexfec

Type name: audio

Subtype name: flexfec

Required parameters:

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

- 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:

- 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.

- 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.

- 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@callstats.io> 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@callstats.io>.

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:

- 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.

- 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:

- 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.

- 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.

- 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@callstats.io> 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@callstats.io>.

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

Provisional registration? (standards tree only): Yes.

5.1.3. Registration of text/flexfec

Type name: text

Subtype name: flexfec

Required parameters:

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

- 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:

- 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.

- 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.

- 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 <vvarun@callstats.io> 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@callstats.io>.

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:

- 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.

- 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:

- 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.

- 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.

- 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@callstats.io> 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@callstats.io>.

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

Provisional registration? (standards tree only): Yes.

5.2. Mapping to SDP Parameters

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

The mapping of the media type specification for "non-interleaved-parityfec" and "interleaved-parityfec" and their parameters in SDP is as follows:

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

- The media subtype goes into the "a=rtpmap" line as the encoding name. The RTP clock rate parameter ("rate") also goes into the "a=rtpmap" line as the clock rate.

- The remaining required payload-format-specific parameters go into the "a=fmtp" line by copying them directly from the media type string as a semicolon-separated list of parameter=value pairs.

SDP examples are provided in Section 7.

5.2.1. Offer-Answer Model Considerations

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

- Each combination of the L and D parameters produces a different FEC data and is not compatible with any other combination. A sender application may desire to offer multiple offers with different sets of L and D values as long as the parameter values are valid. The receiver SHOULD normally choose the offer that has a sufficient amount of interleaving. If multiple such offers exist, the receiver may choose the offer that has the lowest overhead or the one that requires the smallest amount of buffering. The selection depends on the application requirements.

- The value for the repair-window parameter depends on the L and D values and cannot be chosen arbitrarily. More specifically, L and D values determine the lower limit for the repair-window size. The upper limit of the repair-window size does not depend on the L and D values.

- Although combinations with the same L and D values but with different repair-window sizes produce the same FEC data, such combinations are still considered different offers. The size of the repair-window is related to the maximum delay between the transmission of a source packet and the associated repair packet. This directly impacts the buffering requirement on the receiver side and the receiver must consider this when choosing an offer.

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

5.2.2. Declarative Considerations

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

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

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

6. Protection and Recovery Procedures - Parity Codes

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:

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

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

- The first (most significant) 2 bits in the FEC bit string are skipped. The MSK bits in the FEC header are set to the appropriate value, i.e., it depends on the chosen bitmask length.
- The next bit in the FEC bit string is written into the P recovery bit in the FEC header.
- The next bit in the FEC bit string is written into the X recovery bit in the FEC header.
- The next 4 bits of the FEC bit string are written into the CC recovery field in the FEC header.
- The next bit is written into the M recovery bit in the FEC header.
- The next 7 bits of the FEC bit string are written into the PT recovery field in the FEC header.
- The next 16 bits are skipped.
- The next 32 bits of the FEC bit string are written into the TS recovery field in the FEC header.

- The next 16 bits are written into the length recovery field in the FEC header.

- Depending on the chosen MSK value, the bit mask of appropriate length will be set to the appropriate values.

As described in Section 4.2, the SN base field of the FEC header MUST be set to the lowest sequence number of the source packets protected by this repair packet. When MSK represents a bitmask (MSK=00,01,10), the SN base field corresponds to the lowest sequence number indicated in the bitmask. When MSK=11, the following considerations apply: 1) for the interleaved FEC repair packets, this corresponds to the lowest sequence number of the source packets that forms the column, 2) for the non-interleaved FEC repair 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 stream 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 repair packets, respectively, and

$$0 <= i < X_2$$

where X_2 is set to D and L for the interleaved and non-interleaved FEC repair 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 maked 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 repair 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,..., p*_snb+(M-1), p*_snb+M
When N > 0:
  p*_snb, p*_snb+(Mx1), p*_snb+(Mx2),..., p*_snb+(Mx(N-1)), p*_snb+(MxN)
```

6.3.2. Recovering the RTP Header

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

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

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

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

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

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

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

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

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

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

10. Set the Payload type in the new packet to the next 7 bits in the recovered bit string.

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

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

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

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

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

6.3.3. Recovering the RTP Payload

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

1. Append Y bytes to the new packet.

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

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

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

5. Append the recovered bit string (Y octets) to the new packet generated in Section 6.3.2.

6.3.4. Iterative Decoding Algorithm for the 2-D Parity FEC Protection

In 2-D parity FEC protection, the sender generates both non-interleaved and interleaved FEC repair 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 repair 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 repair 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 repair 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 1.1.4. 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 16.

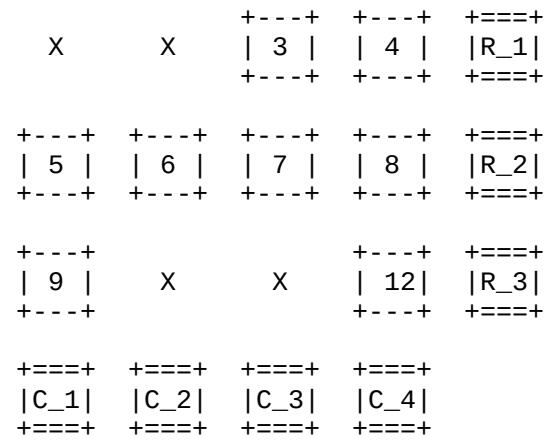


Figure 16: Example loss pattern for the iterative decoding algorithm

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 17.

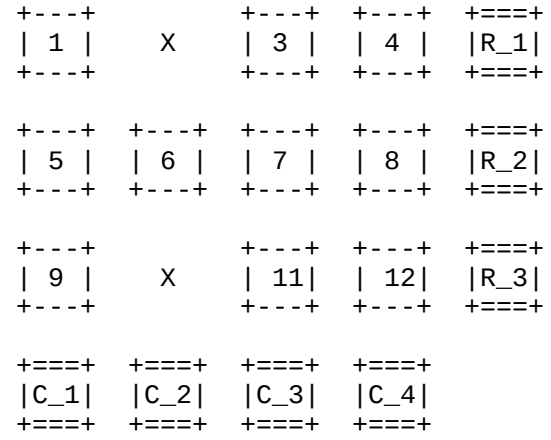


Figure 17: The resulting pattern after the first iteration

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 18.

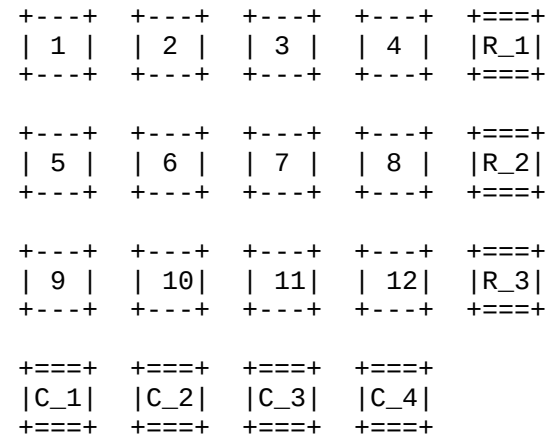


Figure 18: The resulting pattern after the second iteration

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 streams into the network. In particular, in bandwidth-limited networks, FEC repair streams 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.

In a network-friendly implementation, an application SHOULD NOT send/receive FEC repair streams if it knows that sending/receiving those FEC repair streams 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 streams with different levels of FEC protection and the receivers join the corresponding multicast sessions to receive the repair stream(s) that is best for them.

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].

Thanks to Bernard Aboba , Rasmus Brandt , Roni Even , Stefan Holmer , Jonathan Lennox , and Magnus Westerlund for providing valuable feedback on earlier versions of this draft.

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-05

FEC packet format changed as per discussions in IETF97, Seoul.

12.2. draft-ietf-payload-flexible-fec-scheme-03

FEC packet format changed as per discussions in IETF96, Berlin.

Removed section on non-parity codes and flexfec-raptor.

12.3. draft-ietf-payload-flexible-fec-scheme-02

FEC packet format changed as per discussions in IETF94, Tokyo.

Added section on non-parity codes.

Registration of application/flexfec-raptor.

12.4. 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.

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

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

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

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

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

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

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

13. References

13.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, DOI 10.17487/RFC3264, June 2002, <https://www.rfc-editor.org/info/rfc3264>.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <https://www.rfc-editor.org/info/rfc3550>.

[RFC3555] Casner, S. and P. Hoschka, "MIME Type Registration of RTP Payload Formats", RFC 3555, DOI 10.17487/RFC3555, July 2003, <https://www.rfc-editor.org/info/rfc3555>.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, DOI 10.17487/RFC4566, July 2006, <https://www.rfc-editor.org/info/rfc4566>.

[RFC5956] Begen, A., "Forward Error Correction Grouping Semantics in the Session Description Protocol", RFC 5956, DOI 10.17487/RFC5956, September 2010, <https://www.rfc-editor.org/info/rfc5956>.

[RFC6363] Watson, M., Begen, A., and V. Roca, "Forward Error Correction (FEC) Framework", RFC 6363, DOI 10.17487/RFC6363, October 2011, <https://www.rfc-editor.org/info/rfc6363>.

[RFC6709] Carpenter, B., Aboba, B., Ed., and S. Cheshire, "Design Considerations for Protocol Extensions", RFC 6709, DOI 10.17487/RFC6709, September 2012, <https://www.rfc-editor.org/info/rfc6709>.

[RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <https://www.rfc-editor.org/info/rfc6838>.

[RFC7022] Begen, A., Perkins, C., Wing, D., and E. Rescorla, "Guidelines for Choosing RTP Control Protocol (RTCP) Canonical Names (CNAMEs)", RFC 7022, DOI 10.17487/RFC7022, September 2013, <https://www.rfc-editor.org/info/rfc7022>.

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.

[Nagy14] Nagy, M., Singh, V., Ott, J., and L. Eggert, "Congestion Control using FEC for Conversational Multimedia Communication", Proc. of 5th ACM Internation Conference on Multimedia Systems (MMSys 2014) , 3 2014.

[RFC2326] Schulzrinne, H., Rao, A., and R. Lanphier, "Real Time Streaming Protocol (RTSP)", RFC 2326, DOI 10.17487/RFC2326, April 1998, <https://www.rfc-editor.org/info/rfc2326>.

[RFC2733] Rosenberg, J. and H. Schulzrinne, "An RTP Payload Format for Generic Forward Error Correction", RFC 2733, DOI 10.17487/RFC2733, December 1999, <https://www.rfc-editor.org/info/rfc2733>.

[RFC2974] Handley, M., Perkins, C., and E. Whelan, "Session Announcement Protocol", RFC 2974, DOI 10.17487/RFC2974, October 2000, <https://www.rfc-editor.org/info/rfc2974>.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, DOI 10.17487/RFC3711, March 2004, <https://www.rfc-editor.org/info/rfc3711>.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <https://www.rfc-editor.org/info/rfc4301>.

[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, DOI 10.17487/RFC4585, July 2006, <https://www.rfc-editor.org/info/rfc4585>.

[RFC5109] Li, A., Ed., "RTP Payload Format for Generic Forward Error Correction", RFC 5109, DOI 10.17487/RFC5109, December 2007, <https://www.rfc-editor.org/info/rfc5109>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <https://www.rfc-editor.org/info/rfc5246>.

[RFC7656] Lennox, J., Gross, K., Nandakumar, S., Salgueiro, G., and B. Burman, Ed., "A Taxonomy of Semantics and Mechanisms for Real-Time Transport Protocol (RTP) Sources", RFC 7656, DOI 10.17487/RFC7656, November 2015, <https://www.rfc-editor.org/info/rfc7656>.

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

Authors' Addresses

Mo Zanaty
Cisco
Raleigh, NC
USA

Email: mzanaty@cisco.com

Varun Singh
CALLSTATS I/O Oy
Runeberginkatu 4c A 4
Helsinki 00100
Finland

Email: varun.singh@iki.fi
URI: http://www.callstats.io/

Ali Begen
Networked Media
Konya
Turkey

Email: ali.begen@networked.media

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

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

Payload Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 6, 2018

J. Uberti
S. Holmer
M. Flodman
Google
J. Lennox
D. Hong
Vidyo

March 5, 2018

RTP Payload Format for VP9 Video
draft-ietf-payload-vp9-05

Abstract

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

Status of This Memo

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

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

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

This Internet-Draft will expire on September 6, 2018.

Copyright Notice

Copyright (c) 2018 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 3
3. Media Format Description 3
4. Payload Format 5
4.1. RTP Header Usage 5
4.2. VP9 Payload Description 7
4.2.1. Scalability Structure (SS): 11
4.3. VP9 Payload Header 13
4.4. Frame Fragmentation 13
4.5. Scalable encoding considerations 13
4.6. Examples of VP9 RTP Stream 14
4.6.1. Reference picture use for scalable structure 14
5. Feedback Messages and Header Extensions 15
5.1. Reference Picture Selection Indication (RPSI) 15
5.2. Slice Loss Indication (SLI) 15
5.3. Full Intra Request (FIR) 16
5.4. Layer Refresh Request (LRR) 16
5.5. Frame Marking 17
6. Payload Format Parameters 17
6.1. Media Type Definition 18
6.2. SDP Parameters 19
6.2.1. Mapping of Media Subtype Parameters to SDP 19
6.2.2. Offer/Answer Considerations 20
7. Security Considerations 20
8. Congestion Control 20
9. IANA Considerations 20
10. References 21
10.1. Normative References 21
10.2. Informative References 22
Authors’ Addresses 22

1. Introduction

This memo describes an RTP payload specification applicable to the transmission of video streams encoded using the VP9 video codec [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 [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].

TODO: Cite terminology from [VP9-BITSTREAM].

3. Media Format Description

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

VP9 also allows a frame to use another frame of a different resolution as a reference frame. (Specifically, a frame may use any references whose width and height are between 1/16th that of the current frame and twice that of the current frame, inclusive.) 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.

Temporal and spatial scalability layers are associated with non-negative integer IDs. The lowest layer of either type has an ID of 0, and is sometimes referred to as the "base" temporal or spatial layer.

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

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

Within a picture, a frame with spatial layer ID equal to S, where S > 0, can depend on a frame of the same picture with a lower spatial layer ID. This "inter-layer" dependency can result in additional coding gain compared to the case where only traditional "inter-picture" dependency is used, where a frame depends on previously coded frame in time. For simplicity, this payload format assumes that, within a picture and if inter-layer dependency is used, a spatial layer S frame can depend only on the immediately previous spatial layer S-1 frame, when S > 0. Additionally, if inter-picture dependency is used, a spatial layer S frame is assumed to only depend on a previously coded spatial layer S frame.

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 the spatial layer S-1 frame. Given the D bit, a receiver only needs to additionally 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 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 identify 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. 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 Picture Group (PG) MUST be pre-specified as part of the scalability structure (SS) data. In this mode, each packet has an index to refer to one of the described pictures in the PG, from which the pictures referenced by the picture transmitted in the current packet for inter-picture prediction can be identified.

(Editor’s Note: A "Picture Group", as used in this document, is not the same thing as a the term "Group of Pictures" as it is traditionally used in video coding, i.e. to mean an independently-

decoadable run of pictures beginning with a keyframe. Suggestions for better terminology are welcome.)

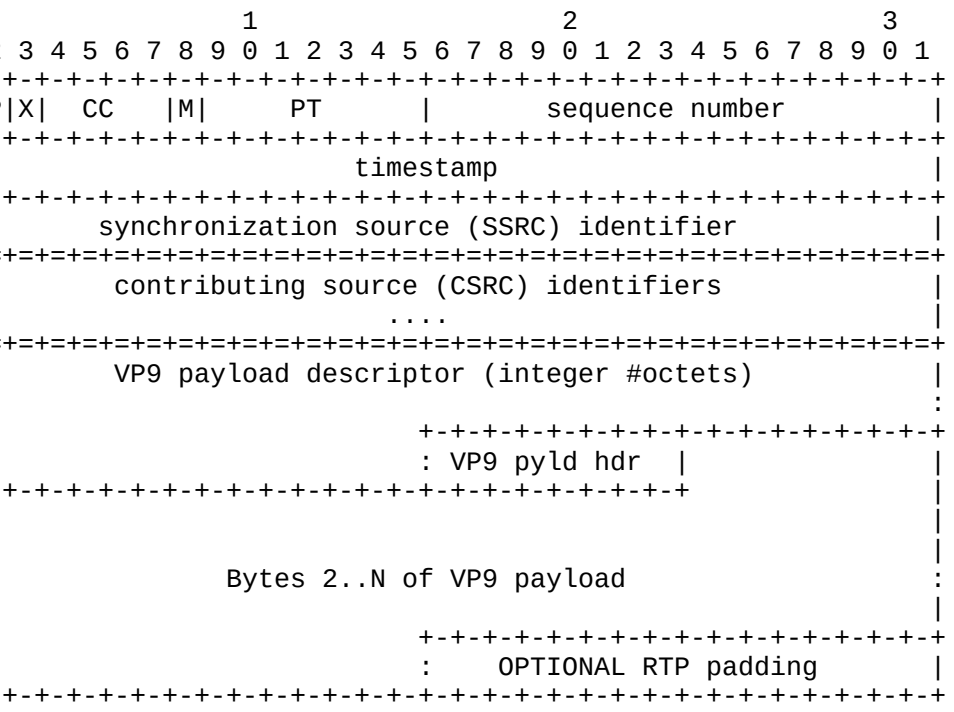
The SS data can also be used to specify the resolution of each spatial layer present in the VP9 stream for both flexible and non-flexible modes.

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 Section 4.2 and Section 4.3. OPTIONAL RTP padding MUST NOT be included unless the P bit is set. The figure specifically shows the format for the first packet in a frame. Subsequent packets will not contain the VP9 payload header, and will have later octets in the frame payload.

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 picture), and 0 otherwise. Unless spatial scalability is in use for this picture, this will have the same value as the E bit described below. Note this bit MUST be set to 1 for the target spatial layer frame if a stream is being rewritten to remove higher spatial layers.

Payload Type (PT): In line with the policy in Section 3 of [RFC3551], applications using the VP9 RTP payload profile MUST assign a dynamic payload type number to be used in each RTP session and provide a mechanism to indicate the mapping. See

Section 6.2 for the mechanism to be used with the Session Description Protocol (SDP) [RFC4566].

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

If a frame has the VP9 show_frame field set to 0 (i.e., it is meant only to populate a reference buffer, without being output) its timestamp MAY alternately be set to be the same as the subsequent frame with show_frame equal to 1. (This will be convenient for playing out pre-encoded content packaged with VP9 "superframes", which typically bundle show_frame==0 frames with a subsequent show_frame==1 frame.) Every frame with show_frame==1, however, MUST have a unique timestamp modulo the 2^32 wrap of the field.

The remaining RTP Fixed Header Fields (V, P, X, CC, sequence number, SSRC and CSRC identifiers) are used as specified in Section 5.1 of [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.

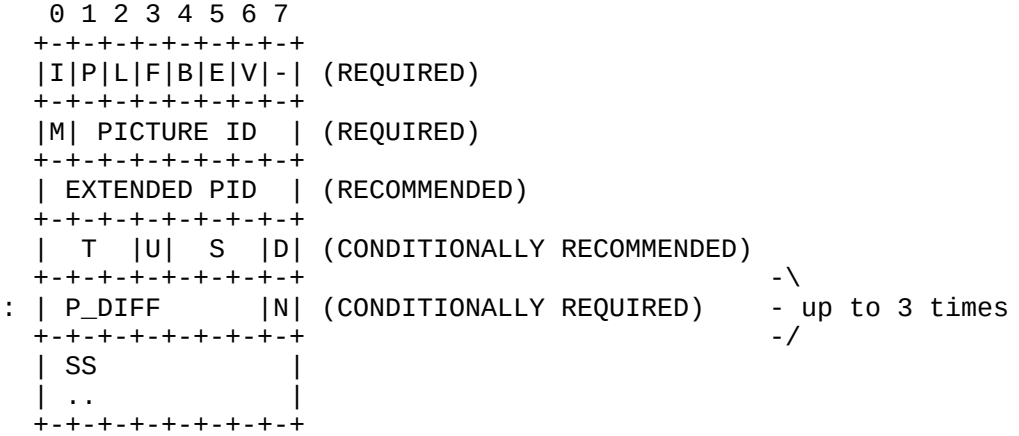


Figure 2

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.

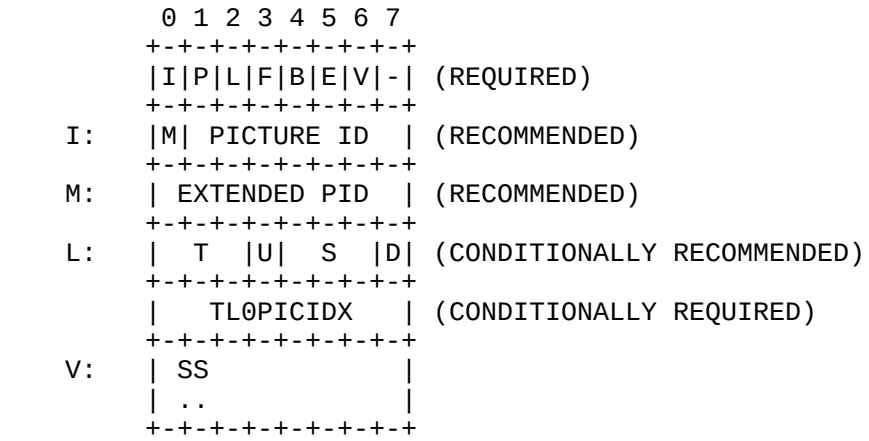


Figure 3

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. If the SS field was present in the stream's most recent start of a keyframe (i.e., non-flexible scalability mode is in use), then the PID MUST also be present in every packet.

P: Inter-picture predicted frame. When set to zero, the frame does not utilize inter-picture prediction. In this case, up-switching to a 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.ietf-avtext-lrr] message (see Section 5.4). When P is set to zero, the T field (described below) MUST also be set to 0 (if present). Note that the P bit does not forbid intra-picture, inter-layer prediction from earlier frames of the same picture, if any.

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 1 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 MUST only change on the first packet of a key picture. A key picture is a picture whose base spatial layer frame is a key frame, and which thus completely resets the encoder state. This packet will have its P bit equal to zero, S or D bit (described below) equal to zero, and B bit (described below) equal to 1.

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

E: End of a frame. MUST be set to 1 for the final RTP packet of a VP9 frame, and 0 otherwise. This enables a decoder to finish decoding the frame, where it otherwise may need to wait for the next packet to explicitly know that the frame is complete. Note that, if spatial scalability is in use, more frames from the same picture 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 in network byte order. The sender may choose between a 7- or 15-bit 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.

In the non-flexible mode (when the F bit is set to 0), this PID is used as an index to the picture group (PG) specified in the SS data below. In this mode, the PID of the key frame corresponds to the first specified frame in the PG. Then subsequent PIDs are mapped to subsequently specified frames in the PG (modulo N_G, specified in the SS data below), respectively.

All frames of the same picture MUST have the same PID value.

Frames (and their corresponding pictures) with the VP9 show_frame field equal to 0 MUST have distinct PID values from subsequent pictures with show_frame equal to 1. Thus, a Picture as defined in this specification is different than a VP9 Superframe.

All frames of the same picture MUST have the same value for show_frame.

Layer indices: This information is optional but recommended whenever encoding with layers. For both flexible and non-flexible modes, one octet is used to specify a layer frame’s temporal layer ID (T) and spatial layer ID (S) as shown both in Figure 2 and Figure 3. 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 frame.

In the non-flexible mode (when the F bit is set to 0), another octet is used to represent temporal layer 0 index (TL0PICIDX), as depicted in Figure 3. The TL0PICIDX is present so that all minimally required frames - the base temporal layer frames - can be tracked.

The T and S fields indicate the temporal and spatial layers and can help middleboxes and and endpoints quickly identify which layer a packet belongs to.

T: The temporal layer ID of current frame. In the case of non-flexible mode, if PID is mapped to a picture in a specified PG, then the value of T MUST match the corresponding T value of the mapped picture in the PG.

U: Switching up point. If this bit is set to 1 for the current picture with temporal layer ID equal to T, then "switch up" to a higher frame rate is possible as subsequent higher temporal layer pictures will not depend on any picture before the

current picture (in coding order) 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 picture. Different frames of the same picture MUST have distinct spatial layer IDs, and frames' spatial layers MUST appear in increasing order within the 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 picture. MUST only be set to zero if current spatial layer S frame does not depend on spatial layer S-1 frame of the same picture. For the base layer frame (with S equal to 0), this D bit MUST be set to zero.

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 pictures, i.e., the pictures with T set to 0. If T is larger than 0, TL0PICIDX indicates which temporal base layer picture the current picture depends on. TL0PICIDX MUST be incremented when T is equal to 0. The index SHOULD start on a random number, and MUST restart at 0 after reaching the maximum number 255.

Reference indices: When P and F are both set to one, indicating a non-key frame in flexible mode, 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 (in 7 bits) specified as the relative PID from the current picture. For example, when P_DIFF=3 on a packet containing the picture with PID 112 means that the picture refers back to the picture with PID 109. This calculation is done modulo the size of the PID field, i.e., either 7 or 15 bits.

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 frame within a picture as well as the inter-picture dependencies for a picture group (PG). 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.

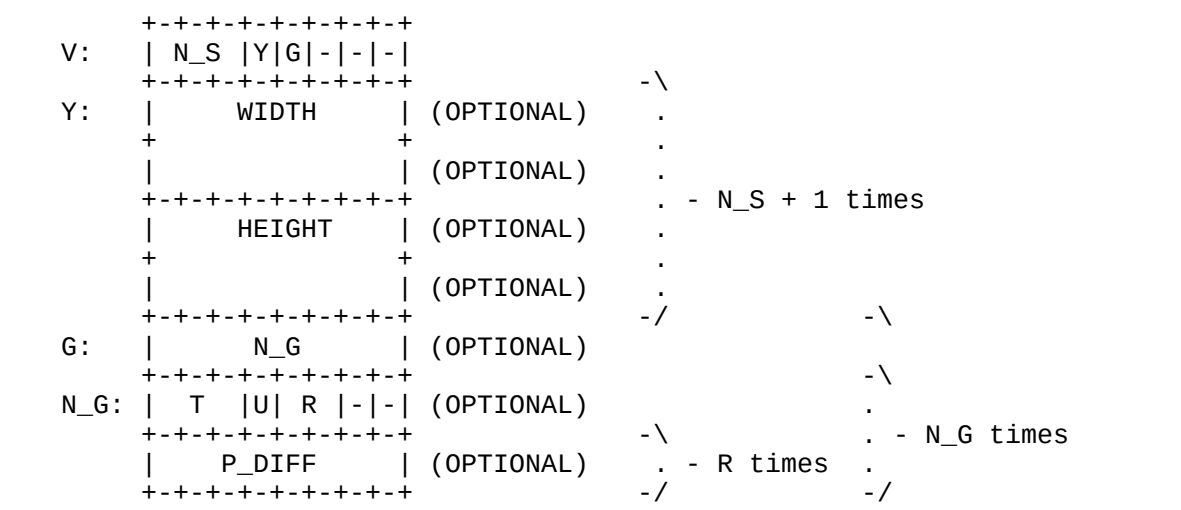


Figure 4

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.

G: PG description present flag.

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

N_G: N_G indicates the number of pictures in a Picture Group (PG). 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 pictures in the PG, each picture’s temporal layer ID (T), switch up point (U), and the R reference indices (P_DIFFs) are specified.

The first picture specified in the PG MUST have T set to 0.

G set to 0 or N_G set to 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 picture, all 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, and B bit equal to 1. The SS data MUST only be changed on the picture that corresponds to the first picture specified in the previous SS data's PG (if the previous SS data's N_G was greater than 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 E bit set. There is no mechanism for finer-grained access to parts of a VP9 frame.

4.5. Scalable encoding considerations

In addition to the use of reference frames, VP9 has several additional forms of inter-frame dependencies, largely involving probability tables for the entropy and tree encoders. In VP9 syntax, the syntax element "error_resilient_mode" resets this additional inter-frame data, allowing a frame's syntax to be decoded independently.

Due to the requirements of scalable streams, a VP9 encoder producing a scalable stream needs to ensure that a frame does not depend on a previous frame (of the same or a previous picture) that can legitimately be removed from the stream. Thus, a frame that follows a removable frame (in full decode order) MUST be encoded with "error_resilient_mode" to true.

For spatially-scalable streams, this means that "error_resilient_mode" needs to be turned on for the base spatial layer; it can however be turned off for higher spatial layers,

assuming they are sent with inter-layer dependency (i.e. with the "D" bit set). For streams that are only temporally-scalable without spatial scalability, "error_resilient_mode" can additionally be turned off for any picture that immediately follows a temporal layer 0 frame.

4.6. Examples of VP9 RTP Stream

TODO: Examples of packet layouts

4.6.1. Reference picture use for scalable structure

As discussed in Section 3, the VP9 codec can maintain up to eight reference frames, of which up to three can be referenced or updated by any new frame. This section illustrates one way that a scalable structure (with three spatial layers and three temporal layers) can be constructed using these reference frames.

Temporal	Spatial	References	Updates
0	0	0	0
0	1	0,1	1
0	2	1,2	2
2	0	0	6
2	1	1,6	7
2	2	2,7	-
1	0	0	3
1	1	1,3	4
1	2	2,4	5
2	0	3	6
2	1	4,6	7
2	2	5,7	-

Example scalability structure

This structure is constructed such that the "U" bit can always be set.

5. Feedback Messages and Header Extensions

5.1. Reference Picture Selection Indication (RPSI)

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 for point to point (unicast) communication. 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.

Note: because all frames of the same picture must have the same inter-picture reference structure, there is no need for a message to specify which frame is being selected.

5.2. Slice Loss Indication (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]:

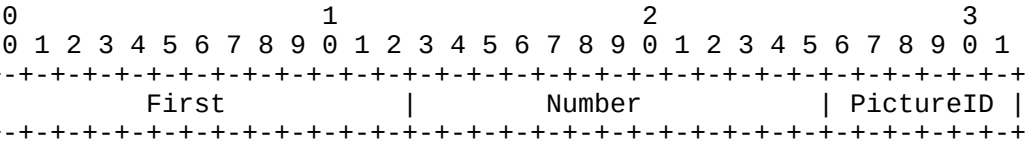


Figure 5

Here, First is the macroblock address (in scan order) of the first lost block and Number is the number of lost blocks, as defined in [RFC4585]. 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 part 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. Full Intra Request (FIR)

The Full Intra Request (FIR) [RFC5104] RTCP feedback message allows a receiver to request a full state refresh of an encoded stream.

Upon receipt of an FIR request, a VP9 sender MUST send a picture with a keyframe for its spatial layer 0 layer frame, and then send frames without inter-picture prediction (P=0) for any higher layer frames.

5.4. Layer Refresh Request (LRR)

The Layer Refresh Request [I-D.ietf-avtext-lrr] allows a receiver to request a single layer of a spatially or temporally encoded stream to be refreshed, without necessarily affecting the stream’s other layers.

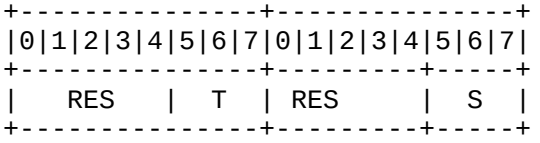


Figure 6

Figure 6 shows the format of LRR’s layer index fields for VP9 streams. The two "RES" fields MUST be set to 0 on transmission and ingnored on reception. See Section 4.2 for details on the T and S fields.

Identification of a layer refresh frame can be derived from the reference IDs of each frame by backtracking the dependency chain until reaching a point where only decodable frames are being

referenced. Therefore it's recommended for both the flexible and the non-flexible mode that, when upgrade frames are being encoded in response to a LRR, those packets should contain layer indices and the reference fields so that the decoder or an MCU can make this derivation.

Example:

LRR {1,0}, {2,1} is sent by an MCU when it is currently relaying {1,0} to a receiver and which wants to upgrade to {2,1}. In response the encoder should encode the next frames in layers {1,1} and {2,1} by only referring to frames in {1,0}, or {0,0}.

In the non-flexible mode, periodic upgrade frames can be defined by the layer structure of the SS, thus periodic upgrade frames can be automatically identified by the picture ID.

5.5. Frame Marking

The Frame Marking RTP header extension [I-D.ietf-avtext-framemarking] is a mechanism to provide information about frames of video streams in a largely codec-independent manner. However, for its extension for scalable codecs, the specific manner in which codec layers are identified needs to be specified specifically for each codec. This section defines how frame marking is used with VP9.

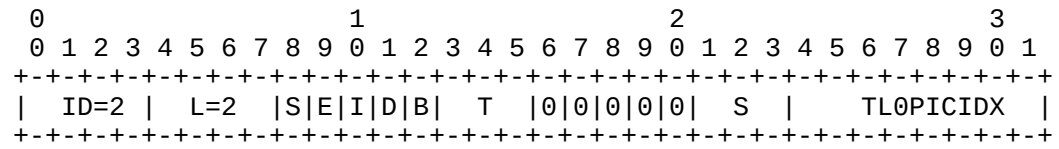


Figure 7

When this header extension is used with VP9, the T and S fields MUST match the values in the packet which the header extension is attached to; see Section 4.2 for details on these fields.

See [I-D.ietf-avtext-framemarking] for explanations of the other fields, which are generic.

6. Payload Format Parameters

This payload format has two optional 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: None.

Optional parameters:
These parameters are used to signal the capabilities of a receiver implementation. If the implementation is willing to receive media, both parameters MUST be provided. 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 int(sqrt(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 [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:

- The media name in the "m=" line of SDP MUST be video.
- The encoding name in the "a=rtpmap" line of SDP MUST be VP9 (the media subtype).
- The clock rate in the "a=rtpmap" line MUST be 90000.
- The parameters "max-fs", and "max-fr", MUST be included in the "a=fmtp" line of SDP 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 such as RTP/AVP [RFC3551], RTP/AVPF [RFC4585], RTP/SAVP [RFC3711], or RTP/SAVPF [RFC5124]. SAVPF [RFC5124]. However, as "Securing the RTP Protocol Framework: Why RTP Does Not Mandate a Single Media Security Solution" [RFC7202] discusses, it is not an RTP payload format's responsibility to discuss or mandate what solutions are used to meet the basic security goals like confidentiality, integrity and source authenticity for RTP in general. This responsibility lays on anyone using RTP in an application. They can find guidance on available security mechanisms in Options for Securing RTP Sessions [RFC7201]. Applications SHOULD use one or more appropriate strong security mechanisms. The rest of this security consideration section discusses the security impacting properties of the payload format itself.

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.ietf-avtext-framemarking]
Zanaty, M., Berger, E., and S. Nandakumar, "Frame Marking RTP Header Extension", draft-ietf-avtext-framemarking-06 (work in progress), October 2017.

[I-D.ietf-avtext-lrr]
Lennox, J., Hong, D., Uberti, J., Holmer, S., and M. Flodman, "The Layer Refresh Request (LRR) RTCP Feedback Message", draft-ietf-avtext-lrr-07 (work in progress), July 2017.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <https://www.rfc-editor.org/info/rfc3550>.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, DOI 10.17487/RFC4566, July 2006, <https://www.rfc-editor.org/info/rfc4566>.

[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, DOI 10.17487/RFC4585, July 2006, <https://www.rfc-editor.org/info/rfc4585>.

[RFC4855] Casner, S., "Media Type Registration of RTP Payload Formats", RFC 4855, DOI 10.17487/RFC4855, February 2007, <https://www.rfc-editor.org/info/rfc4855>.

[RFC5104] Wenger, S., Chandra, U., Westerlund, M., and B. Burman, "Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)", RFC 5104, DOI 10.17487/RFC5104, February 2008, <https://www.rfc-editor.org/info/rfc5104>.

[RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <https://www.rfc-editor.org/info/rfc6838>.

[VP9-BITSTREAM]
Grange, A., de Rivaz, P., and J. Hunt, "VP9 Bitstream & Decoding Process Specification", Version 0.6, March 2016, <https://storage.googleapis.com/downloads.webmproject.org/docs/vp9/vp9-bitstream-specification-v0.6-20160331-draft.pdf>.

10.2. Informative References

[RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, DOI 10.17487/RFC3551, July 2003, <https://www.rfc-editor.org/info/rfc3551>.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, DOI 10.17487/RFC3711, March 2004, <https://www.rfc-editor.org/info/rfc3711>.

[RFC5124] Ott, J. and E. Carrara, "Extended Secure RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/SAVPF)", RFC 5124, DOI 10.17487/RFC5124, February 2008, <https://www.rfc-editor.org/info/rfc5124>.

[RFC7201] Westerlund, M. and C. Perkins, "Options for Securing RTP Sessions", RFC 7201, DOI 10.17487/RFC7201, April 2014, <https://www.rfc-editor.org/info/rfc7201>.

[RFC7202] Perkins, C. and M. Westerlund, "Securing the RTP Framework: Why RTP Does Not Mandate a Single Media Security Solution", RFC 7202, DOI 10.17487/RFC7202, April 2014, <https://www.rfc-editor.org/info/rfc7202>.

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