Internet Engineering Task Force (IETF)
Request for Comments: 6416
Obsoletes: 3016
Category: Standards Track
ISSN: 2070-1721

M. Schmidt
Dolby Laboratories
F. de Bont
Philips Electronics
S. Doehla
Fraunhofer IIS
J. Kim
LG Electronics Inc.

October 2011

RTP Payload Format for MPEG-4 Audio/Visual Streams

Abstract

This document describes Real-time Transport Protocol (RTP) payload formats for carrying each of MPEG-4 Audio and MPEG-4 Visual bitstreams without using MPEG-4 Systems. This document obsoletes RFC 3016. It contains a summary of changes from RFC 3016 and discusses backward compatibility to RFC 3016. It is a necessary revision of RFC 3016 in order to correct misalignments with the 3GPP Packet-switched Streaming Service (PSS) specification regarding the RTP payload format for MPEG-4 Audio.

For the purpose of directly mapping MPEG-4 Audio/Visual bitstreams onto RTP packets, this document provides specifications for the use of RTP header fields and also specifies fragmentation rules. It also provides specifications for Media Type registration and the use of the Session Description Protocol (SDP). The audio payload format described in this document has some limitations related to the signaling of audio codec parameters for the required multiplexing format. Therefore, new system designs should utilize RFC 3640, which does not have these restrictions. Nevertheless, this revision of RFC 3016 is provided to update and complete the specification and to enable interoperable implementations.

Status of This Memo

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6416.

Table of Contents

1. Introduction

The RTP payload formats described in this document specify how MPEG-4 Audio [14496-3] and MPEG-4 Visual streams [14496-2] are to be fragmented and mapped directly onto RTP packets.

These RTP payload formats enable transport of MPEG-4 Audio/Visual streams without using the synchronization and stream management functionality of MPEG-4 Systems [14496-1]. Such RTP payload formats will be used in systems that have intrinsic stream management functionality and thus require no such functionality from MPEG-4 Systems. H.323 [H323] terminals are an example of such systems, where MPEG-4 Audio/Visual streams are not managed by MPEG-4 Systems Object Descriptors but by H.245 [H245]. The streams are directly mapped onto RTP packets without using the MPEG-4 Systems Sync Layer. Other examples are the Session Initiation Protocol (SIP) [RFC3261] and Real Time Streaming Protocol (RTSP) where media type and SDP are used. Media type and SDP usages of the RTP payload formats described in this document are defined to directly specify the attribute of Audio/Visual streams (e.g., media type, packetization format, and codec configuration) without using MPEG-4 Systems. The obvious benefit is that these MPEG-4 Audio/Visual RTP payload formats can be handled in a unified way together with those formats defined for non-MPEG-4 codecs. The disadvantage is that interoperability with environments using MPEG-4 Systems may be difficult; hence, other payload formats may be better suited to those applications.

The semantics of RTP headers in such cases need to be clearly defined, including the association with MPEG-4 Audio/Visual data elements. In addition, it is beneficial to define the fragmentation rules of RTP packets for MPEG-4 Video streams so as to enhance error resiliency by utilizing the error resiliency tools provided inside the MPEG-4 Video stream.

1.1. MPEG-4 Visual RTP Payload Format

MPEG-4 Visual is a visual coding standard with many features, including: high coding efficiency; high error resiliency; and multiple, arbitrary shape object-based coding [14496-2]. It covers a wide range of bitrates from scores of kbit/s to several Mbit/s. It also covers a wide variety of networks, ranging from those guaranteed to be almost error-free to mobile networks with high error rates.

With respect to the fragmentation rules for an MPEG-4 Visual bitstream defined in this document, since MPEG-4 Visual is used for a wide variety of networks, it is desirable not to apply too much restriction on fragmentation, and a fragmentation rule such as "a single video packet shall always be mapped on a single RTP packet" may be inappropriate. On the other hand, careless, media-unaware fragmentation may cause degradation in error resiliency and bandwidth efficiency. The fragmentation rules described in this document are flexible but manage to define the minimum rules for preventing meaningless fragmentation while utilizing the error resiliency functionalities of MPEG-4 Visual.

The fragmentation rule "Different Video Object Planes (VOPs) SHOULD be fragmented into different RTP packets" is made so that the RTP timestamp uniquely indicates the VOP time framing. On the other hand, MPEG-4 video may generate VOPs of very small size, in cases with an empty VOP (vop_coded=0) containing only VOP header or an arbitrary shaped VOP with a small number of coding blocks. To reduce the overhead for such cases, the fragmentation rule permits concatenating multiple VOPs in an RTP packet. (See fragmentation rule (4) in Section 5.2 and the descriptions of marker bit and timestamp in Section 5.1.)

While the additional media-specific RTP header defined for such video coding tools as H.261 [H261] or MPEG-1/2 is effective in helping to recover picture headers corrupted by packet losses, MPEG-4 Visual already has error resiliency functionalities for recovering corrupt headers, and these can be used on RTP/IP networks as well as on other networks (H.223/mobile, MPEG-2 Transport Stream, etc.). Therefore, no extra RTP header fields are defined in this MPEG-4 Visual RTP payload format.

1.2. MPEG-4 Audio RTP Payload Format

MPEG-4 Audio is an audio standard that integrates many different types of audio coding tools. Low-overhead MPEG-4 Audio Transport Multiplex (LATM) manages the sequences of audio data with relatively small overhead. In audio-only applications, then, it is desirable for LATM-based MPEG-4 Audio bitstreams to be directly mapped onto RTP packets without using MPEG-4 Systems.

For MPEG-4 Audio coding tools, as is true for other audio coders, if the payload is a single audio frame, packet loss will not impair the decodability of adjacent packets. Therefore, the additional media-specific header for recovering errors will not be required for MPEG-4 Audio. Existing RTP protection mechanisms, such as Generic Forward Error Correction [RFC5109] and Redundant Audio Data [RFC2198], MAY be applied to improve error resiliency.

1.3. Interoperability with RFC 3016

This specification is not backwards compatible with [RFC3016], as a binary incompatible LATM version is mandated. Existing implementations of RFC 3016 that use a recent LATM version may already comply to this specification and must be considered as not compliant with RFC 3016. The 3GPP PSS service [3GPP] is such an example, as a more recent LATM version is mandated in the 3GPP PSS specification. Existing implementations that use the LATM version as specified in RFC 3016 MUST be updated to comply with this specification.

1.4. Relation with RFC 3640

In this document a payload format for the transport of MPEG-4 Elementary Streams is specified. For MPEG-4 Audio streams "out-of-band" signaling is defined such that a receiver is not obliged to decode the payload data to determine the audio codec and its configuration. The signaling capabilities specified in this document are less explicit than those defined in [RFC3640]. But, the use of the MPEG-4 LATM in various transmission standards justifies its right to exist; see also Section 1.2.

2. Definitions and Abbreviations

This document makes use of terms, specified in [14496-2], [14496-3], and [23003-1]. In addition, the following terms are used in this document and have specific meaning within the context of this document.

Abbreviations:

AAC: Advanced Audio Coding

ASC: AudioSpecificConfig

HE AAC: High Efficiency AAC

LATM: Low-overhead MPEG-4 Audio Transport Multiplex

PS: Parametric Stereo

SBR: Spectral Band Replication

VOP: Video Object Plane

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. Clarifications on Specifying Codec Configurations for MPEG-4 Audio

For MPEG-4 Audio [14496-3] streams, the decoder output configuration can differ from the core codec configuration depending of use of the SBR and PS tools.

The core codec sampling rate is the default audio codec sampling rate. When SBR is used, typically the double value of the core codec sampling rate will be regarded as the definitive sampling rate (i.e., the decoder's output sampling rate)

Note: The exception is down-sampled SBR mode, in which case the SBR sampling rate and core codec sampling rate are identical.

The core codec channel configuration is the default audio codec channel configuration. When PS is used, the core codec channel configuration indicates one channel (i.e., mono) whereas the definitive channel configuration is two channels (i.e. stereo). When MPEG Surround is used, the definitive channel configuration depends on the output of the MPEG Surround decoder.

4. LATM Restrictions for RTP Packetization of MPEG-4 Audio Bitstreams

LATM has several multiplexing features as follows:

- carrying configuration information with audio data,
- concatenating multiple audio frames in one audio stream,
- multiplexing multiple objects (programs), and
- multiplexing scalable layers,

However, in RTP transmission, there is no need for the last two features. Therefore, these two features MUST NOT be used in applications based on RTP packetization specified by this document. Since LATM has been developed for only natural audio coding tools, i.e., not for synthesis tools, it seems difficult to transmit Structured Audio (SA) data and Text-to-Speech Interface (TTSI) data by LATM. Therefore, SA data and TTSI data MUST NOT be transported by the RTP packetization in this document.

For transmission of scalable streams, audio data of each layer SHOULD be packetized onto different RTP streams allowing for the different layers to be treated differently at the IP level, for example, via some means of differentiated service. On the other hand, all configuration data of the scalable streams are contained in one LATM configuration data "StreamMuxConfig", and every scalable layer shares the StreamMuxConfig. The mapping between each layer and its configuration data is achieved by LATM header information attached to the audio data. In order to indicate the dependency information of the scalable streams, the signaling mechanism as specified in [RFC5583] SHOULD be used (see Section 6.2).

5. RTP Packetization of MPEG-4 Visual Bitstreams

This section specifies RTP packetization rules for MPEG-4 Visual content. An MPEG-4 Visual bitstream is mapped directly onto RTP packets without the addition of extra header fields or any removal of Visual syntax elements. The Combined Configuration/Elementary stream mode MUST be used so that configuration information will be carried to the same RTP port as the elementary stream. (See Subclause 6.2.1, "Start codes", of [14496-2].) The configuration information MAY additionally be specified by some out-of-band means. If needed by systems using media type parameters and SDP parameters, e.g., SIP and RTSP, the optional parameter "config" MUST be used to specify the configuration information (see Sections 7.1 and 7.2).

When the short video header mode is used, the RTP payload format for H.263 SHOULD be used. (The format defined in [RFC4629] is RECOMMENDED, but the [RFC4628] format MAY be used for compatibility with older implementations.)

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|X|  CC   |M|     PT      |       sequence number         | RTP
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           timestamp                           | Header
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           synchronization source (SSRC) identifier            |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|            contributing source (CSRC) identifiers             |
|                             ....                              |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|                                                               | RTP
|       MPEG-4 Visual stream (byte aligned)                     | Pay-
|                                                               | load
|                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               :...OPTIONAL RTP padding        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: An RTP Packet for MPEG-4 Visual Stream

5.1. Use of RTP Header Fields for MPEG-4 Visual

Payload Type (PT): The assignment of an RTP payload type for this 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 means of an out-of-band signaling protocol (e.g., H.245, SIP).

Extension (X) bit: Defined by the RTP profile used.

Sequence Number: Incremented by 1 for each RTP data packet sent, starting, for security reasons, with a random initial value.

Marker (M) bit: The marker bit is set to 1 to indicate the last RTP packet (or only RTP packet) of a VOP. When multiple VOPs are carried in the same RTP packet, the marker bit is set to 1.

Timestamp: The timestamp indicates the sampling instance of the VOP contained in the RTP packet. A constant offset, which is random, is added for security reasons.

- o When multiple VOPs are carried in the same RTP packet, the timestamp indicates the earliest of the VOP times within the VOPs carried in the RTP packet. Timestamp information of the rest of the VOPs is derived from the timestamp fields in the VOP header (modulo_time_base and vop_time_increment).

- o If the RTP packet contains only configuration information and/or Group_of_VideoObjectPlane() fields, the timestamp of the next VOP in the coding order is used.

- o If the RTP packet contains only visual_object_sequence_end_code information, the timestamp of the immediately preceding VOP in the coding order is used.

The resolution of the timestamp is set to its default value of 90 kHz, unless specified by out-of-band means (e.g., SDP parameter or media type parameter as defined in Section 7).

Other header fields are used as described in [RFC3550].

5.2. Fragmentation of MPEG-4 Visual Bitstream

A fragmented MPEG-4 Visual bitstream is mapped directly onto the RTP payload without any addition of extra header fields or any removal of Visual syntax elements.

In the following, header means one of the following:

- o Configuration information (Visual Object Sequence Header, Visual Object Header, and Video Object Layer Header)

- o visual_object_sequence_end_code

- o The header of the entry point function for an elementary stream (Group_of_VideoObjectPlane() or the header of VideoObjectPlane(), video_plane_with_short_header(), MeshObject(), or FaceObject())

- o The video packet header (video_packet_header() excluding next_resync_marker())

- o The header of gob_layer()

- o See Subclause 6.2.1 ("Start codes") of [14496-2] for the definition of the configuration information and the entry point functions.

The Combined Configuration/Elementary streams mode is used. The following rules apply for the fragmentation.

(1) Configuration information and Group_of_VideoObjectPlane() fields SHALL be placed at the beginning of the RTP payload (just after the RTP header) or just after the header of the syntactically upper-layer function.

(2) If one or more headers exist in the RTP payload, the RTP payload SHALL begin with the header of the syntactically highest function. Note: The visual_object_sequence_end_code is regarded as the lowest function.

(3) A header SHALL NOT be split into a plurality of RTP packets.

(4) Different VOPs SHOULD be fragmented into different RTP packets so that one RTP packet consists of the data bytes associated with a unique VOP time instance (that is indicated in the timestamp field in the RTP packet header), with the exception that multiple consecutive VOPs MAY be carried within one RTP packet in the decoding order if the size of the VOPs is small.

Note: When multiple VOPs are carried in one RTP payload, the timestamp of the VOPs after the first one may be calculated by the decoder. This operation is necessary only for RTP packets in which the marker bit equals to 1 and the beginning of the RTP payload corresponds to a start code. (See the descriptions of timestamp and marker bit in Section 5.1.)

(5) It is RECOMMENDED that a single video packet is sent as a single RTP packet. The size of a video packet SHOULD be adjusted in such a way that the resulting RTP packet is not larger than the Path MTU. If the video packet is disabled by the coder configuration (by setting resync_marker_disable in the VOL header to 1), or in coding tools where the video packet is not supported, a VOP MAY be split at arbitrary byte positions.

The video packet starts with the VOP header or the video packet header, followed by motion_shape_texture(), and ends with next_resync_marker() or next_start_code().

5.3. Examples of Packetized MPEG-4 Visual Bitstream

Figure 2 shows examples of RTP packets generated based on the criteria described in Section 5.2

(a) is an example of the first RTP packet or the random access point of an MPEG-4 Visual bitstream containing the configuration information. According to criterion (1), the Visual Object Sequence Header (VS header) is placed at the beginning of the RTP payload, preceding the Visual Object Header and the Video Object Layer Header (VO header, VOL header). Since the fragmentation rule defined in Section 5.2 guarantees that the configuration information, starting with visual_object_sequence_start_code, is always placed at the beginning of the RTP payload, RTP receivers can detect the random access point by checking if the first 32-bit field of the RTP payload is visual_object_sequence_start_code.

(b) is another example of the RTP packet containing the configuration information. It differs from example (a) in that the RTP packet also contains a VOP header and a video packet in the VOP following the configuration information. Since the length of the configuration information is relatively short (typically scores of bytes) and an RTP packet containing only the configuration information may thus increase the overhead, the configuration information and the subsequent VOP can be packetized into a single RTP packet.

(c) is an example of an RTP packet that contains Group_of_VideoObjectPlane (GOV). Following criterion (1), the GOV is placed at the beginning of the RTP payload. It would be a waste of RTP/IP header overhead to generate an RTP packet containing only a GOV whose length is 7 bytes. Therefore, the following VOP (or a part of it) can be placed in the same RTP packet as shown in (c).

(d) is an example of the case where one video packet is packetized into one RTP packet. When the packet-loss rate of the underlying network is high, this kind of packetization is recommended. Even when the RTP packet containing the VOP header is discarded by a packet loss, the other RTP packets can be decoded by using the HEC (Header Extension Code) information in the video packet header. No extra RTP header field is necessary.

(e) is an example of the case where more than one video packet is packetized into one RTP packet. This kind of packetization is effective to save the overhead of RTP/IP headers when the bitrate of the underlying network is low. However, it will decrease the packet-loss resiliency because multiple video packets are discarded by a single RTP packet loss. The optimal number of video packets in an RTP packet and the length of the RTP packet can be determined by considering the packet-loss rate and the bitrate of the underlying network.

(f) is an example of the case when the video packet is disabled by setting resync_marker_disable in the VOL header to 1. In this case, a VOP may be split into a plurality of RTP packets at arbitrary byte positions. For example, it is possible to split a VOP into fixed-length packets. This kind of coder configuration and RTP packet fragmentation may be used when the underlying network is guaranteed to be error-free.

Figure 3 shows examples of RTP packets prohibited by the criteria of Section 5.2.

Fragmentation of a header into multiple RTP packets, as in Figure 3(a), will not only increase the overhead of RTP/IP headers but also decrease the error resiliency. Therefore, it is prohibited by criterion (3).

When concatenating more than one video packet into an RTP packet, the VOP header or video_packet_header() is not allowed to be placed in the middle of the RTP payload. The packetization as in Figure 2(b) is not allowed by criterion (2) due to the aspect of the error resiliency. Comparing this example with Figure 2(d), although two video packets are mapped onto two RTP packets in both cases, the packet-loss resiliency is not identical. Namely, if the second RTP packet is lost, both video packets 1 and 2 are lost in the case of Figure 3(b), whereas only video packet 2 is lost in the case of Figure 2(d).

```
    +------+------+------+------+
(a) | RTP  |  VS  |  VO  | VOL  |
    |header|header|header|header|
    +------+------+------+------+

    +------+------+------+------+------+------------+
(b) | RTP  |  VS  |  VO  | VOL  | VOP  |Video Packet|
    |header|header|header|header|header|            |
    +------+------+------+------+------+------------+

    +------+-----+------------------+
(c) | RTP  | GOV |Video Object Plane|
    |header|     |                  |
    +------+-----+------------------+

    +------+------+------------+  +------+------+------------+
(d) | RTP  | VOP  |Video Packet|  | RTP  |  VP  |Video Packet|
    |header|header|    (1)     |  |header|header|    (2)     |
    +------+------+------------+  +------+------+------------+

    +------+------+------------+------+------------+------+------------+
(e) | RTP  |  VP  |Video Packet|  VP  |Video Packet|  VP  |Video Packet|
    |header|header|     (1)    |header|    (2)     |header|    (3)     |
    +------+------+------------+------+------------+------+------------+

    +------+------+------------+  +------+------------+
(f) | RTP  | VOP  |VOP fragment|  | RTP  |VOP fragment|
    |header|header|    (1)     |  |header|    (2)     | . . .
    +------+------+------------+  +------+------------+
```

Figure 2: Examples of RTP Packetized MPEG-4 Visual Bitstream

```
      +------+-------------+  +------+------------+------------+
  (a) | RTP  |First half of|  | RTP  |Last half of|Video Packet|
      |header|  VP header  |  |header|  VP header |            |
      +------+-------------+  +------+------------+------------+

      +------+------+----------+  +------+---------+------+------------+
  (b) | RTP  | VOP  |First half|  | RTP  |Last half|  VP  |Video Packet|
      |header|header| of VP(1) |  |header| of VP(1)|header|    (2)     |
      +------+------+----------+  +------+---------+------+------------+
```

Figure 3: Examples of Prohibited RTP Packetization for MPEG-4 Visual

6. RTP Packetization of MPEG-4 Audio Bitstreams

This section specifies RTP packetization rules for MPEG-4 Audio bitstreams. MPEG-4 Audio streams MUST be formatted LATM (Low-overhead MPEG-4 Audio Transport Multiplex) [14496-3] streams, and the LATM-based streams are then mapped onto RTP packets as described in the sections below.

6.1. RTP Packet Format

LATM-based streams consist of a sequence of audioMuxElements that include one or more PayloadMux elements that carry the audio frames. A complete audioMuxElement or a part of one SHALL be mapped directly onto an RTP payload without any removal of audioMuxElement syntax elements (see Figure 4). The first byte of each audioMuxElement SHALL be located at the first payload location in an RTP packet.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|X|  CC   |M|     PT      |       sequence number         |RTP
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           timestamp                           |Header
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           synchronization source (SSRC) identifier            |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|            contributing source (CSRC) identifiers             |
|                             ....                              |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|                                                               |RTP
:                 audioMuxElement (byte aligned)                :Payload
|                                                               |
|                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               :...OPTIONAL RTP padding        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4 - An RTP packet for MPEG-4 Audio

In order to decode the audioMuxElement, the following muxConfigPresent information is required to be indicated by out-of-band means. When SDP is utilized for this indication, the media type parameter "cpresent" corresponds to the muxConfigPresent information (see Section 7.3). The following restrictions apply:

- o In the out-of-band configuration case, the number of PayloadMux elements contained in each audioMuxElement can only be set once. If more than one PayloadMux element is contained in each audioMuxElement, special care is required to ensure that the last RTP packet remains decodable.

- o To construct the audioMuxElement in the in-band configuration case, non-octet-aligned configuration data is inserted immediately before the one or more PayloadMux elements. Since the generation of RTP payloads with non-octet-aligned data is not possible with RTP hint tracks, as defined by the MP4 file format [14496-12] [14496-14], this document does not support RTP hint tracks for the in-band configuration case.

muxConfigPresent: If this value is set to 1 (in-band mode), the audioMuxElement SHALL include an indication bit "useSameStreamMux" and MAY include the configuration information for audio compression "StreamMuxConfig". The useSameStreamMux bit indicates whether the StreamMuxConfig element in the previous frame is applied in the current frame. If the useSameStreamMux bit indicates to use the StreamMuxConfig from the previous frame, but if the previous frame has been lost, the current frame may not be decodable. Therefore, in case of in-band mode, the StreamMuxConfig element SHOULD be transmitted repeatedly depending on the network condition. On the other hand, if muxConfigPresent is set to 0 (out-of-band mode), the StreamMuxConfig element is required to be transmitted by an out-of-band means. In case of SDP, the media type parameter "config" is utilized (see Section 7.3).

6.2. Use of RTP Header Fields for MPEG-4 Audio

Payload Type (PT): The assignment of an RTP payload type for this packet format is outside the scope of this document and will only be restricted 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 means of an out-of-band signaling protocol (e.g., H.245, SIP). In the dynamic assignment of RTP payload types for scalable streams, the server SHALL assign a different value to each layer. The dependency relationships between the enhanced layer and the base layer MUST be signaled as specified in [RFC5583]. An example of the use of such signaling for scalable audio streams can be found in [RFC5691].

Marker (M) bit: The marker bit indicates audioMuxElement boundaries. It is set to 1 to indicate that the RTP packet contains a complete audioMuxElement or the last fragment of an audioMuxElement.

Timestamp: The timestamp indicates the sampling instance of the first audio frame contained in the RTP packet. Timestamps are RECOMMENDED to start at a random value for security reasons.

Unless specified by an out-of-band means, the resolution of the timestamp is set to its default value of 90 kHz.

Sequence Number: Incremented by 1 for each RTP packet sent, starting, for security reasons, with a random value.

Other header fields are used as described in [RFC3550].

6.3. Fragmentation of MPEG-4 Audio Bitstream

It is RECOMMENDED to put one audioMuxElement in each RTP packet. If the size of an audioMuxElement can be kept small enough that the size of the RTP packet containing it does not exceed the size of the Path MTU, this will be no problem. If it cannot, the audioMuxElement SHALL be fragmented and spread across multiple packets.

7. Media Type Registration for MPEG-4 Audio/Visual Streams

The following sections describe the media type registrations for MPEG-4 Audio/Visual streams, which are registered in accordance with [RFC4855] and use the template of [RFC4288]. Media type registration and SDP usage for the MPEG-4 Visual stream are described in Sections 7.1 and 7.2, respectively, while media type registration and SDP usage for MPEG-4 Audio stream are described in Sections 7.3 and 7.4, respectively.

7.1. Media Type Registration for MPEG-4 Visual

The receiver MUST ignore any unspecified parameter in order to ensure that additional parameters can be added in any future revision of this specification.

Type name: video

Subtype name: MP4V-ES

Required parameters: none

Optional parameters:

"rate": This parameter is used only for RTP transport. It indicates the resolution of the timestamp field in the RTP header. If this parameter is not specified, its default value of 90000 (90 kHz) is used.

"profile-level-id": A decimal representation of MPEG-4 Visual Profile and Level indication value (profile_and_level_indication) defined in Table G-1 of [14496-2]. This parameter MAY be used in the capability exchange or session setup procedure to indicate the MPEG-4 Visual Profile and Level combination of which the MPEG-4 Visual codec is capable. If this parameter is not specified by the procedure, its default value of 1 (Simple Profile/Level 1) is used.

"config": This parameter SHALL be used to indicate the configuration of the corresponding MPEG-4 Visual bitstream. It SHALL NOT be used to indicate the codec capability in the capability exchange procedure. It is a hexadecimal representation of an octet string that expresses the MPEG-4 Visual configuration information, as defined in Subclause 6.2.1 ("Start codes") of [14496-2]. The configuration information is mapped onto the octet string most significant bit (MSB) first. The first bit of the configuration information SHALL be located at the MSB of the first octet. The configuration information indicated by this parameter SHALL be the same as the configuration information in the corresponding MPEG-4 Visual stream, except for first_half_vbv_occupancy and latter_half_vbv_occupancy (if they exist), which may vary in the repeated configuration information inside an MPEG-4 Visual stream. (See Subclause 6.2.1, "Start codes", of [14496-2].)

Published specification:

The specifications for MPEG-4 Visual streams are presented in [14496-2]. The RTP payload format is described in [RFC6416].

Encoding considerations:

Video bitstreams MUST be generated according to MPEG-4 Visual specifications [14496-2]. A video bitstream is binary data and MUST be encoded for non-binary transport (for email, the Base64 encoding is sufficient). This type is also defined for transfer via RTP. The RTP packets MUST be packetized according to the MPEG-4 Visual RTP payload format defined in [RFC6416].

Security considerations:

See Section 10 of [RFC6416].

Interoperability considerations:

MPEG-4 Visual provides a large and rich set of tools for the coding of visual objects. For effective implementation of the standard, subsets of the MPEG-4 Visual tool sets have been provided for use in specific applications. These subsets, called 'Profiles', limit the size of the tool set a decoder is required to implement. In order to restrict computational complexity, one or more Levels are set for each Profile. A Profile@Level combination allows:

* a codec builder to implement only the subset of the standard he needs, while maintaining interworking with other MPEG-4 devices included in the same combination, and

* checking whether MPEG-4 devices comply with the standard ('conformance testing').

The visual stream SHALL be compliant with the MPEG-4 Visual Profile@Level specified by the parameter "profile-level-id". Interoperability between a sender and a receiver may be achieved by specifying the parameter "profile-level-id" or by arranging a capability exchange/announcement procedure for this parameter.

Applications that use this media type:

Audio and visual streaming and conferencing tools

Additional information: none

Person and email address to contact for further information:

See Authors' Addresses section at the end of [RFC6416].

Intended usage: COMMON

Author:

See Authors' Addresses section at the end of [RFC6416].

Change controller:

IETF Audio/Video Transport Payloads working group delegated from the IESG.

7.2. Mapping to SDP for MPEG-4 Visual

The media type video/MP4V-ES string is mapped to fields in SDP [RFC4566], as follows:

- The media type (video) goes in SDP "m=" as the media name.

- The Media subtype (MP4V-ES) goes in SDP "a=rtpmap" as the encoding name.

- The optional parameter "rate" goes in "a=rtpmap" as the "clock rate".

- The optional parameter "profile-level-id" and "config" go in the "a=fmtp" line to indicate the coder capability and configuration, respectively. These parameters are expressed as a string, in the form of a semicolon-separated list of parameter=value pairs.

  Example usages for the "profile-level-id" parameter are:
  1 : MPEG-4 Visual Simple Profile/Level 1
  34 : MPEG-4 Visual Core Profile/Level 2
  145: MPEG-4 Visual Advanced Real Time Simple Profile/Level 1

7.2.1. Declarative SDP Usage for MPEG-4 Visual

The following are some examples of media representations in SDP:

Simple Profile/Level 1, rate=90000(90 kHz), "profile-level-id" and "config" are present in "a=fmtp" line:

```
m=video 49170/2 RTP/AVP 98
a=rtpmap:98 MP4V-ES/90000
a=fmtp:98 profile-level-id=1;config=000001B001000001B50900000100000
   00120008440FA282C2090A21F
```

Core Profile/Level 2, rate=90000(90 kHz), "profile-level-id" is present in "a=fmtp" line:

```
m=video 49170/2 RTP/AVP 98
a=rtpmap:98 MP4V-ES/90000
a=fmtp:98 profile-level-id=34
```

Advance Real Time Simple Profile/Level 1, rate=90000(90 kHz), "profile-level-id" is present in "a=fmtp" line:

```
m=video 49170/2 RTP/AVP 98
a=rtpmap:98 MP4V-ES/90000
a=fmtp:98 profile-level-id=145
```

7.3. Media Type Registration for MPEG-4 Audio

The receiver MUST ignore any unspecified parameter, to ensure that additional parameters can be added in any future revision of this specification.

Type name: audio

Subtype name: MP4A-LATM

Required parameters:

"rate": the "rate" parameter indicates the RTP timestamp "clock rate". The default value is 90000. Other rates MAY be indicated only if they are set to the same value as the audio sampling rate (number of samples per second).

In the presence of SBR, the sampling rates for the core encoder/ decoder and the SBR tool are different in most cases. Therefore, this parameter SHALL NOT be considered as the definitive sampling rate. If this parameter is used, the server must follow the rules below:

* When the presence of SBR is not explicitly signaled by the optional SDP parameters such as "object", "profile-level-id", or "config", this parameter SHALL be set to the core codec sampling rate.

* When the presence of SBR is explicitly signaled by the optional SDP parameters such as "object", "profile-level-id", or "config", this parameter SHALL be set to the SBR sampling rate.

NOTE: The optional parameter "SBR-enabled" in SDP "a=fmtp" is useful for implicit HE AAC / HE AAC v2 signaling. But the "SBR-enabled" parameter can also be used in the case of explicit HE AAC / HE AAC v2 signaling. Therefore, its existence (in itself) is not the criteria to determine whether or HE AAC / HE AAC v2 signaling is explicit.

Optional parameters:

"profile-level-id": a decimal representation of MPEG-4 Audio Profile Level indication value defined in [14496-3]. This parameter indicates which MPEG-4 Audio tool subsets the decoder is capable of using. If this parameter is not specified in the capability exchange or session setup procedure, its default value of 30 (Natural Audio Profile/Level 1) is used.

"MPS-profile-level-id": a decimal representation of the MPEG Surround Profile Level indication as defined in [14496-3]. This parameter indicates the support of the MPEG Surround profile and level by the decoder to be capable to decode the stream.

"object": a decimal representation of the MPEG-4 Audio Object Type value defined in [14496-3]. This parameter specifies the tool to be used by the decoder. It CAN be used to limit the capability within the specified "profile-level-id".

"bitrate": the data rate for the audio bitstream.

"cpresent": a boolean parameter that indicates whether audio payload configuration data has been multiplexed into an RTP payload (see Section 6.1). A 0 indicates the configuration data has not been multiplexed into an RTP payload, and in that case, the "config" parameter MUST be present; a 1 indicates that it has been multiplexed. The default if the parameter is omitted is 1. If this parameter is set to 1 and the "config" parameter is present, the multiplexed configuration data and the value of the "config" parameter SHALL be consistent.

"config": a hexadecimal representation of an octet string that expresses the audio payload configuration data "StreamMuxConfig", as defined in [14496-3]. Configuration data is mapped onto the octet string in an MSB-first basis. The first bit of the configuration data SHALL be located at the MSB of the first octet. In the last octet, zero-padding bits, if necessary, SHALL follow the configuration data. Senders MUST set the StreamMuxConfig elements taraBufferFullness and latmBufferFullness to their largest respective value, indicating that buffer fullness measures are not used in SDP. Receivers MUST ignore the value of these two elements contained in the "config" parameter.

"MPS-asc": a hexadecimal representation of an octet string that expresses audio payload configuration data "AudioSpecificConfig", as defined in [14496-3]. If this parameter is not present, the relevant signaling is performed by other means (e.g., in-band or contained in the "config" string).

The same mapping rules as for the "config" parameter apply.

"ptime": duration of each packet in milliseconds.

"SBR-enabled": a boolean parameter that indicates whether SBR-data can be expected in the RTP-payload of a stream. This parameter is relevant for an SBR-capable decoder if the presence of SBR cannot be detected from an out-of-band decoder configuration (e.g., contained in the "config" string).

If this parameter is set to 0, a decoder MAY expect that SBR is not used. If this parameter is set to 1, a decoder CAN up-sample the audio data with the SBR tool, regardless of whether or not SBR data is present in the stream.

If the presence of SBR cannot be detected from out-of-band configuration and the "SBR-enabled" parameter is not present, the parameter defaults to 1 for an SBR-capable decoder. If the resulting output sampling rate or the computational complexity is not supported, the SBR tool can be disabled or run in down-sampled mode.

The timestamp resolution at the RTP layer is determined by the "rate" parameter.

Published specification:

Encoding specifications are provided in [14496-3]. The RTP payload format specification is described in [RFC6416].

Encoding considerations:

This type is only defined for transfer via RTP.

Security considerations:

See Section 10 of [RFC6416].

Interoperability considerations:

MPEG-4 Audio provides a large and rich set of tools for the coding of audio objects. For effective implementation of the standard, subsets of the MPEG-4 Audio tool sets similar to those used in MPEG-4 Visual have been provided (see Section 7.1).

The audio stream SHALL be compliant with the MPEG-4 Audio Profile@ Level specified by the parameters "profile-level-id" and "MPS-profile-level-id". Interoperability between a sender and a receiver may be achieved by specifying the parameters "profile-level-id" and "MPS-profile-level-id" or by arranging in the capability exchange procedure to set this parameter mutually to the same value. Furthermore, the "object" parameter can be used to limit the capability within the specified Profile@Level in the capability exchange.

Applications that use this media type:

Audio and video streaming and conferencing tools.

Additional information: none

Personal and email address to contact for further information:

See Authors' Addresses section at the end of [RFC6416].

Intended usage: COMMON

Author:

See Authors' Addresses section at the end of [RFC6416].

Change controller:

IETF Audio/Video Transport Payloads working group delegated from the IESG.

7.4. Mapping to SDP for MPEG-4 Audio

The media type audio/MP4A-LATM string is mapped to fields in SDP [RFC4566], as follows:

- o The media type (audio) goes in SDP "m=" as the media name.
- o The Media subtype (MP4A-LATM) goes in SDP "a=rtpmap" as the encoding name.
- o The required parameter "rate" goes in "a=rtpmap" as the "clock rate".
- o The optional parameter "ptime" goes in SDP "a=ptime" attribute.
- o The optional parameters "profile-level-id", "MPS-profile-level-id", and "object" go in the "a=fmtp" line to indicate the coder capability.

The following are some examples of the "profile-level-id" value:
1 : Main Audio Profile Level 1
9 : Speech Audio Profile Level 1
15: High Quality Audio Profile Level 2
30: Natural Audio Profile Level 1
44: High Efficiency AAC Profile Level 2
48: High Efficiency AAC v2 Profile Level 2
55: Baseline MPEG Surround Profile (see ISO/IEC 23003-1) Level 3

The optional payload-format-specific parameters "bitrate", "cpresent", "config", "MPS-asc", and "SBR-enabled" also go in the "a=fmtp" line. These parameters are expressed as a string, in the form of a semicolon-separated list of parameter=value pairs.

7.4.1. Declarative SDP Usage for MPEG-4 Audio

The following sections contain some examples of the media representation in SDP.

Note that the "a=fmtp" line in some of the examples has been wrapped to fit the page; they would comprise a single line in the SDP file.

7.4.1.1. Example: In-Band Configuration

In this example, the audio configuration data appears in the RTP payload exclusively (i.e., the MPEG-4 audio configuration is known when a StreamMuxConfig element appears within the RTP payload).

```
m=audio 49230 RTP/AVP 96
a=rtpmap:96 MP4A-LATM/90000
a=fmtp:96 object=2; cpresent=1
```

The "clock rate" is set to 90 kHz. This is the default value, and the real audio sampling rate is known when the audio configuration data is received.

7.4.1.2. Example: 6 kbit/s CELP

This example shows a 6 kbit/s CELP (Code-Excited Linear Prediction) bitstream (with an audio sampling rate of 8 kHz).

```
m=audio 49230 RTP/AVP 96
a=rtpmap:96 MP4A-LATM/8000
a=fmtp:96 profile-level-id=9; object=8; cpresent=0;
  config=40008B18388380
a=ptime:20
```

In this example, audio configuration data is not multiplexed into the RTP payload and is described only in SDP. Furthermore, the "clock rate" is set to the audio sampling rate.

7.4.1.3. Example: 64 kbit/s AAC LC Stereo

This example shows a 64 kbit/s AAC LC stereo bitstream (with an audio sampling rate of 24 kHz).

```
m=audio 49230 RTP/AVP 96
a=rtpmap:96 MP4A-LATM/24000/2
a=fmtp:96 profile-level-id=1; bitrate=64000; cpresent=0;
  object=2; config=400026203fc0
```

In this example, audio configuration data is not multiplexed into the RTP payload and is described only in SDP. Furthermore, the "clock rate" is set to the audio sampling rate.

In this example, the presence of SBR cannot be determined by the SDP parameter set. The "clock rate" represents the core codec sampling rate. An SBR-enabled decoder can use the SBR tool to up-sample the audio data if the complexity and resulting output sampling rate permit.

7.4.1.4. Example: Use of the "SBR-enabled" Parameter

These two examples are identical to the example above with the exception of the "SBR-enabled" parameter. The presence of SBR is not signaled by the SDP parameters "object", "profile-level-id", and "config", but instead the "SBR-enabled" parameter is present. The "rate" parameter and the StreamMuxConfig contain the core codec sampling rate.

This example shows "SBR-enabled=0", with definitive and core codec sampling rates of 24 kHz.

```
m=audio 49230 RTP/AVP 96
a=rtpmap:96 MP4A-LATM/24000/2
a=fmtp:96 profile-level-id=1; bitrate=64000; cpresent=0;
  SBR-enabled=0; config=400026203fc0
```

This example shows "SBR-enabled=1", with core codec sampling rate of 24 kHz, and definitive and SBR sampling rates of 48 kHz:

```
m=audio 49230 RTP/AVP 96
a=rtpmap:96 MP4A-LATM/24000/2
a=fmtp:96 profile-level-id=1; bitrate=64000; cpresent=0;
  SBR-enabled=1; config=400026203fc0
```

In this example, the "clock rate" is still 24000, and this information is used for RTP timestamp calculation. The value of 24000 is used to support old AAC decoders. This makes the decoder supporting only AAC understand the HE AAC coded data, although only plain AAC is supported. A HE AAC decoder is able to generate output data with the SBR sampling rate.

7.4.1.5. Example: Hierarchical Signaling of SBR

When the presence of SBR is explicitly signaled by the SDP parameters "object", "profile-level-id", or "config", as in the example below, the StreamMuxConfig contains both the core codec sampling rate and the SBR sampling rate.

```
m=audio 49230 RTP/AVP 96
a=rtpmap:96 MP4A-LATM/48000/2
a=fmtp:96 profile-level-id=44; bitrate=64000; cpresent=0;
  config=40005623101fe0; SBR-enabled=1
```

This "config" string uses the explicit signaling mode 2.A (hierarchical signaling; see [14496-3]. This means that the AOT (Audio Object Type) is SBR (5) and SFI (Sampling Frequency Index) is 6 (24000 Hz), which refers to the underlying core codec sampling frequency. CC (Channel Configuration) is stereo (2), and the ESFI (Extension Sampling Frequency Index)=3 (48000) is referring to the sampling frequency of the extension tool (SBR).

7.4.1.6. Example: HE AAC v2 Signaling

HE AAC v2 decoders are required to always produce a stereo signal from a mono signal. Hence, there is no parameter necessary to signal the presence of PS.

This example shows "SBR-enabled=1" with 1 channel signaled in the "a=rtpmap" line and within the "config" parameter. The core codec sampling rate is 24 kHz; the definitive and SBR sampling rates are 48 kHz. The core codec channel configuration is mono; the PS channel configuration is stereo.

```
m=audio 49230 RTP/AVP 110
a=rtpmap:110 MP4A-LATM/24000/1
a=fmtp:110 profile-level-id=15; object=2; cpresent=0;
  config=400026103fc0; SBR-enabled=1
```

7.4.1.7. Example: Hierarchical Signaling of PS

This example shows 48 kHz stereo audio input.

```
m=audio 49230 RTP/AVP 110
a=rtpmap:110 MP4A-LATM/48000/2
a=fmtp:110 profile-level-id=48; cpresent=0; config=4001d613101fe0
```

The "config" parameter indicates explicit hierarchical signaling of PS and SBR. This configuration method is not supported by legacy AAC an HE AAC decoders, and these are therefore unable to decode the coded data.

7.4.1.8. Example: MPEG Surround

The following examples show how MPEG Surround configuration data can be signaled using SDP. The configuration is carried within the "config" string in the first example by using two different layers. The general parameters in this example are: AudioMuxVersion=1; allStreamsSameTimeFraming=1; numSubFrames=0; numProgram=0; numLayer=1. The first layer describes the HE AAC payload and signals the following parameters: ascLen=25; audioObjectType=2 (AAC LC); extensionAudioObjectType=5 (SBR); samplingFrequencyIndex=6 (24 kHz); extensionSamplingFrequencyIndex=3 (48 kHz); channelConfiguration=2 (2.0 channels). The second layer describes the MPEG Surround payload and specifies the following parameters: ascLen=110; AudioObjectType=30 (MPEG Surround); samplingFrequencyIndex=3 (48 kHz); channelConfiguration=6 (5.1 channels); sacPayloadEmbedding=1; SpatialSpecificConfig=(48 kHz; 32 slots; 525 tree; ResCoding=1; ResBands=[7,7,7,7]).

In this example, the signaling is carried by using two different LATM layers. The MPEG Surround payload is carried together with the AAC payload in a single layer as indicated by the sacPayloadEmbedding Flag.

```
m=audio 49230 RTP/AVP 96
a=rtpmap:96 MP4A-LATM/48000
a=fmtp:96 profile-level-id=1; bitrate=64000; cpresent=0;
  SBR-enabled=1;
  config=8FF8004192B11880FF0DDE3699F2408C00536C02313CF3CE0FF0
```

7.4.1.9. Example: MPEG Surround with Extended SDP Parameters

The following example is an extension of the configuration given above by the MPEG-Surround-specific parameters. The "MPS-asc" parameter specifies the MPEG Surround Baseline Profile at Level 3 (PLI55), and the "MPS-asc" string contains the hexadecimal representation of the MPEG Surround ASC [audioObjectType=30 (MPEG Surround); samplingFrequencyIndex=0x3 (48 kHz); channelConfiguration=6 (5.1 channels); sacPayloadEmbedding=1; SpatialSpecificConfig=(48 kHz; 32 slots; 525 tree; ResCoding=1; ResBands=[0,13,13,13])].

```
m=audio 49230 RTP/AVP 96
a=rtpmap:96 MP4A-LATM/48000
a=fmtp:96 profile-level-id=44; bitrate=64000; cpresent=0;
  config=40005623101fe0; MPS-profile-level-id=55;
  MPS-asc=F1B4CF920442029B501185B6DA00;
```

7.4.1.10. Example: MPEG Surround with Single-Layer Configuration

The following example shows how MPEG Surround configuration data can be signaled using the SDP "config" parameter. The configuration is carried within the "config" string using a single layer. The general parameters in this example are: AudioMuxVersion=1; allStreamsSameTimeFraming=1; numSubFrames=0; numProgram=0; numLayer=0. The single layer describes the combination of HE AAC and MPEG Surround payload and signals the following parameters: ascLen=101; audioObjectType=2 (AAC LC); extensionAudioObjectType=5 (SBR); samplingFrequencyIndex=7 (22.05 kHz); extensionSamplingFrequencyIndex=7 (44.1 kHz); channelConfiguration=2 (2.0 channels). A backward-compatible extension according to [14496-3/Amd.1] signals the presence of MPEG Surround payload data and specifies the following parameters: SpatialSpecificConfig=(44.1 kHz; 32 slots; 525 tree; ResCoding=0).

In this example, the signaling is carried by using a single LATM layer. The MPEG Surround payload is carried together with the HE AAC payload in a single layer.

```
m=audio 49230 RTP/AVP 96
a=rtpmap:96 MP4A-LATM/44100
a=fmtp:96 profile-level-id=44; bitrate=64000; cpresent=0;
  SBR-enabled=1; config=8FF8000652B920876A83A1F440884053620FF0;
  MPS-profile-level-id=55
```

8. IANA Considerations

This document updates the media subtypes "MP4A-LATM" and "MP4V-ES" from RFC 3016. The new registrations are in Sections 7.1 and 7.3 of this document.

9. Acknowledgements

The authors would like to thank Yoshihiro Kikuchi, Yoshinori Matsui, Toshiyuki Nomura, Shigeru Fukunaga, and Hideaki Kimata for their work on RFC 3016, and Ali Begen, Keith Drage, Roni Even, and Qin Wu for their valuable input and comments on this document.

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 document are confidentiality, integrity, and source authenticity. Confidentiality is achieved by encryption of the RTP payload, and integrity of the RTP packets is achieved through a suitable cryptographic integrity protection mechanism. 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 whether or not an RTP packet is from a member of the RTP session.

Note that most MPEG-4 codecs define an extension mechanism to transmit extra data within a stream that is gracefully skipped by decoders that do not support this extra data. This may be used to transmit unwanted data in an otherwise valid stream.

The appropriate mechanism to provide security to RTP and payloads following this 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 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] (e.g., for RTP over TCP), but other alternatives may also exist.

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. The complete MPEG-4 System allows for transport of a wide range of content, including Java applets (MPEG-J) and scripts. Since this payload format is restricted to audio and video streams, it is not possible to transport such active content in this format.

11. Differences to RFC 3016

The RTP payload format for MPEG-4 Audio as specified in RFC 3016 is used by the 3GPP PSS service [3GPP]. However, there are some misalignments between RFC 3016 and the 3GPP PSS specification that are addressed by this update:

- o The audio payload format (LATM) referenced in this document is the newer format specified in [14496-3], which is binary compatible to the format used in [3GPP]. This newer format is not binary compatible with the LATM referenced in RFC 3016, which is specified in [14496-3:1999/Amd.1:2000].

- o The audio signaling format (StreamMuxConfig) referenced in this document is binary compatible to the format used in [3GPP]. The StreamMuxConfig element has also been revised by MPEG since RFC 3016.

- o The use of an audio parameter "SBR-enabled" is now defined in this document, which is used by 3GPP implementations [3GPP]. RFC 3016 does not define this parameter.

- o The "rate" parameter is defined unambiguously in this document for the case of presence of SBR (Spectral Band Replication). In RFC 3016, the definition of the "rate" parameter is ambiguous.

- o The number of audio channels parameter is defined unambiguously in this document for the case of presence of PS (Parametric Stereo). At the time RFC 3016 was written, PS was not yet defined.

Furthermore, some comments have been addressed and signaling support for MPEG Surround [23003-1] was added.

Below is a summary of the changes in requirements by this update:

- o In the dynamic assignment of RTP payload types for scalable MPEG-4 Audio streams, the server SHALL assign a different value to each layer.

- o The dependency relationships between the enhanced layer and the base layer for scalable MPEG-4 Audio streams MUST be signaled as specified in [RFC5583].

- o If the size of an audioMuxElement is so large that the size of the RTP packet containing it does exceed the size of the Path MTU, the audioMuxElement SHALL be fragmented and spread across multiple packets.

- o The receiver MUST ignore any unspecified parameter in order to ensure that additional parameters can be added in any future revision of this specification.

12. References

12.1. Normative References

[14496-2] MPEG, "ISO/IEC International Standard 14496-2 - Coding of audio-visual objects, Part 2: Visual", 2003.

[14496-3] MPEG, "ISO/IEC International Standard 14496-3 - Coding of audio-visual objects, Part 3 Audio", 2009.

[14496-3/Amd.1] MPEG, "ISO/IEC International Standard 14496-3 - Coding of audio-visual objects, Part 3: Audio, Amendment 1: HD-AAC profile and MPEG Surround signaling", 2009.

[23003-1] MPEG, "ISO/IEC International Standard 23003-1 - MPEG Surround (MPEG D)", 2007.

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

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

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

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

[RFC4629] Ott, H., Bormann, C., Sullivan, G., Wenger, S., and R. Even, "RTP Payload Format for ITU-T Rec", RFC 4629, January 2007.

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

[RFC5583] Schierl, T. and S. Wenger, "Signaling Media Decoding Dependency in the Session Description Protocol (SDP)", RFC 5583, July 2009.

12.2. Informative References

[14496-1] MPEG, "ISO/IEC International Standard 14496-1 - Coding of audio-visual objects, Part 1 Systems", 2004.

[14496-12] MPEG, "ISO/IEC International Standard 14496-12 - Coding of audio-visual objects, Part 12 ISO base media file format".

[14496-14] MPEG, "ISO/IEC International Standard 14496-14 - Coding of audio-visual objects, Part 12 MP4 file format".

[14496-3:1999/Amd.1:2000] MPEG, "ISO/IEC International Standard 14496-3 - Coding of audio-visual objects, Part 3 Audio, Amendment 1: Audio extensions", 2000.

[3GPP] 3GPP, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Transparent end-to-end Packet-switched Streaming Service (PSS); Protocols and codecs (Release 9)", 3GPP TS 26.234 V9.5.0, December 2010.

[H245] International Telecommunication Union, "Control protocol for multimedia communication", ITU Recommendation H.245, December 2009.

[H261] International Telecommunication Union, "Video codec for audiovisual services at p x 64 kbit/s", ITU Recommendation H.261, March 1993.

[H323] International Telecommunication Union, "Packet-based multimedia communications systems", ITU Recommendation H.323, December 2009.

[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.

[RFC3016] Kikuchi, Y., Nomura, T., Fukunaga, S., Matsui, Y., and H. Kimata, "RTP Payload Format for MPEG-4 Audio/Visual Streams", RFC 3016, November 2000.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[RFC3640] van der Meer, J., Mackie, D., Swaminathan, V., Singer, D., and P. Gentric, "RTP Payload Format for Transport of MPEG-4 Elementary Streams", RFC 3640, November 2003.

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

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

[RFC4628] Even, R., "RTP Payload Format for H.263 Moving RFC 2190 to Historic Status", RFC 4628, January 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.

[RFC5691] de Bont, F., Doehla, S., Schmidt, M., and R. Sperschneider, "RTP Payload Format for Elementary Streams with MPEG Surround Multi-Channel Audio", RFC 5691, October 2009.

Authors' Addresses

Malte Schmidt
Dolby Laboratories
Deutschherrnstr. 15-19
90537 Nuernberg
DE

Phone: +49 911 928 91 42
EMail: malte.schmidt@dolby.com

Frans de Bont
Philips Electronics
High Tech Campus 36
5656 AE Eindhoven
NL

Phone: +31 40 2740234
EMail: frans.de.bont@philips.com

Stefan Doehla
Fraunhofer IIS
Am Wolfmantel 33
91058 Erlangen
DE

Phone: +49 9131 776 6042
EMail: stefan.doehla@iis.fraunhofer.de

Jaehwan Kim
LG Electronics Inc.
VCS/HE, 16Fl. LG Twin Towers
Yoido-Dong, YoungDungPo-Gu,
Seoul 150-721
Korea

Phone: +82 10 6225 0619
EMail: kjh1905m@naver.com