A Framework for the delivery of MPEG-4 over IP-based Protocols

Abstract

This document forms an umbrella specification for the carriage and operation of MPEG-4 multimedia sessions over IP-based protocols, including RTP, RTSP, and HTTP, among others. It addresses IP Multicast as well.

It also serves to document the standard MIME types associated with MPEG-4 files.

1 Introduction

MPEG-4 is a standard designed for the representation and delivery of
multimedia information over a variety of transport protocols. It includes interactive scene management, visual and audio representations as well as systems functionality like multiplexing, synchronization, and an object descriptor framework.

This document provides a number of specifications for the detailed mapping of MPEG-4 into several IP-based protocols, as well as references to other specifications.

Open issues: it might be desirable to signal to the terminal the amount of buffering assumed by the encoding/transmission process (in addition to any network jitter).

Editor’s note: the sections that apply to FlexMux have not yet been harmonized with the proposed FlexMux format. Some of the information related to FlexMux (e.g. MIME names, FlexMux structures) should probably be in that draft and removed from here.

Glossary of terms and acronyms

AAC - MPEG-4 advanced audio codec
AU - access unit in an ES (the smallest media data unit to which timing can be attributed).
BIFS - binary format for scenes; the MPEG-4 scene composition system
CELP - MPEG-4 speech codec
CTS - composition time stamp
DTS - decoding time stamp
ES - elementary stream
ESID - elementary stream ID
FCR - flexmux clock reference
FlexMux - a multiplex of several PDUs into a single unit; not used for multiplexing in RTP
IOD - initial object descriptor; the ‘hook’ to the MPEG-4 streams needed to start a session
OCR - object clock reference; an external clock reference for an MPEG-4 stream
OD - object descriptor; declares and defines an MPEG-4 stream
SL - synchronization layer
SL Packet - synchronization layer protocol data unit, in MPEG-4 systems

2 Use of RTP

There are a number of Internet Drafts describing RTP packetization schemes for MPEG-4 data [5] [6] [7] [8] [9]. This draft does not specify any new one. Media-aware packetization (e.g. video frames split at recoverable sub-frame boundaries) is a principle in RTP, and thus it is likely that several RTP schemes will be needed, to suit
both the different kinds of media - audio, video, etc. - and different encodings (e.g. AAC and CELP audio codecs) [11].

This specification requires that, no matter what packetization scheme is used, there are a number of common characteristics that all MUST have: however, such characteristics depend on the fact that the RTP Session contains a single elementary stream or a flexmux stream.

In case an RTP Session contains a single elementary stream the following characteristics apply:

2.1] The RTP timestamp corresponds to the presentation time (e.g. CTS) of the earliest AU within the packet.

2.2] RTP packets have sequence numbers in transmission order. The payloads logically or physically have SL Sequence numbers, which are in decoding order, for each elementary stream.

2.3] The MPEG-4 timescale (clock ticks per second), which is timeStamRes in the case of MPEG-4 Systems, MUST be used as the RTP timescale, e.g. as declared in SDP for an RTP stream.

2.4] To achieve a base level of interoperability, and to ensure that any MPEG-4 stream may be carried, all senders and receivers MUST implement a default RTP payload mapping scheme. It is highly desirable that this default scheme is common for both pure Audio and Visual streams as well as for SL Packetized streams. This default scheme is not yet identified.

2.5] Streams SHOULD be synchronized using RTP techniques (notable RTCP sender reports). When the MPEG-4 OCR is used, it is logically mapped to the NTP time axis used in RTCP.

2.6] The RTP packetization schemes may be used for MPEG-4 elementary streams ‘standing alone’ (e.g. without MPEG-4 systems, including BIFS); or they may be used within an overall presentation using the object descriptor framework. In the latter case, an SLConfigDescriptor is sent describing the stream. Logically, each RTP stream is passed through a mapping function which is specific to the payload format used; this mapping function yields an SL packetized stream. The SLConfigDescriptor describes this logical stream, not the actual bits in the RTP payload. For example, the RTP sequence number may be used to make the SLPacketHeader sequence number; other SL fields may be set in this way, dynamically, or from static values in the payload specification. For example, as all RTP packets carry a composition time-stamp, the flag in the SL header indicating its presence can normally be statically defined as ‘true’. Each payload format for MPEG-4 content MUST specify the mapping
function for the formation of the SLConfigDescriptor and the
SLPacketHeader.

In the case of the draft by Kikuchi-san et al., the mapping will be
defined in a new section.

In case an RTP Session contains a flexmultiplexed stream the
following characteristics apply:

2.6] There is a single payload format for the carriage of Flexmux
Streams over RTP [5]. Senders and receivers MAY implement this
scheme.

2.7] The RTP timestamp corresponds to the FCR if present at the
Flexmux level.

2.8] The MPEG-4 Flexmux timescale (FCR resolution in ticks per
second) SHOULD be used as the RTP timescale (as can be declared in
SDP).

2.9] the MPEG-4 FCR is logically mapped to the NTP time axis used in
RTCP.

Other payload formats MAY be used. They are signalled as dynamic
payload IDs, defined by a suitable name (e.g. a payload name in an
SDP RTPMAP attribute). In particular, the development of specialized
RTP payloads for video (e.g. respecting video packets) and audio
(e.g. providing interleave) is expected. It is possible that these
schemes can be compatible with the default scheme required here.

There may be a choice of RTP payload formats for a given stream (e.g.
as an elementary stream, an SL-packetized stream, using FlexMux, and
so on). It is recommended that
* terminals implementing a given sub-system (e.g. video) accept at
  least an ES and the default SL packings [8] of that stream, if
  they exist; for example, this means accepting the draft by
  Kikuchi et al. and also the SL draft by Civanlar et al. for
  MPEG-4 video;
* terminals implementing a given payload format accept any stream
  over that format for which they have a decoder, even if that
  packing is not normally the ‘best’ packing.

Future versions of this specification will identify the single
standard RTP packing format for each MPEG-4 stream type. However, at
the time of writing the RTP payload format specifications are still
being defined, and the set is incomplete. These recommendations will
form the basis for improved interoperability.
For those streams requiring a certain Quality of Service (specifiable appropriately), the recommendation is to further investigate possible solutions such as the leverage of existing work in the IETF in this area (including, but not limited to FEC, re-transmission, or repetition). However, techniques in data-dependent error correction, or combined source/channel coding solutions make other schemes attractive [7]. Also, it is recommended that requirement such as efficient grouping mechanisms (i.e. the ability to send in a single RTP packet multiple consecutive Aus, each with its own SL information) and low overhead are also taken into account.

3 SDP Information

This specification considers only MPEG-4 Systems related issues. The usage of elementary streams in other contexts is not addressed here: codepoints for this case are specified in [6], and in other places.

This specification currently assumes that any session described by SDP (e.g. in SAP, as a file download, as a DESCRIBE over RTSP) has at most one MPEG-4 session. It is desirable that this restriction be lifted.

3.1] Senders SHOULD alert receivers that an MPEG-4 session is included, by means of an SDP attribute that is general (i.e. before any "media" lines). This takes the form of an attribute line:

a=mpeg4-iod [<location>]

location: In an RTSP session, this is an optional attribute. If not supplied, the IOD is retrieved over the RTSP session by using DESCRIBE with an accept of type application/mpeg4-iod. Where the SDP information is supplied by some other means (e.g. as a file, in SAP), the location is obligatory. The location should be a URL enclosed in double-quotes, which will supply the IOD (e.g. small ones may be encoded using "data:", otherwise "http:" or other suitable file-access URL). The InitialObjectDescriptor is defined in sub-clause 8.6.3.1 of ISO/IEC 14496-1.

3.3] New encoding names for the a = rtpmap attribute It is recommended that, no matter what payload format is used, each media stream be placed in a media section that is appropriate. For example, a payload format which can carry both video and audio streams may be used in sections of SDP starting both with "m=video" and "m=audio". The MIME name for the payload format is thus registered under all applicable branches.
a = rtpmap:<payload> <name>/<time scale>/<parameters>

payload is the dynamic payload number
The <name> is defined and documented in the IETF specification for
the payload format; for example, mpeg4-SL might indicate the
encoding type of the media, one MPEG-4 SL packetized stream, or
mpeg4-flexmux might indicate the encoding type of the media, one
MPEG-4 FlexMux stream.

time scale is the time scale of the RTP time stamps
parameters if used, is defined in the RTP payload format

3.3] The mapping of RTP streams to elementary streams needs to cover
the Flexmux case as well as the single stream. Within the SDP
information, a stream-specific attribute SHOULD be present for each
MPEG-4 stream. It takes one of two forms, depending on whether a
single elementary stream, or a flexmux, is carried.

3.4] In case of a single elementary stream, the following attribute
is defined:

a=mpeg4-esid a

a is the ESID.

3.5] In case of a flexmux stream, the following attribute is defined:

a=mpeg4-esids m1:a, m2:b ...

where m1, m2 are flexmux channels and a, b are ESids

3.5] In case of a flexmux stream, the following attribute is defined:

a = mpeg4-flexmuxinfo: <location>
a = mpeg4-muxcodetable: <location>

The first form is used to define both the ES mapping and the
muxcodetable, the second the muxcodetable only. The mapping of ESs
to streams and the formatting of the muxcodetable needs to be
harmonized with the draft on FlexMux.

<location> is a URL enclosed in double quotes, that will supply the
required flexmux list of descriptors. If they are small, a DATA: URL
will probably suffice to carry them in-line. If not, the URL should
use a file-retrieval scheme (e.g. HTTP, FTP). The data at the
indicated URL consists of some number of concatenated descriptors,
complete, in binary format (but note that DATA URLs allow for base64
encoding of binary data, which would be needed here). These
descriptors have an intrinsic length, so simple concatenation suffices. The MPEG-4 descriptors related to FlexMux description can be MPEG-4 FlexMuxChannel, MPEG-4 MuxCode, MPEG-4 MultiplexBuffer. The MPEG-4 Muxcodetable is defined in MPEG-4 systems.

The list of MPEG-4 descriptors cannot be empty. Private descriptors can complete it. The MIME name used for this data is defined below.

3.6] Other SDP attributes should, if used, carry values consistent with those carried in MPEG-4 systems (for example, bit rate).

4 MIME Types

4.1] The historical approach for MPEG data is to declare it under "video", and this approach is followed for MPEG-4. For presentations with audio information and no visual aspect, the "audio" top-level mime type may be used; otherwise, "video" is used.

4.2] Amendment 1 of the MPEG-4 standard (also known as version 2) includes a standard file type for encapsulating MPEG-4 data. This file type can be used in a number of ways: perhaps the most important are its use as an interchange format for MPEG-4 data, its use as a content-download format, and as the format read by streaming media servers.

These first two uses will be greatly facilitated if there is a standard MIME type for serving these files (e.g. over HTTP).

The MPEG-4 standard is broad, and therefore the type of data that may be in such a file can vary. In brief, simple compressed video and audio (using a number of different compression algorithms) can be included; interactive scene information; meta-data about the presentation; references to MPEG-4 media streams outside the file and so on.

The MIME types to be assigned to MP4 files SHOULD be "audio/mp4", and "video/mp4", based on the criteria in 4.1. In either case, these indicate files conforming to the "MP4" specification (ISO/IEC 14496-1:2000, systems file format).

4.3] When an MP4 file is served (e.g. over HTTP) or otherwise must be identified by a MIME type, the type "video/mp4" SHOULD be used. The types "audio/mp4" MAY be used when the MPEG-4 presentation contained within the MP4 file has no visual presentation and refers to a pure audio presentation.
4.4] When a visual MPEG-4 ES is served (e.g. over HTTP or otherwise) and must be identified by a MIME type, the type "video/MPEG4-visual" SHALL be used. This MIME type may require optional parameters to carry all necessary information to configure a receiver: therefore no further meta-information (such as that defined by the MP4 file format or by the MPEG-4 Object Descriptor framework) has to be provided in the data, and the data itself merely represents the media content. The format of the bit-stream, including timing etc., is defined in ISO/IEC 14496-2.

4.5] In some cases, the initial object descriptor needs to be identified with a MIME type. In this case, the type "application/mpeg4-iod" SHALL be used.

4.6] When a flexmux stream is served (e.g. over HTTP) or otherwise must be identified by a MIME type, the type "application/mpeg4-flexmux" SHALL be used. These files consist of concatenated flexmux PDUs in transmission order.

4.7] In some cases, the information needed by a flexmux decoder needs to be identified with a MIME type. In this case, the type "application/mpeg4-flexmuxinfo" SHOULD be used.

4.8] The payload names used in an RTPMAP attribute within SDP, to specify the mapping of payload number to its definition, also come from the MIME namespace. Each of the RTP payload mappings defined above has a distinct name. It is recommended that visual streams be identified under "video", and audio streams be identified under "audio", and otherwise "application" be used.

Given the broad and general nature of MPEG-4, and the interactive environment, it is hard to say that there are no security considerations. However, none are known to the author at this time, and the standard was developed with the intent that there be none.

<table>
<thead>
<tr>
<th>MIME media type name:</th>
<th>video, and audio</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIME subtype name:</td>
<td>mp4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MIME media type name:</th>
<th>application</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIME subtype name:</td>
<td>mpeg4-iod, mpeg4-flexmux, mpeg4-flexmuxinfo</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Required parameters:</th>
<th>none</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional parameters:</td>
<td>none</td>
</tr>
<tr>
<td>Encoding considerations:</td>
<td>base64 generally preferred; files are binary and should be transmitted without CR/LF conversion, 7-bit</td>
</tr>
</tbody>
</table>
Security considerations: None known at the time of writing
Interoperability considerations: A number of interoperating implementations exist within the MPEG-4 community; and that community has reference software for reading and writing the file format.
Applications: Multimedia
Additional information:
Magic number(s): none
File extension(s): mp4 and mpg4 are both declared at <http://pitch.nist.gov/nics/>
Macintosh File Type Code(s): mpg4 is registered with Apple
Person to contact for info: David Singer, singer@apple.com
Intended usage: Common
Author/Change controller: David Singer, MPEG-4 file format chair

5 RTSP usage

This specification considers only MPEG-4 Systems related issues. The usage of elementary audio or visual streams in other context does not require any specific statement about RTSP.

RTSP may be used as a session control protocol for sessions which carry MPEG-4 information. When RTSP is used as a session-control protocol:

5.1] RTP SHOULD be used as the transport protocol.

5.2] The initial DESCRIBE format SHOULD be SDP. If the SDP information reveals that an IOD is needed, and the terminal does not already have it, then a second DESCRIBE accepting an IOD SHOULD be performed (see above).

5.3] Note that if all MPEG-4 streams are closed (TEARDOWN) then the RTSP session ID will be lost. The next (re-)opened stream will supply a new session ID. Care should be taken that the target of the URL has not changed in the interval; new DESCRIBEs may be needed.

6 Multicast
This specification considers only MPEG-4 Systems related issues.

When using IP Multicast, the SDP information describing the MPEG-4 Session SHOULD be made available to the terminal.

In addition, elementary stream descriptors may use URLs to directly address ESs. The goal of such URL would be to convey information to enable the terminal to directly connect to the RTP channel carrying the ES. No matter what URL scheme is used ("rtp:" ....) information shall be conveyed for the information which would otherwise be needed from SDP, including but not limited to
- IP Multicast address
- Port number
- Any such attributes above as may be needed.

For these reasons, it is recommended that any multicast session be described by SDP. The default protocol stack SHALL be used, or more parameters are required to identify the protocol stack.

Acknowledgments

This draft has benefited greatly by contributions from many people, including Mike Coleman, Jean-Claude Duford, Peter Westerink, Carsten Herpel, Olivier Avaro, Paul Christ, Zvi Lifshitz, and many others. Their insight, foresight, and contribution is gratefully acknowledged. Little has been invented here by the author; this is mostly a collation of greatness that has gone before.
References


Authors’ Contact Information
David Singer
Email: singer@apple.com
Tel: +1 408 974 3162

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

Young-Kwon LIM
E-mail: young@techway.co.kr
TEL: +82-42-863-7800

mp4cast
(MPEG-4 Internet Broadcasting Solution Consortium)
1001-1 Daechi-Dong Gangnam-Gu
Seoul, 305-333, Korea