Network Working Group
Request for Comments: 3640
Category: Standards Track

J. van der Meer
Philips Electronics
D. Mackie
Apple Computer
V. Swaminathan
Sun Microsystems Inc.
D. Singer
Apple Computer
P. Gentric
Philips Electronics

November 2003

RTP Payload Format for Transport of MPEG-4 Elementary Streams

Status of this Memo

Copyright Notice

Abstract

The Motion Picture Experts Group (MPEG) Committee (ISO/IEC JTC1/SC29 WG11) is a working group in ISO that produced the MPEG-4 standard. MPEG defines tools to compress content such as audio-visual information into elementary streams. This specification defines a simple, but generic RTP payload format for transport of any non-multiplexed MPEG-4 elementary stream.

Table of Contents

1. Introduction 3
2. Carriage of MPEG-4 Elementary Streams Over RTP 4
   2.1. Signaling by MIME Format Parameters 4
   2.2. MPEG Access Units 5
   2.3. Concatenation of Access Units 5
   2.4. Fragmentation of Access Units 6
   2.5. Interleaving 6
   2.6. Time Stamp Information 7
   2.7. State Indication of MPEG-4 System Streams 8
   2.8. Random Access Indication 8

2.9. Carriage of Auxiliary Information 8
2.10. MIME Format Parameters and Configuring Conditional Field 8
2.11. Global Structure of Payload Format 9
2.12. Modes to Transport MPEG-4 Streams 9
2.13. Alignment with RFC 3016 10
3. Payload Format 10
3.1. Usage of RTP Header Fields and RTCP 10
3.2. RTP Payload Structure 11
3.2.1. The AU Header Section 11
3.2.1.1. The AU-header 12
3.2.2. The Auxiliary Section 14
3.2.3. The Access Unit Data Section 15
3.2.3.1. Fragmentation. 16
3.2.3.2. Interleaving 16
3.2.3.3. Constraints for Interleaving 17
3.2.3.4. Crucial and Non-Crucial AUs with MPEG-4 System Data 20
3.3. Usage of this Specification. 21
3.3.1. General. 21
3.3.2. The Generic Mode 22
3.3.3. Constant Bit Rate CELP 22
3.3.4. Variable Bit Rate CELP 23
3.3.5. Low Bit Rate AAC 24
3.3.6. High Bit Rate AAC. 25
3.3.7. Additional Modes 26
4. IANA Considerations. 27
4.1. MIME Type Registration 27
4.2. Registration of Mode Definitions with IANA 33
4.3. Concatenation of Parameters. 33
4.4. Usage of SDP 34
4.4.1. The a=fmtp Keyword 34
5. Security Considerations. 34
6. Acknowledgements 35
APPENDIX: Usage of this Payload Format. 36
Appendix A. Interleave Analysis 36
A. Examples of Delay Analysis with Interleave. 36
A.1. Introduction 36
A.2. De-interleaving and Error Concealment 36
A.3. Simple Group Interleave 36
A.3.1. Introduction 36
A.3.2. Determining the De-interleave Buffer Size . . . 37
A.3.3. Determining the Maximum Displacement 37
A.4. More Subtle Group Interleave 38
A.4.1. Introduction 38
A.4.2. Determining the De-interleave Buffer Size. . . . 38
A.4.3. Determining the Maximum Displacement 39
A.5. Continuous Interleave 39
A.5.1. Introduction 39

A.5.2. Determining the De-interleave Buffer Size . . . 40
A.5.3. Determining the Maximum Displacement 40
References 41
Normative References 41
Informative References 41
Authors' Addresses 42
Full Copyright Statement 43

1. Introduction

The MPEG Committee is Working Group 11 (WG11) in ISO/IEC JTC1 SC29 that specified the MPEG-1, MPEG-2 and, more recently, the MPEG-4 standards [1]. The MPEG-4 standard specifies compression of audio-visual data into, for example an audio or video elementary stream. In the MPEG-4 standard, these streams take the form of audio-visual objects that may be arranged into an audio-visual scene by means of a scene description. Each MPEG-4 elementary stream consists of a sequence of Access Units; examples of an Access Unit (AU) are an audio frame and a video picture.

This specification defines a general and configurable payload structure to transport MPEG-4 elementary streams, in particular MPEG-4 audio (including speech) streams, MPEG-4 video streams and also MPEG-4 systems streams, such as BIFS (BInary Format for Scenes), OCI (Object Content Information), OD (Object Descriptor) and IPMP (Intellectual Property Management and Protection) streams. The RTP payload defined in this document is simple to implement and reasonably efficient. It allows for optional interleaving of Access Units (such as audio frames) to increase error resiliency in packet loss.

Some types of MPEG-4 elementary streams include "crucial" information whose loss cannot be tolerated. However, RTP does not provide reliable transmission, so receipt of that crucial information is not assured. Section 3.2.3.4 specifies how stream state is conveyed so that the receiver can detect the loss of crucial information and cease decoding until the next random access point has been received. Applications transmitting streams that include crucial information, such as OD commands, BIFS commands, or programmatic content such as MPEG-J (Java) and ECMAScript, should include random access points, at a suitable periodicity depending upon the probability of loss, in order to reduce stream corruption to an acceptable level. An example is the carousel mechanism as defined by MPEG in ISO/IEC 14496-1 [1].

Such applications may also employ additional protocols or services to reduce the probability of loss. At the RTP layer, these measures include payload formats and profiles for retransmission or forward error correction (such as in RFC 2733 [10]), that must be employed

with due consideration to congestion control. Another solution that may be appropriate for some applications is to carry RTP over TCP (such as in RFC 2326 [8], section 10.12). At the network layer, resource allocation or preferential service may be available to reduce the probability of loss. For a general description of methods to repair streaming media, see RFC 2354 [9].

Though the RTP payload format defined in this document is capable of transporting any MPEG-4 stream, other, more specific, formats may exist, such as RFC 3016 [12] for transport of MPEG-4 video (ISO/IEC 14496 [1] part 2).

Configuration of the payload is provided to accommodate the transportation of any MPEG-4 stream at any possible bit rate. However, for a specific MPEG-4 elementary stream typically only very few configurations are needed. So as to allow for the design of simplified, but dedicated receivers, this specification requires that specific modes be defined for transport of MPEG-4 streams. This document defines modes for MPEG-4 CELP and AAC streams, as well as a generic mode that can be used to transport any MPEG-4 stream. In the future, new RFCs are expected to specify additional modes for the transportation of MPEG-4 streams.

The RTP payload format defined in this document specifies carriage of system-related information that is often equivalent to the information that may be contained in the MPEG-4 Sync Layer (SL) as defined in MPEG-4 Systems [1]. This document does not prescribe how to transcode or map information from the SL to fields defined in the RTP payload format. Such processing, if any, is left to the discretion of the application. However, to anticipate the need for the transportation of any additional system-related information in the future, an auxiliary field can be configured that may carry any such data.

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 BCP 14, RFC 2119 [4].

2. Carriage of MPEG-4 Elementary Streams over RTP

2.1. Signaling by MIME Format Parameters

With this payload format, a single MPEG-4 elementary stream can be transported. Information on the type of MPEG-4 stream carried in the payload is conveyed by MIME format parameters, as in an SDP [5] message or by other means (see section 4). These MIME format parameters specify the configuration of the payload. To allow for simplified and dedicated receivers, a MIME format parameter is

available to signal a specific mode of using this payload. A mode definition MAY include the type of MPEG-4 elementary stream, as well as the applied configuration, so as to avoid the need for receivers to parse all MIME format parameters. The applied mode MUST be signaled.

2.2. MPEG Access Units

For carriage of compressed audio-visual data, MPEG defines Access Units. An MPEG Access Unit (AU) is the smallest data entity to which timing information is attributed. In the case of audio, an Access Unit may represent an audio frame and in the case of video, a picture. MPEG Access Units are octet-aligned by definition. If, for example, an audio frame is not octet-aligned, up to 7 zero-padding bits MUST be inserted at the end of the frame to achieve the octet-aligned Access Units, as required by the MPEG-4 specification. MPEG-4 decoders MUST be able to decode AUs in which such padding is applied.

Consistent with the MPEG-4 specification, this document requires that each MPEG-4 part 2 video Access Unit include all the coded data of a picture, any video stream headers that may precede the coded picture data, and any video stream stuffing that may follow it, up to but not including the startcode indicating the start of a new video stream or the next Access Unit.

2.3. Concatenation of Access Units

Frequently it is possible to carry multiple Access Units in one RTP packet. This is particularly useful for audio; for example, when AAC is used for encoding a stereo signal at 64 kbits/sec, AAC frames contain on average, approximately 200 octets. On a LAN with a 1500 octet MTU, this would allow an average of 7 complete AAC frames to be carried per RTP packet.

Access Units may have a fixed size in octets, but a variable size is also possible. To facilitate parsing in the case of multiple concatenated AUs in one RTP packet, the size of each AU is made known to the receiver. When concatenating in the case of a constant AU size, this size is communicated "out of band" through a MIME format parameter. When concatenating in case of variable size AUs, the RTP payload carries "in band" an AU size field for each contained AU.

In combination with the RTP payload length, the size information allows the RTP payload to be split by the receiver back into the individual AUs.

To simplify the implementation of RTP receivers, it is required that when multiple AUs are carried in an RTP packet, each AU MUST be complete, i.e., the number of AUs in an RTP packet MUST be integral.

In addition, an AU MUST NOT be repeated in other RTP packets; hence repetition of an AU is only possible when using a duplicate RTP packet.

2.4. Fragmentation of Access Units

MPEG allows for very large Access Units. Since most IP networks have significantly smaller MTU sizes, this payload format allows for the fragmentation of an Access Unit over multiple RTP packets. Hence, when an IP packet is lost after IP-level fragmentation, only an AU fragment may get lost instead of the entire AU. To simplify the implementation of RTP receivers, an RTP packet SHALL either carry one or more complete Access Units or a single fragment of one AU, i.e., packets MUST NOT contain fragments of multiple Access Units.

2.5. Interleaving

When an RTP packet carries a contiguous sequence of Access Units, the loss of such a packet can result in a "decoding gap" for the user. One method of alleviating this problem is to allow for the Access Units to be interleaved in the RTP packets. For a modest cost in latency and implementation complexity, significant error resiliency to packet loss can be achieved.

To support optional interleaving of Access Units, this payload format allows for index information to be sent for each Access Unit. After informing receivers about buffer resources to allocate for de-interleaving, the RTP sender is free to choose the interleaving pattern without propagating this information a priori to the receiver(s). Indeed, the sender could dynamically adjust the interleaving pattern based on the Access Unit size, error rates, etc. The RTP receiver does not need to know the interleaving pattern used; it only needs to extract the index information of the Access Unit and insert the Access Unit into the appropriate sequence in the decoding or rendering queue. An example of interleaving is given below.

For example, if we assume that an RTP packet contains 3 AUs, and that the AUs are numbered 0, 1, 2, 3, 4, and so forth, and if an interleaving group length of 9 is chosen, then RTP packet(i) contains the following AU(n):

```
RTP packet(0):  AU(0),  AU(3),  AU(6)
RTP packet(1):  AU(1),  AU(4),  AU(7)
RTP packet(2):  AU(2),  AU(5),  AU(8)
RTP packet(3):  AU(9),  AU(12), AU(15)
RTP packet(4):  AU(10), AU(13), AU(16)  Etc.
```

2.6. Time Stamp Information

The RTP time stamp MUST carry the sampling instant of the first AU (fragment) in the RTP packet. When multiple AUs are carried within an RTP packet, the time stamps of subsequent AUs can be calculated if the frame period of each AU is known. For audio and video, this is possible if the frame rate is constant. However, in some cases it is not possible to make such a calculation (for example, for variable frame rate video, or for MPEG-4 BIFS streams carrying composition information). To support such cases, this payload format can be configured to carry a time stamp in the RTP payload for each contained Access Unit. A time stamp MAY be conveyed in the RTP payload only for non-first AUs in the RTP packet, and SHALL NOT be conveyed for the first AU (fragment), as the time stamp for the first AU in the RTP packet is carried by the RTP time stamp.

MPEG-4 defines two types of time stamps: the composition time stamp (CTS) and the decoding time stamp (DTS). The CTS represents the sampling instant of an AU, and hence the CTS is equivalent to the RTP time stamp. The DTS may be used in MPEG-4 video streams that use bi-directional coding, i.e., when pictures are predicted in both forward and backward direction by using either a reference picture in the past, or a reference picture in the future. The DTS cannot be carried in the RTP header. In some cases, the DTS can be derived from the RTP time stamp using frame rate information; this requires deep parsing in the video stream, which may be considered objectionable. If the video frame rate is variable, the required information may not even be present in the video stream. For both reasons, the capability has been defined to optionally carry the DTS in the RTP payload for each contained Access Unit.

To keep the coding of time stamps efficient, each time stamp contained in the RTP payload is coded as a difference. For the CTS, the offset from the RTP time stamps is provided, and for the DTS, the offset from the CTS.

2.7. State Indication of MPEG-4 System Streams

ISO/IEC 14496-1 defines states for MPEG-4 system streams. So as to convey state information when transporting MPEG-4 system streams, this payload format allows for the optional carriage in the RTP payload of the stream state for each contained Access Unit. Stream states are used to signal "crucial" AUs that carry information whose loss cannot be tolerated and are also useful when repeating AUs according to the carousel mechanism defined in ISO/IEC 14496-1.

2.8. Random Access Indication

Random access to the content of MPEG-4 elementary streams may be possible at some but not all Access Units. To signal Access Units where random access is possible, a random access point flag can optionally be carried in the RTP payload for each contained Access Unit. Carriage of random access points is particularly useful for MPEG-4 system streams in combination with the stream state.

2.9. Carriage of Auxiliary Information

This payload format defines a specific field to carry auxiliary data. The auxiliary data field is preceded by a field that specifies the length of the auxiliary data, so as to facilitate the skipping of data without parsing it. The coding of the auxiliary data is not defined in this document; instead, the format, meaning and signaling of auxiliary information is expected to be specified in one or more future RFCs. Auxiliary information MUST NOT be transmitted until its format, meaning and signaling have been specified and its use has been signaled. Receivers that have knowledge of the auxiliary data MAY decode the auxiliary data, but receivers without knowledge of such data MUST skip the auxiliary data field.

2.10. MIME Format Parameters and Configuring Conditional Fields

To support the features described in the previous sections, several fields are defined for carriage in the RTP payload. However, their use strongly depends on the type of MPEG-4 elementary stream that is carried. Sometimes a specific field is needed with a certain length, while in other cases such a field is not needed. To be efficient in either case, the fields to support these features are configurable by means of MIME format parameters. In general, a MIME format parameter defines the presence and length of the associated field. A length of zero indicates absence of the field. As a consequence, parsing of the payload requires knowledge of MIME format parameters. The MIME format parameters are conveyed to the receiver via SDP [5] messages, as specified in section 4.4.1, or through other means.

2.11. Global Structure of Payload Format

The RTP payload following the RTP header, contains three octet-aligned data sections, of which the first two MAY be empty, see Figure 1.

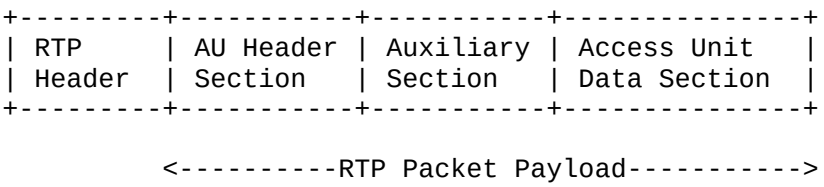

Figure 1: Data sections within an RTP packet

The first data section is the AU (Access Unit) Header Section, that contains one or more AU-headers; however, each AU-header MAY be empty, in which case the entire AU Header Section is empty. The second section is the Auxiliary Section, containing auxiliary data; this section MAY also be configured empty. The third section is the Access Unit Data Section, containing either a single fragment of one Access Unit or one or more complete Access Units. The Access Unit Data Section MUST NOT be empty.

2.12. Modes to Transport MPEG-4 Streams

While it is possible to build fully configurable receivers capable of receiving any MPEG-4 stream, this specification also allows for the design of simplified, but dedicated receivers, that are for example, capable of receiving only one type of MPEG-4 stream. This is achieved by requiring that specific modes be defined in order to use this specification. Each mode may define constraints for transport of one or more types of MPEG-4 streams, for instance on the payload configuration.

The applied mode MUST be signaled. Signaling the mode is particularly important for receivers that are only capable of decoding one or more specific modes. Such receivers need to determine whether the applied mode is supported, so as to avoid problems with processing of payloads that are beyond the capabilities of the receiver.

In this document several modes are defined for the transportation of MPEG-4 CELP and AAC streams, as well as a generic mode that can be used for any MPEG-4 stream. In the future, new RFCs may specify other modes of using this specification. However, each mode MUST be in full compliance with this specification (see section 3.3.7).

2.13. Alignment with RFC 3016

This payload can be configured as nearly identical to the payload format defined in RFC 3016 [12] for the MPEG-4 video configurations recommended in RFC 3016. Hence, receivers that comply with RFC 3016 can decode such RTP payload, provided that additional packets containing video decoder configuration (VO, VOL, VOSH) are inserted in the stream, as required by RFC 3016 [12]. Conversely, receivers that comply with the specification in this document SHOULD be able to decode payloads, names and parameters defined for MPEG-4 video in RFC 3016 [12]. In this respect, it is strongly RECOMMENDED that the implementation provide the ability to ignore "in band" video decoder configuration packets that may be found in streams conforming to the RFC 3016 video payload.

Note the "out of band" availability of the video decoder configuration is optional in RFC 3016 [12]. To achieve maximum interoperability with the RTP payload format defined in this document, applications that use RFC 3016 to transport MPEG-4 video (part 2) are recommended to make the video decoder configuration available as a MIME parameter.

3. Payload Format

3.1. Usage of RTP Header Fields and RTCP

Payload Type (PT): The assignment of an RTP payload type for this packet format is outside the scope of this document; it is specified by the RTP profile under which this payload format is used, or signaled dynamically out-of-band (e.g., using SDP).

Marker (M) bit: The M bit is set to 1 to indicate that the RTP packet payload contains either the final fragment of a fragmented Access Unit or one or more complete Access Units.

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

Sequence Number: The RTP sequence number SHOULD be generated by the sender in the usual manner with a constant random offset.

Timestamp: Indicates the sampling instant of the first AU contained in the RTP payload. This sampling instant is equivalent to the CTS in the MPEG-4 time domain. When using SDP, the clock rate of the RTP time stamp MUST be expressed using the "rtpmap" attribute. If an MPEG-4 audio stream is transported, the rate SHOULD be set to the same value as the sampling rate of the audio stream. If an MPEG-4 video stream is transported, it is RECOMMENDED that the rate be set to 90 kHz.

In all cases, the sender SHALL make sure that RTP time stamps are identical only if the RTP time stamp refers to fragments of the same Access Unit.

According to RFC 3550 [2] (section 5.1), it is RECOMMENDED that RTP time stamps start at a random value for security reasons. This is not an issue for synchronization of multiple RTP streams. However, when streams from multiple sources are to be synchronized (for example one stream from local storage, another from an RTP streaming server), synchronization may become impossible if the receiver only knows the original time stamp relationships. In such cases the time stamp relationship required for obtaining synchronization may be provided by out of band means. The format of such information, as well as methods to convey such information, are beyond the scope of this specification.

SSRC: set as described in RFC 3550 [2].

CC and CSRC fields are used as described in RFC 3550 [2].

RTCP SHOULD be used as defined in RFC 3550 [2]. Note that time stamps in RTCP Sender Reports may be used to synchronize multiple MPEG-4 elementary streams and also to synchronize MPEG-4 streams with non-MPEG-4 streams, in case the delivery of these streams uses RTP.

3.2. RTP Payload Structure

3.2.1. The AU Header Section

When present, the AU Header Section consists of the AU-headers-length field, followed by a number of AU-headers, see Figure 2.

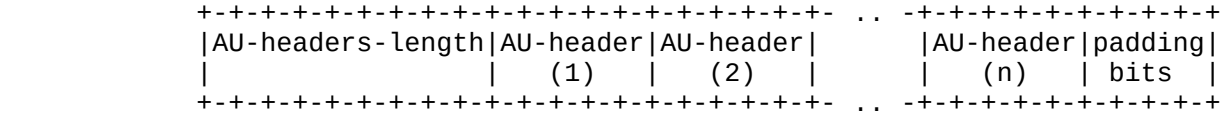

Figure 2: The AU Header Section

The AU-headers are configured using MIME format parameters and MAY be empty. If the AU-header is configured empty, the AU-headers-length field SHALL NOT be present and consequently the AU Header Section is empty. If the AU-header is not configured empty, then the AU-headers-length is a two octet field that specifies the length in bits of the immediately following AU-headers, excluding the padding bits.

Each AU-header is associated with a single Access Unit (fragment) contained in the Access Unit Data Section in the same RTP packet.

For each contained Access Unit (fragment), there is exactly one AU-header. Within the AU Header Section, the AU-headers are bit-wise concatenated in the order in which the Access Units are contained in the Access Unit Data Section. Hence, the n-th AU-header refers to the n-th AU (fragment). If the concatenated AU-headers consume a non-integer number of octets, up to 7 zero-padding bits MUST be inserted at the end in order to achieve octet-alignment of the AU Header Section.

3.2.1.1. The AU-header

Each AU-header may contain the fields given in Figure 3. The length in bits of the fields, with the exception of the CTS-flag, the DTS-flag and the RAP-flag fields, is defined by MIME format parameters; see section 4.1. If a MIME format parameter has the default value of zero, then the associated field is not present. The number of bits for fields that are present and that represent the value of a parameter MUST be chosen large enough to correctly encode the largest value of that parameter during the session.

If present, the fields MUST occur in the mutual order given in Figure 3. In the general case, a receiver can only discover the size of an AU-header by parsing it since the presence of the CTS-delta and DTS-delta fields is signaled by the value of the CTS-flag and DTS-flag, respectively.

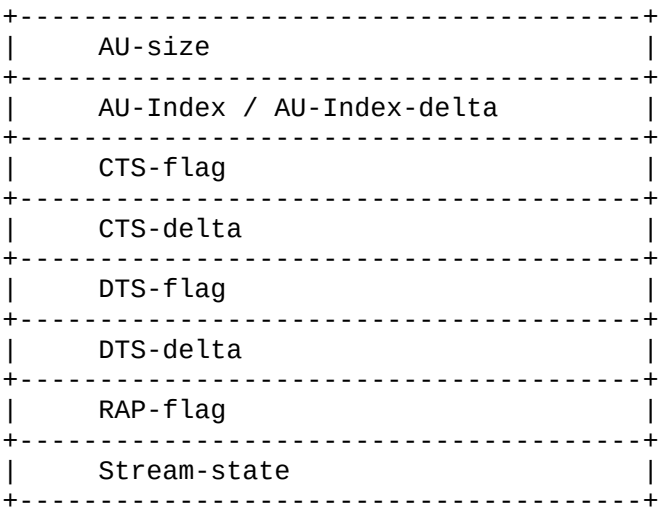

Figure 3: The fields in the AU-header. If used, the AU-Index field only occurs in the first AU-header within an AU Header Section; in any other AU-header, the AU-Index-delta field occurs instead.

AU-size: Indicates the size in octets of the associated Access Unit in the Access Unit Data Section in the same RTP packet. When the AU-size is associated with an AU fragment, the AU size indicates the size of the entire AU and not the size of the fragment. In this case, the size of the fragment is known from the size of the AU data section. This can be exploited to determine whether a packet contains an entire AU or a fragment, which is particularly useful after losing a packet carrying the last fragment of an AU.

AU-Index: Indicates the serial number of the associated Access Unit (fragment). For each (in decoding order) consecutive AU or AU fragment, the serial number is incremented by 1. When present, the AU-Index field occurs in the first AU-header in the AU Header Section, but MUST NOT occur in any subsequent (non-first) AU-header in that Section. To encode the serial number in any such non-first AU-header, the AU-Index-delta field is used.

AU-Index-delta: The AU-Index-delta field is an unsigned integer that specifies the serial number of the associated AU as the difference with respect to the serial number of the previous Access Unit. Hence, for the n-th (n>1) AU, the serial number is found from:

AU-Index(n) = AU-Index(n-1) + AU-Index-delta(n) + 1

If the AU-Index field is present in the first AU-header in the AU Header Section, then the AU-Index-delta field MUST be present in any subsequent (non-first) AU-header. When the AU-Index-delta is coded with the value 0, it indicates that the Access Units are consecutive in decoding order. An AU-Index-delta value larger than 0 signals that interleaving is applied.

CTS-flag: Indicates whether the CTS-delta field is present. A value of 1 indicates that the field is present, a value of 0 indicates that it is not present.

The CTS-flag field MUST be present in each AU-header if the length of the CTS-delta field is signaled to be larger than zero. In that case, the CTS-flag field MUST have the value 0 in the first AU-header and MAY have the value 1 in all non-first AU-headers. The CTS-flag field SHOULD be 0 for any non-first fragment of an Access Unit.

CTS-delta: Encodes the CTS by specifying the value of CTS as a 2's complement offset (delta) from the time stamp in the RTP header of this RTP packet. The CTS MUST use the same clock rate as the time stamp in the RTP header.

DTS-flag: Indicates whether the DTS-delta field is present. A value of 1 indicates that DTS-delta is present, a value of 0 indicates that it is not present.

The DTS-flag field MUST be present in each AU-header if the length of the DTS-delta field is signaled to be larger than zero. The DTS-flag field MUST have the same value for all fragments of an Access Unit.

DTS-delta: Specifies the value of the DTS as a 2's complement offset (delta) from the CTS. The DTS MUST use the same clock rate as the time stamp in the RTP header. The DTS-delta field MUST have the same value for all fragments of an Access Unit.

RAP-flag: When set to 1, indicates that the associated Access Unit provides a random access point to the content of the stream. If an Access Unit is fragmented, the RAP flag, if present, MUST be set to 0 for each non-first fragment of the AU.

Stream-state: Specifies the state of the stream for an AU of an MPEG-4 system stream; each state is identified by a value of a modulo counter. In ISO/IEC 14496-1, MPEG-4 system streams use the AU_SequenceNumber to signal stream states. When the stream state changes, the value of the stream-state MUST be incremented by one.

Note: no relation is required between stream-states of different streams.

3.2.2. The Auxiliary Section

The Auxiliary Section consists of the auxiliary-data-size field followed by the auxiliary-data field. Receivers MAY (but are not required to) parse the auxiliary-data field; to facilitate skipping of the auxiliary-data field by receivers, the auxiliary-data-size field indicates the length in bits of the auxiliary-data. If the concatenation of the auxiliary-data-size and the auxiliary-data fields consume a non-integer number of octets, up to 7 zero padding bits MUST be inserted immediately after the auxiliary data in order to achieve octet-alignment. See Figure 4.

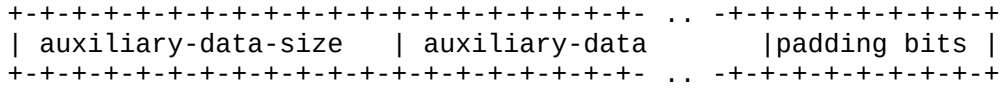

Figure 4: The fields in the Auxiliary Section

The length in bits of the auxiliary-data-size field is configurable by a MIME format parameter; see section 4.1. The default length of zero indicates that the entire Auxiliary Section is absent.

auxiliary-data-size: specifies the length in bits of the immediately following auxiliary-data field;

auxiliary-data: the auxiliary-data field contains data of a format not defined by this specification.

3.2.3. The Access Unit Data Section

The Access Unit Data Section contains an integer number of complete Access Units or a single fragment of one AU. The Access Unit Data Section is never empty. If data of more than one Access Unit is present, then the AUs are concatenated into a contiguous string of octets. See Figure 5. The AUs inside the Access Unit Data Section MUST be in decoding order, though not necessarily contiguous in the case of interleaving.

The size and number of Access Units SHOULD be adjusted such that the resulting RTP packet is not larger than the path MTU. To handle larger packets, this payload format relies on lower layers for fragmentation, which may result in reduced performance.

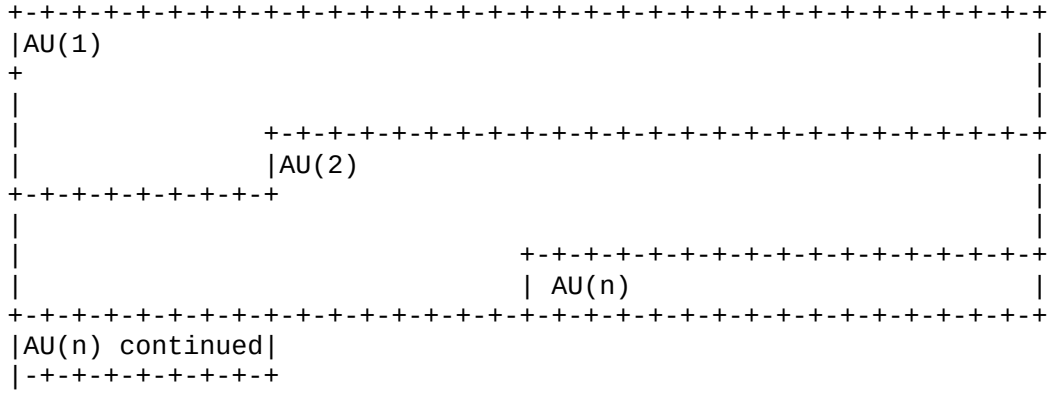

Figure 5: Access Unit Data Section; each AU is octet-aligned.

When multiple Access Units are carried, the size of each AU MUST be made available to the receiver. If the AU size is variable, then the size of each AU MUST be indicated in the AU-size field of the corresponding AU-header. However, if the AU size is constant for a stream, this mechanism SHOULD NOT be used; instead, the fixed size SHOULD be signaled by the MIME format parameter "constantSize"; see section 4.1.

The absence of both AU-size in the AU-header and the constantSize MIME format parameter indicates the carriage of a single AU (fragment), i.e., that a single Access Unit (fragment) is transported in each RTP packet for that stream.

3.2.3.1. Fragmentation

A packet SHALL carry either one or more complete Access Units, or a single fragment of an Access Unit. Fragments of the same Access Unit have the same time stamp but different RTP sequence numbers. The marker bit in the RTP header is 1 on the last fragment of an Access Unit, and 0 on all other fragments.

3.2.3.2. Interleaving

Unless prohibited by the signaled mode, a sender MAY interleave Access Units. Receivers that are capable of receiving modes that support interleaving MUST be able to decode interleaved Access Units.

When a sender interleaves Access Units, it needs to provide sufficient information to enable a receiver to unambiguously reconstruct the original order, even in the case of out-of-order packets, packet loss or duplication. The information that senders need to provide depends on whether or not the Access Units have a constant time duration. Access Units have a constant time duration, if:

TS(i+1) - TS(i) = constant

for any i, where:
i indicates the index of the AU in the original order, and
TS(i) denotes the time stamp of AU(i)

The MIME parameter "constantDuration" SHOULD be used to signal that Access Units have a constant time duration; see section 4.1.

If the "constantDuration" parameter is present, the receiver can reconstruct the original Access Unit timing based solely on the RTP timestamp and AU-Index-delta. Accordingly, when transmitting Access Units of constant duration, the AU-Index, if present, MUST be set to

the value 0. Receivers of constant duration Access Units MUST use the RTP timestamp to determine the index of the first AU in the RTP packet. The AU-Index-delta header and the signaled "constantDuration" are used to reconstruct AU timing.

If the "constantDuration" parameter is not present, then senders MAY signal AUs of constant duration by coding the AU-Index with zero in each RTP packet. In the absence of the constantDuration parameter receivers MUST conclude that the AUs have constant duration if the AU-index is zero in two consecutive RTP packets.

When transmitting Access Units of variable duration, then the "constantDuration" parameter MUST NOT be present, and the transmitter MUST use the AU-Index to encode the index information required for re-ordering, and the receiver MUST use that value to determine the index of each AU in the RTP packet. The number of bits of the AU-Index field MUST be chosen so that valid index information is provided at the applied interleaving scheme, without causing problems due to roll-over of the AU-Index field. In addition, the CTS-delta MUST be coded in the AU header for each non-first AU in the RTP packet, so that receivers can place the AUs correctly in time.

When interleaving is applied, a de-interleave buffer is needed in receivers to put the Access Units in their correct logical consecutive decoding order. This requires the computation of the time stamp for each Access Unit. In case of a constant time duration per Access Unit, the time stamp of the i-th access unit in an RTP packet with RTP time stamp T is calculated as follows:

```
Timestamp[0] = T
Timestamp[i, i > 0] = T +(Sum(for k=1 to i of (AU-Index-delta[k]
                      + 1))) * access-unit-duration
```

When AU-Index-delta is always 0, this reduces to T + i * (access-unit-duration). This is the non-interleaved case, where the frames are consecutive in decoding order. Note that the AU-Index field (present for the first Access Unit) is indeed not needed in this calculation.

3.2.3.3. Constraints for Interleaving

The size of the packets should be suitably chosen to be appropriate to both the path MTU and the capacity of the receiver's de-interleave buffer. The maximum packet size for a session SHOULD be chosen to not exceed the path MTU.

To allow receivers to allocate sufficient resources for de-interleaving, senders MUST provide the information to receivers as specified in this section.

AUs enter the decoder in decoding order. The de-interleave buffer is used to re-order a stream of interleaved AUs back into decoding order. When interleaving is applied, the decoding of "early" AUs has to be postponed until all AUs that precede it in decoding order are present. Therefore, these "early" AUs are stored in the de-interleave buffer. As an example in Figure 6, the interleaving pattern from section 2.5 is considered.

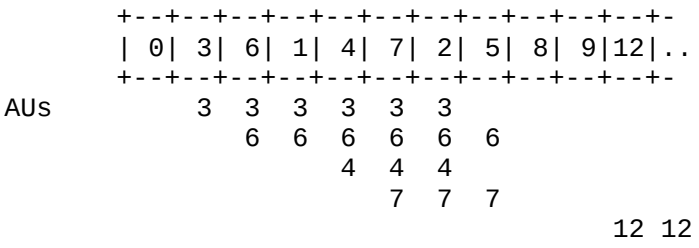

Figure 6: Storage of "early" AUs in the de-interleave buffer per interleaved AU.

AU(3) is to be delivered to the decoder after AU(0), AU(1) and AU(2); of these AUs, AU(2) arrives from the network last and hence AU(3) needs to be stored until AU(2) is present in the pattern. Similarly, AU(6) is to be stored until AU(5) is present, while AU(4) and AU(7) are to be stored until AU(2) and AU(5) are present, respectively. Note that the fullness of the de-interleave buffer varies in time. In Figure 6, the de-interleave buffer contains at most 4, but often less AUs.

So as to give a rough indication of the resources needed in the receiver for de-interleaving, the maximum displacement in time of an AU is defined. For any AU(j) in the pattern, each AU(i) with i<j that is not yet present can be determined. The maximum displacement in time of an AU is the maximum difference between the time stamp of an AU in the pattern and the time stamp of the earliest AU that is not yet present. In other words, when considering a sequence of interleaved AUs, then:

Maximum displacement = max{TS(i) - TS(j)}

for any i and any j>i, where:
i and j indicate the index of the AU in the interleaving pattern, and
TS denotes the time stamp of the AU.

As an example in Figure 7, the interleaving pattern from section 2.5 is considered. For each AU in the pattern, the index is given of the earliest of any earlier AUs not yet present. Hence for each AU(n) in the interleaving pattern the smallest index k (with k<n) of not yet delivered AUs is indicated. A "-" indicates that all previous AUs are present. If the AU period is constant, the maximum displacement equals 5 AU periods, as found for AU(6) and AU(7).

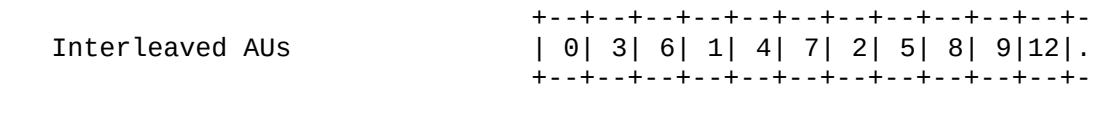

Figure 7: For each AU in the interleaving pattern, the earliest of any earlier AUs not yet present

When interleaving, senders MUST signal the maximum displacement in time during the session via the MIME format parameter "maxDisplacement"; see section 4.1.

An estimate of the size of the de-interleave buffer is found by multiplying the maximum displacement by the maximum bit rate:

size(de-interleave buffer) = {(maxDisplacement) * Rate(max)} / (RTP clock frequency),

where:
Rate(max) is the maximum bit-rate of the transported stream.

Note that receivers can derive Rate(max) from the MIME format parameters streamType, profile-level-id, and config.

However, this calculation estimates the size of the de-interleave buffer and the required size may differ from the calculated value. If this calculation under-estimates the size of the de-interleave buffer, then senders, when interleaving, MUST signal a size of the de-interleave buffer via the MIME format parameter "de-interleaveBufferSize"; see section 4.1. If the calculation over-estimates the size of the de-interleave buffer, then senders, when interleaving, MAY signal a size of the de-interleave buffer via the MIME format parameter "de-interleaveBufferSize".

The signaled size of the de-interleave buffer MUST be large enough to contain all "early" AUs at any point in time during the session. That is:

```
minimum de-interleave buffer size = max [sum {if TS(i) > TS(j) then
                                    AU-size(i) else 0}]

    for any j and any i<j, where:
       i and j indicate the index of an AU in the interleaving
             pattern,
       TS(i) denotes the time stamp of AU(i), and
       AU-size(i) denotes the size of AU(i) in number of octets.
```

If the "de-interleaveBufferSize" parameter is present, then the applied buffer for de-interleaving in a receiver MUST have a size that is at least equal to the signaled size of the de-interleave buffer, else a size that is at least equal to the calculated size of the de-interleave buffer.

No matter what interleaving scheme is used, the scheme must be analyzed to calculate the applicable maxDisplacement value, as well as the required size of the de-interleave buffer. Senders SHOULD signal values that are not larger than the strictly required values; if larger values are signaled, the receiver will buffer excessively.

Note that for low bit-rate material, the applied interleaving may make packets shorter than the MTU size.

3.2.3.4. Crucial and Non-Crucial AUs with MPEG-4 System Data

Some Access Units with MPEG-4 system data, called "crucial" AUs, carry information whose loss cannot be tolerated, either in the presentation or in the decoder. At each crucial AU in an MPEG-4 system stream, the stream state changes. The stream-state MAY remain constant at non-crucial AUs. In ISO/IEC 14496-1, MPEG-4 system streams use the AU_SequenceNumber to signal stream states.

Example: Given three AUs, AU1 = "Insertion of node X", AU2 = "Set position of node X", AU3 = "Set position of node X". AU1 is crucial, since if it is lost, AU2 cannot be executed. However, AU2 is not crucial, since AU3 can be executed even if AU2 is lost.

When a crucial AU is (possibly) lost, the stream is corrupted. For example, when an AU is lost and the stream state has changed at the next received AU, then it is possible that the lost AU was crucial. Once corrupted, the stream remains corrupted until the next random access point. Note that loss of non-crucial AUs does not corrupt the stream. When a decoder starts receiving a stream, the decoder MUST

consider the stream corrupted until an AU is received that provides a random access point.

An AU that provides a random access point, as signaled by the RAP-flag, may or may not be crucial. Non-crucial RAP AUs provide a "repeated" random access point for use by decoders that recently joined the stream or that need to re-start decoding after a stream corruption. Non-crucial RAP AUs MUST include all updates since the last crucial RAP AU.

Upon receiving AUs, decoders are to react as follows:

a) if the RAP-flag is set to 1 and the stream-state changes, then the AU is a crucial RAP AU, and the AU MUST be decoded.

b) if the RAP-flag is set to 1 and the stream state does not change, then the AU is a non-crucial RAP AU, and the receiver SHOULD decode it if the stream is corrupted. Otherwise, the decoder MUST ignore the AU.

c) if the RAP-flag is set to 0, then the AU MUST be decoded, unless the stream is corrupted, in which case the AU MUST be ignored.

3.3. Usage of this Specification

3.3.1. General

Usage of this specification requires definition of a mode. A mode defines how to use this specification, as deemed appropriate. Senders MUST signal the applied mode via the MIME format parameter "mode", as specified in section 4.1. This specification defines a generic mode that can be used for any MPEG-4 stream, as well as specific modes for the transportation of MPEG-4 CELP and MPEG-4 AAC streams, defined in ISO/IEC 14496-3 [1].

When use of this payload format is signaled using SDP [5], an "rtpmap" attribute is part of that signaling. The same requirements apply for the rtpmap attribute in any mode compliant to this specification. The general form of an rtpmap attribute is:

```
a=rtpmap:<payload type> <encoding name>/<clock rate>[/<encoding
          parameters>]
```

For audio streams, <encoding parameters> specifies the number of audio channels: 2 for stereo material (see RFC 2327 [5]) and 1 for mono. Provided no additional parameters are needed, this parameter may be omitted for mono material, hence its default value is 1.

3.3.2. The Generic Mode

The generic mode can be used for any MPEG-4 stream. In this mode, no mode-specific constraints are applied; hence, in the generic mode, the full flexibility of this specification can be exploited. The generic mode is signaled by mode=generic.

An example is given below for the transportation of a BIFS-Anim stream. In this example carriage of multiple BIFS-Anim Access Units is allowed in one RTP packet. The AU-header contains the AU-size field, the CTS-flag and, if the CTS flag is set to 1, the CTS-delta field. The number of bits of the AU-size and the CTS-delta fields are 10 and 16, respectively. The AU-header also contains the RAP-flag and the Stream-state of 4 bits. This results in an AU-header with a total size of two or four octets per BIFS-Anim AU. The RTP time stamp uses a 1 kHz clock. Note that the media type name is video, because the BIFS-Anim stream is part of an audio-visual presentation. For conventions on media type names, see section 4.1.

In detail:

```
m=video 49230 RTP/AVP 96
a=rtpmap:96 mpeg4-generic/1000
a=fmtp:96 streamtype=3; profile-level-id=1807; mode=generic;
objectType=2; config=0842237F24001FB400094002C0; sizeLength=10;
CTSDeltaLength=16; randomAccessIndication=1;
streamStateIndication=4
```

Note: The a=fmtp line has been wrapped to fit the page, it comprises a single line in the SDP file.

The hexadecimal value of the "config" parameter is the BIFSConfiguration() as defined in ISO/IEC 14496-1. The BIFSConfiguration() specifies that the BIFS stream is a BIFS-Anim stream. For the description of MIME parameters, see section 4.1.

3.3.3. Constant Bit-rate CELP

This mode is signaled by mode=CELP-cbr. In this mode, one or more complete CELP frames of fixed size can be transported in one RTP packet; interleaving MUST NOT be used with this mode. The RTP payload consists of one or more concatenated CELP frames, each of equal size. CELP frames MUST NOT be fragmented when using this mode. Both the AU Header Section and the Auxiliary Section MUST be empty.

The MIME format parameter constantSize MUST be provided to specify the length of each CELP frame.

For example:

```
m=audio 49230 RTP/AVP 96
a=rtpmap:96 mpeg4-generic/16000/1
a=fmtp:96 streamtype=5; profile-level-id=14; mode=CELP-cbr; config=
440E00; constantSize=27; constantDuration=240
```

Note: The a=fmtp line has been wrapped to fit the page, it comprises a single line in the SDP file.

The hexadecimal value of the "config" parameter is the AudioSpecificConfig()as defined in ISO/IEC 14496-3. AudioSpecificConfig() specifies a mono CELP stream with a sampling rate of 16 kHz at a fixed bitrate of 14.4 kb/s and 6 sub-frames per CELP frame. For the description of MIME parameters, see section 4.1.

3.3.4. Variable Bit-rate CELP

This mode is signaled by mode=CELP-vbr. With this mode, one or more complete CELP frames of variable size can be transported in one RTP packet with OPTIONAL interleaving. In this mode, the largest possible value for AU-size is greater than the maximum CELP frame size. Because CELP frames are very small, there is no support for fragmentation of CELP frames. Hence, CELP frames MUST NOT be fragmented when using this mode.

In this mode, the RTP payload consists of the AU Header Section, followed by one or more concatenated CELP frames. The Auxiliary Section MUST be empty. For each CELP frame contained in the payload, there MUST be a one octet AU-header in the AU Header Section to provide:

a) the size of each CELP frame in the payload and

b) index information for computing the sequence (and hence timing) of each CELP frame.

Transport of CELP frames requires that the AU-size field be coded with 6 bits. Therefore, in this mode 6 bits are allocated to the AU-size field, and 2 bits to the AU-Index(-delta) field. Each AU-Index field MUST be coded with the value 0. In the AU Header Section, the concatenated AU-headers are preceded by the 16-bit AU-headers-length field, as specified in section 3.2.1.

In addition to the required MIME format parameters, the following parameters MUST be present: sizeLength, indexLength, and indexDeltaLength. CELP frames always have a fixed duration per Access Unit; when interleaving in this mode, this specific duration

MUST be signaled by the MIME format parameter constantDuration. In addition, the parameter maxDisplacement MUST be present when interleaving.

For example:

```
m=audio 49230 RTP/AVP 96
a=rtpmap:96 mpeg4-generic/16000/1
a=fmtp:96 streamtype=5; profile-level-id=14; mode=CELP-vbr; config=
440F20; sizeLength=6; indexLength=2; indexDeltaLength=2;
constantDuration=160; maxDisplacement=5
```

Note: The a=fmtp line has been wrapped to fit the page; it comprises a single line in the SDP file.

The hexadecimal value of the "config" parameter is the AudioSpecificConfig() as defined in ISO/IEC 14496-3. AudioSpecificConfig() specifies a mono CELP stream with a sampling rate of 16 kHz, at a bitrate that varies between 13.9 and 16.2 kb/s and with 4 sub-frames per CELP frame. For the description of MIME parameters, see section 4.1.

3.3.5. Low Bit-rate AAC

This mode is signaled by mode=AAC-lbr. This mode supports the transportation of one or more complete AAC frames of variable size. In this mode, the AAC frames are allowed to be interleaved and hence receivers MUST support de-interleaving. The maximum size of an AAC frame in this mode is 63 octets. AAC frames MUST NOT be fragmented when using this mode. Hence, when using this mode, encoders MUST ensure that the size of each AAC frame is at most 63 octets.

The payload configuration in this mode is the same as in the variable bit-rate CELP mode as defined in 3.3.4. The RTP payload consists of the AU Header Section, followed by concatenated AAC frames. The Auxiliary Section MUST be empty. For each AAC frame contained in the payload, the one octet AU-header MUST provide:

a) the size of each AAC frame in the payload and

b) index information for computing the sequence (and hence timing) of each AAC frame.

In the AU-header Section, the concatenated AU-headers MUST be preceded by the 16-bit AU-headers-length field, as specified in section 3.2.1.

In addition to the required MIME format parameters, the following parameters MUST be present: sizeLength, indexLength, and indexDeltaLength. AAC frames always have a fixed duration per Access Unit; when interleaving in this mode, this specific duration MUST be signaled by the MIME format parameter constantDuration. In addition, the parameter maxDisplacement MUST be present when interleaving.

For example:

```
m=audio 49230 RTP/AVP 96
a=rtpmap:96 mpeg4-generic/22050/1
a=fmtp:96 streamtype=5; profile-level-id=14; mode=AAC-lbr; config=
1388; sizeLength=6; indexLength=2; indexDeltaLength=2;
constantDuration=1024; maxDisplacement=5
```

Note: The a=fmtp line has been wrapped to fit the page; it comprises a single line in the SDP file.

The hexadecimal value of the "config" parameter is the AudioSpecificConfig(), as defined in ISO/IEC 14496-3. AudioSpecificConfig() specifies a mono AAC stream with a sampling rate of 22.05 kHz. For the description of MIME parameters, see section 4.1.

3.3.6. High Bit-rate AAC

This mode is signaled by mode=AAC-hbr. This mode supports the transportation of variable size AAC frames. In one RTP packet, either one or more complete AAC frames are carried, or a single fragment of an AAC frame is carried. In this mode, the AAC frames are allowed to be interleaved and hence receivers MUST support de-interleaving. The maximum size of an AAC frame in this mode is 8191 octets.

In this mode, the RTP payload consists of the AU Header Section, followed by either one AAC frame, several concatenated AAC frames or one fragmented AAC frame. The Auxiliary Section MUST be empty. For each AAC frame contained in the payload, there MUST be an AU-header in the AU Header Section to provide:

a) the size of each AAC frame in the payload and

b) index information for computing the sequence (and hence timing) of each AAC frame.

To code the maximum size of an AAC frame requires 13 bits. Therefore, in this configuration 13 bits are allocated to the AU-size, and 3 bits to the AU-Index(-delta) field. Thus, each AU-header

has a size of 2 octets. Each AU-Index field MUST be coded with the value 0. In the AU Header Section, the concatenated AU-headers MUST be preceded by the 16-bit AU-headers-length field, as specified in section 3.2.1.

In addition to the required MIME format parameters, the following parameters MUST be present: sizeLength, indexLength, and indexDeltaLength. AAC frames always have a fixed duration per Access Unit; when interleaving in this mode, this specific duration MUST be signaled by the MIME format parameter constantDuration. In addition, the parameter maxDisplacement MUST be present when interleaving.

For example:

```
m=audio 49230 RTP/AVP 96
a=rtpmap:96 mpeg4-generic/48000/6
a=fmtp:96 streamtype=5; profile-level-id=16; mode=AAC-hbr;
config=11B0; sizeLength=13; indexLength=3;
indexDeltaLength=3; constantDuration=1024
```

Note: The a=fmtp line has been wrapped to fit the page; it comprises a single line in the SDP file.

The hexadecimal value of the "config" parameter is the AudioSpecificConfig(), as defined in ISO/IEC 14496-3. AudioSpecificConfig() specifies a 5.1 channel AAC stream with a sampling rate of 48 kHz. For the description of MIME parameters, see section 4.1.

3.3.7. Additional Modes

This specification only defines the modes specified in sections 3.3.2 through 3.3.6. Additional modes are expected to be defined in future RFCs. Each additional mode MUST be in full compliance with this specification.

Any new mode MUST be defined such that an implementation including all the features of this specification can decode the payload format corresponding to this new mode. For this reason, a mode MUST NOT specify new default values for MIME parameters. In particular, MIME parameters that configure the RTP payload MUST be present (unless they have the default value), even if its presence is redundant in case the mode assigns a fixed value to a parameter. A mode may additionally define that some MIME parameters are required instead of optional, that some MIME parameters have fixed values (or ranges), and that there are rules restricting its usage.

4. IANA Considerations

This section describes the MIME types and names associated with this payload format. Section 4.1 registers the MIME types, as per RFC 2048 [3].

This format may require additional information about the mapping to be made available to the receiver. This is done using parameters described in the next section.

4.1. MIME Type Registration

MIME media type name: "video" or "audio" or "application"

"video" MUST be used for MPEG-4 Visual streams (ISO/IEC 14496-2) or MPEG-4 Systems streams (ISO/IEC 14496-1) that convey information needed for an audio/visual presentation.

"audio" MUST be used for MPEG-4 Audio streams (ISO/IEC 14496-3) or MPEG-4 Systems streams that convey information needed for an audio only presentation.

"application" MUST be used for MPEG-4 Systems streams (ISO/IEC 14496-1) that serve purposes other than audio/visual presentation, e.g., in some cases when MPEG-J (Java) streams are transmitted.

Depending on the required payload configuration, MIME format parameters may need to be available to the receiver. This is done using the parameters described in the next section. There are required and optional parameters.

Optional parameters are of two types: general parameters and configuration parameters. The configuration parameters are used to configure the fields in the AU Header section and in the auxiliary section. The absence of any configuration parameter is equivalent to the associated field set to its default value, which is always zero. The absence of all configuration parameters results in a default "basic" configuration with an empty AU-header section and an empty auxiliary section in each RTP packet.

MIME subtype name: mpeg4-generic

Required parameters:

MIME format parameters are not case dependent; for clarity however, both upper and lower case are used in the names of the parameters described in this specification.

streamType:
The integer value that indicates the type of MPEG-4 stream that is carried; its coding corresponds to the values of the streamType, as defined in Table 9 (streamType Values) in ISO/IEC 14496-1.

profile-level-id:
A decimal representation of the MPEG-4 Profile Level indication. This parameter MUST be used in the capability exchange or session set-up procedure to indicate the MPEG-4 Profile and Level combination of which the relevant MPEG-4 media codec is capable.

For MPEG-4 Audio streams, this parameter is the decimal value from Table 5 (audioProfileLevelIndication Values) in ISO/IEC 14496-1, indicating which MPEG-4 Audio tool subsets are required to decode the audio stream.

For MPEG-4 Visual streams, this parameter is the decimal value from Table G-1 (FLC table for profile and level indication) of ISO/IEC 14496-2 [1], indicating which MPEG-4 Visual tool subsets are required to decode the visual stream.

For BIFS streams, this parameter is the decimal value obtained from (SPLI + 256*GPLI), where:
SPLI is the decimal value from Table 4 in ISO/IEC 14496-1 with the applied sceneProfileLevelIndication;
GPLI is the decimal value from Table 7 in ISO/IEC 14496-1 with the applied graphicsProfileLevelIndication.

For MPEG-J streams, this parameter is the decimal value from table 13 (MPEGJProfileLevelIndication) in ISO/IEC 14496-1, indicating the profile and level of the MPEG-J stream.

For OD streams, this parameter is the decimal value from table 3 (ODProfileLevelIndication) in ISO/IEC 14496-1, indicating the profile and level of the OD stream.

For IPMP streams, this parameter has either the decimal value 0, indicating an unspecified profile and level, or a value larger than zero, indicating an MPEG-4 IPMP profile and level as defined in a future MPEG-4 specification.

For Clock Reference streams and Object Content Info streams, this parameter has the decimal value zero, indicating that profile and level information is conveyed through the OD framework.

config:
A hexadecimal representation of an octet string that expresses the media payload configuration. Configuration data is mapped onto the hexadecimal 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, if necessary to achieve octet-alignment, up to 7 zero-valued padding bits shall follow the configuration data.

For MPEG-4 Audio streams, config is the audio object type specific decoder configuration data AudioSpecificConfig(), as defined in ISO/IEC 14496-3. For Structured Audio, the AudioSpecificConfig() may be conveyed by other means, not defined by this specification. If the AudioSpecificConfig() is conveyed by other means for Structured Audio, then the config MUST be a quoted empty hexadecimal octet string, as follows: config="".

Note that a future mode of using this RTP payload format for Structured Audio may define such other means.

For MPEG-4 Visual streams, config is the MPEG-4 Visual configuration information as defined in subclause 6.2.1, Start codes of ISO/IEC 14496-2. 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 it exists, which may vary in the repeated configuration information inside an MPEG-4 Visual stream (See 6.2.1 Start codes of ISO/IEC 14496-2).

For BIFS streams, this is the BIFSConfig() information as defined in ISO/IEC 14496-1. Version 1 of BIFSConfig is defined in section 9.3.5.2, and version 2 is defined in section 9.3.5.3. The MIME format parameter objectType signals the version of BIFSConfig.

For IPMP streams, this is either a quoted empty hexadecimal octet string, indicating the absence of any decoder configuration information (config=""), or the IPMPConfiguration() as will be defined in a future MPEG-4 IPMP specification.

For Object Content Info (OCI) streams, this is the OCIDecoderConfiguration() information of the OCI stream, as defined in section 8.4.2.4 in ISO/IEC 14496-1.

For OD streams, Clock Reference streams and MPEG-J streams, this is a quoted empty hexadecimal octet string (config=""), as no information on the decoder configuration is required.

mode:
The mode in which this specification is used. The following modes can be signaled:

mode=generic,
mode=CELP-cbr,
mode=CELP-vbr,
mode=AAC-lbr and
mode=AAC-hbr.

Other modes are expected to be defined in future RFCs. See also section 3.3.7 and 4.2 of RFC 3640.

Optional general parameters:

objectType:
The decimal value from Table 8 in ISO/IEC 14496-1, indicating the value of the objectTypeIndication of the transported stream. For BIFS streams, this parameter MUST be present to signal the version of BIFSConfiguration(). Note that objectTypeIndication may signal a non-MPEG-4 stream and that the RTP payload format defined in this document may not be suitable for carrying a stream that is not defined by MPEG-4. The objectType parameter SHOULD NOT be set to a value that signals a stream that cannot be carried by this payload format.

constantSize:
The constant size in octets of each Access Unit for this stream. The constantSize and the sizeLength parameters MUST NOT be simultaneously present.

constantDuration:
The constant duration of each Access Unit for this stream, measured with the same units as the RTP time stamp.

maxDisplacement:
The decimal representation of the maximum displacement in time of an interleaved AU, as defined in section 3.2.3.3, expressed in units of the RTP time stamp clock.

This parameter MUST be present when interleaving is applied.

de-interleaveBufferSize:
The decimal representation in number of octets of the size of the de-interleave buffer, described in section 3.2.3.3. When interleaving, this parameter MUST be present if the calculation of the de-interleave buffer size given in 3.2.3.3 and based on maxDisplacement and rate(max) under-estimates the size of the de-interleave buffer. If this calculation does not under-estimate the size of the de-interleave buffer, then the de-interleaveBufferSize parameter SHOULD NOT be present.

Optional configuration parameters:

sizeLength:
The number of bits on which the AU-size field is encoded in the AU-header. The sizeLength and the constantSize parameters MUST NOT be simultaneously present.

indexLength:
The number of bits on which the AU-Index is encoded in the first AU-header. The default value of zero indicates the absence of the AU-Index field in each first AU-header.

indexDeltaLength:
The number of bits on which the AU-Index-delta field is encoded in any non-first AU-header. The default value of zero indicates the absence of the AU-Index-delta field in each non-first AU-header.

CTSDeltaLength:
The number of bits on which the CTS-delta field is encoded in the AU-header.

DTSDeltaLength:
The number of bits on which the DTS-delta field is encoded in the AU-header.

randomAccessIndication:
A decimal value of zero or one, indicating whether the RAP-flag is present in the AU-header. The decimal value of one indicates presence of the RAP-flag, the default value zero indicates its absence.

streamStateIndication:
The number of bits on which the Stream-state field is encoded in the AU-header. This parameter MAY be present when transporting MPEG-4 system streams, and SHALL NOT be present for MPEG-4 audio and MPEG-4 video streams.

auxiliaryDataSizeLength:
The number of bits that is used to encode the auxiliary-data-size field.

Applications MAY use more parameters, in addition to those defined above. Each additional parameter MUST be registered with IANA to ensure that there is not a clash of names. Each additional parameter MUST be accompanied by a specification in the form of an RFC, MPEG standard, or other permanent and readily available reference (the "Specification Required" policy defined in RFC 2434 [6]). Receivers MUST tolerate the presence of such additional parameters, but these parameters SHALL NOT impact the decoding of receivers that comply with this specification.

Encoding considerations:
This MIME subtype is defined for RTP transport only. System bitstreams MUST be generated according to MPEG-4 Systems specifications (ISO/IEC 14496-1). Video bitstreams MUST be generated according to MPEG-4 Visual specifications (ISO/IEC 14496-2). Audio bitstreams MUST be generated according to MPEG-4 Audio specifications (ISO/IEC 14496-3). The RTP packets MUST be packetized according to the RTP payload format defined in RFC 3640.

Security considerations:
As defined in section 5 of RFC 3640.

Interoperability considerations:
MPEG-4 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 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 that implement the same combination, and

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

A stream SHALL be compliant with the MPEG-4 Profile@Level specified by the parameter "profile-level-id". Interoperability between a sender and a receiver is achieved by specifying the parameter "profile-level-id" in MIME content. In the capability exchange/announcement procedure, this parameter may mutually be set to the same value.

Published specification:
The specifications for MPEG-4 streams are presented in ISO/IEC 14496-1, 14496-2, and 14496-3. The RTP payload format is described in RFC 3640.

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

Additional information: none

Magic number(s): none

File extension(s):
None. A file format with the extension .mp4 has been defined for MPEG-4 content but is not directly correlated with this MIME type for which the sole purpose is RTP transport.

Macintosh File Type Code(s): none

Person & email address to contact for further information:
Authors of RFC 3640, IETF Audio/Video Transport working group.

Intended usage: COMMON

Author/Change controller:
Authors of RFC 3640, IETF Audio/Video Transport working group.

4.2. Registration of Mode Definitions with IANA

This specification can be used in a number of modes. The mode of operation is signaled using the "mode" MIME parameter, with the initial set of values specified in section 4.1. New modes may be defined at any time, as described in section 3.3.7. These modes MUST be registered with IANA, to ensure that there is not a clash of names.

A new mode registration MUST be accompanied by a specification in the form of an RFC, MPEG standard, or other permanent and readily available reference (the "Specification Required" policy defined in RFC 2434 [6]).

4.3. Concatenation of Parameters

Multiple parameters SHOULD be expressed as a MIME media type string, in the form of a semicolon-separated list of parameter=value pairs (for parameter usage examples see sections 3.3.2 up to 3.3.6).

4.4. Usage of SDP

4.4.1. The a=fmtp Keyword

It is assumed that one typical way to transport the above-described parameters associated with this payload format is via an SDP message [5] for example transported to the client in reply to an RTSP DESCRIBE [8] or via SAP [11]. In that case, the (a=fmtp) keyword MUST be used as described in RFC 2327 [5], section 6, the syntax then being:

a=fmtp:<format> <parameter name>=<value>[; <parameter name>=<value>]

5. Security Considerations

RTP packets using the payload format defined in this specification are subject to the security considerations discussed in the RTP specification [2]. 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 packet processing complexity of this payload type (i.e., excluding media data processing) does not exhibit any significant non-uniformity in the receiver side to cause a denial-of-service threat.

However, it is possible to inject non-compliant MPEG streams (Audio, Video, and Systems) so that the receiver/decoder's buffers are overloaded, which might compromise the functionality of the receiver or even crash it. This is especially true for end-to-end systems like MPEG, where the buffer models are precisely defined.

MPEG-4 Systems support stream types including commands that are executed on the terminal, like OD commands, BIFS commands, etc. and programmatic content like MPEG-J (Java(TM) Byte Code) and MPEG-4 scripts. It is possible to use one or more of the above in a manner non-compliant to MPEG to crash the receiver or make it temporarily unavailable. Senders that transport MPEG-4 content SHOULD ensure that such content is MPEG compliant, as defined in the compliance part of IEC/ISO 14496 [1]. Receivers that support MPEG-4 content should prevent malfunctioning of the receiver in case of non MPEG compliant content.

Authentication mechanisms can be used to validate the sender and the data to prevent security problems due to non-compliant malignant MPEG-4 streams.

In ISO/IEC 14496-1, a security model is defined for MPEG-4 Systems streams carrying MPEG-J access units that comprise Java(TM) classes and objects. MPEG-J defines a set of Java APIs and a secure execution model. MPEG-J content can call this set of APIs and Java(TM) methods from a set of Java packages supported in the receiver within the defined security model. According to this security model, downloaded byte code is forbidden to load libraries, define native methods, start programs, read or write files, or read system properties. Receivers can implement intelligent filters to validate the buffer requirements or parametric (OD, BIFS, etc.) or programmatic (MPEG-J, MPEG-4 scripts) commands in the streams. However, this can increase the complexity significantly.

Implementors of MPEG-4 streaming over RTP who also implement MPEG-4 scripts (subset of ECMAScript) MUST ensure that the action of such scripts is limited solely to the domain of the single presentation in which they reside (thus disallowing session to session communication, access to local resources and storage, etc). Though loading static network-located resources (such as media) into the presentation should be permitted, network access by scripts MUST be restricted to such a (media) download.

6. Acknowledgements

This document evolved into RFC 3640 after several revisions. Thanks to contributions from people in the ISMA forum, the IETF AVT Working Group and the 4-on-IP ad-hoc group within MPEG. The authors wish to thank all people involved, particularly Andrea Basso, Stephen Casner, M. Reha Civanlar, Carsten Herpel, John Lazaro, Zvi Lifshitz, Young-kwon Lim, Alex MacAulay, Bill May, Colin Perkins, Dorairaj V and Stephan Wenger for their valuable comments and support.

APPENDIX: Usage of this Payload Format

Appendix A. Interleave Analysis

A. Examples of Delay Analysis with Interleave

A.1. Introduction

Interleaving issues are discussed in this appendix. Some general notes are provided on de-interleaving and error concealment, while a number of interleaving patterns are examined, in particular for determining the size of the de-interleave buffer and the maximum displacement of access units in time. In these examples, the maximum displacement is cited in terms of an access unit count, for ease of reading. In actual streams, it is signaled in units of the RTP time stamp clock.

A.2. De-interleaving and Error Concealment

This appendix does not describe any details on de-interleaving and error concealment, as the control of the AU decoding and error concealment process has little to do with interleaving. If the next AU to be decoded is present and there is sufficient storage available for the decoded AU, then decode it immediately. If not, wait. When the decoding deadline is reached (i.e., the time when decoding must begin in order to be completed by the time the AU is to be presented), or if the decoder is some hardware that presents a constant delay between initiation of decoding of an AU and presentation of that AU, then decoding must begin at that deadline time.

If the next AU to be decoded is not present when the decoding deadline is reached, then that AU is lost so the receiver must take whatever error concealment measures are deemed appropriate. The play-out delay may need to be adjusted at that point (especially if other AUs have also missed their deadline recently). Or, if it was a momentary delay, and maintaining the latency is important, then the receiver should minimize the glitch and continue processing with the next AU.

A.3. Simple Group Interleave

A.3.1. Introduction

An example of regular interleave is when packets are formed into groups. If the 'stride' of the interleave (the distance between interleaved AUs) is N, packet 0 could contain AU(0), AU(N), AU(2N), and so on; packet 1 could contain AU(1), AU(1+N), AU(1+2N), and so

on. If there are M access units in a packet, then there are M*N access units in the group.

An example with N=M=3 follows; note that this is the same example as given in section 2.5 and that a fixed time duration per Access Unit is assumed:

Packet	Time stamp	Carried AUs	AU-Index, AU-Index-delta
P(0)	T[0]	0, 3, 6	0, 2, 2
P(1)	T[1]	1, 4, 7	0, 2, 2
P(2)	T[2]	2, 5, 8	0, 2, 2
P(3)	T[9]	9,12,15	0, 2, 2

In this example, the AU-Index is present in the first AU-header and coded with the value 0, as required for fixed duration AUs. The position of the first AU of each packet within the group is defined by the RTP time stamp, while the AU-Index-delta field indicates the position of subsequent AUs relative to the first AU in the packet. All AU-Index-delta fields are coded with the value N-1, equal to 2 in this example. Hence the RTP time stamp and the AU-Index-delta are used to reconstruct the original order. See also section 3.2.3.2.

A.3.2. Determining the De-interleave Buffer Size

For the regular pattern as in this example, Figure 6 in section 3.2.3.3 shows that the de-interleave buffer stores at most 4 AUs. A de-interleaveBufferSize value that is at least equal to the total number of octets of any 4 "early" AUs that are stored at the same time may be signaled.

A.3.3. Determining the Maximum Displacement

For the regular pattern as in this example, Figure 7 in section 3.3 shows that the maximum displacement in time equals 5 AU periods. Hence, the minimum maxDisplacement value that must be signaled is 5 AU periods. In case each AU has the same size, this maxDisplacement value over-estimates the de-interleave buffer size with one AU. However, note that in case of variable AU sizes, the total size of any 4 "early" AUs that must be stored at the same time may exceed maxDisplacement times the maximum bitrate, in which case the de-interleaveBufferSize must be signaled.

A.4. More Subtle Group Interleave

A.4.1. Introduction

Another example of forming packets with group interleave is given below. In this example, the packets are formed such that the loss of two subsequent RTP packets does not cause the loss of two subsequent AUs. Note that in this example, the RTP time stamps of packet 3 and packet 4 are earlier than the RTP time stamps of packets 1 and 2, respectively; a fixed time duration per Access Unit is assumed.

Packet	Time stamp	Carried AUs	AU-Index, AU-Index-delta
0	T[0]	0, 5	0, 4
1	T[2]	2, 7	0, 4
2	T[4]	4, 9	0, 4
3	T[1]	1, 6	0, 4
4	T[3]	3, 8	0, 4
5	T[10]	10, 15	0, 4

and so on ..

In this example, the AU-Index is present in the first AU-header and coded with the value 0, as required for AUs with a fixed duration. To reconstruct the original order, the RTP time stamp and the AU-Index-delta (coded with the value 4) are used. See also section 3.2.3.2.

A.4.2. Determining the De-interleave Buffer Size

From Figure 8, it can be to determined that at most 5 "early" AUs are to be stored. If the AUs are of constant size, then this value equals 5 times the AU size. The minimum size of the de-interleave buffer equals the maximum total number of octets of the "early" AUs that are to be stored at the same time. This gives the minimum value of the de-interleaveBufferSize that may be signaled.

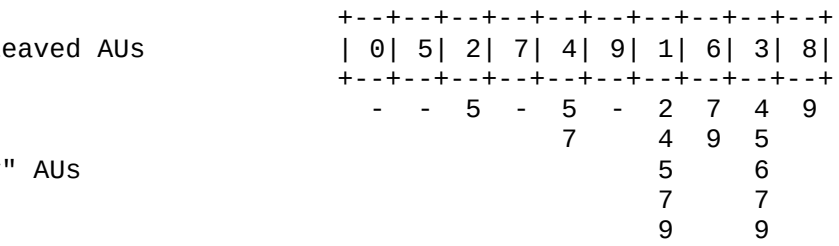

Figure 8: Storage of "early" AUs in the de-interleave buffer per interleaved AU.

A.4.3. Determining the Maximum Displacement

From Figure 9, it can be seen that the maximum displacement in time equals 8 AU periods. Hence the minimum maxDisplacement value to be signaled is 8 AU periods.

Interleaved AUs	0	5	2	7	4	9	1	6	3	8
Earliest not yet present AU	-	1	1	1	1	1	-	3	-	-

Figure 9: For each AU in the interleaving pattern, the earliest of any earlier AUs not yet present

In case each AU has the same size, the found maxDisplacement value over-estimates the de-interleave buffer size with three AUs. However, in case of variable AU sizes, the total size of any 5 "early" AUs stored at the same time may exceed maxDisplacement times the maximum bitrate, in which case de-interleaveBufferSize must be signaled.

A.5. Continuous Interleave

A.5.1. Introduction

In continuous interleave, once the scheme is 'primed', the number of AUs in a packet exceeds the 'stride' (the distance between them). This shortens the buffering needed, smoothes the data-flow, and gives slightly larger packets -- and thus lower overhead -- for the same interleave. For example, here is a continuous interleave also over a stride of 3 AUs, but with 4 AUs per packet, for a run of 20 AUs. This shows both how the scheme 'starts up' and how it finishes. Once again, the example assumes fixed time duration per Access Unit.

Packet	Time-stamp	Carried AUs				AU-Index, AU-Index-delta			
0	T[0]				0	0			
1	T[1]			1	4	0	2		
2	T[2]		2	5	8	0	2	2	
3	T[3]	3	6	9	12	0	2	2	2
4	T[7]	7	10	13	16	0	2	2	2
5	T[11]	11	14	17	20	0	2	2	2
6	T[15]	15	18			0	2		
7	T[19]	19				0			

In this example, the AU-Index is present in the first AU-header and coded with the value 0, as required for AUs with a fixed duration. To reconstruct the original order, the RTP time stamp and the

AU-Index-delta (coded with the value 2) are used. See also 3.2.3.2. Note that this example has RTP time-stamps in increasing order.

A.5.2. Determining the De-interleave Buffer Size

For this example the de-interleave buffer size can be derived from Figure 10. The maximum number of "early" AUs is 3. If the AUs are of constant size, then the de-interleave buffer size equals 3 times the AU size. Compared to the example in A.2, for constant size AUs the de-interleave buffer size is reduced from 4 to 3 times the AU size, while maintaining the same 'stride'.

```
                    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
Interleaved AUs     | 0| 1| 4| 2| 5| 8| 3| 6| 9|12| 7|10|13|16|
                    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
                      -  -  -  4  -  -  4  8  -  -  8 12  -  -
                                        5           9
"Early" AUs                             8          12
```

Figure 10: Storage of "early" AUs in the de-interleave buffer per interleaved AU.

A.5.3. Determining the Maximum Displacement

For this example, the maximum displacement has a value of 5 AU periods. See Figure 11. Compared to the example in A.2, the maximum displacement does not decrease, though in fact less de-interleave buffering is required.

```
                    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
Interleaved AUs     | 0| 1| 4| 2| 5| 8| 3| 6| 9|12| 7|10|13|16|
                    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
Earliest not yet
     present AU       -  -  2  -  3  3  -  -  7  7  -  - 11 11
```

Figure 11: For each AU in the interleaving pattern, the earliest of any earlier AUs not yet present

References

Normative References

[1] ISO/IEC International Standard 14496 (MPEG-4); "Information technology - Coding of audio-visual objects", January 2000

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

[3] Freed, N., Klensin, J. and J. Postel, "Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures", BCP 13, RFC 2048, November 1996.

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

[5] Handley, M. and V. Jacobson, "SDP: Session Description Protocol", RFC 2327, April 1998.

[6] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

Informative References

[7] Hoffman, D., Fernando, G., Goyal, V. and M. Civanlar, "RTP Payload Format for MPEG1/MPEG2 Video", RFC 2250, January 1998.

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

[9] Perkins, C. and O. Hodson, "Options for Repair of Streaming Media", RFC 2354, June 1998.

[10] Schulzrinne, H. and J. Rosenberg, "An RTP Payload Format for Generic Forward Error Correction", RFC 2733, December 1999.

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

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

Authors' Addresses

Jan van der Meer
Philips Electronics
Prof Holstlaan 4
Building WAH-1
5600 JZ Eindhoven
Netherlands

EMail: jan.vandermeer@philips.com

David Mackie
Apple Computer, Inc.
One Infinite Loop, MS:302-3KS
Cupertino CA 95014

EMail: dmackie@apple.com

Viswanathan Swaminathan
Sun Microsystems Inc.
2600 Casey Avenue
Mountain View, CA 94043

EMail: viswanathan.swaminathan@sun.com

David Singer
Apple Computer, Inc.
One Infinite Loop, MS:302-3MT
Cupertino CA 95014

EMail: singer@apple.com

Philippe Gentric
Philips Electronics
51 rue Carnot
92156 Suresnes
France

EMail: philippe.gentric@philips.com

Full Copyright Statement

Acknowledgement

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