Payload Working Group   J. Downs, Ed.
Internet-Draft   PAR Government Systems Corp.
Intended status: Standards Track   J. Arbeiter, Ed.
Expires: September 2, 2012   March 1, 2012

RTP Payload Format for SMPTE 336M Encoded Data
draft-ietf-payload-rtp-klv-04

Abstract

This document specifies the payload format for packetization of KLV (Key-Length-Value) Encoded Data, as defined by the Society of Motion Picture and Television Engineers (SMPTE) in SMPTE 336M, into the Real-time Transport Protocol (RTP).

Table of Contents

1. Introduction 3
2. Conventions, Definitions and Acronyms 3
3. Media Format Background 3
4. Payload Format 4
4.1. RTP Header Usage 4
4.2. Payload Data 5
4.2.1. The KLVunit 5
4.2.2. KLVunit Mapping to RTP Packet Payload 5
4.3. Implementation Considerations 6
4.3.1. Loss of Data 6
4.3.1.1. Damaged KLVunits 6
4.3.1.2. Treatment of Damaged KLVunits 8
5. Congestion Control 8
6. Payload Format Parameters 8
6.1. Media Type Definition 8
6.2. Mapping to SDP 9
6.2.1. Offer/Answer Model and Declarative Considerations . . 9
7. IANA Considerations 10
8. Security Considerations 10
9. References 11
9.1. Normative References 11
9.2. Informative References 11
Authors' Addresses 12

1. Introduction

This document specifies the payload format for packetization of KLV (Key-Length-Value) Encoded Data, as defined by the Society of Motion Picture and Television Engineers (SMPTE) in [SMPTE336M], into the Real-time Transport Protocol (RTP) [RFC3550].

The payload format is defined in such a way that arbitrary KLV data can be carried. No restrictions are placed on which KLV data keys can be used.

A brief description of SMPTE 336M, KLV Encoded Data, is given. The payload format itself, including use of the RTP header fields, is specified in Section 4. The media type and IANA considerations are also described. This document concludes with security considerations relevant to this payload format.

2. Conventions, Definitions and Acronyms

The term "Universal Label Key" is used in this document to refer to a fixed-length, 16-byte SMPTE-administered Universal Label (see [SMPTE298M]) that is used as an identifying key in a KLV item.

The term "KLV item" is used in this document to refer to one single Universal Label Key, length, and value triplet encoded as described in [SMPTE336M].

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 [RFC2119].

3. Media Format Background

[SMPTE336M], Data Encoding Protocol Using Key-Length-Value, defines a byte-level data encoding protocol for representing data items and data groups. This encoding protocol definition is independent of the application or transportation method used.

SMPTE 336M data encoding can be applied to a wide variety of binary data. This encoding has been used to provide diverse and rich metadata sets that describe or enhance associated video presentations. Use of SMPTE 336M encoded metadata in conjunction with video has enabled improvements in multimedia presentations, content management and distribution, archival and retrieval, and production workflow.

The SMPTE 336M standard defines a Key-Length-Value (KLV) triplet as a data interchange protocol for data items or data groups where the Key identifies the data, the Length specifies the length of the data and the Value is the data itself. The KLV protocol provides a common interchange point for all compliant applications irrespective of the method of implementation or transport.

The Key of a KLV triplet (a Universal Label Key) is coded using a fixed-length 16-byte SMPTE-administered Universal Label. [SMPTE298M] further details the structure of 16-byte SMPTE-administered Universal Labels. Universal Label Keys are maintained in registries published by SMPTE (see, for example, [SMPTE335M] and [SMPTERP210]).

The standard also provides methods for combining associated KLV triplets in data sets where the set of KLV triplets is itself coded with KLV data coding protocol. Such sets can be coded in either full form (Universal Sets) or in one of four increasingly bit-efficient forms (Global Sets, Local Sets, Variable Length Packs and Defined Length Packs). The standard provides a definition of each of these data constructs.

Additionally, the standard defines the use of KLV coding to provide a means to carry information that is registered with a non-SMPTE external agency.

4. Payload Format

The main goal of the payload format design for SMPTE 336M data is to provide carriage of SMPTE 336M data over RTP in a simple, yet robust manner. All forms of SMPTE 336M data can be carried by the payload format. The payload format maintains simplicity by using only the standard RTP headers and not defining any payload headers.

SMPTE 336M KLV data is broken into KLVunits. A KLVunit is simply a logical grouping of otherwise unframed KLV data, grouped based on source data timing (see Section 4.2.1). Each KLVunit is then placed into one or more RTP packet payloads. The RTP header marker bit is used to assist receivers in locating the boundaries of KLVunits.

4.1. RTP Header Usage

This payload format uses the RTP packet header fields as described in the table below:

Field	Usage
Timestamp	The RTP Timestamp encodes the instant along a presentation timeline that the entire KLVunit encoded in the packet payload is to be presented. When one KLVunit is placed in multiple RTP packets, the RTP timestamp of all packets comprising that KLVunit MUST be the same. The timestamp clock frequency is defined as a parameter to the payload format (Section 6).
M-bit	The RTP header marker bit (M) is used to demarcate KLVunits. Senders MUST set the marker bit to '1' for any RTP packet which contains the final byte of a KLVunit. For all other packets, senders MUST set the RTP header marker bit to '0'. This allows receivers to pass a KLVunit for parsing/decoding immediately upon receipt of the last RTP packet comprising the KLVunit. Without this, a receiver would need to wait for the next RTP packet with a different timestamp to arrive, thus signaling the end of one KLVunit and the start of another.

The remaining RTP header fields are used as specified in [RFC3550].

4.2. Payload Data

4.2.1. The KLVunit

A KLVunit is a logical collection of all KLV items that are to be presented at a specific time. A KLVunit is comprised of one or more KLV items. Compound items (sets, packs) are allowed as per [SMPTE336M], but the contents of a compound item MUST NOT be split across two KLVunits. Multiple KLV items in a KLVunit occur one after another with no padding or stuffing between items.

4.2.2. KLVunit Mapping to RTP Packet Payload

An RTP packet payload SHALL contain one, and only one, KLVunit or a fragment thereof. KLVunits small enough to fit into a single RTP packet (RTP packet size is up to implementation but should consider underlying transport/network factors such as MTU limitations) are placed directly into the payload of the RTP packet, with the first byte of the KLVunit (which is the first byte of a KLV Universal Label Key) being the first byte of the RTP packet payload.

KLVunits too large to fit into a single RTP packet payload MAY span multiple RTP packet payloads. When this is done, the KLVunit data

MUST be sent in sequential byte order, such that when all RTP packets comprising the KLVunit are arranged in sequence number order, concatenating the payload data together exactly reproduces the original KLVunit.

Additionally, when a KLVunit is fragmented across multiple RTP packets, all RTP packets transporting the fragments of a KLVunit MUST have the same timestamp.

KLVunits are bounded with changes in RTP packet timestamps. The marker (M) bit in the RTP packet headers marks the last RTP packet comprising a KLVunit (see Section 4.1).

4.3. Implementation Considerations

4.3.1. Loss of Data

RTP is generally deployed in network environments where packet loss might occur. RTP header fields enable detection of lost packets, as described in [RFC3550]. When transmitting payload data described by this payload format, packet loss can cause the loss of whole KLVunits or portions thereof.

4.3.1.1. Damaged KLVunits

A damaged KLVunit is any KLVunit that was carried in one or more RTP packets that have been lost. When a lost packet is detected (through use of the sequence number header field), the receiver:

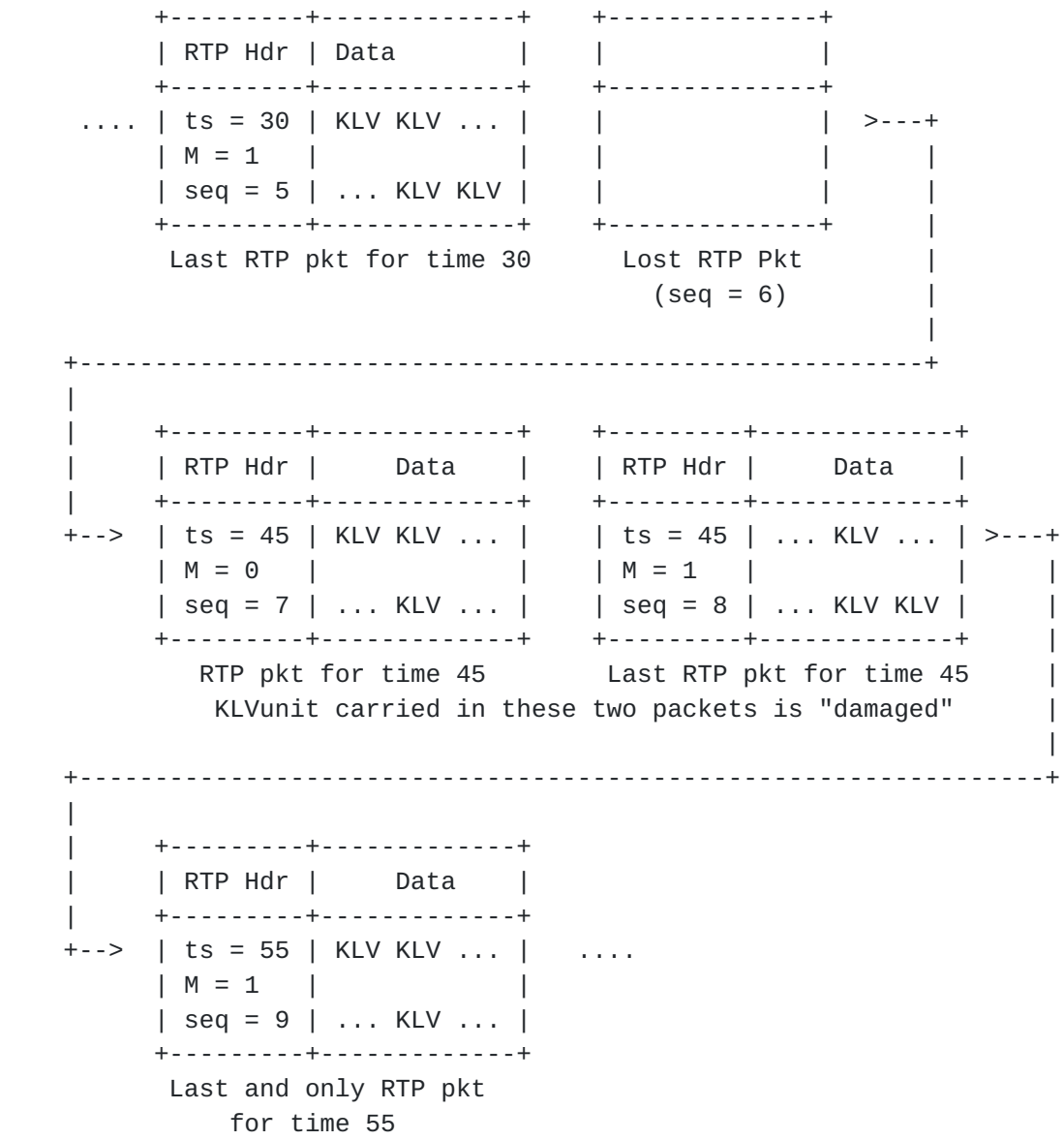
- MUST consider the KLVunit partially received before a lost packet as damaged. This damaged KLVunit includes all packets prior to the lost one (in sequence number order) back to, but not including, the most recent packet in which the M-bit in the RTP header was set to '1'.

- MUST consider the first KLVunit received after a lost packet as damaged. This damaged KLVunit includes the first packet after the lost one (in sequence number order) and, if the first packet has its M-bit in the RTP header is set to '0', all subsequent packets up to and including the next one with the M-bit in the RTP header set to '1'.

The above applies regardless of the M-bit value in the RTP header of the lost packet itself. This enables very basic receivers to look solely at the M-bit to determine the outer boundaries of damaged KLVunits. For example, when a packet with the M-bit set to '1' is lost, the KLVunit that the lost packet would have terminated is considered damaged, as is the KLVunit comprised of packets received

subsequent to the lost packet (up to and including the next received packet with M-bit set to '1').

The example below illustrates how a receiver would handle a lost packet in another possible packet sequence:



In this example, the packets with sequence numbers 7 and 8 contain portions of a KLVunit with timestamp of 45. This KLVunit is considered "damaged" due to the missing RTP packet with sequence number 6, which might have been part of this KLVunit. The KLVunit for timestamp 30 (ended in packet with sequence number 5) is unaffected by the missing packet. The KLVunit for timestamp 55,

carried in the packet with sequence number 9, is also unaffected by the missing packet and is considered complete and intact.

4.3.1.2. Treatment of Damaged KLVunits

SMPTE 336M KLV data streams are built in such a way that it is possible to partially recover from errors or missing data in a stream. Exact specifics of how damaged KLVunits are handled are left to each implementation, as different implementations can have differing capabilities and robustness in their downstream KLV payload processing. Because some implementations can be particularly limited in their capacity to handle damaged KLVunits, receivers MAY drop damaged KLVunits entirely.

5. Congestion Control

The general congestion control considerations for transporting RTP data apply; see RTP [RFC3550] and any applicable RTP profile like AVP [RFC3551].

Further, SMPTE 336M data can be encoded in different schemes which reduce the overhead associated with individual data items within the overall stream. SMPTE 336M grouping constructs, such as local sets and data packs, provide a mechanism to reduce bandwidth requirements.

6. Payload Format Parameters

This RTP payload format is identified using the application/smpte336m media type which is registered in accordance with [RFC4855] and using the template of [RFC4288].

6.1. Media Type Definition

Type name: application

Subtype name: smpte336m

Required parameters:

rate: RTP timestamp clock rate. Typically chosen based on sampling rate of metadata being transmitted, but other rates can be specified.

Optional parameters: None

Encoding considerations: This media type is framed and binary; see Section 4.8 of [RFC4288].

Security considerations: See Section 8 of RFCXXXX (note to RFC editor: please replace XXXX with the number assigned to this RFC).

Interoperability considerations: Data items in smpte336m can be very diverse. Receivers might only be capable of interpreting a subset of the possible data items; unrecognized items are skipped. Agreement on data items to be used out of band, via application profile or similar, is typical.

Published specification: RFCXXXX

Applications that use this media type: Streaming of metadata associated with simultaneously streamed video and transmission of [SMPTE336M] based media formats (e.g. MXF [SMPTE377M]).

Additional Information: none

Person & email address to contact for further information: J. Downs <jeff_downs@partech.com>; IETF Payload Working Group <payload@ietf.org>

Intended usage: COMMON

Restrictions on usage: This media type depends on RTP framing, and hence is only defined for transfer via RTP ([RFC3550]). Transport within other framing protocols is not defined at this time.

Author:

J. Downs <jeff_downs@partech.com>

J. Arbeiter <jimsgti@gmail.com>

Change controller: IETF Payload working group delegated from the IESG.

6.2. Mapping to SDP

The mapping of the above defined payload format media type and its parameters SHALL be done according to Section 3 of [RFC4855].

6.2.1. Offer/Answer Model and Declarative Considerations

This payload format has no configuration or optional format parameters. Thus, when offering SMPTE 336M Encoded Data over RTP

using Session Description Protocol (SDP) in an Offer/Answer model [RFC3264] or in a declarative manner (e.g., SDP in the Real-time Streaming Protocol (RTSP) [RFC2326] or the Session Announcement Protocol (SAP) [RFC2974]), there are no specific considerations.

7. IANA Considerations

This memo requests that IANA registers application/smpte336m as specified in Section 6.1. The media type is also requested to be added to the IANA registry for "RTP Payload Format MIME types" (http://www.iana.org/assignments/rtp-parameters).

8. Security Considerations

RTP packets using the payload format defined in this specification are subject to the security considerations discussed in the RTP specification [RFC3550], and in any applicable RTP profile. The main security considerations for the RTP packet carrying the RTP payload format defined within this memo are confidentiality, integrity and source authenticity. Confidentiality is achieved by encryption of the RTP payload. Integrity of the RTP packets through suitable cryptographic integrity protection mechanism. Cryptographic systems may also allow the authentication of the source of the payload. A suitable security mechanism for this RTP payload format should provide confidentiality, integrity protection and at least source authentication capable of determining if an RTP packet is from a member of the RTP session or not.

Note that the appropriate mechanism to provide security to RTP and payloads following this memo may vary. It is dependent on the application, the transport, and the signaling protocol employed. Therefore a single mechanism is not sufficient, although if suitable the usage of SRTP [RFC3711] is recommended. Other mechanisms that may be used are IPsec [RFC4301] and TLS [RFC5246] (RTP over TCP), but also other alternatives may exist.

This RTP payload format presents the possibility for significant non-uniformity in the receiver-side computational complexity during processing of SMPTE 336M payload data. Because the length of SMPTE 336M encoded data items is essentially unbounded, receivers must take care when allocating resources used in processing. It is trivial to construct pathological data that would cause a naive decoder to allocate large amounts of resources, resulting in denial-of-service threats. Receivers SHOULD place limits on resource allocation that are within the bounds set forth by any application profile in use.

This RTP payload format does not contain any inherently active content. However, individual SMPTE 336M KLV items could be defined to convey active content in a particular application. Therefore, receivers capable of decoding and interpreting such data items should use appropriate caution and security practices. In particular, accepting active content from streams that lack authenticity or integrity protection mechanisms places a receiver at risk of attacks using spoofed packets. Receivers not capable of decoding such data items are not at risk; unknown data items are skipped over and discarded according to SMPTE 336M processing rules.

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, July 2003.

[RFC4288] Freed, N. and J. Klensin, "Media Type Specifications and Registration Procedures", BCP 13, RFC 4288, December 2005.

[RFC4855] Casner, S., "Media Type Registration of RTP Payload Formats", RFC 4855, February 2007.

9.2. Informative References

[RFC2326] Schulzrinne, H., Rao, A., and R. Lanphier, "Real Time Streaming Protocol (RTSP)", RFC 2326, April 1998.

[RFC2974] Handley, M., Perkins, C., and E. Whelan, "Session Announcement Protocol", RFC 2974, October 2000.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[SMPTE298M] Society of Motion Picture and Television Engineers, "ANSI/ SMPTE 298M-1997: Universal Labels for Unique Identification of Digital Data", 1997, <http://www.smpte.org>.

[SMPTE335M] Society of Motion Picture and Television Engineers, "SMPTE 335M-2001: Metadata Dictionary Structure", 2001, <http://www.smpte.org>.

[SMPTE336M] Society of Motion Picture and Television Engineers, "SMPTE336M-2007: Data Encoding Protocol Using Key-Length-Value", 2007, <http://www.smpte.org>.

[SMPTE377M] Society of Motion Picture and Television Engineers, "SMPTE 377M-2004: Material Exchange Format (MXF) File Format Specification", 2004, <http://www.smpte.org>.

[SMPTERP210] Society of Motion Picture and Television Engineers, "SMPTE RP 210v12: Metadata Dictionary Registry of Metadata Element Descriptions", 2010, <http://www.smpte.org>.

Authors' Addresses

J. Downs (editor)
PAR Government Systems Corp.
US

Phone:
Email: jeff_downs@partech.com

J. Arbeiter (editor)
US

Phone:
Email: jimsgti@gmail.com