Network Working Group
Request for Comments: 3016
Category: Standards Track

Y. Kikuchi
Toshiba
T. Nomura
NEC
S. Fukunaga
Oki
Y. Matsui
Matsushita
H. Kimata
NTT

November 2000

RTP Payload Format for MPEG-4 Audio/Visual Streams

Status of this Memo

Copyright Notice

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. For the purpose of directly mapping MPEG-4 Audio/Visual bitstreams onto RTP packets, it provides specifications for the use of RTP header fields and also specifies fragmentation rules. It also provides specifications for Multipurpose Internet Mail Extensions (MIME) type registrations and the use of Session Description Protocol (SDP).

1. Introduction

The RTP payload formats described in this document specify how MPEG-4 Audio [3][5] and MPEG-4 Visual streams [2][4] 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 [6]. 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 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. The streams are directly mapped onto RTP packets without using MPEG-4 Systems Sync Layer. Other examples are SIP and RTSP where MIME and SDP are used. MIME types 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 an 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, 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 resilience 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 new features: high coding efficiency; high error resiliency; multiple, arbitrary shape object-based coding; etc. [2]. It covers a wide range of bitrates from scores of Kbps to several Mbps. 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 resilience functionalities of MPEG-4 Visual.

The fragmentation rule recommends not to map more than one VOP in an RTP packet 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 3.2 and marker bit and timestamp in section 3.1.)

While the additional media specific RTP header defined for such video coding tools as H.261 or MPEG-1/2 is effective in helping to recover picture headers corrupted by packet losses, MPEG-4 Visual has already error resilience 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/TS, 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 a new kind of 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 the RTP packets without using MPEG-4 Systems.

While LATM has several multiplexing features as follows;

- Carrying configuration information with audio data,
- Concatenation of multiple audio frames in one audio stream,
- Multiplexing multiple objects (programs),
- Multiplexing scalable layers,

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 packets 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, a restriction is applied to the dynamic assignment rule of payload type (PT) values (see section 4.2).

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 (RFC 2733) and Redundant Audio Data (RFC 2198), MAY be applied to improve error resiliency.

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [7].

3. RTP Packetization of MPEG-4 Visual bitstream

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 6.2.1 "Start codes" of ISO/IEC 14496-2 [2][9][4]) The configuration information MAY additionally be specified by some out-of-band means. If needed for an H.323 terminal, H.245 codepoint "decoderConfigurationInformation" MUST be used for this purpose. If needed by systems using MIME content type and SDP parameters, e.g., SIP and RTSP, the optional parameter "config" MUST be used to specify the configuration information (see 5.1 and 5.2).

When the short video header mode is used, the RTP payload format for H.263 SHOULD be used (the format defined in RFC 2429 is RECOMMENDED, but the RFC 2190 format MAY be used for compatibility with older implementations).

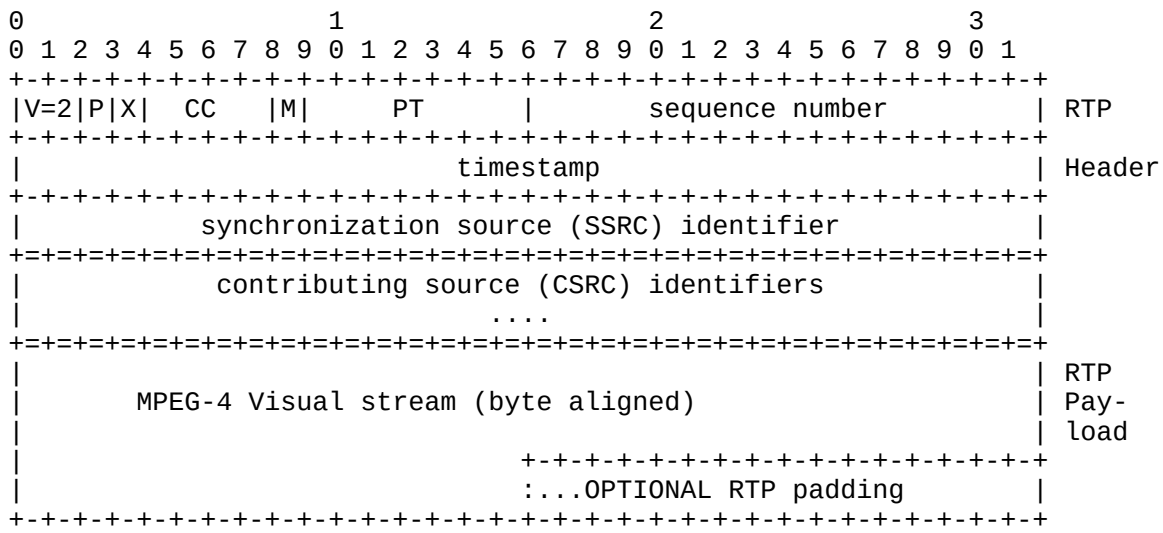

Figure 1 - An RTP packet for MPEG-4 Visual stream

3.1 Use of RTP header fields for MPEG-4 Visual

Payload Type (PT): The assignment of an RTP payload type for this new packet format is outside the scope of this document, and will not be specified here. It is expected that the RTP profile for a particular class of applications will assign a payload type for this encoding, or if that is not done then a payload type in the dynamic range SHALL be chosen by means of an out of band signaling protocol (e.g., H.245, SIP, etc).

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

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

Marker (M) bit: The marker bit is set to one 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 one.

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.

- 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 are derived from the timestamp fields in the VOP header (modulo_time_base and vop_time_increment).
- 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.
- 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 90kHz, unless specified by an out-of-band means (e.g., SDP parameter or MIME parameter as defined in section 5).

Other header fields are used as described in RFC 1889 [8].

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

In the following, header means one of the following:

- Configuration information (Visual Object Sequence Header, Visual Object Header and Video Object Layer Header)
- visual_object_sequence_end_code
- 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())
- The video packet header (video_packet_header() excluding next_resync_marker())
- The header of gob_layer()
  See 6.2.1 "Start codes" of ISO/IEC 14496-2 [2][9][4] for the definition of the configuration information and the entry point functions.

(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 one and the beginning of RTP payload corresponds to a start code. (See timestamp and marker bit in section 3.1.)

(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. Note: Rule (5) does not apply when 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. In this case, 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().

3.3 Examples of packetized MPEG-4 Visual bitstream

Figure 2 shows examples of RTP packets generated based on the criteria described in 3.2

(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 3.2 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 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 immediately following GOV and/or (a part of) VOP can be packetized into a single RTP packet as in this example.

(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, (a part of) the following VOP 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 bit-rate 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 considering the packet-loss rate and the bit-rate 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. On the other hand, it is not recommended to use it in error-prone environment since it provides only poor packet loss resiliency.

Figure 3 shows examples of RTP packets prohibited by the criteria of 3.2.

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

When concatenating more than one video packets into an RTP packet, VOP header or video_packet_header() shall not be placed in the middle of the RTP payload. The packetization as in (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).

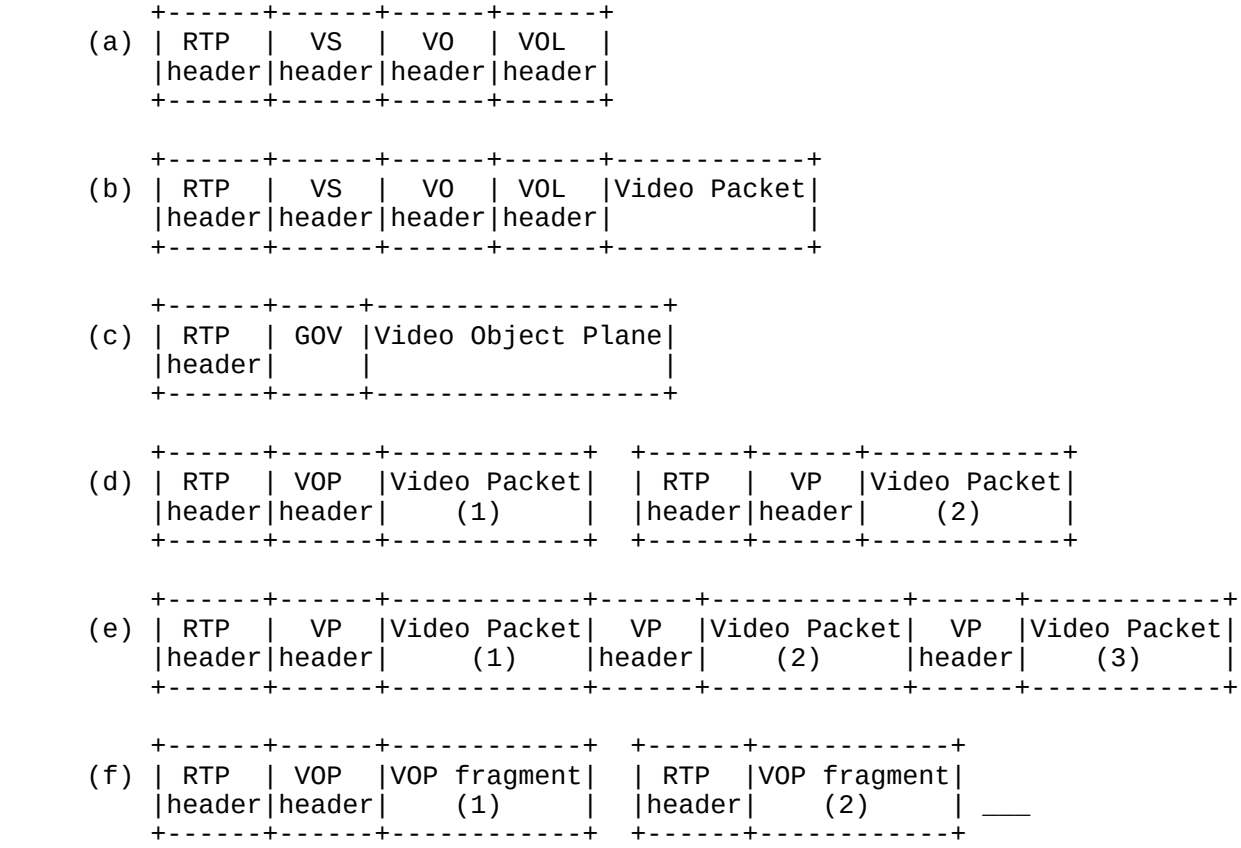

Figure 2 - Examples of RTP packetized MPEG-4 Visual bitstream

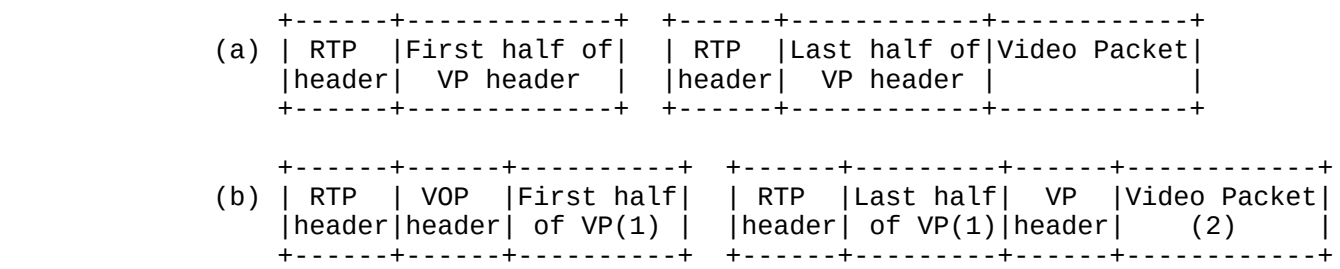

Figure 3 - Examples of prohibited RTP packetization for MPEG-4 Visual bitstream

4. RTP Packetization of MPEG-4 Audio bitstream

This section specifies RTP packetization rules for MPEG-4 Audio bitstreams. MPEG-4 Audio streams MUST be formatted by LATM (Low-overhead MPEG-4 Audio Transport Multiplex) tool [5], and the LATM-based streams are then mapped onto RTP packets as described the three sections below.

4.1 RTP Packet Format

LATM-based streams consist of a sequence of audioMuxElements that include one or more 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.

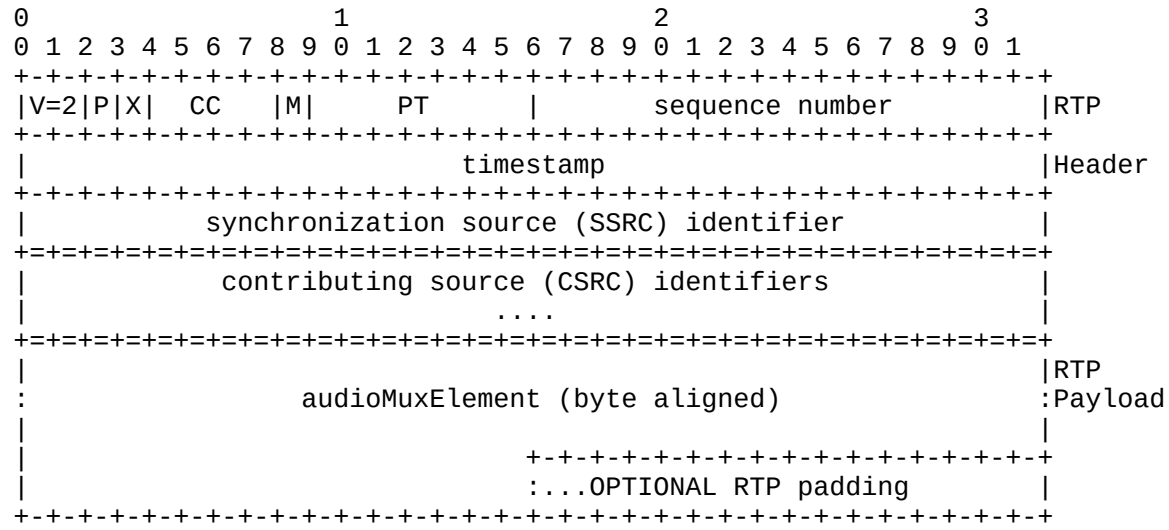

Figure 4 - An RTP packet for MPEG-4 Audio

In order to decode the audioMuxElement, the following muxConfigPresent information is required to be indicated by an out-of-band means. When SDP is utilized for this indication, MIME parameter "cpresent" corresponds to the muxConfigPresent information (see section 5.3).

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-band mode), the StreamMuxConfig element is required to be transmitted by an out-of-band means. In case of SDP, MIME parameter "config" is utilized (see section 5.3).

4.2 Use of RTP Header Fields for MPEG-4 Audio

Payload Type (PT): The assignment of an RTP payload type for this new packet format is outside the scope of this document, and will not be specified here. It is expected that the RTP profile for a particular class of applications will assign a payload type for this encoding,

or if that is not done then a payload type in the dynamic range shall be chosen by means of an out of band signaling protocol (e.g., H.245, SIP, etc). In the dynamic assignment of RTP payload types for scalable streams, a different value SHOULD be assigned to each layer. The assigned values SHOULD be in order of enhance layer dependency, where the base layer has the smallest value.

Marker (M) bit: The marker bit indicates audioMuxElement boundaries. It is set to one 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 one for each RTP packet sent, starting, for security reasons, with a random value.

Other header fields are used as described in RFC 1889 [8].

4.3 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 MAY be fragmented and spread across multiple packets.

5. MIME type registration for MPEG-4 Audio/Visual streams

The following sections describe the MIME type registrations for MPEG-4 Audio/Visual streams. MIME type registration and SDP usage for the MPEG-4 Visual stream are described in Sections 5.1 and 5.2, respectively, while MIME type registration and SDP usage for MPEG-4 Audio stream are described in Sections 5.3 and 5.4, respectively.

5.1 MIME type registration for MPEG-4 Visual

MIME media type name: video

MIME 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 (90kHz) 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 ISO/IEC 14496-2 [2][4]. This parameter MAY be used in the capability exchange or session setup procedure to indicate 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 ISO/IEC14496-2 [2][4][9]. The configuration information is mapped onto the octet string in an MSB-first basis. 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 exist, which may vary in the repeated configuration information inside an MPEG-4 Visual stream (See 6.2.1 Start codes of ISO/IEC14496-2).

Example usages for these parameters are:

- MPEG-4 Visual Simple Profile/Level 1:
  Content-type: video/mp4v-es; profile-level-id=1

- MPEG-4 Visual Core Profile/Level 2:
  Content-type: video/mp4v-es; profile-level-id=34

- MPEG-4 Visual Advanced Real Time Simple Profile/Level 1:
  Content-type: video/mp4v-es; profile-level-id=145

Published specification:
The specifications for MPEG-4 Visual streams are presented in ISO/IEC 14469-2 [2][4][9]. The RTP payload format is described in RFC 3016.

Encoding considerations:
Video bitstreams MUST be generated according to MPEG-4 Visual specifications (ISO/IEC 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 RFC 3016.

Security considerations:
See section 6 of RFC 3016.

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:

o 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

o 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" in MIME content, or by arranging in the capability exchange/announcement procedure to set this parameter mutually to the same value.

Applications which use this media type:
Audio and visual streaming and conferencing tools, Internet messaging and Email applications.

Additional information: none

Person & email address to contact for further information:
The authors of RFC 3016. (See section 8.)

Intended usage: COMMON

Author/Change controller:
The authors of RFC 3016. (See section 8.)

5.2 SDP usage of MPEG-4 Visual

The MIME media type video/MP4V-ES string is mapped to fields in the Session Description Protocol (SDP), RFC 2327, as follows:

- The MIME type (video) goes in SDP "m=" as the media name.
- The MIME 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 MIME media type string, in the form of as a semicolon separated list of parameter=value pairs.

The following are some examples of media representation in SDP:

Simple Profile/Level 1, rate=90000(90kHz), "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=000001B001000001B509000001000000012
   0008440FA282C2090A21F
```

Core Profile/Level 2, rate=90000(90kHz), "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(90kHz), "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
```

5.3 MIME type registration of MPEG-4 Audio

MIME media type name: audio

MIME subtype name: MP4A-LATM

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

Optional parameters:
profile-level-id: a decimal representation of MPEG-4 Audio Profile Level indication value defined in ISO/IEC 14496-1 ([6] and its amendments). 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.

object: a decimal representation of the MPEG-4 Audio Object Type value defined in ISO/IEC 14496-3 [5]. This parameter specifies the tool to be used by the coder. It CAN be used to limit the capability within the specified "profile-level-id".

bitrate: the data rate for the audio bit stream.

cpresent: a boolean parameter indicates whether audio payload configuration data has been multiplexed into an RTP payload (see section 4.1). A 0 indicates the configuration data has not been multiplexed into an RTP payload, a 1 indicates that it has. The default if the parameter is omitted is 1.

config: a hexadecimal representation of an octet string that expresses the audio payload configuration data "StreamMuxConfig", as defined in ISO/IEC 14496-3 [5] (see section 4.1). 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.

ptime: RECOMMENDED duration of each packet in milliseconds.

Published specification:
Payload format specifications are described in this document. Encoding specifications are provided in ISO/IEC 14496-3 [3][5].

Encoding considerations:
This type is only defined for transfer via RTP.

Security considerations:
See Section 6 of RFC 3016.

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 5.1).

The audio stream SHALL be compliant with the MPEG-4 Audio 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" in MIME content, 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 capability exchange.

Applications which use this media type:
Audio and video streaming and conferencing tools.

Additional information: none

Personal & email address to contact for further information:
See Section 8 of RFC 3016.

Intended usage: COMMON

Author/Change controller:
See Section 8 of RFC 3016.

5.4 SDP usage of MPEG-4 Audio

The MIME media type audio/MP4A-LATM string is mapped to fields in the Session Description Protocol (SDP), RFC 2327, as follows:

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

"bitrate", "cpresent" and "config" go in the "a=fmtp" line. These parameters are expressed as a MIME media type string, in the form of as a semicolon separated list of parameter=value pairs.

The following are some examples of the media representation in SDP:

For 6 kb/s CELP bitstreams (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=9128B1071070
a=ptime:20
```

For 64 kb/s AAC LC stereo bitstreams (with an audio sampling rate of 24 kHz),

```
m=audio 49230 RTP/AVP 96
a=rtpmap:96 MP4A-LATM/24000
a=fmtp:96 profile-level-id=1; bitrate=64000; cpresent=0;
config=9122620000
```

In the above two examples, 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.

If the clock rate has been set to its default value and it is necessary to obtain the audio sampling rate, this can be done by parsing the "config" parameter (see the following example).

```
m=audio 49230 RTP/AVP 96
a=rtpmap:96 MP4A-LATM/90000
a=fmtp:96 object=8; cpresent=0; config=9128B1071070
```

The following example shows that the audio configuration data appears in the RTP payload.

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

6. Security Considerations

RTP packets using the payload format defined in this specification are subject to the security considerations discussed in the RTP specification [8]. This implies that confidentiality of the media streams is achieved by encryption. Because the data compression used with this payload format is applied end-to-end, encryption may be performed on the compressed data so there is no conflict between the two operations.

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.

7. References

1 Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

2 ISO/IEC 14496-2:1999, "Information technology - Coding of audio-visual objects - Part2: Visual".

3 ISO/IEC 14496-3:1999, "Information technology - Coding of audio-visual objects - Part3: Audio".

4 ISO/IEC 14496-2:1999/Amd.1:2000, "Information technology - Coding of audio-visual objects - Part 2: Visual, Amendment 1: Visual extensions".

5 ISO/IEC 14496-3:1999/Amd.1:2000, "Information technology - Coding of audio-visual objects - Part3: Audio, Amendment 1: Audio extensions".

6 ISO/IEC 14496-1:1999, "Information technology - Coding of audio-visual objects - Part1: Systems".

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

8 Schulzrinne, H., Casner, S., Frederick, R. and V. Jacobson "RTP: A Transport Protocol for Real Time Applications", RFC 1889, January 1996.

9 ISO/IEC 14496-2:1999/Cor.1:2000, "Information technology - Coding of audio-visual objects - Part2: Visual, Technical corrigendum 1".

8. Authors' Addresses

Yoshihiro Kikuchi
Toshiba corporation
1, Komukai Toshiba-cho, Saiwai-ku, Kawasaki, 212-8582, Japan

EMail: yoshihiro.kikuchi@toshiba.co.jp

Yoshinori Matsui
Matsushita Electric Industrial Co., LTD.
1006, Kadoma, Kadoma-shi, Osaka, Japan

EMail: matsui@drl.mei.co.jp

Toshiyuki Nomura
NEC Corporation
4-1-1,Miyazaki,Miyamae-ku,Kawasaki,JAPAN

EMail: t-nomura@ccm.cl.nec.co.jp

Shigeru Fukunaga
Oki Electric Industry Co., Ltd.
1-2-27 Shiromi, Chuo-ku, Osaka 540-6025 Japan.

EMail: fukunaga444@oki.co.jp

Hideaki Kimata
Nippon Telegraph and Telephone Corporation
1-1, Hikari-no-oka, Yokosuka-shi, Kanagawa, Japan

EMail: kimata@nttvdt.hil.ntt.co.jp

9. Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.