

RTP Payload Format for MPEG1/MPEG2 Video

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Abstract

This memo describes a packetization scheme for MPEG video and audio streams. The scheme proposed can be used to transport such a video or audio flow over the transport protocols supported by RTP. Two approaches are described. The first is designed to support maximum interoperability with MPEG System environments. The second is designed to provide maximum compatibility with other RTP-encapsulated media streams and future conference control work of the IETF.

1. Introduction

ISO/IEC JTC1/SC29 WG11 (also referred to as the MPEG committee) has defined the MPEG1 standard (ISO/IEC 11172)[1] and the MPEG2 standard (ISO/IEC 13818)[2]. This memo describes a packetization scheme to transport MPEG video and audio streams using the Real-time Transport Protocol (RTP), version 2 [3, 4].

The MPEG1 specification is defined in three parts: System, Video and Audio. It is designed primarily for CD-ROM-based applications, and is optimized for approximately 1.5 Mbits/sec combined data rates. The video and audio portions of the specification describe the basic format of the video or audio stream. These formats define the Elementary Streams (ES). The MPEG1 System specification defines an encapsulation of the ES that contains Presentation Time Stamps (PTS), Decoding Time Stamps and System Clock references, and performs multiplexing of MPEG1 compressed video and audio ES's with user data.

The MPEG2 specification is structured in a similar way. However, it hasn't been restricted only to CD-ROM applications. The MPEG2 System specification defines two system stream formats: the MPEG2 Transport Stream (MTS) and the MPEG2 Program Stream (MPS). The MTS is tailored for communicating or storing one or more programs of MPEG2 compressed data and also other data in relatively error-prone environments. The MPS is tailored for relatively error-free environments.

We seek to achieve interoperability among 4 types of end-systems in the following specification. The 4 types are:

1. Transmitting Interworking Unit (TIU)

Receives MPEG information from a native MTS system for distribution over packet networks using a native RTP-based system layer (such as an IP-based internetwork). Examples: real-time encoder, MTS satellite link to Internet, video server with MTS-encoded source material.

2. Receiving Interworking Unit (RIU)

Receives MPEG information in real time from an RTP-based network for forwarding to a native MTS environment. Examples: Internet-based video server to MTS-based cable distribution plant.

3. Transmitting Internet End-System (TAES)

Transmits MPEG information generated or stored within the internet end-system itself, or received from internet-based computer networks. Example: video server.

4. Receiving Internet End-System (RAES)

Receives MPEG information over an RTP-based internet for consumption at the internet end-system or forwarding to traditional computer network. Example: desktop PC or workstation viewing training video.

Each of the 2 types of transmitters must work with each of the 2 types of receivers. Because it is probable that the TAES, and certain that the RAES, will be based on existing and planned internet-connected computers, it is highly desirable for the interoperable protocol to be based on RTP.

Because of the range of applications that might employ MPEG streams, we propose to define two payload formats.

Much interest in the MPEG community is in the use of one of the MPEG System encodings, and hence, in [Section 2](#) we propose encapsulations of MPEG1 System streams and MPEG2 Transport and Program Streams with RTP. This profile supports the full semantics of MPEG System and offers basic interoperability among all four end-system types.

When operating only among internet-based end-systems (i.e., TAES and RAES) a payload format that provides greater compatibility with the Internet architecture is desired, deferring some of the system issues to other protocols being defined in the Internet community (such as the MMUSIC WG). In [Section 3](#) we propose an encapsulation of compressed video and audio data (referred to in MPEG documentation as "Elementary Streams" (ES)) complying with either MPEG1 or MPEG2. Here, neither of the System standards of MPEG1 or MPEG2 are utilized. The ES's are directly encapsulated with RTP.

Throughout this specification, we make extensive use of MPEG terminology. The reader should consult the primary MPEG references for definitive descriptions of this terminology.

2. Encapsulation of MPEG System and Transport Streams

Each RTP packet will contain a timestamp derived from the sender's 90KHz clock reference. This clock is synchronized to the system stream Program Clock Reference (PCR) or System Clock Reference (SCR) and represents the target transmission time of the first byte of the packet payload. The RTP timestamp will not be passed to the MPEG decoder. This use of the timestamp is somewhat different than normally is the case in RTP, in that it is not considered to be the media display or presentation timestamp. The primary purposes of the RTP timestamp will be to estimate and reduce any network-induced jitter and to synchronize relative time drift between the transmitter and receiver.

For MPEG2 Transport Streams the RTP payload will contain an integral number of MPEG transport packets. To avoid end system inefficiencies, data from multiple small MTS packets (normally fixed in size at 188 bytes) are aggregated into a single RTP packet. The number of transport packets contained is computed by dividing RTP payload length by the length of an MTS packet (188).

For MPEG2 Program streams and MPEG1 system streams there are no packetization restrictions; these streams are treated as a packetized stream of bytes.

2.1 RTP header usage

The RTP header fields are used as follows:

Payload Type: Distinct payload types should be assigned for of MPEG1 System Streams, MPEG2 Program Streams and MPEG2 Transport Streams. See [4] for payload type assignments.

M bit: Set to 1 whenever the timestamp is discontinuous (such as might happen when a sender switches from one data source to another). This allows the receiver and any intervening RTP mixers or translators that are synchronizing to the flow to ignore the difference between this timestamp and any previous timestamp in their clock phase detectors.

timestamp: 32 bit 90K Hz timestamp representing the target transmission time for the first byte of the packet.

3. Encapsulation of MPEG Elementary Streams

The following ES types may be encapsulated directly in RTP:

- (a) MPEG1 Video (ISO/IEC 11172-2)
- (b) MPEG2 Video (ISO/IEC 13818-2)
- (c) MPEG1 Audio (ISO/IEC 11172-3)
- (d) MPEG2 Audio (ISO/IEC 13818-3)

A distinct RTP payload type is assigned to MPEG1/MPEG2 Video and MPEG1/MPEG2 Audio, respectively. Further indication as to whether the data is MPEG1 or MPEG2 need not be provided in the RTP or MPEG-specific headers of this encapsulation, as this information is available in the ES headers.

Presentation Time Stamps (PTS) of 32 bits with an accuracy of 90 kHz shall be carried in the fixed RTP header. All packets that make up a audio or video frame shall have the same time stamp.

3.1 MPEG Video elementary streams

MPEG1 Video can be distinguished from MPEG2 Video at the video sequence header, i.e. for MPEG2 Video a `sequence_header()` is followed by `sequence_extension()`. The particular profile and level of MPEG2 Video (MAIN_Profile@MAIN_Level, HIGH_Profile@HIGH_Level, etc) are determined by the `profile_and_level_indicator` field of the `sequence_extension` header of MPEG2 Video.

The MPEG bit-stream semantics were designed for relatively error-free environments, and there is significant amount of dependency (both

temporal and spatial) within the stream such that loss of some data make other uncorrupted data useless. The format as defined in this encapsulation uses application layer framing information plus additional information in the RTP stream-specific header to allow for certain recovery mechanisms. Appendix 1 suggests several recovery strategies based on the properties of this encapsulation.

Since MPEG pictures can be large, they will normally be fragmented into packets of size less than a typical LAN/WAN MTU. The following fragmentation rules apply:

1. The MPEG Video_Sequence_Header, when present, will always be at the beginning of an RTP payload.
2. An MPEG GOP_header, when present, will always be at the beginning of the RTP payload, or will follow a Video_Sequence_Header.
3. An MPEG Picture_Header, when present, will always be at the beginning of a RTP payload, or will follow a GOP_header.

Each ES header must be completely contained within the packet. Consequently, a minimum RTP payload size of 261 bytes must be supported to contain the largest single header defined in the ES (that is, the extension_data() header containing the quant_matrix_extension()). Otherwise, there are no restrictions on where headers may appear within packet payloads.

In MPEG, each picture is made up of one or more "slices," and a slice is intended to be the unit of recovery from data loss or corruption. An MPEG-compliant decoder will normally advance to the beginning of next slice whenever an error is encountered in the stream. MPEG slice begin and end bits are provided in the encapsulation header to facilitate this.

The beginning of a slice must either be the first data in a packet (after any MPEG ES headers) or must follow after some integral number of slices in a packet. This requirement insures that the beginning of the next slice after one with a missing packet can be found without requiring that the receiver scan the packet contents. Slices may be fragmented across packets as long as all the above rules are met.

An implementation based on this encapsulation assumes that the Video_Sequence_Header is repeated periodically in the MPEG bit-stream. In practice (though not required by MPEG standard) this is used to allow channel switching and to receive and start decoding a continuously relayed MPEG bit-stream at arbitrary points in the media stream. It is suggested that when playing back from an MPEG stream from a file format (where the Video_Sequence_Header may only be

represented at the beginning of the stream) that the first Video_Sequence_Header (preceded by an end-of-stream indicator) be saved by the packetizer for periodic injection in to the network stream.

3.2 MPEG Audio elementary streams

MPEG1 Audio can be distinguished from MPEG2 Audio from the MPEG ancillary_data() header. For either MPEG1 or MPEG2 Audio, distinct Presentation Time Stamps may be present for frames which correspond to either 384 samples for Layer-I, or 1152 samples for Layer-II or Layer-III. The actual number of bytes required to represent this number of samples will vary depending on the encoder parameters.

Multiple audio frames may be encapsulated within one RTP packet. In this case, an integral number of audio frames must be contained within the packet and the fragmentation header defined in [Section 3.5](#) shall be set to 0.

Also, if relatively short packets are to be used, one frame may be so large that it may straddle multiple RTP packets. For example, for Layer-II MPEG audio sampled at a rate of 44.1 KHz each frame would represent a time slot of 26.1 msec. At this sampling rate if the compressed bit-rate is 384 kbits/sec (i.e. 48 kBytes/sec) then the average audio frame size would be 1.25 KBytes. If packets were to be 500 Bytes long, then each audio frame would straddle 3 RTP packets. The audio fragmentation indicator header (See [Section 3.5](#)) shall be present for an MPEG1/2 Audio payload type to provide for this fragmentation.

3.3 RTP Fixed Header for MPEG ES encapsulation

The RTP header fields are used as follows:

Payload Type: Distinct payload types should be assigned for video elementary streams and audio elementary streams. See [4] for payload type assignments.

M bit: For video, set to 1 on packet containing MPEG frame end code, 0 otherwise. For audio, set to 1 on first packet of a "talk-spurt," 0 otherwise.

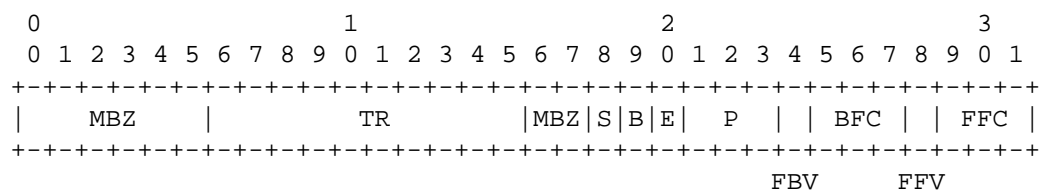
PT: MPEG video or audio stream ID.

timestamp: 32-bit 90K Hz timestamp representing presentation time of MPEG picture or audio frame. Same for all packets that make up a picture or audio frame. May not be monotonically increasing in video stream if B pictures

present in stream. For packets that contain only a video sequence and/or GOP header, the timestamp is that of the subsequent picture.

3.4 MPEG Video-specific header

This header shall be attached to each RTP packet after the RTP fixed header.



MBZ: Unused. Must be set to zero in current specification. This space is reserved for future use.

TR: Temporal-Reference (10 bits). The temporal reference of the current picture within the current GOP. This value ranges from 0-1023 and is constant for all RTP packets of a given picture.

MBZ: Unused. Must be set to zero in current specification. This space is reserved for future use.

S: Sequence-header-present (1 bit). Normally 0 and set to 1 at the occurrence of each MPEG sequence header. Used to detect presence of sequence header in RTP packet.

B: Beginning-of-slice (BS) (1 bit). Set when the start of the packet payload is a slice start code, or when a slice start code is preceded only by one or more of a Video_Sequence_Header, GOP_header and/or Picture_Header.

E: End-of-slice (ES) (1 bit). Set when the last byte of the payload is the end of an MPEG slice.

P: Picture-Type (3 bits). I (1), P (2), B (3) or D (4). This value is constant for each RTP packet of a given picture. Value 000B is forbidden and 101B - 111B are reserved to support future extensions to the MPEG ES specification.

FBV: full_pel_backward_vector

BFC: backward_f_code

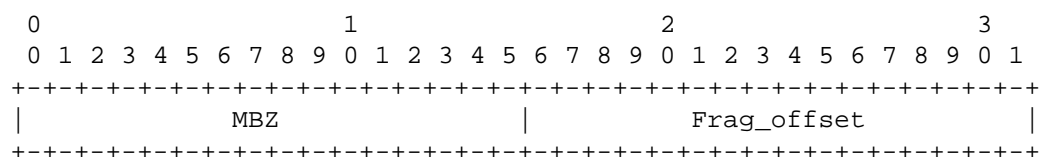
FFV: full_pel_forward_vector

FFC: forward_f_code

Obtained from the most recent picture header, and are constant for each RTP packet of a given picture. None of these values are used for I frames and must be set to zero in the RTP header. For P frames only the last two values are present and FBV and BFC must be set to zero in the RTP header. For B frames all the four values are present.

3.5 MPEG Audio-specific header

This header shall be attached to each RTP packet at the start of the payload and after any RTP headers for an MPEG1/2 Audio payload type.



Frag_offset: Byte offset into the audio frame for the data in this packet.

Appendix 1. Error Recovery and Resynchronization Strategies.

The following error recovery and resynchronization strategies are intended to be guidelines only. A compliant receiver is free to employ alternative (or no) strategies.

When initially decoding an RTP-encapsulated MPEG Elementary Stream, the receiver may discard all packets until the Sequence-header-present bit is set to 1. At this point, sufficient state information is contained in the stream to allow processing by an MPEG decoder.

Loss of packets containing the GOP_header and/or Picture_Header are detected by an unexpected change in the Temporal-Reference and Picture-Type values. Consider the following example GOP sequence:

```
In display order: 0B 1B 2I 3B 4B 5P 6B 7B 8P GOP_HDR 0B ...
In stream order:  2I 0B 1B 5P 3B 4B 8P 6B 7B GOP_HDR 2I ...
```

Consider also two counters:

```
ref_pic_temp (Reference Picture (I,P) Temporal Reference)
dep_pic_temp (Dependent Picture (B) Temporal Reference)
```

At each GOP beginning, set these counters to the temporal reference value of the corresponding picture type. For our example GOP sequence, `ref_pic_temp` = 2 and `dep_pic_temp` = 0. Keep incrementing BOTH counters by unity with each following picture. `Ref_pic_temp` should match the temporal references of the I and P frames, and `dep_pic_temp` should match the temporal references of the B frames.

```
dep_pic_temp: - 0 1 2 3 4 5 6 7          8 9
In stream order: 2I 0B 1B 5P 3B 4B 8P 6B 7B GOP_H 2I 0B 1B ...
ref_pic_temp:  2 3 4 5 6 7 8 9 10 ^    11
               ----- |    ^
                   Match Drop |
                           Mismatch
                           in ref_pic_temp
```

The loss of a GOP header can be detected by matching the appropriate counter (based on picture type) to the temporal reference value. A mismatch indicates a lost GOP header. If desired, a GOP header can be re-constructed using a "null" time_code, repeating the closed_gop flag from previous GOP headers, and setting the broken_link flag to 1.

The loss of a Picture_Header can also be detected by a mismatch in the Temporal Reference contained in the RTP packet from the appropriate `dep_pic_temp` or `ref_pic_temp` counters at the receiver.

After scanning to the next Beginning-of-slice the Picture_Header is reconstructed from the P, TR, FBV, BFC, FFV and FFC contained in that packet, and from stream-dependent default values.

Any time an RTP packet is lost (as indicated by a gap in the RTP sequence number), the receiver may discard all packets until the Beginning-of-slice bit is set. At this point, sufficient state information is contained in the stream to allow processing by an MPEG decoder starting at the next slice boundary (possibly after reconstruction of the GOP_header and/or Picture_Header as described above).

References

- [1] ISO/IEC International Standard 11172; "Coding of moving pictures and associated audio for digital storage media up to about 1,5 Mbits/s", November 1993.
- [2] ISO/IEC International Standard 13818; "Generic coding of moving pictures and associated audio information", November 1994.
- [3] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", [RFC 1889](#), January 1996.
- [4] H. Schulzrinne, "RTP Profile for Audio and Video Conferences with Minimal Control", [RFC 1890](#), January 1996.

Authors' Addresses

Gerard Fernando
Sun Microsystems, Inc.
Mail-stop UMPK14-305
2550 Garcia Avenue
Mountain View, California 94043-1100
USA

Phone: +1 415-786-6373
EMail: gerard.fernando@eng.sun.com

Vivek Goyal
Precept Software, Inc.
1072 Arastradero Rd,
Palo Alto, CA 94304
USA

Phone: +1 415-845-5200
EMail: goyal@precept.com

Don Hoffman
Sun Microsystems, Inc.
Mail-stop UMPK14-305
2550 Garcia Avenue
Mountain View, California 94043-1100
USA

Phone: +1 503-297-1580
EMail: don.hoffman@eng.sun.com