Audio Video Transport Working	J. Arbeiter, Ed.	
Group	Harris Corporation	
Internet-Draft	J. Downs, Ed.	
Intended status: Standards Track	PAR Government Systems Corp.	
Expires: May 20, 2011	November 16, 2010	

TOC

RTP Payload Format for SMPTE 336M Encoded Data
draft-ietf-avt-rtp-klv-01

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).

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.
Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.
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.”
This Internet-Draft will expire on May 20, 2011.

Copyright Notice

Table of Contents

1. Introduction

TOC

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] (SMPTE, "SMPTE336M-2007: Data Encoding Protocol Using Key-Length-Value," 2007.), into the Real-time Transport Protocol (RTP) [RFC3550] (Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," July 2003.).
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 (Payload Format). 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

TOC

The term "KLV item" is used in this document to refer to one single universal key, length, and value triplet, or one single SMPTE Universal Label, encoded as described in [SMPTE336M] (SMPTE, “SMPTE336M-2007: Data Encoding Protocol Using Key-Length-Value,” 2007.).
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] (Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” March 1997.).

---

3. Description of SMPTE 336M Data

TOC

[SMPTE336M] (SMPTE, “SMPTE336M-2007: Data Encoding Protocol Using Key-Length-Value,” 2007.), 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 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.
The standard also describes implications of KLV coding including the use of a SMPTE Universal Label as a value within a KLV coding triplet or whose meaning is entirely conveyed by the SMPTE UL itself. The two kinds of usage for such standalone SMPTE Universal Labels are a) as a value in a K L V construct and b) as a Key that has no Length and no Value.

The standard also defines the use of KLV coding to provide a means to carry information that is registered with a non-SMPTE external agency. The encoding byte range (length of the payload) may accommodate unusually large volumes of data. Consequently, a specific application of KLV encoding may require only a limited operating data range and those details shall be defined in a relevant application document.

---

4. Payload Format

TOC

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 (see Section 4.2.1 (The KLVunit)) based on source data timing. 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

TOC

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 SHALL be defined as a parameter to the payload format (Payload Format Parameters).
M-bit	The RTP header marker bit (M) SHALL be set to '1' for any RTP packet which contains the final byte of a KLVunit. For all other packets, the RTP header marker bit SHALL be set 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] (Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications,” July 2003.).

4.2. Payload Data

TOC

4.2.1. The KLVunit

TOC

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] (SMPTE, “SMPTE336M-2007: Data Encoding Protocol Using Key-Length-Value,” 2007.), 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

TOC

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 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 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 (RTP Header Usage)).

4.3. Implementation Considerations

TOC

4.3.1. Loss of Data

TOC

RTP is generally deployed in network environments where packet loss may occur. RTP header fields enable detection of lost packets, as described in [RFC3550] (Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications,” July 2003.). 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

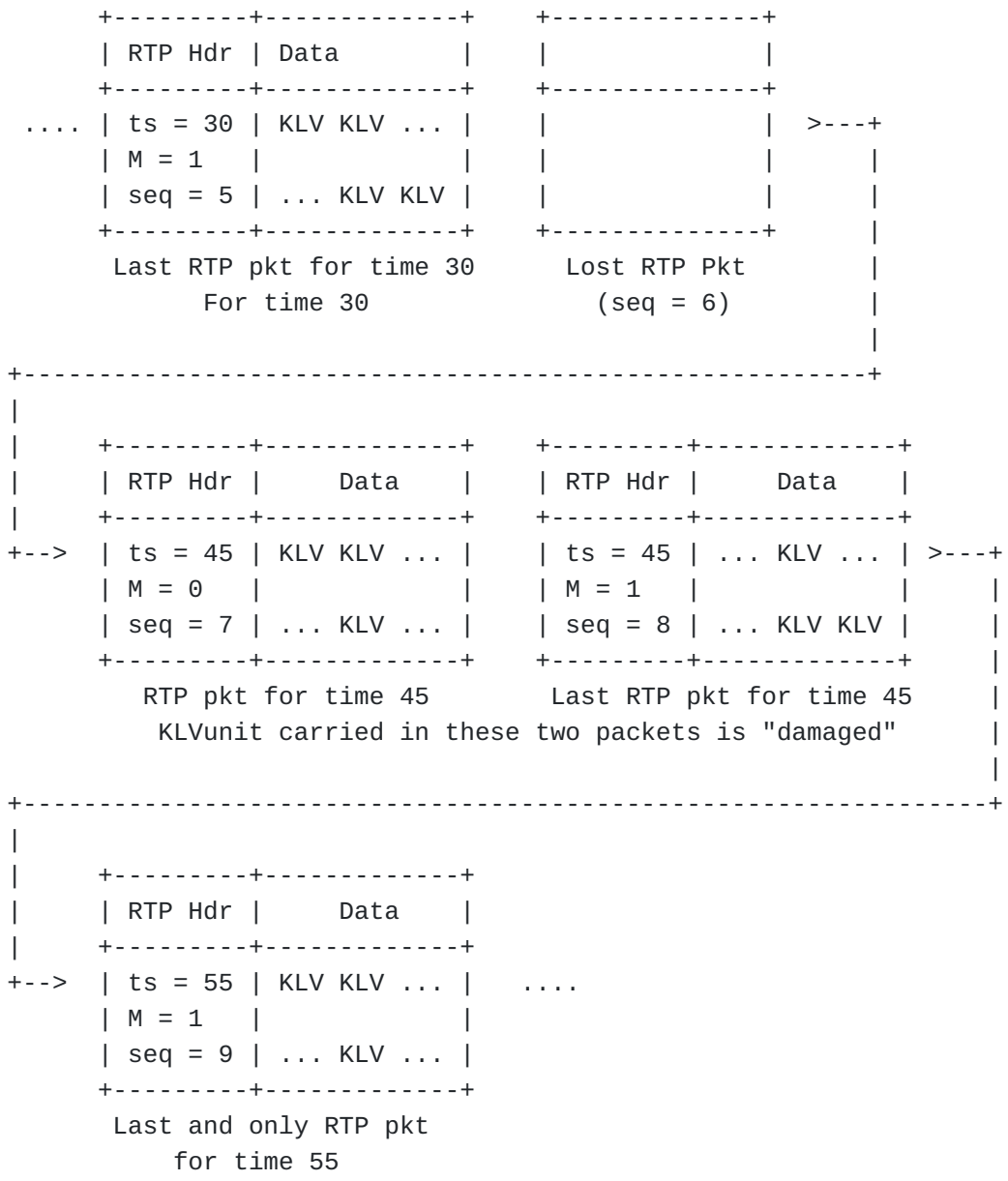
TOC

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:

- SHOULD consider the KLVunit carried in the prior packet (in sequence number order) as damaged unless that prior packet's M bit in the RTP header was set to '1'.

- SHOULD consider all subsequent packets (in sequence number order) up to and including the next one with the M-bit in the RTP header set to '1' as part of a damaged KLVunit.

The example below illustrates how a receiver would handle a lost packet in one 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 may 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.

TOC

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 may have differing capabilities and robustness in their downstream KLV payload processing. Because some implementations may be particularly limited in their capacity to handle damaged KLVunits, receivers MAY drop damaged KLVunits entirely.

---

5. Congestion Control

TOC

The general congestion control considerations for transporting RTP data apply; see RTP [RFC3550] (Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," July 2003.) and any applicable RTP profile like AVP [RFC3551] (Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control," July 2003.).
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

TOC

This RTP payload format is identified using the media type application/smpte336m, which is registered in accordance with [RFC4855] (Casner, S., "Media Type Registration of RTP Payload Formats," February 2007.) and using the template of [RFC4288] (Freed, N. and J. Klensin, "Media Type Specifications and Registration Procedures," December 2005.).

6.1. Media Type Definition

TOC

> 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 may be specified.

Optional parameters:

Encoding considerations: This media type is framed and binary; see Section 4.8 of [RFC4288] (Freed, N. and J. Klensin, “Media Type Specifications and Registration Procedures,” December 2005.).

Security considerations: See Section 8 (Security Considerations) of XXXX.

Interoperability considerations: Data items in smpte336m can be very diverse. Receivers may 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: XXXX

Applications that use this media type: Audio and video streaming and conferencing tools

Additional Information: none

Person & email address to contact for further information: J. Arbeiter <jarbeite@harris.com>

Intended usage: COMMON

Restrictions on usage: This media type depends on RTP framing, and hence is only defined for transfer via RTP ([RFC3550] (Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications,” July 2003.)). Transport within other framing protocols is not defined at this time.

Author:

> J. Arbeiter <jarbeite@harris.com>
>
> J. Downs <jeff_downs@partech.com>

Change controller: IETF Audio/Video Transport working group delegated from the IESG

TOC

6.2. Mapping to SDP

The mapping of the above defined payload format media type and its parameters to SDP [RFC4566] (Handley, M., Jacobson, V., and C. Perkins, “SDP: Session Description Protocol,” July 2006.) SHALL be done according to Section 3 of [RFC4855] (Casner, S., “Media Type Registration of RTP Payload Formats,” February 2007.).

---

7. IANA Considerations

TOC

This memo requests that IANA registers application/smpte336m as specified in Section 6.1 (Media Type Definition). 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

TOC

RTP packets using the payload format defined in this specification are subject to the security considerations discussed in the RTP specification [RFC3550] (Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications,” July 2003.), 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 system 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 signalling protocol employed. Therefore a single mechanism is not sufficient, although if suitable the usage of SRTP [RFC3711] (Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, “The Secure Real-time Transport Protocol (SRTP),” March 2004.) is recommended. Other mechanism that may be used are IPsec [RFC4301] (Kent, S. and K. Seo, “Security Architecture for the Internet Protocol,” December 2005.) and TLS [RFC5246] (Dierks, T. and E. Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.2,” August 2008.) (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 are encouraged to 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 inheritly 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. 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

TOC

9.1. Normative References

TOC

[RFC2119]	Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” BCP 14, RFC 2119, March 1997 (TXT, HTML, XML).
[RFC3550]	Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications,” STD 64, RFC 3550, July 2003 (TXT, PS, PDF).
[RFC3551]	Schulzrinne, H. and S. Casner, “RTP Profile for Audio and Video Conferences with Minimal Control,” STD 65, RFC 3551, July 2003 (TXT, PS, PDF).
[RFC4288]	Freed, N. and J. Klensin, “Media Type Specifications and Registration Procedures,” BCP 13, RFC 4288, December 2005 (TXT).
[RFC4566]	Handley, M., Jacobson, V., and C. Perkins, “SDP: Session Description Protocol,” RFC 4566, July 2006 (TXT).
[RFC4855]	Casner, S., “Media Type Registration of RTP Payload Formats,” RFC 4855, February 2007 (TXT).
[SMPTE336M]	SMPTE, “SMPTE336M-2007: Data Encoding Protocol Using Key-Length-Value,” 2007.

9.2. Informative References

TOC

[RFC3711]	Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, “The Secure Real-time Transport Protocol (SRTP),” RFC 3711, March 2004 (TXT).
[RFC4301]	Kent, S. and K. Seo, “Security Architecture for the Internet Protocol,” RFC 4301, December 2005 (TXT).
[RFC5246]	Dierks, T. and E. Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.2,” RFC 5246, August 2008 (TXT).

Authors' Addresses

TOC

	J. Arbeiter (editor)
	Harris Corporation
	US
Phone:	
Email:	jarbeite@harris.com
	J. Downs (editor)
	PAR Government Systems Corp.
	US
Phone:	
Email:	jeff_downs@partech.com