RTP Payload Format for MPEG-4 FlexMultiplexed Streams

1. Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as 'work in progress.'

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

2. Abstract

This document describes a payload format for transporting MPEG-4 encoded and multiplexed 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 and multiplexed 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.

RTP PAYLOAD FORMAT FOR MPEG-4 FLEXMULTIPLEXED STREAMS
1. STATUS OF THIS MEMO 1
2. ABSTRACT 1
3. INTRODUCTION TO THE MPEG-4 STANDARD 3
4. OVERVIEW OF MPEG-4 END-SYSTEM ARCHITECTURE 4
5. BENEFITS OF USING RTP FOR TRANSPORT:5
3. Introduction to the MPEG-4 standard

The MPEG-4 standard (ISO/IEC 14496) can be represented in a layered architecture, where three layers can be identified as follows:

- **Compression Layer**: Elementary Streams (ES) encoding
  - MPEG-4 part 2 Visual
  - MPEG-4 part 3 Audio
  - MPEG-4 part 1 Bifs,OD,IPMP,OCI

- **Sync Layer (SL)**: Elementary streams management and synchronisation

- **Delivery Layer (DMIF)**: Provides Flexmultiplexing of SL streams and transparent access to the delivery technology

Although the Delivery Layer mostly focuses on the control plane, it also encompasses multiplexing tools and defines a bitstream syntax, called the Flexmux, to multiplex MPEG-4 SL streams. The reconstruction of the correct timing of MPEG-4 bitstreams is supported both by the MPEG-4 SL stream syntax and by the MPEG-4 FlexMux stream syntax. The reconstruction of the correct timing
of an MPEG-4 Flexmux stream is possible under various QoS constraints related to the reduction of network jitter. MPEG makes the assumption that all Flexmux packets transmitted over the network should have a nearly constant transmission delay. The reconstruction of the correct timing of the MPEG-4 Flexmux streams is based on the fact that this assumption can be verified, in accordance with the required accuracy of the reconstructed timing of the MPEG-4 FlexMux stream. This document will specify a RTP payload format to enable the carriage of Flexmux streams.

4. Overview of MPEG-4 End-System Architecture

The Compression Layer processes the traditional individual audio/visual elementary streams (ES) and some associated 'systems' elementary streams (ES) such as Bifs, OD, IPMP and OCI elementary streams. The MPEG-4 audio/visual ES syntaxes are defined in [2] and [3]. The 'systems' ES syntaxes are described in [1]: the Bifs ES syntax allows a dynamic scene description. The OD ES syntax allows the description of the hierarchical relations, location and properties of different ESs through a dynamic set of Object Descriptors. The 'system' ES may require to be carried with a better protection than the traditional audio/visual ESs. The compressed content produced by this layer are organised into Elementary Streams (ESs). The compression layer is unaware of a specific delivery technology, but it can react to the characteristics of a particular delivery layer such as the path-MTU or packet loss or bit error characteristics. The MPEG-4 SL stream syntax is defined in [1]. It provides a unique and homogeneous encapsulation of any of the MPEG-4 ESs. That layer primarily provides the synchronisation between ESs. ESs are organised in Access Units (AU). An AU is the smallest element to which timestamps can be assigned. Integer or fractional AUs are then encapsulated in SL packets. The MPEG-4 Delivery Layer consists of the Delivery Multimedia Integration Framework defined in [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 DAI interface between the SL layer and the DMIF layer is called the DMIF Application Interface. This interface supports content location independent protocols firstly for establishing the MPEG-4 session and secondly for accessing to transport channels. DMIF monitors transport channels on the QoS requirements assigned to the SL streams, and supports the multiplexing of the SL streams, by the means
5. Benefits of using RTP for transport:

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

6. Conventions used in this document

6.1 general:

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].

6.2 MPEG-4 glossary:

AU : Access Unit
Bifs: Binary format for scene
DMIF: Delivery Multimedia Integration Framework,
DAI: DMIF Application Interface, ES: Elementary stream,
IPMP: Intellectual Property Management and Protection,
OCI: Object Content Information, OD: Object descriptor,
SL: Synchronization layer

C.Roux & al. expires 14/08/01 5

RTP payload format for MPEG-4 FlexMultiplexed streams 23/02/01

7. The RTP packet

7.1 The RTP packet header

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 2
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|X|   CC  |M|      PT     |     sequence number           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          timestamp                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         synchronization source (SSRC) identifier              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        contributing source (CSRC) identifiers                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                                                               |
|                    RTP Packet Payload                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

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

7.2 RTP header fields usage streams:
Payload Type (PT): The assignment of a particular RTP payload type to this new packet format, is outside the scope of this document, and is not specified here. If the dynamic payload type assignment is used, it can be specified by some out of band means (e.g. SDP, according to the syntax proposed in the paragraph 9) that the MPEG-4 FlexMux payload format is used for the corresponding RTP packets. The specification of the usage of MPEG-4 FlexMux payload format can also include, if needed, the specification of the usage of the MPEG-4 FlexMux signaling format.

Marker (M) bit: set to zero.

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

Sequence Number: Increment by one for each RTP data packet sent. It starts with a random initial value for security reasons.

Timestamp: it represents the target transmission time for the first byte of the RTP packet. Unless specified by an out of band means (e.g. SDP), the resolution of the timestamp is set to its default (90KHz).

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

7.3 The two RTP packet payloads

The MPEG-4 FlexMux payload format encompasses two payload types:
the first one for MPEG-4 FlexMux packets and the second one for MPEG-4 FlexMux signaling.

When the Payload type specifies the usage of the MPEG-4 FlexMux payload format, the RTP packet payload is built from an integer number of complete, defined in [1].

When the Payload type specifies the usage of the MPEG-4 FlexMux signaling payload format, the RTP packet payload is built from an integer number of descriptors, defined in the following 7.4. paragraph.

7.3.1 payload for the MPEG-4 FlexMux payload format

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         FlexMux Packet Header |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                     |
|            FlexMux Packet Payload (byte aligned)              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    FlexMux Packet Header      |     FlexMux                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
|                        Packet                                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  
|             Payload           :...optional RTP padding        |
```
7.3.2 payload for the MPEG-4 FlexMux signaling payload format

They are used to describe in-band a FlexMux stream characteristics. Their use is mostly static. Only one of these descriptors (the FlexMux codetable entry descriptor) may have a dynamic use, when the FlexMux stream characteristic is to be dynamic.

They are built from a signaling descriptor header followed by a signaling descriptor content.

The signaling descriptor header structure is identical for all the signaling descriptors.

Their overall length is an integer number of bytes.

7.4 MPEG-4 FlexMux signaling descriptors

7.4.1 signaling descriptor usage:

The signaling descriptor header is built from a tag field on one byte, followed by a length field on one byte.
0x60 for FlexMux Channel Table descriptor
0x61 for FlexMux buffersize descriptor
0x62 for FlexMux timing descriptor
0x63 for FlexMux codetable entry descriptor
0x64 for FlexMux declaration descriptor

values from 0x00 to 0x5F, & value 0xFF are forbidden
values from 0x65 to 0xBF are ISO reserved
values from 0xC0 to 0xDF are IETF reserved
values from 0xE0 to 0xFE are Ad Hoc values

LENGTH - is the byte length of the descriptor’s content that follows.

7.4.2 FlexMux declaration descriptor

This descriptor is used to allow identification of some different FlexMux streams within an MPEG-4 scene. This can be the case when scalable coding is used.

When no FlexMux descriptor is used, the default TYPE is the first FlexMux tool, and the default MODE is static.

0 1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|      MUXID    |   TYPE|  MODE |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
Figure 7 - declaration descriptor content

MUXID - is the identifier of the FlexMux stream

TYPE - is the type of the Multiplexing tool used to generate the FlexMux stream. As they are two FlexMux tools defined, indicated type values shall be either 0 (for the first FlexMux tool), 1 (for the second FlexMux tool).

7.4.3 FlexMux timing descriptor

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| FCR-ES-ID | FCRRESOLUTION |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| FCRRESOLUTION (end) | FCRLENGTH | FMXRATELENGTH |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
Figure 8 - timing descriptor content

FCR-ES-ID (16 bits) is the ES_ID associated to this clock reference stream.

FCRRESOLUTION (32 bits) is the resolution of the object time base in cycles per second.
FCRLLENGTH - (8 bits) is the length of the fmxClockReference field in FlexMux packets with index = 238. A length of zero shall indicate that no FlexMux packets with index = 238 are present in this FlexMux stream. FCRLNGTH shall take values between zero and 64.

FMXRATELENGTH Å (8 bits) is the length of the fmxRate field in FlexMux packets with index = 238. FMXRATELENGTH shall take values between 1 and 32.

7.4.4 FlexMux Channel Table descriptor

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[Diagram of FlexMuxChannel Table descriptor content]

C. Roux & al. expires 14/08/01

RTP payload format for MPEG-4 FlexMultiplexed streams 23/02/01

<ES1>, .., <ESn> - these 16-bit fields specify the identifiers of ISO/IEC 14496-1 SL-packetized streams defined in [1].

<M1>, .., <M2>, .. This 8-bit fields specify the number of the FlexMux channels used for these SL-packetized streams.

7.4.5 FlexMux codetable entry descriptor

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[Diagram of FlexMuxcodetable table descriptor content]

MUXCODE - the number through which this FlexMux codetable entry is referenced

VERSION - the version of the FlexMux codetable entry. Only the latest received version of a FlexMux codetable entry is valid.

SUBSTRUCTCNT is the number of substructures of this FlexMux codetable entry.

SLOTCNT - the number of slots with data from different FlexMux channels that are described by this substructure

RCNT - indicates how often this substructure is to be repeated.
A zero indicates that this substructure is to be repeated infinitely. Zero is only permitted in the last substructure of a FlexMux codetable entry.

M1,..,Mi - the FlexMux channels to which the data in this slot belongs.

NB1,..,NBi - the number of data bytes in this slot associated to the FlexMux channel M1, Mi. This number of bytes corresponds to one SL packet. SL packets are defined in [1].

Figure 11 - FlexMux buffersize descriptor content

DEFAULTSIZE - the default size of multiplex buffers for each individual channel in a FlexMux stream. FlexMux channels that use a different buffer size may signal this using the following Mi,SIZEi assignments

M1,M2,M3 - the FlexMux channels

SIZE1, SIZE2, SIZE3 - the exact sizes of FlexMux buffers, for FlexMux channels M1,M2 and M3 of this FlexMux stream. Sizes are expressed in bytes.

8. Flex Multiplexing

8.1 Some of the advantages:

1. 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. The use of one session per elementary stream cannot be much cost effective, both on the server side and on the client side in terms of performance, when the number of elementary streams will increase within a scene.

2. The use of one single session for a multiplexed bitstream enables to send a bunch of ESs that are tightly synchronized together. Some of these ESs can themselves be Bifs and OD ESs when a scene description is used with Audio-Visual ES, and some other ESs can be OCI ES, and even IPMP ES when such systems are involved.
3. The FlexMultiplexing management supports embedding multiple SL packets into one FlexMux packet, by the use of FlexMux codetable entries.

4. If the multiplexing policy used is smoothing at most the multiplexed SL streams, mutual synchronization between these SL streams can be easily preserved when packet losses occur.

5. The use of the FlexMux technology enables possible interconnection between Internet network and digital television network, as MPEG normatively defines the use the MPEG-4 FlexMux syntax to carry MPEG-4 over MPEG-2 transport channels[8].

6. The reconstruction of the correct timing of the FlexMux stream is possible, if some QoS requirements are supported.

7. The overall MPEG-4 receiver buffer size is reduced, as MPEG-4 compliant Flexmultiplexed streams, by the use of the MPEG-4 timestamps, respect the MPEG-4 system decoder model.

8. The overall bandwidth management is easier. The FlexMux syntax allows having piecewise constant bitrate FlexMux bitstream, with an inband signaling mechanism.

9. Protection can be enhanced by means of repetitions of vital SL packets.

10. Content providers are able to bundle together a single stream with assurance that associated streams will be kept together and synchronized.

8.2 Disadvantages:

The major disadvantage with the packetization of the MPEG-4 Flexmultiplexed streams is the added packet header overhead. MPEG-4 does not support a reduction mechanism of the carried MPEG-4 Flexmultiplexed streams packet headers. This issue needs certainly be resolved using a mechanism similar to what was proposed with [7]. During their transport, FlexMux packets need not be compliant to the MPEG-4 standard, it is only when they are delivered to the Flexdemultiplexer that the MPEG-4 compliance point is defined.

8.3. Transport of MPEG-4 FlexMux streams

An MPEG-4 FlexMux stream is mapped directly onto the RTP payload without any addition of extra header fields or removal of any FlexMux packet header syntactic elements. Each RTP packet will contain a timestamp derived from the sender’s clock reference. This clock is synchronized to the FlexMux Clock Reference (FCR) and represents the target transmission time of the first byte of the RTP packet payload.
On the receiving side, the RTP packet timestamp will not be passed to the MPEG-4 Flexdemultiplexer. This use of the timestamp is slightly different from the normal use in RTP, in that it is not considered to be the media display time-stamp. The first purpose of this RTP timestamp will then be to reduce (after estimation) the network jitter, and the relative time drift between the transmitter and the receiver.

There are packetization restrictions due to the fact that no synchronization pattern is part of the FlexMux packet header: An RTP packet payload should start with the start of a FlexMux packet. An RTP packet will contain an integer number of FlexMux packets.

The FlexMux characteristics (declaration descriptor, timing descriptor, Channel Table descriptor, codetable entry descriptor, buffersize descriptor) may be provided by out of band means (e.g. SDP), or by the inband signaling mechanism supported by the use of the FlexMuxChannel signalling descriptors.

The FlexMux declaration descriptor, FlexMux timing descriptor, FlexMux Channel Table descriptor, FlexMux buffersize descriptor, may only be used to define statically the FlexMux stream characteristics.

The FlexMux codetable entry descriptor may change dynamically.

The IP packet marking facility may be needed. If this is the case, as it is based on the ‘degradationPriority’ field present in each FlexMux packet, all the FlexMux packets grouped in the same RTP packet should have the ‘degradationPriority’ field filled with the same value.

The size of the FlexMux packets should be adjusted such that the resulting RTP packet (embedding one or several FlexMux packets) is not larger than the path-MTU.

Protection mechanisms for FlexMux streams within RTP packets are outside of the scope of this specification.

C.Roux & al.  expires 14/08/01  14

RTP payload format for MPEG-4 FlexMultiplexed streams  23/02/01

9. SDP syntax

9.1 attributes

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.
<name> when equal to mpeg4-flexmux indicates the encoding type of the media, one MPEG-4 FlexMux stream. When equal to mpeg4-flexsig indicates the encoding type of the media, one MPEG-4 FlexMux signaling stream.

*time scale* is the time scale of the RTP time stamps.

*parameters* if used, can be a way to specialise *name*.

In order to be able to define either in the clear, or in binary format, or to point out to some already defined characteristics (MPEG-4 FlexMux descriptors) of a FlexMux stream, the following attribute is defined

\[ a = \text{mpeg4-flexmuxinfo: <location of descriptor list>} \]

*location of descriptor list* is a URL enclosed in double quotes, that will supply the required flexmux list of MPEG-4 FlexMux 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). The data at the indicated URL consists of some number of concatenated complete FlexMux descriptors. These descriptors have an intrinsic length, so simple concatenation suffices.

Examples:

- \[ a = \text{mpeg4-flexmuxinfo:"http://example.com/FlexMuxdescriptor.dat"} \]
- \[ a = \text{mpeg4-flexmuxinfo:"http://example.com/FlexMuxdescriptor.xml"} \]
- \[ a = \text{mpeg4-flexmuxinfo:"data:application/mpeg4-flexmuxinfo,7ab3742134bab347"} \]
- \[ a = \text{mpeg4-flexmuxinfo:"data:text/xml;<FlexMuxdescriptor.../>"} \]

C.Roux & al. expires 14/08/01

RTP payload format for MPEG-4 FlexMultiplexed streams 23/02/01

The MPEG-4 FlexMux descriptors related to FlexMux description are defined within the paragraph 7.4. The list of MPEG-4 descriptors cannot be empty. Ad Hoc descriptors can complete it. The MIME name used for this data is defined below.

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

9.2 MIME Types

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.

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 packets in transmission order.

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.

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.

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

Required parameters: none
Optional parameters: none
Encoding considerations: base64 generally preferred; files are binary and should be transmitted without CR/LF conversion, 7-bit stripping etc.

10. RTSP usage:

When RTSP is used as a session-control protocol:
RTP SHOULD be used as the transport protocol.
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.

Note that if an MPEG-4 FlexMux stream is closed (TEARDOWN) then the RTSP session ID will be lost. The next (re-)opened FlexMux stream will supply a new session ID. New DESCRIBEs may be needed.

11. 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.

12. References

[8] B. Thompson, T. Koren, D. Wing, Tunneling multiplexed
    Compressed RTP ('TCRTP'), work in progress,

C. Roux & al. expires 14/08/01
RTP payload format for MPEG-4 FlexMultiplexed streams 23/02/01

    over IP-based Protocols, work in progress,
    draft-singer-mpeg4-ip-01.txt, October 2000.
[10] Handley, Jacobson, SDP: Session Description Protocol,

13. Authors’ Addresses

Jan Van der Meer
Philips Digital Networks
Cederlaan 4
5600 JB Eindhoven
Netherlands
e-mail: jan.vandermeer@philips.com

C. Roux, D. Curet,
E. Gouleau, S. Relier
France Telecom
rue du Clos Courtel
35512 Cesson-Sevigne
France
e-mails: catherine.roux, dominique.curet
emmanuel.gouleau, stephanie.relier@rd.francetelecom.fr

Pierre Clement
Thomcast
40 r Bray
35510 Cesson-Sevigne
France
pierre.clement@thomcast.thomson-csf.com

Guy Cherry
nCUBE
110 Marsh Drive, Suite 200
Foster City, California 94404-1184
USA
guyc@ncube.com

C. Roux & al. expires 14/08/01