Internet Engineering Task Force
INTERNET DRAFT
File: draft-ietf-avt-rtp-mpeg4-01.txt

Civanlar-AT&T/Balabanian-Nortel
Basso-AT&T/Casner-Cisco
Herpel-Thomson/Perkins-UCL

February 25, 1999
Expires: August 25, 1999

RTP Payload Format for MPEG-4 Streams

STATUS OF THIS MEMO

Abstract

This document describes a payload format for transporting MPEG-4 encoded data using RTP. MPEG-4 is a recent standard from ISO/IEC for the coding of natural and synthetic audio-visual data. Several services provided by RTP are beneficial for MPEG-4 encoded data transport over the Internet. Additionally, the use of RTP makes it possible to synchronize MPEG-4 data with other real-time data types.

This specification is a product of the Audio/Video Transport working group within the Internet Engineering Task Force and ISO/IEC MPEG-4 ad hoc group on MPEG-4 over Internet. Comments are solicited and should be addressed to the working group's mailing list at rem-conf@es.net and/or the authors.

1. Introduction

MPEG-4 is a recent standard from ISO/IEC for the coding of natural and synthetic audio-visual data in the form of audiovisual objects that are arranged into an audiovisual scene by means of a scene description [1][2][3][4]. This draft specifies an RTP [5] payload format for transporting MPEG-4 encoded data streams.

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 [6].

The benefits of using RTP for MPEG-4 data stream transport include:

i. Ability to synchronize MPEG-4 streams with other RTP payloads

ii. Monitoring MPEG-4 delivery performance through RTCP

iii. Combining MPEG-4 and other real-time data streams received from multiple end-systems into a set of consolidated streams through RTP mixers

iv. Converting data types, etc. through the use of RTP translators.

1.1 Overview of MPEG-4 End-System Architecture

Fig. 1 below shows the general layered architecture of MPEG-4 terminals. The Compression Layer processes individual audio-visual media streams. The MPEG-4 compression schemes are defined in the ISO/IEC specifications 14496-2 [2] and 14496-3 [3]. The compression schemes in MPEG-4 achieve efficient encoding over a bandwidth ranging from several Kbps to many Mbps. The audio-visual content compressed by this layer is organized into Elementary Streams (ESs). The MPEG-4 standard specifies MPEG-4 compliant streams. Within the constraint of this compliance the compression layer is unaware of a specific delivery technology, but it can be made to react to the characteristics of a particular delivery layer such as the path-MTU or loss characteristics. Also, some compressors can be designed to be delivery specific for implementation efficiency. In such cases the compressor may work in a non-optimal fashion with delivery technologies that are different than the one it is specifically designed to operate with.

The hierarchical relations, location and properties of ESs in a presentation are described by a dynamic set of Object Descriptors (ODs). Each OD groups one or more ES Descriptors referring to a single content item (audio-visual object). Hence, multiple alternative or hierarchical representations of each content item are possible.

ODs are themselves conveyed through one or more ESs. A complete set of ODs can be seen as an MPEG-4 resource or session description at a stream level. The resource description may itself be hierarchical, i.e. an ES conveying an OD may describe other ESs conveying other ODs.

The session description is accompanied by a dynamic scene description, Binary Format for Scene (BIFS), again conveyed through one or more ESs. At this level, content is identified in terms of audio-visual objects. The spatiotemporal location of each object is defined by BIFS. The audio-visual content of those objects that are synthetic and static are described by BIFS also. Natural and animated synthetic objects may refer to an OD that points to one or more ESs that carry the coded representation of the object or its animation data.

By conveying the session (or resource) description as well as the scene (or content composition) description through their own ESs, it is made possible to change portions of the content composition and the number and properties of media streams that carry the audio-visual content separately and dynamically at well known instants in time.

One or more initial Scene Description streams and the corresponding OD streams has to be pointed to by an initial object descriptor (IOD). The IOD needs to be made available to the receivers through some out-of-band means which are not defined in this document.

A homogeneous encapsulation of ESs carrying media or control (ODs, BIFS) data is defined by the Sync Layer (SL) that primarily provides the synchronization between streams. The Compression Layer organizes the ESs in Access Units (AU), the smallest elements that can be attributed individual timestamps. Integer or fractional AUs are then encapsulated in SL packets. All consecutive data from one stream is called an SL-packetized stream at this layer. The interface between the compression layer and the SL is called the Elementary Stream Interface (ESI). The ESI is informative.

The Delivery Layer in MPEG-4 consists of the Delivery Multimedia Integration Framework defined in ISO/IEC 14496-6 [4]. This layer is media unaware but delivery technology aware. It provides transparent access to and delivery of content irrespective of the technologies used. The interface between the SL and DMIF is called the DMIF Application Interface (DAI). It offers content location independent procedures for establishing MPEG-4 sessions and access to transport channels. The specification of this payload format is considered as a part of the MPEG-4 Delivery Layer.

```
media aware        +-----------------------------------------+
delivery unaware   |           COMPRESSION LAYER             |
14496-2 Visual     |streams from as low as Kbps to multi-Mbps|
14496-3 Audio      +-----------------------------------------+  Elementary
                                                                Stream
================================================================Interface
                                                                (ESI)
                  +-------------------------------------------+
media and         |                SYNC LAYER                 |
delivery unaware  | manages elementary streams, their synch-  |
14496-1 Systems   | ronization and hierarchical relations     |
                  +-------------------------------------------+ DMIF
                                                                Application
================================================================Interface
                                                                (DAI)
                  +-------------------------------------------+
delivery aware    |               DELIVERY LAYER              |
media  unaware    |provides transparent access to and delivery|
14496-6 DMIF      | of content irrespective of delivery       |
                  |                 technology                |
                  +-------------------------------------------+
```

Figure 1: General MPEG-4 terminal architecture

1.2 MPEG-4 Elementary Stream Data Packetization

The ESs from the encoders are fed into the SL with indications of AU boundaries, random access points, desired composition time and the current time.

The Sync Layer fragments the ESs into SL packets, each containing a header which encodes information conveyed through the ESI. If the AU is larger than an SL packet, subsequent packets containing remaining parts of the AU are generated with subset headers until the complete AU is packetized.

The syntax of the Sync Layer is not fixed and can be adapted to the needs of the stream to be transported. This includes the possibility to select the presence or absence of individual syntax elements as well as configuration of their length in bits. The configuration for each individual stream is conveyed in an SLConfigDescriptor, which is an integral part of the ES Descriptor for this stream.

2. Analysis of the alternatives for carrying MPEG-4 over IP

2.1 MPEG-4 over UDP

Considering that the MPEG-4 SL defines several transport related functions such as timing, sequence numbering, etc., this seems to be the most straightforward alternative for carrying MPEG-4 data over IP. One group of problems with this approach, however, stems from the monolithic architecture of MPEG-4. No other multimedia data stream (including those carried with RTP) can be synchronized with MPEG-4 data carried directly over UDP. Furthermore, the dynamic scene and session control concepts can't be extended to non-MPEG-4 data.

Even if the coordination with non-MPEG-4 data is overlooked, carrying MPEG-4 data over UDP has the following additional shortcomings:

i. Mechanisms need to be defined to protect sensitive parts of MPEG-4 data. Some of these (like FEC) are already defined for RTP.

ii. There is no defined technique for synchronizing MPEG-4 streams from different servers in the variable delay environment of the Internet.

iii. MPEG-4 streams originating from two servers may collide (their sources may become unresolvable at the destination) in a multicast session.

iv. An MPEG-4 backchannel needs to be defined for quality feedback similar to that provided by RTCP.

v. RTP mixers and translators can't be used.

The backchannel problem may be alleviated by developing a reception reporting protocol like RTCP. Such an effort may benefit from RTCP design knowledge, but needs extensions.

2.2 RTP header followed by full MPEG-4 headers

This alternative may be implemented by using the send time or the composition time coming from the reference clock as the RTP timestamp. This way no new feedback protocol needs to be defined for MPEG-4's backchannel, but RTCP may not be sufficient for MPEG-4's feedback requirements which are still in the definition stage. Additionally, due to the duplication of header information, such as the sequence numbers and time stamps, this alternative causes unnecessary increases in the overhead. Scene description or dynamic session control can't be extended to non-MPEG-4 streams also.

2.3 MPEG-4 ESs over RTP with individual payload types

This is the most suitable alternative for coordination with the existing Internet multimedia transport techniques and does not use MPEG-4 systems at all. Complete implementation of it requires definition of potentially many payload types and might lead to constructing new session and scene description mechanisms. Considering the size of the work involved which essentially reconstructs MPEG-4 systems, this may only be a long term alternative if no other solution can be found.

2.4 RTP header followed by a reduced SL header

The inefficiency of the approach described in 2.2 can be fixed by using a reduced SL header that does not carry duplicate information following the RTP header.

2.5 Recommendation

Based on the above analysis, the best compromise is to map the MPEG-4 SL packets onto RTP packets, such that the common pieces of the headers reside in the RTP header that is followed by an optional reduced SL header providing the MPEG-4 specific information. The details of this payload format are described in the next section.

3. Payload Format

The RTP Payload consists of a single SL packet, including an SL packet header without the sequenceNumber and compositionTimeStamp fields. Use of all other fields in the SL packet headers that the RTP header does not duplicate (including the decodingTimeStamp) is OPTIONAL. Packets SHOULD be sent in the decoding order.

If the resulting, smaller, SL packet header consumes a non-integer number of bytes, zero padding bits MUST be inserted at the end of the SL header to byte-align the SL packet payload.

The size of the SL packets 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 not be desirable.

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

Figure 2 - An RTP packet for MPEG-4

3.1 RTP Header Fields Usage:

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.

Marker (M) bit: Set to one to mark the last fragment (or only fragment) of an AU.

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

Sequence Number: Derived from the sequenceNumber field of the SL packet by adding a constant random offset. If the sequenceNumber has less than 16-bit length, the MSBs MUST initially be filled with a random value that is incremented by one each time the sequenceNumber value of the SL packet returns to zero. If the value sequenceNumber=0 is encountered in multiple consecutive SL packets, indicating a deliberate duplication of the SL packet, the sequence number SHOULD be incremented by one for each of these packets after the first one.

In implementations where full SL packets are generated first and then packetised in RTP, the sequenceNumber MUST be removed from the SL packet header by bit-shifting the subsequent header elements towards the beginning of the header. When unpacking the RTP packet this process can be reversed with the knowledge of the SLConfigDescriptor. For using this payload format, MPEG-4 implementations that do not produce the full SL packet in the first place, but rather produce the RTP header and stripped down (perhaps null) SL header directly are preferable.

However, the choice between generating SL packets and converting, or generating RTP directly is an implementation detail, and does not affect what goes on the wire. Both forms will interwork.

If no sequenceNumber field is configured for this stream (no sequenceNumber field present in the SL packet header), then the RTP packetizer MUST generate its own sequence numbers.

Timestamp: Set to the value in the compositionTimeStamp field of the SL packet, if present. If compositionTimeStamp has less than 32 bits length, the MSBs of timestamp MUST be set to zero.

Although it is available from the SL configuration data, the resolution of the timestamp may need to be conveyed explicitly through some out-of-band means to be used by network elements which are not MPEG-4 aware.

If compositionTimeStamp has more than 32 bits length, this payload format cannot be used.

In case compositionTimeStamp is not present in the current SL packet, but has been present in a previous SL packet, this same value MUST be taken again as the timestamp.

If compositionTimeStamp is never present in SL packets for this stream, the RTP packetizer SHOULD convey a reading of a local clock at the time the RTP packet is created.

Similar to handling of the sequence numbers in implementations that generate full SL packets, the compositionTimeStamp, if present, MUST then be removed from the SL packet header by bit-shifting the subsequent header elements towards the beginning of the SL packet header. When unpacking the RTP packet this process can be reversed with the knowledge of the SLConfigDescriptor and by evaluating the compositionTimeStampFlag.

Timestamps are recommended to start at a random value for security reasons [5, Section 5.1].

SSRC: set as described in RFC1889 [5]. A mapping between the ES identifiers (ESIDs) and SSRCs should be provided through out-of-band means.

CC and CSRC fields are used as described in RFC 1889 [5].

RTCP SHOULD be used as defined in RFC 1889 [5].

4. Multiplexing

Since a typical MPEG-4 session may involve a large number of objects, that may be as many as a few hundred, transporting each ES as an individual RTP session may not always be practical. Allocating and controlling hundreds of destination addresses for each MPEG-4 session may pose insurmountable session administration problems. The input/output processing overhead at the end-points will be extremely high also. Additionally, low delay transmission of low bitrate data streams, e.g. facial animation parameters, results in extremely high header overheads.

To solve these problems, MPEG-4 data transport requires a multiplexing scheme that allows selective bundling of several ESs. This is beyond the scope of the payload format defined here. MPEG-4's Flexmux multiplexing scheme may be used for this purpose by defining an additional RTP payload format for "multiplexed MPEG-4 streams." On the other hand, considering that many other payload types may have similar needs, a better approach may be to develop a generic RTP multiplexing scheme usable for MPEG-4 data. The generic multiplexing scheme reported in [7] is a candidate for this approach.

For MPEG-4 applications, the multiplexing technique needs to address the following requirements:

i. The ESs multiplexed in one stream can change frequently during a session. Consequently, the coding type, individual packet size and temporal relationships between the multiplexed data units must be handled dynamically.

ii. The multiplexing scheme should have a mechanism to determine the ES identifier (ES_ID) for each of the multiplexed packets. ES_ID is not a part of the SL header.

iii. In general, an SL packet does not contain information about its size. The multiplexing scheme should be able to delineate the multiplexed packets whose lengths may vary from a few bytes to close to the path-MTU.

5. Security Considerations

RTP packets using the payload format defined in this specification are subject to the security considerations discussed in the RTP specification [5]. 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.

This payload type does not exhibit any significant non-uniformity in the receiver side computational complexity for packet processing to cause a potential denial-of-service threat.

6. References

[1] ISO/IEC 14496-1 FDIS MPEG-4 Systems November 1998

[2] ISO/IEC 14496-2 FDIS MPEG-4 Visual November 1998

[3] ISO/IEC 14496-3 FDIS MPEG-4 Audio November 1998

[4] ISO/IEC 14496-6 FDIS Delivery Multimedia Integration Framework, November 1998.

[5] Schulzrinne, Casner, Frederick, Jacobson RTP: A Transport Protocol for Real Time Applications RFC 1889, Internet Engineering Task Force, January 1996.

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

[7] M. Handley, "GeRM: Generic RTP Multiplexing," work in progress, draft-ietf-avt-germ-00.txt, November 1998.

7. Authors’ Addresses

M. Reha Civanlar
AT&T Labs - Research
100 Schultz Drive
Red Bank, NJ 07701
USA
e-mail: civanlar@research.att.com

Vahe Balabanian
Nortel
P.O.Box 3511, St. C
Ottawa, Ontario
Canada K1Y 4H7
e-mail: balabani@nortelnetworks.com

Andrea Basso
AT&T Labs - Research
100 Schultz Drive
Red Bank, NJ 07701
USA
e-mail: basso@research.att.com

Stephen L. Casner
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134
USA
e-mail: casner@cisco.com

Carsten Herpel
THOMSON multimedia
Karl-Wiechert-Allee 74
30625 Hannover
Germany
e-mail: herpelc@thmulti.com

Colin Perkins
Department of Computer Science
University College London
Gower Street
London WC1E 6BT
United Kingdom
e-mail: C.Perkins@cs.ucl.ac.uk