Network Working Group
Request for Comments: 5404
Category: Standards Track

M. Westerlund
I. Johansson
Ericsson AB

January 2009

RTP Payload Format for G.719

Status of This Memo

Copyright Notice

Abstract

This document specifies the payload format for packetization of the G.719 full-band codec encoded audio signals into the Real-time Transport Protocol (RTP). The payload format supports transmission of multiple channels, multiple frames per payload, and interleaving.

Table of Contents

1. Introduction

This document specifies the payload format for packetization of the G.719 full-band (FB) codec encoded audio signals into the Real-time Transport Protocol (RTP) [RFC3550]. The payload format supports transmission of multiple channels, multiple frames per payload, and packet loss robustness methods using redundancy or interleaving.

This document starts with conventions, a brief description of the codec, and the payload format's capabilities. The payload format is specified in Section 5. Examples can be found in Section 6. The media type and its mappings to the Session Description Protocol (SDP) and usage in SDP offer/answer are then specified. The document ends with considerations regarding congestion control and security.

2. Definitions and Conventions

The term "frame-block" is used in this document to describe the time-synchronized set of audio frames in a multi-channel audio session. In particular, in an N-channel session, a frame-block will contain N audio frames, one from each of the channels, and all N speech frames represent exactly the same time period.

This document contains depictions of bit fields. The most significant bit is always leftmost in the figure on each row and has the lowest enumeration. For fields that are depicted over multiple rows, the upper row is more significant than the next.

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].

3. G.719 Description

The ITU-T G.719 full-band codec is a transform coder based on Modulated Lapped Transform (MLT). G.719 is a low-complexity full-bandwidth codec for conversational speech and audio coding. The encoder input and decoder output are sampled at 48 kHz. The codec enables full-bandwidth from 20 Hz to 20 kHz, encoding of speech, music, and general audio content at rates from 32 kbit/s up to 128 kbit/s. The codec operates on 20-ms frames and has an algorithmic delay of 40 ms.

The codec provides excellent quality for speech, music, and other types of audio. Some of the applications for which this coder is suitable are:

- Real-time communications such as video conferencing and telephony
- Streaming audio
- Archival and messaging

The encoding and decoding algorithm can change the bitrate at any 20-ms frame boundary. The encoder receives the audio sampled at 48 kHz. The support of other sampling rates is possible by re-sampling the input signal to the codec's sampling rate, i.e., 48 kHz; however, this functionality is not part of the standard.

The encoding is performed on equally sized frames. For each frame, the encoder decides between two encoding modes, a transient mode and a stationary mode. The decision is based on statistics derived from the input signal. The stationary mode uses a long MLT that leads to a spectrum of 960 coefficients, while the transient encoding mode uses a short MLT (higher time resolution transform) that results in 4 spectra (4 x 240 = 960 coefficients). The encoding of the spectrum is done in two steps. First, the spectral envelope is computed, quantized, and Huffman encoded. The envelope is computed on a non-uniform frequency subdivision. From the coded spectral envelope, a weighted spectral envelope is derived and is used for bit allocation; this process is also repeated at the decoder. Thus, only the spectral envelope is transmitted. The output of the bit allocation is used in order to quantize the spectra. In addition, for stationary frames, the encoder estimates the amount of noise level. The decoder applies the reverse operation upon reception of the bit stream. The non-coded coefficients (i.e., no bits allocated) are replaced by entries of a noise codebook that is built based on the decoded coefficients.

4. Payload Format Capabilities

This payload format has a number of capabilities, and this section discusses them in some detail.

4.1. Multi-Rate Encoding and Rate Adaptation

G.719 supports a multi-rate encoding capability that enables on a per-frame basis variation of the encoding rate. This enables support for bitrate adaptation and congestion control. The possibility to aggregate multiple audio frames into a single RTP payload is another dimension of adaptation. The RTP and payload format overhead can thus be reduced by the aggregation at the cost of increased delay and reduced packet-loss robustness.

4.2. Support for Multi-Channel Sessions

The RTP payload format defined in this document supports multi-channel audio content (e.g., stereophonic or surround audio sessions). Although the G.719 codec itself does not support encoding of multi-channel audio content into a single bit stream, it can be used to separately encode and decode each of the individual channels. To transport (or store) the separately encoded multi-channel content, the audio frames for all channels that are framed and encoded for the same 20-ms period are logically collected in a "frame-block".

At the session setup, out-of-band signaling must be used to indicate the number of channels in the payload type. The order of the audio frames within the frame-block depends on the number of the channels and follows the definition in Section 4.1 of the RTP/AVP profile [RFC3551]. When using SDP for signaling, the number of channels is specified in the rtpmap attribute.

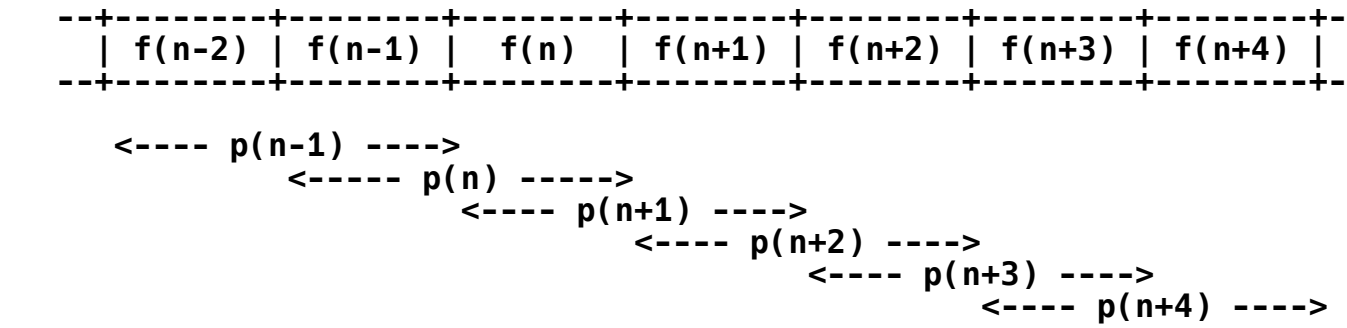
4.3. Robustness against Packet Loss

The payload format supports several means, including forward error correction (FEC) and frame interleaving, to increase robustness against packet loss.

4.3.1. Use of Forward Error Correction (FEC)

Generic forward error correction within RTP is defined, for example, in RFC 5109 [RFC5109]. Audio redundancy coding is defined in RFC 2198 [RFC2198]. Either scheme can be used to add redundant information to the RTP packet stream and make it more resilient to packet losses, at the expense of a higher bitrate. Please see either of the RFCs for a discussion of the implications of the higher bitrate to network congestion.

In addition to these media-unaware mechanisms, this memo specifies a G.719-specific form of audio redundancy coding, which may be beneficial in terms of packetization overhead. Conceptually, previously transmitted transport frames are aggregated together with new ones. A sliding window can be used to group the frames to be sent in each payload. However, irregular or non-consecutive patterns are also possible by inserting NO_DATA frames between primary and redundant transmissions. Figure 1 below shows an example.



Here, each frame is retransmitted once in the following RTP payload packet. f(n-2)...f(n+4) denote a sequence of audio frames, and p(n-1)...p(n+4) a sequence of payload packets.

The mechanism described does not really require signaling at the session setup. However, signaling has been defined to allow for the sender to voluntarily bind the buffering and delay requirements. If nothing is signaled, the use of this mechanism is allowed and unbounded. For a certain timestamp, the receiver may receive multiple copies of a frame containing encoded audio data, even at different encoding rates. The cost of this scheme is bandwidth and the receiver delay necessary to allow the redundant copy to arrive.

This redundancy scheme provides a functionality similar to the one described in RFC 2198, but it works only if both original frames and redundant representations are G.719 frames. When the use of other media coding schemes is desirable, one has to resort to RFC 2198.

The sender is responsible for selecting an appropriate amount of redundancy based on feedback about the channel conditions, e.g., in the RTP Control Protocol (RTCP) [RFC3550] receiver reports. The sender is also responsible for avoiding congestion, which may be exacerbated by redundancy (see Section 9 for more details).

4.3.2. Use of Frame Interleaving

To decrease protocol overhead, the payload design allows several audio transport frames to be encapsulated into a single RTP packet. One of the drawbacks of such an approach is that in the case of packet loss, several consecutive frames are lost. Consecutive frame loss normally renders error concealment less efficient and usually causes clearly audible and annoying distortions in the reconstructed audio. Interleaving of transport frames can improve the audio quality in such cases by distributing the consecutive losses into a number of isolated frame losses, which are easier to conceal.

However, interleaving and bundling several frames per payload also increases end-to-end delay and sets higher buffering requirements. Therefore, interleaving is not appropriate for all use cases or devices. Streaming applications should most likely be able to exploit interleaving to improve audio quality in lossy transmission conditions.

Note that this payload design supports the use of frame interleaving as an option. The usage of this feature needs to be negotiated in the session setup.

The interleaving supported by this format is rather flexible. For example, a continuous pattern can be defined, as depicted in Figure 2.

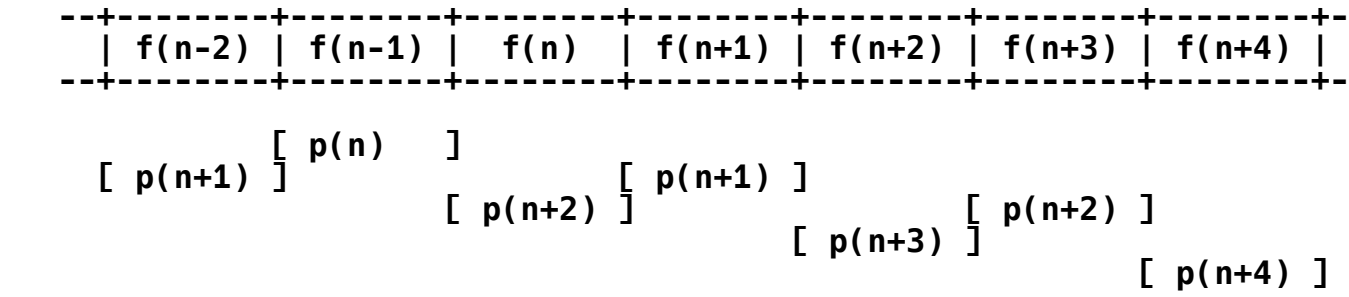


Figure 2: An example of interleaving pattern that has constant delay

In Figure 2, the consecutive frames, denoted f(n-2) to f(n+4), are aggregated into packets p(n) to p(n+4), each packet carrying two frames. This approach provides an interleaving pattern that allows for constant delay in both the interleaving and de-interleaving processes. The de-interleaving buffer needs to have room for at least three frames, including the one that is ready to be consumed. The storage space for three frames is needed, for example, when f(n) is the next frame to be decoded: since frame f(n) was received in packet p(n+2), which also carried frame f(n+3), both these frames are stored in the buffer. Furthermore, frame f(n+1) received in the previous packet, p(n+1), is also in the de-interleaving buffer. Note also that in this example the buffer occupancy varies: when frame f(n+1) is the next one to be decoded, there are only two frames, f(n+1) and f(n+3), in the buffer.

5. Payload Format

The main purpose of the payload design for G.719 is to maximize the potential of the codec to its fullest degree with as minimal overhead as possible. In the design, both basic and interleaved modes have

been included, as the codec is suitable both for conversational and other low-delay applications as well as streaming, where more delay is acceptable.

The main structural difference between the basic and interleaved modes is the extension of the table of contents entries with frame displacement fields in the interleaved mode. The basic mode supports aggregation of multiple consecutive frames in a payload. The interleaved mode supports aggregation of multiple frames that are non-consecutive in time. In both modes, it is possible to have frames encoded with different frame types in the same payload.

The payload format also supports the usage of G.719 for carrying multi-channel content using one discrete encoder per channel all using the same bitrate. In this case, a complete frame-block with data from all channels is included in the RTP payload. The data is the concatenation of all the encoded audio frames in the order specified for that number of included channels. Also, interleaving is done on complete frame-blocks rather than on individual audio frames.

5.1. RTP Header Usage

The RTP timestamp corresponds to the sampling instant of the first sample encoded for the first frame-block in the packet. The timestamp clock frequency SHALL be 48000 Hz. The timestamp is also used to recover the correct decoding order of the frame-blocks.

The RTP header marker bit (M) SHALL be set to 1 whenever the first frame-block carried in the packet is the first frame-block in a talkspurt (see definition of the talkspurt in Section 4.1 of [RFC3551]). For all other packets, the marker bit SHALL be set to zero (M=0).

The assignment of an RTP payload type for the format defined in this memo is outside the scope of this document. The RTP profiles in use currently mandate binding the payload type dynamically for this payload format. This is basically necessary because the payload type expresses the configuration of the payload itself, i.e., basic or interleaved mode, and the number of channels carried.

The remaining RTP header fields are used as specified in [RFC3550].

5.2. Payload Structure

The payload consists of one or more table of contents (ToC) entries followed by the audio data corresponding to the ToC entries. The following sections describe both the basic mode and the interleaved

mode. Each ToC entry MUST be padded to a byte boundary to ensure octet alignment. The rules regarding maximum payload size given in Section 3.2 of [RFC5405] SHOULD be followed.

5.2.1. Basic ToC Element

All the different formats and modes in this document use a common basic ToC that may be extended in the different options described below.

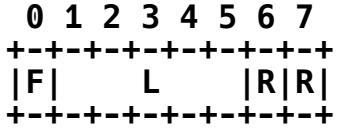


Figure 3: Basic TOC element

F (1 bit): If set to 1, indicates that this ToC entry is followed by another ToC entry; if set to zero, indicates that this ToC entry is the last one in the ToC.

L (5 bits): A field that gives the frame length of each individual frame within the frame-block.

L	length(bytes)
0	0 NO_DATA
1-7	N/A (reserved)
8-22	80+10*(L-8)
23-27	240+20*(L-23)
28-31	N/A (reserved)

Figure 4: How to map L values to frame lengths

L=0 (NO_DATA) is used to indicate an empty frame, which is useful if frames are missing (e.g., at re-packetization), or to insert gaps when sending redundant frames together with primary frames in the same payload.
The value range [1..7] and [28..31] inclusive is reserved for future use in this document version; if these values occur in a ToC, the entire packet SHOULD be treated as invalid and discarded.
A few examples are given below where the frame size and the corresponding codec bitrate is computed based on the value L.

L	Bytes	Codec Bitrate(kbps)
8	80	32
9	90	36
10	100	40
12	120	48
16	160	64
22	220	88
23	240	96
25	280	112
27	320	128

Figure 5: Examples of L values and corresponding frame lengths

This encoding yields a granularity of 4 kbps between 32 and 88 kbps and a granularity of 8 kbps between 88 and 128 kbps with a defined range of 32-128 kbps for the codec data.

R (2 bits): Reserved bits. SHALL be set to zero on sending and SHALL be ignored on reception.

5.3. Basic Mode

The basic ToC element shown in Figure 3 is followed by a 1-octet field for the number of frame-blocks (#frames) to form the ToC entry. The frame-blocks field tells how many frame-blocks of the same length the ToC entry relates to.

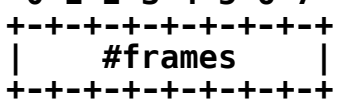


Figure 6: Number of frame-blocks field

5.4. Interleaved Mode

The basic ToC is followed by a 1-octet field for the number of frame-blocks (#frames) and then the DIS fields to form a ToC entry in interleaved mode. The frame-blocks field tells how many frame-blocks of the same length the ToC relates to. The DIS fields, one for each frame-block indicated by the #frames field, express the interleaving distance between audio frames carried in the payload. If necessary to achieve octet alignment, a 4-bit padding is added.

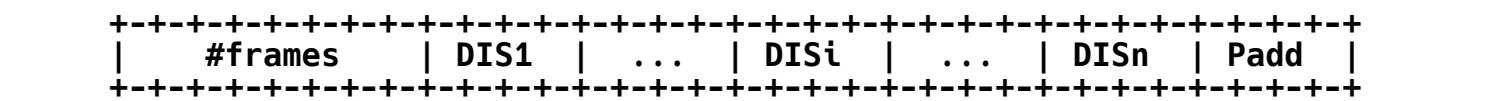


Figure 7: Number of frame-block + interleave fields

DIS1...DISn (4 bits): A list of n (n=#frames) displacement fields indicating the displacement of the i:th (i=1..n) audio frame-block relative to the preceding frame-block in the payload, in units of 20-ms long audio frame-blocks). The 4-bit unsigned integer displacement values may be between zero and 15 indicating the number of audio frame-blocks in decoding order between the (i-1):th and the i:th frame in the payload. Note that for the first ToC entry of the payload, the value of DIS1 is meaningless. It SHALL be set to zero by a sender and SHALL be ignored by a receiver. This frame-block's location in the decoding order is uniquely defined by the RTP timestamp. Note that for subsequent ToC entries DIS1 indicates the number of frames between the last frame of the previous group and the first frame of this group.

Padd (4 bits): To ensure octet alignment, 4 padding bits SHALL be included at the end of the ToC entry in case there is an odd number of frame-blocks in the group referenced by this ToC entry. These bits SHALL be set to zero and SHALL be ignored by the receiver. If a group containing an even number of frames is referenced by this ToC entry, these padding bits SHALL NOT be included in the payload.

5.5. Audio Data

The audio data part follows the table of contents. All the octets comprising an audio frame SHALL be appended to the payload as a unit. For each frame-block, the audio frames are concatenated in the order indicated by the table in Section 4.1 of [RFC3551] for the number of channels configured for the payload type in use. So the first channel (leftmost) indicated comes first followed by the next channel. The audio frame-blocks are packetized in increasing timestamp order within each group of frame-blocks (per ToC entry), i.e., oldest frame-block first. The groups of frame-blocks are packetized in the same order as their corresponding ToC entries.

The audio frames are specified in ITU recommendation [ITU-T-G719].

The G.719 bit stream is split into a sequence of octets and transmitted in order from the leftmost (most significant (MSB)) bit to the rightmost (least significant (LSB)) bit.

5.6. Implementation Considerations

An application implementing this payload format MUST understand all the payload parameters specified in this specification. Any mapping of the parameters to a signaling protocol MUST support all parameters. So an implementation of this payload format in an application using SDP is required to understand all the payload parameters in their SDP-mapped form. This requirement ensures that an implementation always can decide whether it is capable of communicating when the communicating entities support this version of the specification.

Basic mode SHALL be implemented and the interleaved mode SHOULD be implemented. The implementation burden of both is rather small, and supporting both ensures interoperability. However, interleaving is not mandated as it has limited applicability for conversational applications that require tight delay boundaries.

5.6.1. Receiving Redundant Frames

The reception of redundant audio frames, i.e., more than one audio frame from the same source for the same time slot, MUST be supported by the implementation. In the case that the receiver gets multiple audio frames in different bitrates for the same time slot, it is RECOMMENDED that the receiver keeps the one with the highest bitrate.

5.6.2. Interleaving

The use of interleaving requires further considerations. As presented in the example in Section 4.3.2, a given interleaving pattern requires a certain amount of the de-interleaving buffer. This buffer space, expressed in a number of transport frame slots, is indicated by the "interleaving" media type parameter. The number of frame slots needed can be converted into actual memory requirements by considering the 320 bytes per frame used by the highest bitrate of G.719.

The information about the frame buffer size is not always sufficient to determine when it is appropriate to start consuming frames from the interleaving buffer. Additional information is needed when the interleaving pattern changes. The "int-delay" media type parameter is defined to convey this information. It allows a sender to indicate the minimal media time that needs to be present in the buffer before the decoder can start consuming frames from the buffer. Because the sender has full control over the interleaving pattern, it can calculate this value. In certain cases (for example, if joining a multicast session with interleaving mid-session), a receiver may initially receive only part of the packets in the interleaving

pattern. This initial partial reception (in frame sequence order) of frames can yield too few frames for acceptable quality from the audio decoding. This problem also arises when using encryption for access control, and the receiver does not have the previous key. Although the G.719 is robust and thus tolerant to a high random frame erasure rate, it would have difficulties handling consecutive frame losses at startup. Thus, some special implementation considerations are described.

In order to handle this type of startup efficiently, decoding can start provided that:

1. There are at least two consecutive frames available.

2. More than or equal to half the frames are available in the time period from where decoding was planned to start and the most forward received decoding.

After receiving a number of packets, in the worst case as many packets as the interleaving pattern covers, the previously described effects disappear and normal decoding is resumed. Similar issues arise when a receiver leaves a session or has lost access to the stream. If the receiver leaves the session, this would be a minor issue since playout is normally stopped. The sender can avoid this type of problem in many sessions by starting and ending interleaving patterns correctly when risks of losses occur. One such example is a key-change done for access control to encrypted streams. If only some keys are provided to clients and there is a risk they will receive content for which they do not have the key, it is recommended that interleaving patterns do not overlap key changes.

5.6.3. Decoding Validation

If the receiver finds a mismatch between the size of a received payload and the size indicated by the ToC of the payload, the receiver SHOULD discard the packet. This is recommended because decoding a frame parsed from a payload based on erroneous ToC data could severely degrade the audio quality.

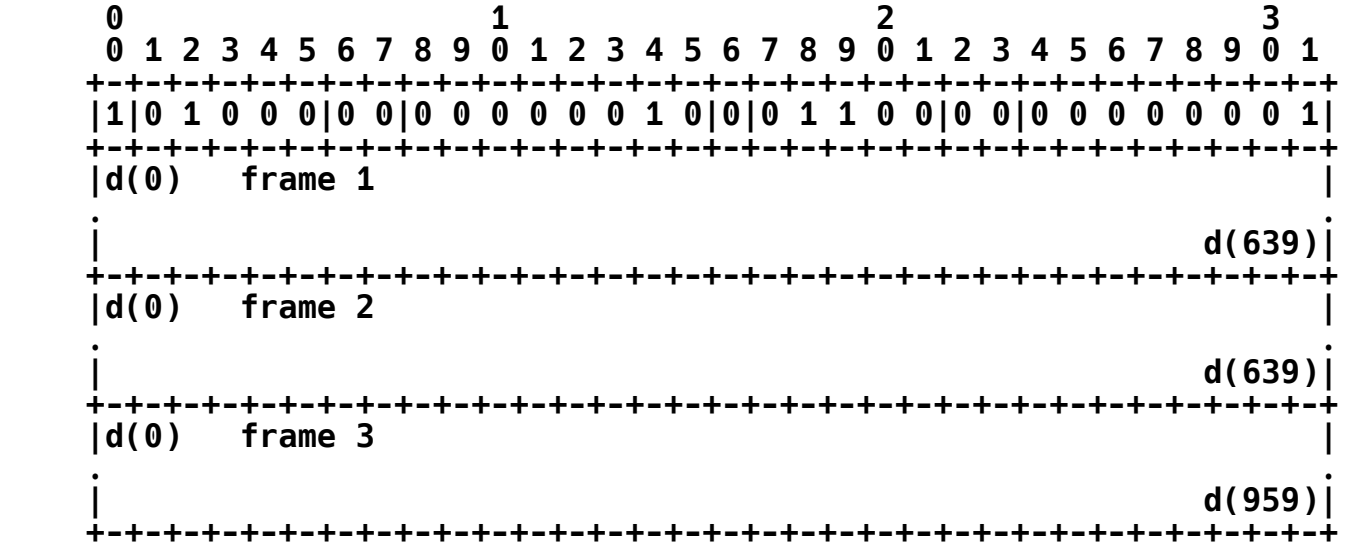
6. Payload Examples

A few examples to highlight the payload format follow.

6.1. 3 Mono Frames with 2 Different Bitrates

The first example is a payload consisting of 3 mono frames where the first 2 frames correspond to a bitrate of 32 kbps (80 bytes/frame) and the last is 48 kbps (120 bytes/frame).

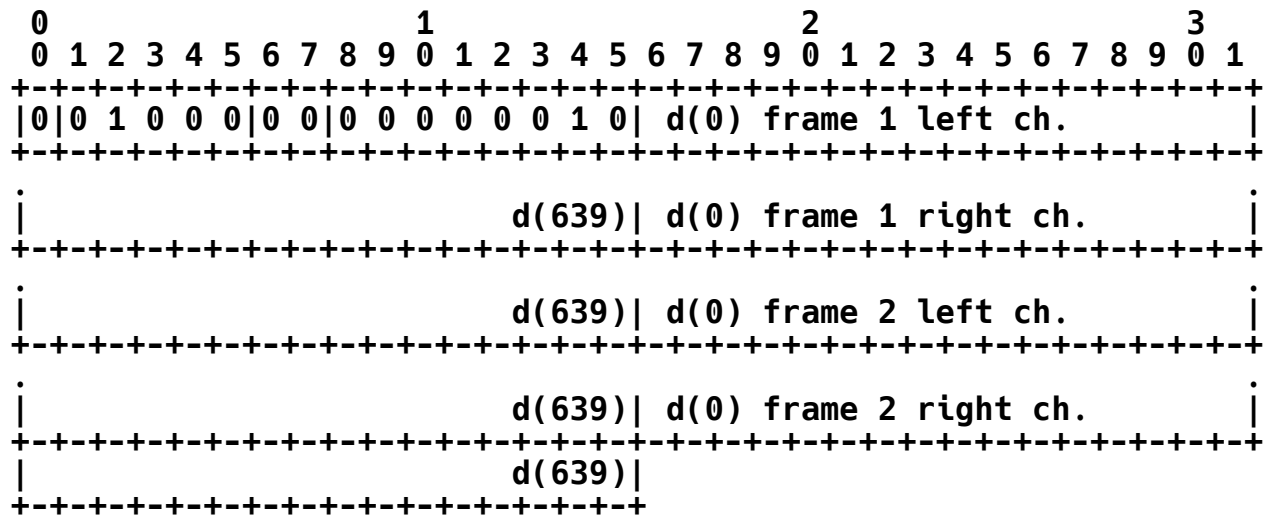
The first 32 bits are ToC fields.
Bit 0 is '1' as another ToC field follows.
Bits 1..5 are '01000' = 80 bytes/frame.
Bits 8..15 are '00000010' = 2 frame-blocks with 80 bytes/frame.
Bit 16 is '0', no more ToC follows.
Bits 17..21 are '01100' = 120 bytes/frame.
Bits 24..31 are '00000001' = 1 frame-block with 120 bytes/frame.



6.2. 2 Stereo Frame-Blocks of the Same Bitrate

The second example is a payload consisting of 2 stereo frames that correspond to a bitrate of 32 kbps (80 bytes/frame) per channel. The receiver calculates the number of frames in the audio block by multiplying the value of the "channels" parameter (2) with the #frames field value (2) to derive that there are 4 audio frames in the payload.

The first 16 bits is the ToC field.
Bit 0 is '0' as no ToC field follows.
Bits 1..5 are '01000' = 80 bytes/frame.
Bits 8..15 are '00000010' = 2 frame-blocks with 80 bytes/frame.

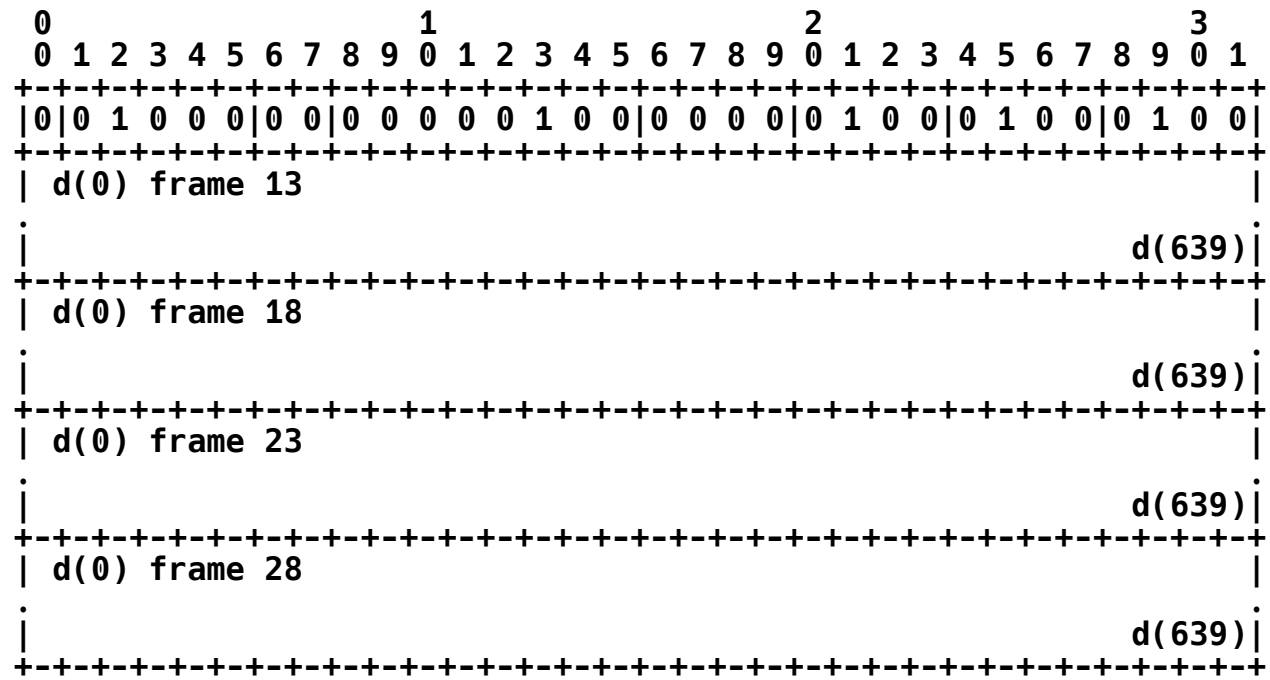


6.3. 4 Mono Frames Interleaved

The third example is a payload consisting of 4 mono frames that correspond to a bitrate of 32 kbps (80 bytes/frame) interleaved. A pattern of interleaving for constant delay when aggregating 4 frames is used in the example below. The actual packet illustrated is packet n, while the previous and following packets' frame-block content is shown to illustrate the pattern.

```
Packet n-3:  1,  6, 11, 16
Packet n-2:  5, 10, 15, 20
Packet n-1:  9, 14, 19, 24
Packet   n: 13, 18, 23, 28
Packet n+1: 17, 22, 27, 32
Packet n+2: 21, 26, 31, 36
```

The first 32 bits are the ToC field.
Bit 0 is '0' as there is no ToC field following.
Bits 1..5 are '01000' = 80 bytes/frame.
Bits 8..15 are '00000100' = 4 frame-blocks with 80 bytes/frame.
Bits 16..19 are '0000' = DIS1 (0).
Bits 20..23 are '0100' = DIS2 (4).
Bits 24..27 are '0100' = DIS3 (4).
Bits 28..31 are '0100' = DIS4 (4).



7. Payload Format Parameters

This RTP payload format is identified using the media type audio/G719, which is registered in accordance with [RFC4855] and uses the template of [RFC4288].

7.1. Media Type Definition

The media type for the G.719 codec is allocated from the IETF tree since G.719 has the potential to become a widely used audio codec in general Voice over IP (VoIP), teleconferencing, and streaming applications. This media type registration covers real-time transfer via RTP.

Note, any unspecified parameter MUST be ignored by the receiver to ensure that additional parameters can be added in any future revision of this specification.

Type name: audio

Subtype name: G719

Required parameters: none

Optional parameters:

interleaving: Indicates that interleaved mode SHALL be used for the payload. The parameter specifies the number of frame-block slots available in a de-interleaving buffer (including the frame that is ready to be consumed) for each source. Its value is equal to one plus the maximum number of frames that can precede any frame in transmission order and follow the frame in RTP timestamp order. The value MUST be greater than zero. If this parameter is not present, interleaved mode SHALL NOT be used.

int-delay: The minimal media time delay in milliseconds that is needed to avoid underrun in the de-interleaving buffer before starting decoding, i.e., the difference in RTP timestamp ticks between the earliest and latest audio frame present in the de-interleaving buffer expressed in milliseconds. The value is a stream property and provided per source. The allowed values are zero to the largest value expressible by an unsigned 16-bit integer (65535). Please note that in practice, the largest value that can be used is equal to the declared size of the interleaving buffer of the receiver. If the value for some reason is larger than the receiver buffer declared by or for the receiver, this value defaults to the size of the receiver buffer. For sources for which this value hasn't been provided, the value defaults to the size of the receiver buffer. The format is a comma-separated list of synchronization source (SSRC) ":" delay in ms pairs, which in ABNF [RFC5234] is expressed as:

```
int-delay = "int-delay:" source-delay *("," source-delay)

source-delay = SSRC ":" delay-value

SSRC = 1*8HEXDIG ; The 32-bit SSRC encoded in hex format

delay-value = 1*5DIGIT ; The delay value in milliseconds

Example: int-delay=ABCD1234:1000,4321DCB:640

NOTE: No white space allowed in the parameter before the end of
all the value pairs
```

max-red: The maximum duration in milliseconds that elapses between the primary (first) transmission of a frame and any redundant transmission that the sender will use. This parameter allows a receiver to have a bounded delay when redundancy is used. Allowed values are between zero (no redundancy will be used) and 65535. If the parameter is omitted, no limitation on the use of redundancy is present.

channels: The number of audio channels. The possible values (1-6) and their respective channel order is specified in Section 4.1 of [RFC3551]. If omitted, it has the default value of 1.

CBR: Constant Bitrate (CBR) indicates the exact codec bitrate in bits per second (not including the overhead from packetization, RTP header, or lower layers) that the codec MUST use. "CBR" is to be used when the dynamic rate cannot be supported (one case is, e.g., gateway to H.320). "CBR" is mostly used for gateways to circuit switch networks. Therefore, the "CBR" is the rate not including any FEC as specified in Section 4.3.1. If FEC is to be used, the "b=" parameter MUST be used to allow the extra bitrate needed to send the redundant information. It is RECOMMENDED that this parameter is only used when necessary to establish a working communication. The usage of this parameter has implications for congestion control that need to be considered; see Section 9.

ptime: see [RFC4566].

maxptime: see [RFC4566].

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

Security considerations: See Section 10 of RFC 5404.

Interoperability considerations: The support of the Interleaving mode is not mandatory and needs to be negotiated. See Section 7.2 for how to do that for SDP-based protocols.

Published specification: RFC 5404

Applications that use this media type: Real-time audio applications like Voice over IP and teleconference, and multi-media streaming.

Additional information: none

Person & email address to contact for further information:
Ingemar Johansson
<ingemar.s.johansson@ericsson.com>

Intended usage: COMMON

Restrictions on usage: This media type 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:
Ingemar Johansson <ingemar.s.johansson@ericsson.com>
Magnus Westerlund <magnus.westerlund@ericsson.com>

Change controller: IETF Audio/Video Transport working group delegated from the IESG.

Additionally, note that file storage of G.719-encoded audio in ISO base media file format is specified in Annex A of [ITU-T-G719]. Thus, media file formats such as MP4 (audio/mp4 or video/mp4) [RFC4337] and 3GP (audio/3GPP and video/3GPP) [RFC3839] can contain G.719-encoded audio.

7.2. Mapping to SDP

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 G.719 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 RTP clock rate in "a=rtpmap" MUST be 48000, and the encoding parameter "channels" (Section 7.1) MUST either be explicitly set to N or omitted, implying a default value of 1. The values of N that are allowed are specified in Section 4.1 in [RFC3551].
- 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.

7.2.1. Offer/Answer Considerations

The following considerations apply when using SDP offer/answer procedures to negotiate the use of G.719 payload in RTP:

- Each combination of the RTP payload transport format configuration parameters ("interleaving" and "channels") is unique in its bit pattern and not compatible with any other combination. When creating an offer in an application desiring to use the more advanced features (interleaving or more than one channel), the offerer is RECOMMENDED to also offer a payload type containing

only the configuration with a single channel. If multiple configurations are of interest to the application, they may all be offered; however, care should be taken not to offer too many payload types. An SDP answerer MUST include, in the SDP answer for a payload type, the following parameters unmodified from the SDP offer (unless it removes the payload type): "interleaving" and "channels". However, the value of the "interleaving" parameter MAY be changed. The SDP offerer and answerer MUST generate G.719 packets as described by these parameters.

- The "interleaving" and "int-delay" parameters' values have a specific relationship that needs to be considered. It also depends on the directionality of the streams and their delivery method. The high-level explanation that can be understood from the definition is that the value of "interleaving" declares the size of the receiver buffer, while "int-delay" is a stream property provided by the sender to inform how much buffer space it in practice is using for the stream it sends.

  - For media streams that are sent over multicast, the value of "interleaving" SHALL NOT be changed by the answerer. It shall either be accepted or the payload type deleted. The value of the "int-delay" parameter is a stream property and provided by the offer/answer agent that intends to send media with this payload type, and for each stream coming from that agent (one or more). The value MUST be between zero and what corresponds to the buffer size declared by the value of the "interleaving" parameter.

  - For unicast streams that the offerer declares as send-only, the value of the "interleaving" parameter is the size that the answerer is RECOMMENDED to use by the offerer. The answerer MAY change it to any allowed value. The "int-delay" parameter value will be the one the offerer intends to use unless the answerer reduces the value of the "interleaving" parameter below what is needed for that "int-delay" value. If the "interleaving" value in the answer is smaller than the offer's "int-delay" value, the "int-delay" value is per default reduced to be corresponding to the "interleaving" value. If the offerer is not satisfied with this, he will need to perform another round of offer/answer. As the answerer will not send any media, it doesn't include any "int-delay" in the answer.

  - For unicast streams that the offerer declares as recvonly, the value of "interleaving" in the offer will be the offerer's size of the interleaving buffer. The answerer indicates its preferred size of the interleaving buffer for any future round of offer/answer. The offerer will not provide any "int-delay"

parameter as it is not sending any media. The answerer is recommended to include in its answer an "int-delay" parameter to declare what the property is for the stream it is going to send. The answer is expected to be capable of selecting a valid parameter value that is between zero and the declared maximum number of slots in the de-interleaving buffer.

* For unicast streams that the offer declares as sendrecv streams, the value of the "interleaving" parameter in the offer will be the offerer's size of the interleaving buffer. The answerer will in the answer indicate the size of its actual interleaving buffer. It is recommended that this value is at least as big as the offer's. The offerer is recommended to include an "int-delay" parameter that is selected based on the answerer having at least as much interleaving space as the offerer unless nothing else is known. As the offerer's interleaving buffer size is not yet known, this may fail, in which case the default rule is to downgrade the value of the "int-delay" to correspond to the full size of the answerer's interleaving buffer. If the offerer isn't satisfied with this, it will need to initiate another round of offer/answer. The answerer is recommended in its answer to include an "int-delay" parameter to declare what the property is for the stream(s) it is going to send. The answer is expected to be capable of selecting a valid parameter value that is between zero and the declared maximum number of slots in the de-interleaving buffer.

- 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]. The "maxptime" parameter MUST be handled in the same way.

- The parameter "max-red" is a stream property parameter. For sendonly or sendrecv unicast media streams, the parameter declares the limitation on redundancy that the stream sender will use. For recvonly streams, it indicates the desired value for the stream sent to the receiver. The answerer MAY change the value, but is RECOMMENDED to use the same limitation as the offer declares. In the case of multicast, the offerer MAY declare a limitation; this SHALL be answered using the same value. A media sender using this payload format is RECOMMENDED to always include the "max-red" parameter. This information is likely to simplify the media stream handling in the receiver. This is especially true if no redundancy will be used, in which case "max-red" is set to zero.

- Any unknown parameter in an offer SHALL be removed in the answer.

- The "b=" SDP parameter SHOULD be used to negotiate the maximum bandwidth to be used for the audio stream. The offerer may offer a maximum rate and the answer may contain a lower rate. If no "b=" parameter is present in the offer or answer, it implies a rate up to 128 kbps.

- The parameter "CBR" is a receiver capability; i.e., only receivers that really require a constant bitrate should use it. Usage of this parameter has a negative impact on the possibility to perform congestion control; see Section 9. For recvonly and sendrecv streams, it indicates the desired constant bitrate that the receiver wants to accept. A sender MUST be able to send a constant bitrate stream since it is a subset of the variable bitrate capability. If the offer includes this parameter, the answerer MUST send G.719 audio at the constant bitrate if it is within the allowed session bitrate ("b=" parameter). If the answerer cannot support the stated CBR, this payload type must be refused in the answer. The answerer SHOULD only include this parameter if the answerer itself requires to receive at a constant bitrate, even if the offer did not include the "CBR" parameter. In this case, the offerer SHALL send at the constant bitrate, but SHALL be able to accept media at a variable bitrate. An answerer is RECOMMEND to use the same CBR as in the offer, as symmetric usage is more likely to work. If both sides require a particular CBR, there is the possibility of communication failure when one or both sides can't transmit the requested rate. In this case, the agent detecting this issue will have to perform a second round of offer/answer to try to find another working configuration or end the established session. In case the offer contained a "CBR" parameter but the answer does not, then the offerer is free to transmit at any rate to the answerer, but the answerer is restricted to the declared rate.

7.2.2. Declarative SDP Considerations

In declarative usage, like SDP in the Real Time Streaming Protocol (RTSP) [RFC2326] or the Session Announcement Protocol (SAP) [RFC2974], the parameters SHALL be interpreted as follows:

- The payload format configuration parameters ("interleaving" and "channels") are all declarative, and a participant MUST use the configuration(s) that is provided for the session. More than one configuration may be provided if necessary by declaring multiple RTP payload types; however, the number of types should be kept small.

- It might not be possible to know the SSRC values that are going to be used by the sources at the time of sending the SDP. This is not a major issue as the size of the interleaving buffer can be tailored towards the values that are actually going to be used, thus ensuring that the default values for "int-delay" are not resulting in too much extra buffering.

- Any "maxptime" and "ptime" values should be selected with care to ensure that the session's participants can achieve reasonable performance.

- The parameter "CBR" if included applies to all RTP streams using that payload type for which a particular CBR is declared. Usage of this parameter has a negative impact on the possibility to perform congestion control; see Section 9.

8. IANA Considerations

One media type (audio/G719) has been defined and registered in the media types registry; see Section 7.1.

9. Congestion Control

The general congestion control considerations for transporting RTP data apply; see RTP [RFC3550] and any applicable RTP profile like AVP [RFC3551]. However, the multi-rate capability of G.719 audio coding provides a mechanism that may help to control congestion, since the bandwidth demand can be adjusted (within the limits of the codec) by selecting a different encoding bitrate.

The number of frames encapsulated in each RTP payload highly influences the overall bandwidth of the RTP stream due to header overhead constraints. Packetizing more frames in each RTP payload can reduce the number of packets sent and hence the header overhead, at the expense of increased delay and reduced error robustness. If forward error correction (FEC) is used, the amount of FEC-induced redundancy needs to be regulated such that the use of FEC itself does not cause a congestion problem. In other words, a sender SHALL NOT increase the total bitrate when adding redundancy in response to packet loss, and needs instead to adjust it down in accordance to the congestion control algorithm being run. Thus, when adding redundancy, the media bitrate will need to be reduced to provide room for the redundancy.

The "CBR" signaling parameter allows a receiver to lock down an RTP payload type to use a single encoding rate. As this prevents the codec rate from being lowered when congestion is experienced, the sender is constrained to either change the packetization or abort the

transmission. Since these responses to congestion are severely limited, implementations SHOULD NOT use the "CBR" parameter unless they are interacting with a device that cannot support a variable bitrate (e.g., a gateway to H.320 systems). When using CBR mode, a receiver MUST monitor the packet loss rate to ensure congestion is not caused, following the guidelines in Section 2 of RFC 3551.

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 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, the transport, and the signaling protocol employed. Therefore, a single mechanism is not sufficient, although if suitable, usage of 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.

The use of interleaving in conjunction with encryption can have a negative impact on confidentiality for a short period of time. Consider the following packets (in brackets) containing frame numbers as indicated: {10, 14, 18}, {13, 17, 21}, {16, 20, 24} (a popular continuous diagonal interleaving pattern). The originator wishes to deny some participants the ability to hear material starting at time 16. Simply changing the key on the packet with the timestamp at or after 16, and denying that new key to those participants, does not achieve this; frames 17, 18, and 21 have been supplied in prior packets under the prior key, and error concealment may make the audio intelligible at least as far as frame 18 or 19, and possibly further.

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.

11. Acknowledgements

The authors would like to thank Roni Even and Anisse Taleb for their help with this document. We would also like to thank the people who have provided feedback: Colin Perkins, Mark Baker, and Stephen Botzko.

12. References

12.1. Normative References

[ITU-T-G719] ITU-T, "Specification : ITU-T G.719 extension for 20 kHz fullband audio", April 2008.

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

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

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

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

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

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[RFC5405] Eggert, L. and G. Fairhurst, "Unicast UDP Usage Guidelines for Application Designers", BCP 145, RFC 5405, November 2008.

12.2. Informative References

[RFC2198] Perkins, C., Kouvelas, I., Hodson, O., Hardman, V., Handley, M., Bolot, J., Vega-Garcia, A., and S. Fosse-Parisis, "RTP Payload for Redundant Audio Data", RFC 2198, September 1997.

[RFC2326] Schulzrinne, H., Rao, A., and R. Lanphier, "Real Time Streaming Protocol (RTSP)", RFC 2326, April 1998.

[RFC2974] Handley, M., Perkins, C., and E. Whelan, "Session Announcement Protocol", RFC 2974, October 2000.

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

[RFC3839] Castagno, R. and D. Singer, "MIME Type Registrations for 3rd Generation Partnership Project (3GPP) Multimedia files", RFC 3839, July 2004.

[RFC4288] Freed, N. and J. Klensin, "Media Type Specifications and Registration Procedures", BCP 13, RFC 4288, December 2005.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC4337] Y Lim and D. Singer, "MIME Type Registration for MPEG-4", RFC 4337, March 2006.

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

[RFC5109] Li, A., "RTP Payload Format for Generic Forward Error Correction", RFC 5109, December 2007.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

Authors' Addresses

Magnus Westerlund
Ericsson AB
Torshamnsgatan 21-23
SE-164 83 Stockholm
SWEDEN

Phone: +46 10 7190000
EMail: magnus.westerlund@ericsson.com

Ingemar Johansson
Ericsson AB
Laboratoriegrand 11
SE-971 28 Lulea
SWEDEN

Phone: +46 10 7190000
EMail: ingemar.s.johansson@ericsson.com