Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: February 5, 2021

M. Zanaty
E. Berger
S. Nandakumar
Cisco Systems

August 4, 2020

Frame Marking RTP Header Extension
draft-ietf-avtext-framemarking-11

Abstract

This document describes a Frame Marking RTP header extension used to convey information about video frames that is critical for error recovery and packet forwarding in RTP middleboxes or network nodes. It is most useful when media is encrypted, and essential when the middlebox or node has no access to the media decryption keys. It is also useful for codec-agnostic processing of encrypted or unencrypted media, while it also supports extensions for codec-specific information.

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 https://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 February 5, 2021.

Copyright Notice

Table of Contents

1. Introduction

Many widely deployed RTP [RFC3550] topologies [RFC7667] used in modern voice and video conferencing systems include a centralized component that acts as an RTP switch. It receives voice and video streams from each participant, which may be encrypted using SRTP [RFC3711], or extensions that provide participants with private media [I-D.ietf-perc-private-media-framework] via end-to-end encryption where the switch has no access to media decryption keys. The goal is to provide a set of streams back to the participants which enable them to render the right media content. In a simple video configuration, for example, the goal will be that each participant sees and hears just the active speaker. In that case, the goal of the switch is to receive the voice and video streams from each participant, determine the active speaker based on energy in the voice packets, possibly using the client-to-mixer audio level RTP header extension [RFC6464], and select the corresponding video stream for transmission to participants; see Figure 1.

In this document, an "RTP switch" is used as a common short term for the terms "switching RTP mixer", "source projecting middlebox", "source forwarding unit/middlebox" and "video switching MCU" as discussed in [RFC7667].

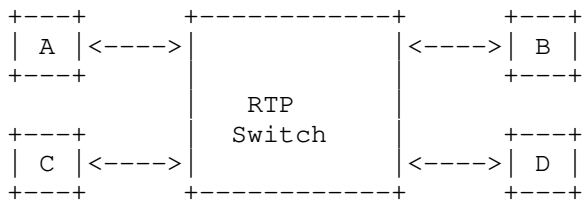


Figure 1: RTP switch

In order to properly support switching of video streams, the RTP switch typically needs some critical information about video frames in order to start and stop forwarding streams.

- Because of inter-frame dependencies, it should ideally switch video streams at a point where the first frame from the new speaker can be decoded by recipients without prior frames, e.g switch on an intra-frame.
- In many cases, the switch may need to drop frames in order to realize congestion control techniques, and needs to know which frames can be dropped with minimal impact to video quality.
- For scalable streams with dependent layers, the switch may need to selectively forward specific layers to specific recipients due to recipient bandwidth or decoder limits.
- Furthermore, it is highly desirable to do this in a payload format-agnostic way which is not specific to each different video codec. Most modern video codecs share common concepts around frame types and other critical information to make this codec-agnostic handling possible.
- It is also desirable to be able to do this for SRTP without requiring the video switch to decrypt the packets. SRTP will encrypt the RTP payload format contents and consequently this data is not usable for the switching function without decryption, which may not even be possible in the case of end-to-end encryption of private media [I-D.ietf-perc-private-media-framework].

By providing meta-information about the RTP streams outside the encrypted media payload, an RTP switch can do codec-agnostic selective forwarding without decrypting the payload. This document specifies the necessary meta-information in an RTP header extension.

2. Key Words for Normative Requirements

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. Frame Marking RTP Header Extension

This specification uses RTP header extensions as defined in [RFC8285]. A subset of meta-information from the video stream is provided as an RTP header extension to allow an RTP switch to do generic selective forwarding of video streams encoded with potentially different video codecs.

The Frame Marking RTP header extension is encoded using the one-byte header or two-byte header as described in [RFC8285]. The one-byte header format is used for examples in this memo. The two-byte header format is used when other two-byte header extensions are present in the same RTP packet, since mixing one-byte and two-byte extensions is not possible in the same RTP packet.

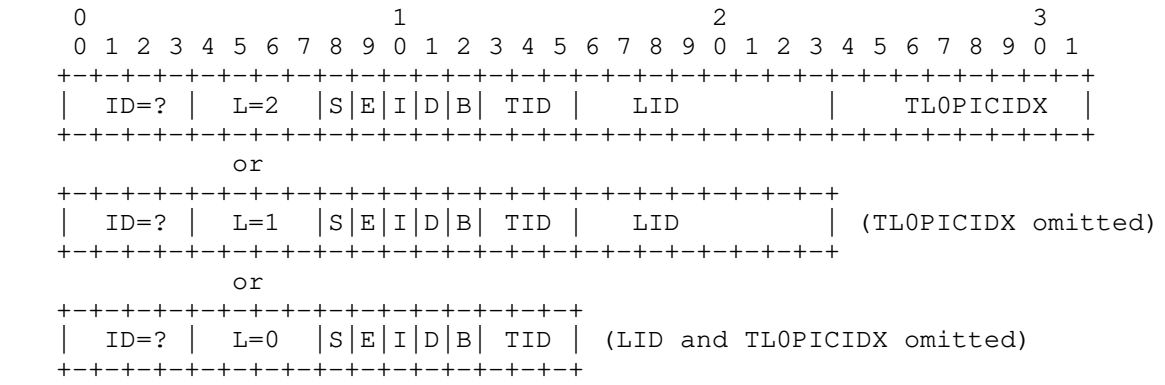
This extension is only specified for Source (not Redundancy) RTP Streams [RFC7656] that carry video payloads. It is not specified for audio payloads, nor is it specified for Redundancy RTP Streams. The (separate) specifications for Redundancy RTP Streams often include provisions for recovering any header extensions that were part of the original source packet. Such provisions SHALL be followed to recover the Frame Marking RTP header extension of the original source packet. Source packet frame markings may be useful when generating Redundancy RTP Streams; for example, the I and D bits can be used to generate extra or no redundancy, respectively, and redundancy schemes with source blocks can align source block boundaries with Independent frame boundaries as marked by the I bit.

A frame, in the context of this specification, is the set of RTP packets with the same RTP timestamp from a specific RTP synchronization source (SSRC). A frame within a layer is the set of RTP packets with the same RTP timestamp, SSRC, Temporal ID (TID), and Layer ID (LID).

3.1. Long Extension for Scalable Streams

The following RTP header extension is RECOMMENDED for scalable streams. It MAY also be used for non-scalable streams, in which case TID, LID and TL0PICIDX MUST be 0 or omitted. The ID is assigned per [RFC8285], and the length is encoded as L=2 which indicates 3 octets of data when nothing is omitted, or L=1 for 2 octets when TL0PICIDX

is omitted, or L=0 for 1 octet when both LID and TL0PICIDX are omitted.



The following information are extracted from the media payload and sent in the Frame Marking RTP header extension.

- S: Start of Frame (1 bit) - MUST be 1 in the first packet in a frame within a layer; otherwise MUST be 0.
- E: End of Frame (1 bit) - MUST be 1 in the last packet in a frame within a layer; otherwise MUST be 0. Note that the RTP header marker bit MAY be used to infer the last packet of the highest enhancement layer, in payload formats with such semantics.
- I: Independent Frame (1 bit) - MUST be 1 for a frame within a layer that can be decoded independent of temporally prior frames, e.g. intra-frame, VPX keyframe, H.264 IDR [RFC6184], H.265 IDR/CRA/BLA/RAP [RFC7798]; otherwise MUST be 0. Note that this bit only signals temporal independence, so it can be 1 in spatial or quality enhancement layers that depend on temporally co-located layers but not temporally prior frames.
- D: Discardable Frame (1 bit) - MUST be 1 for a frame within a layer the sender knows can be discarded, and still provide a decodable media stream; otherwise MUST be 0.
- B: Base Layer Sync (1 bit) - When TID is not 0, this MUST be 1 if the sender knows this frame within a layer only depends on the base temporal layer; otherwise MUST be 0. When TID is 0 or if no scalability is used, this MUST be 0.
- TID: Temporal ID (3 bits) - Identifies the temporal layer/sub-layer encoded, starting with 0 for the base layer, and increasing with higher temporal fidelity. If no scalability is used, this MUST be 0. It is implicitly 0 in the short extension format.
- LID: Layer ID (8 bits) - Identifies the spatial and quality layer encoded, starting with 0 for the base layer, and increasing with higher fidelity. If no scalability is used, this MUST be 0 or omitted to reduce length. When omitted, TL0PICIDX MUST also be

omitted. It is implicitly 0 in the short extension format or when omitted in the long extension format.

- o TL0PICIDX: Temporal Layer 0 Picture Index (8 bits) - When TID is 0 and LID is 0, this is a cyclic counter labeling base layer frames. When TID is not 0 or LID is not 0, this indicates a dependency on the given index, such that this frame within this layer depends on the frame with this label in the layer with TID 0 and LID 0. If no scalability is used, or the cyclic counter is unknown, this MUST be omitted to reduce length. Note that 0 is a valid index value for TL0PICIDX.

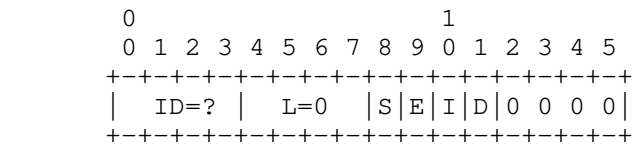
The layer information contained in TID and LID convey useful aspects of the layer structure that can be utilized in selective forwarding.

Without further information about the layer structure, these TID/LID identifiers can only be used for relative priority of layers and implicit dependencies between layers. They convey a layer hierarchy with TID=0 and LID=0 identifying the base layer. Higher values of TID identify higher temporal layers with higher frame rates. Higher values of LID identify higher spatial and/or quality layers with higher resolutions and/or bitrates. Implicit dependencies between layers assume that a layer with a given TID/LID MAY depend on layer(s) with the same or lower TID/LID, but MUST NOT depend on layer(s) with higher TID/LID.

With further information, for example, possible future RTCP SDES items that convey full layer structure information, it may be possible to map these TIDs and LIDs to specific absolute frame rates, resolutions and bitrates, as well as explicit dependencies between layers. Such additional layer information may be useful for forwarding decisions in the RTP switch, but is beyond the scope of this memo. The relative layer information is still useful for many selective forwarding decisions even without such additional layer information.

3.2. Short Extension for Non-Scalable Streams

The following RTP header extension is RECOMMENDED for non-scalable streams. It is identical to the shortest form of the extension for scalable streams, except the last four bits (B and TID) are replaced with zeros. It MAY also be used for scalable streams if the sender has limited or no information about stream scalability. The ID is assigned per [RFC8285], and the length is encoded as L=0 which indicates 1 octet of data.



The following information are extracted from the media payload and sent in the Frame Marking RTP header extension.

- o S: Start of Frame (1 bit) - MUST be 1 in the first packet in a frame; otherwise MUST be 0.
- o E: End of Frame (1 bit) - MUST be 1 in the last packet in a frame; otherwise MUST be 0. SHOULD match the RTP header marker bit in payload formats with such semantics for marking end of frame.
- o I: Independent Frame (1 bit) - MUST be 1 for frames that can be decoded independent of temporally prior frames, e.g. intra-frame, VPX keyframe, H.264 IDR [RFC6184], H.265 IDR/CRA/BLA/IRAP [RFC7798]; otherwise MUST be 0.
- o D: Discardable Frame (1 bit) - MUST be 1 for frames the sender knows can be discarded, and still provide a decodable media stream; otherwise MUST be 0.
- o The remaining (4 bits) - are reserved/fixed values and not used for non-scalable streams; they MUST be set to 0 upon transmission and ignored upon reception.

3.3. Layer ID Mappings for Scalable Streams

This section maps the specific Layer ID information contained in specific scalable codecs to the generic LID and TID fields.

Note that non-scalable streams have no Layer ID information and thus no mappings.

3.3.1. H265 LID Mapping

The following shows the H265 [RFC7798] LayerID (6 bits) and TID (3 bits) from the NAL unit header mapped to the generic LID and TID fields.

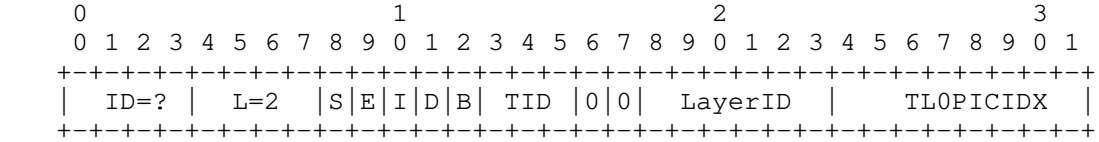
The S and E bits MUST match the correspondingly named bits in PACI:PHES:TSCI payload structures.

The I bit MUST be 1 when the NAL unit type is 16-23 (inclusive) or 32-34 (inclusive), or an aggregation packet or fragmentation unit encapsulating any of these types, otherwise it MUST be 0. These

ranges cover intra (IRAP) frames as well as critical parameter sets (VPS, SPS, PPS).

The D bit MUST be 1 when the NAL unit type is 0, 2, 4, 6, 8, 10, 12, 14, or 38, or an aggregation packet or fragmentation unit encapsulating only these types, otherwise it MUST be 0. These ranges cover non-reference frames as well as filler data.

The B bit can not be determined reliably from simple inspection of payload headers, and therefore is determined by implementation-specific means. For example, internal codec interfaces may provide information to set this reliably.



3.3.2. H264-SVC LID Mapping

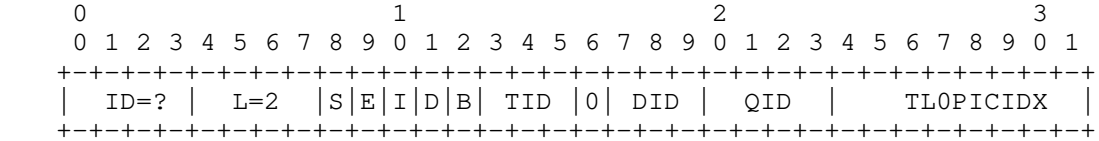
The following shows H264-SVC [RFC6190] Layer encoding information (3 bits for spatial/dependency layer, 4 bits for quality layer and 3 bits for temporal layer) mapped to the generic LID and TID fields.

The S, E, I and D bits MUST match the correspondingly named bits in PACSI payload structures.

The I bit MUST be 1 when the NAL unit type is 5, 7, 8, 13, or 15, or an aggregation packet or fragmentation unit encapsulating any of these types, otherwise it MUST be 0. These ranges cover intra (IDR) frames as well as critical parameter sets (SPS/PPS variants).

The D bit MUST be 1 when the NAL unit header NRI field is 0, or an aggregation packet or fragmentation unit encapsulating only NAL units with NRI=0, otherwise it MUST be 0. The NRI=0 condition signals non-reference frames.

The B bit can not be determined reliably from simple inspection of payload headers, and therefore is determined by implementation-specific means. For example, internal codec interfaces may provide information to set this reliably.



3.3.3. H264 (AVC) LID Mapping

The following shows the header extension for H264 (AVC) [RFC6184] that contains only temporal layer information.

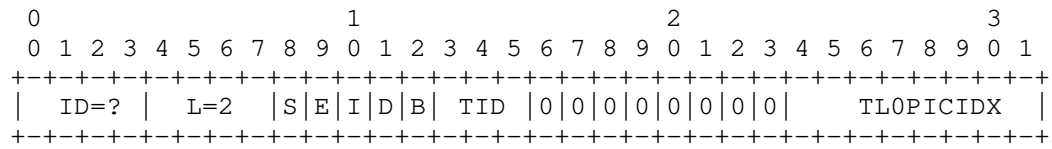
The S bit MUST be 1 when the timestamp in the RTP header differs from the timestamp in the prior RTP sequence number from the same SSRC, otherwise it MUST be 0.

The E bit MUST match the M bit in the RTP header.

The I bit MUST be 1 when the NAL unit type is 5, 7, or 8, or an aggregation packet or fragmentation unit encapsulating any of these types, otherwise it MUST be 0. These ranges cover intra (IDR) frames as well as critical parameter sets (SPS/PPS).

The D bit MUST be 1 when the NAL unit header NRI field is 0, or an aggregation packet or fragmentation unit encapsulating only NAL units with NRI=0, otherwise it MUST be 0. The NRI=0 condition signals non-reference frames.

The B bit can not be determined reliably from simple inspection of payload headers, and therefore is determined by implementation-specific means. For example, internal codec interfaces may provide information to set this reliably.



3.3.4. VP8 LID Mapping

The following shows the header extension for VP8 [RFC7741] that contains only temporal layer information.

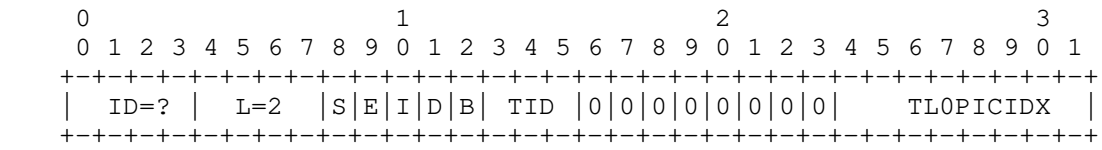
The S bit MUST match the correspondingly named bit in the VP8 payload descriptor when PID=0, otherwise it MUST be 0.

The E bit MUST match the M bit in the RTP header.

The I bit MUST match the inverse of the P bit in the VP8 payload header.

The D bit MUST match the N bit in the VP8 payload descriptor.

The B bit MUST match the Y bit in the VP8 payload descriptor.



3.3.5. Future Codec LID Mapping

The RTP payload format specification for future video codecs SHOULD include a section describing the LID mapping and TID mapping for the codec. For example, the LID/TID mapping for the VP9 codec is described in the VP9 RTP Payload Format [I-D.ietf-payload-vp9].

3.4. Signaling Information

The URI for declaring this header extension in an extmap attribute is "urn:ietf:params:rtp-hdrext:framemarking". It does not contain any extension attributes.

An example attribute line in SDP:

```
a=extmap:3 urn:ietf:params:rtp-hdrext:framemarking
```

3.5. Usage Considerations

The header extension values MUST represent what is already in the RTP payload.

When an RTP switch needs to discard a received video frame due to congestion control considerations, it is RECOMMENDED that it preferably drop frames marked with the D (Discardable) bit set, or the highest values of TID and LID, which indicate the highest temporal and spatial/quality enhancement layers, since those typically have fewer dependenices on them than lower layers.

When an RTP switch wants to forward a new video stream to a receiver, it is RECOMMENDED to select the new video stream from the first switching point with the I (Independent) bit set in all spatial layers and forward the same. An RTP switch can request a media source to generate a switching point by sending Full Intra Request (RTCP FIR) as defined in [RFC5104], for example.

3.5.1. Relation to Layer Refresh Request (LRR)

Receivers can use the Layer Refresh Request (LRR) [I-D.ietf-avtext-lrr] RTCP feedback message to upgrade to a higher layer in scalable encodings. The TID/LID values and formats used in

LRR messages MUST correspond to the same values and formats specified in Section 3.1.

Because frame marking can only be used with temporally-nested streams, temporal-layer LRR refreshes are unnecessary for frame-marked streams. Other refreshes can be detected based on the I bit being set for the specific spatial layers.

3.5.2. Scalability Structures

The LID and TID information is most useful for fixed scalability structures, such as nested hierarchical temporal layering structures, where each temporal layer only references lower temporal layers or the base temporal layer. The LID and TID information is less useful, or even not useful at all, for complex, irregular scalability structures that do not conform to common, fixed patterns of inter-layer dependencies and referencing structures. Therefore it is RECOMMENDED to use LID and TID information for RTP switch forwarding decisions only in the case of temporally nested scalability structures, and it is NOT RECOMMENDED for other (more complex or irregular) scalability structures.

4. Security Considerations

In the Secure Real-Time Transport Protocol (SRTP) [RFC3711], RTP header extensions are authenticated but usually not encrypted. When header extensions are used some of the payload type information are exposed and visible to middle boxes. The encrypted media data is not exposed, so this is not seen as a high risk exposure.

5. Acknowledgements

Many thanks to Bernard Aboba, Jonathan Lennox, Stephan Wenger, Dale Worley, and Magnus Westerlund for their inputs.

6. IANA Considerations

This document defines a new extension URI to the RTP Compact HeaderExtensions sub-registry of the Real-Time Transport Protocol (RTP) Parameters registry, according to the following data:

Extension URI: urn:ietf:params:rtp-hdrext:framemarkinginfo
Description: Frame marking information for video streams
Contact: mzanaty@cisco.com
Reference: RFC XXXX

Note to RFC Editor: please replace RFC XXXX with the number of this RFC.

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC6184] Wang, Y., Even, R., Kristensen, T., and R. Jesup, "RTP Payload Format for H.264 Video", RFC 6184, DOI 10.17487/RFC6184, May 2011, <https://www.rfc-editor.org/info/rfc6184>.

[RFC6190] Wenger, S., Wang, Y., Schierl, T., and A. Eleftheriadis, "RTP Payload Format for Scalable Video Coding", RFC 6190, DOI 10.17487/RFC6190, May 2011, <https://www.rfc-editor.org/info/rfc6190>.

[RFC7741] Westin, P., Lundin, H., Glover, M., Uberti, J., and F. Galligan, "RTP Payload Format for VP8 Video", RFC 7741, DOI 10.17487/RFC7741, March 2016, <https://www.rfc-editor.org/info/rfc7741>.

[RFC7798] Wang, Y., Sanchez, Y., Schierl, T., Wenger, S., and M. Hannuksela, "RTP Payload Format for High Efficiency Video Coding (HEVC)", RFC 7798, DOI 10.17487/RFC7798, March 2016, <https://www.rfc-editor.org/info/rfc7798>.

[RFC8285] Singer, D., Desineni, H., and R. Even, Ed., "A General Mechanism for RTP Header Extensions", RFC 8285, DOI 10.17487/RFC8285, October 2017, <https://www.rfc-editor.org/info/rfc8285>.

7.2. Informative References

[I-D.ietf-avtext-lrr] Lennox, J., Hong, D., Uberti, J., Holmer, S., and M. Flodman, "The Layer Refresh Request (LRR) RTCP Feedback Message", draft-ietf-avtext-lrr-07 (work in progress), July 2017.

[I-D.ietf-payload-vp9] Uberti, J., Holmer, S., Flodman, M., Hong, D., and J. Lennox, "RTP Payload Format for VP9 Video", draft-ietf-payload-vp9-10 (work in progress), July 2020.

[I-D.ietf-perc-private-media-framework]
Jones, P., Benham, D., and C. Groves, "A Solution Framework for Private Media in Privacy Enhanced RTP Conferencing (PERC)", draft-ietf-perc-private-media-framework-12 (work in progress), June 2019.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <https://www.rfc-editor.org/info/rfc3550>.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, DOI 10.17487/RFC3711, March 2004, <https://www.rfc-editor.org/info/rfc3711>.

[RFC5104] Wenger, S., Chandra, U., Westerlund, M., and B. Burman, "Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)", RFC 5104, DOI 10.17487/RFC5104, February 2008, <https://www.rfc-editor.org/info/rfc5104>.

[RFC6464] Lennox, J., Ed., Ivov, E., and E. Marocco, "A Real-time Transport Protocol (RTP) Header Extension for Client-to-Mixer Audio Level Indication", RFC 6464, DOI 10.17487/RFC6464, December 2011, <https://www.rfc-editor.org/info/rfc6464>.

[RFC7656] Lennox, J., Gross, K., Nandakumar, S., Salgueiro, G., and B. Burman, Ed., "A Taxonomy of Semantics and Mechanisms for Real-Time Transport Protocol (RTP) Sources", RFC 7656, DOI 10.17487/RFC7656, November 2015, <https://www.rfc-editor.org/info/rfc7656>.

[RFC7667] Westerlund, M. and S. Wenger, "RTP Topologies", RFC 7667, DOI 10.17487/RFC7667, November 2015, <https://www.rfc-editor.org/info/rfc7667>.

Authors' Addresses

Mo Zanaty
Cisco Systems
170 West Tasman Drive
San Jose, CA 95134
US

Email: mzanaty@cisco.com

Espen Berger
Cisco Systems

Phone: +47 98228179
Email: espeberg@cisco.com

Suhas Nandakumar
Cisco Systems
170 West Tasman Drive
San Jose, CA 95134
US

Email: snandaku@cisco.com

avtcore
Internet-Draft
Intended status: Standards Track
Expires: May 31, 2021

S. Lugan
intoPIX
A. Descampe
UCL
C. Damman
intoPIX
T. Richter
IIS
T. Bruylants
intoPIX

November 27, 2020

RTP Payload Format for ISO/IEC 21122 (JPEG XS)
draft-ietf-payload-rtp-jpegxs-07

Abstract

This document specifies a Real-Time Transport Protocol (RTP) payload format to be used for transporting JPEG XS (ISO/IEC 21122) encoded video. JPEG XS is a low-latency, lightweight image coding system. Compared to an uncompressed video use case, it allows higher resolutions and frame rates, while offering visually lossless quality, reduced power consumption, and end-to-end latency confined to a fraction of a frame.

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 https://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 31, 2021.

Copyright Notice

Table of Contents

1. Introduction

This document specifies a payload format for packetization of JPEG XS [ISO21122-1] encoded video signals into the Real-time Transport Protocol (RTP) [RFC3550].

The JPEG XS coding system offers compression and recompression of image sequences with very moderate computational resources while remaining robust under multiple compression and decompression cycles and mixing of content sources, e.g. embedding of subtitles, overlays or logos. Typical target compression ratios ensuring visually lossless quality are in the range of 2:1 to 10:1, depending on the nature of the source material. The end-to-end latency can be confined to a fraction of a frame, typically between a small number of lines down to below a single line.

2. Conventions, Definitions, and Abbreviations

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 [RFC2119].

Application Data Unit (ADU)
: The unit of source data provided as payload to the transport layer, and corresponding, in this RTP payload definition, to a single JPEG XS frame.

Colour specification box (CS box)
: A ISO colour specification box defined in ISO/IEC 21122-3 [ISO21122-3] that includes colour-related metadata required to correctly display JPEG XS frames, such as colour primaries, transfer characteristics and matrix coefficients.

EOC marker
: A marker that consists of the two bytes 0xff11 indicating the end of a JPEG XS codestream.

JPEG XS codestream
: A sequence of bytes representing a compressed image formatted according to JPEG XS Part-1 [ISO21122-1].

JPEG XS codestream header
: A sequence of bytes, starting with a SOC marker, at the beginning of each JPEG XS codestream encoded in multiple markers and marker segments that does not carry entropy coded data, but metadata such as the frame dimension and component precision.

JPEG XS frame
: A JPEG XS picture segment in the case of a progressive frame, or, in the case of an interlaced frame, the concatenation of two JPEG XS picture segments.

JPEG XS header segment

The concatenation of a video support box, as defined in ISO/IEC 21122-3 [ISO21122-3], a colour specification box, as defined in ISO/IEC 21122-3 as well [ISO21122-3] and a JPEG XS codestream header.

JPEG XS picture segment
The concatenation of a video support box, as defined in ISO/IEC 21122-3 [ISO21122-3], a colour specification box, as defined in ISO/IEC 21122-3 as well [ISO21122-3] and a JPEG XS codestream.

JPEG XS stream
A sequence of JPEG XS frames.

Marker
A two-byte functional sequence that is part of a JPEG XS codestream starting with a 0xff byte and a subsequent byte defining its function.

Marker segment
A marker along with a 16-bit marker size and payload data following the size.

Packetization unit
A portion of an Application Data Unit whose boundaries coincide with boundaries of RTP packet payloads (excluding payload header), i.e. the first (resp. last) byte of a packetization unit is the first (resp. last) byte of a RTP packet payload (excluding its payload header).

Slice
The smallest independently decodable unit of a JPEG XS codestream, bearing in mind that it decodes to wavelet coefficients which still require inverse wavelet filtering to give an image.

SOC marker
A marker that consists of the two bytes 0xff10 indicating the start of a JPEG XS codestream.

Video support box (VS box)
A ISO video support box defined in ISO/IEC 21122-3 [ISO21122-3] that includes metadata required to play back a JPEG XS stream, such as its maximum bitrate, its subsampling structure, its buffer model and its frame rate.

3. Media Format Description

3.1. Image Data Structures

JPEG XS is a low-latency lightweight image coding system for coding continuous-tone grayscale or continuous-tone colour digital images.

This coding system provides an efficient representation of image signals through the mathematical tool of wavelet analysis. The wavelet filter process separates each component into multiple bands, where each band consists of multiple coefficients describing the image signal of a given component within a frequency domain specific to the wavelet filter type, i.e. the particular filter corresponding to the band.

Wavelet coefficients are grouped into precincts, where each precinct includes all coefficients over all bands that contribute to a spatial region of the image.

One or multiple precincts are furthermore combined into slices consisting of an integer number of precincts. Precincts do not cross slice boundaries, and wavelet coefficients in precincts that are part of different slices can be decoded independently from each other. Note, however, that the wavelet transformation runs across slice boundaries. A slice always extends over the full width of the image, but may only cover parts of its height.

3.2. Codestream

A JPEG XS codestream header, followed by several slices, and terminated by an EOC marker form a JPEG XS codestream.

The overall codestream format, including the definition of all markers, is further defined in ISO/IEC 21122-1 [ISO21122-1]. It represents sample values of a single image, bare any interpretation relative to a colour space.

3.3. Video support box and colour specification box

While the information defined in the codestream is sufficient to reconstruct the sample values of one image, the interpretation of the samples remains undefined by the codestream itself. This interpretation is given by the video support box and the colour specification box which contain significant information to correctly play the JPEG XS stream. The layout and syntax of these boxes, together with their content, are defined in ISO/IEC 21122-3 [ISO21122-3]. The video support box provides information on the maximum bitrate, the frame rate, the frame mode (progressive or

interlaced), the subsampling image format, the timecode of the current JPEG XS frame, the profile, level and sublevel used (as defined in ISO/IEC 21122-2 [ISO21122-2]), and optionally on the buffer model and the mastering display metadata. The colour specification box indicates the colour primaries, transfer characteristics, matrix coefficients and video full range flag needed to specify the colour space of the video stream.

3.4. JPEG XS Frame

The concatenation of a video support box, a colour specification box, and a JPEG XS codestream forms a JPEG XS picture segment.

In the case of a progressive video stream, each JPEG XS frame consists of one single JPEG XS picture segment.

In the case of an interlaced video stream, each JPEG XS frame is made of two concatenated JPEG XS picture segments. The codestream of each segment corresponds exclusively to one of the two fields of the interlaced frame. Both picture segments SHALL contain identical boxes (i.e. concatenation of the video support box and the colour specification box is byte exact the same for both picture segments).

Note that the interlaced mode as signaled by the frat field in the video support box indicates either progressive, interlaced top-field first, or interlaced bottom-field first mode. Thus, its value too SHALL be identical in both picture segments.

4. RTP Payload Format

This section specifies the payload format for JPEG XS streams over the Real-time Transport Protocol (RTP) [RFC3550].

In order to be transported over RTP, each JPEG XS stream is transported in a distinct RTP stream, identified by a distinct SSRC.

A JPEG XS stream is divided into Application Data Units (ADUs), each ADU corresponding to a single JPEG XS frame.

4.1. RTP packetization

An ADU is made of several packetization units. If a packetization unit is bigger than the maximum size of a RTP packet payload, the unit is split into multiple RTP packet payloads, as illustrated in Figure 1. As seen there, each packet SHALL contain (part of) one and only one packetization unit. A packetization unit may extend over multiple packets. The payload of every packet SHALL have the same size (based e.g. on the Maximum Transfer Unit of the network), except

(possibly) the last packet of a packetization unit. The boundaries of a packetization unit SHALL coincide with the boundaries of the payload of a packet (excluding the payload header), i.e. the first (resp. last) byte of the packetization unit SHALL be the first (resp. last) byte of the payload (excluding its header).

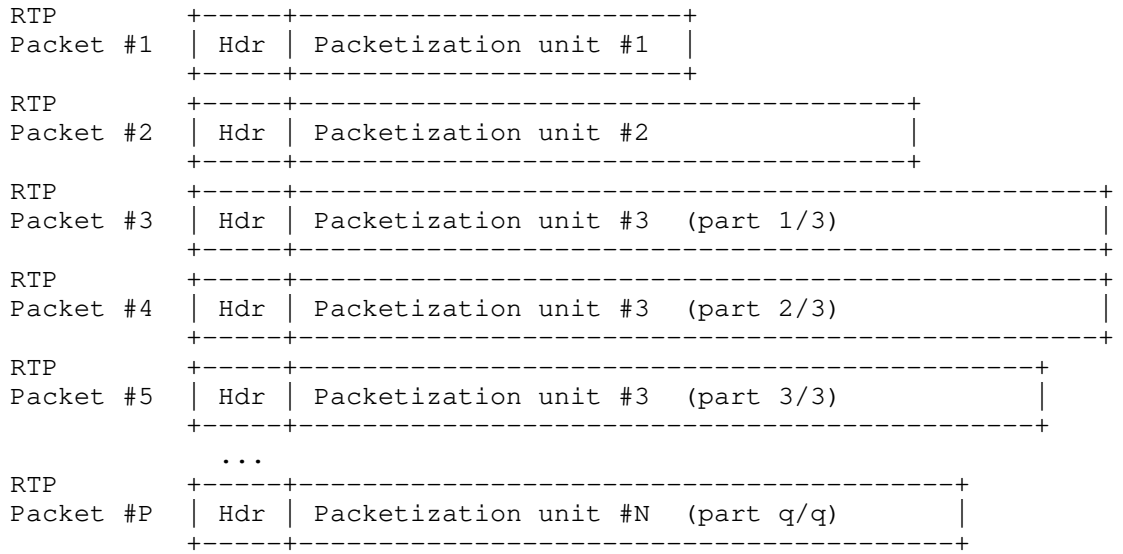


Figure 1: Example of ADU packetization

There are two different packetization modes defined for this RTP payload format.

1. Codestream packetization mode: in this mode, the packetization unit SHALL be the entire JPEG XS picture segment (i.e. codestream preceeded by boxes). This means that a progressive frame will have a single packetization unit, while an interlaced frame will have two. The progressive case is illustrated in Figure 2.

2. Slice packetization mode: in this mode, the packetization unit SHALL be the slice, i.e. there SHALL be data from no more than one slice per RTP packet. The first packetization unit SHALL be made of the JPEG XS header segment (i.e. the concatenation of the VS box, the CS box and the JPEG XS codestream header). This first unit is then followed by successive units, each containing one and only one slice. The packetization unit containing the last slice of a JPEG XS codestream SHALL also contain the EOC marker immediately following this last slice. This is illustrated in Figure 3. In the case of an interlaced frame, the

JPEG XS header segment of the second field SHALL be in its own packetization unit.

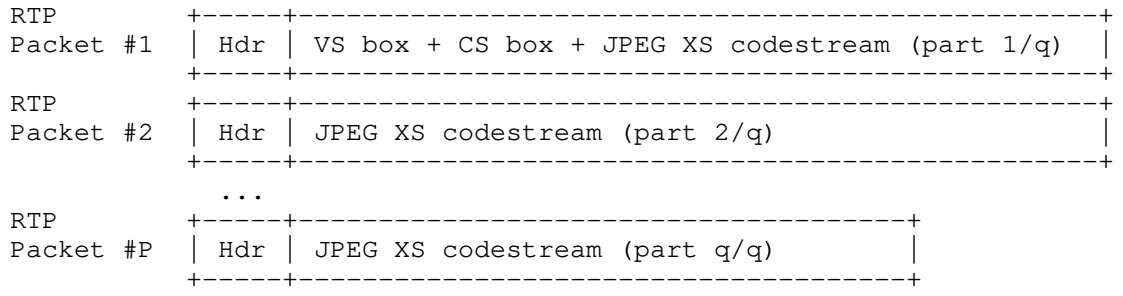


Figure 2: Example of codestream packetization mode

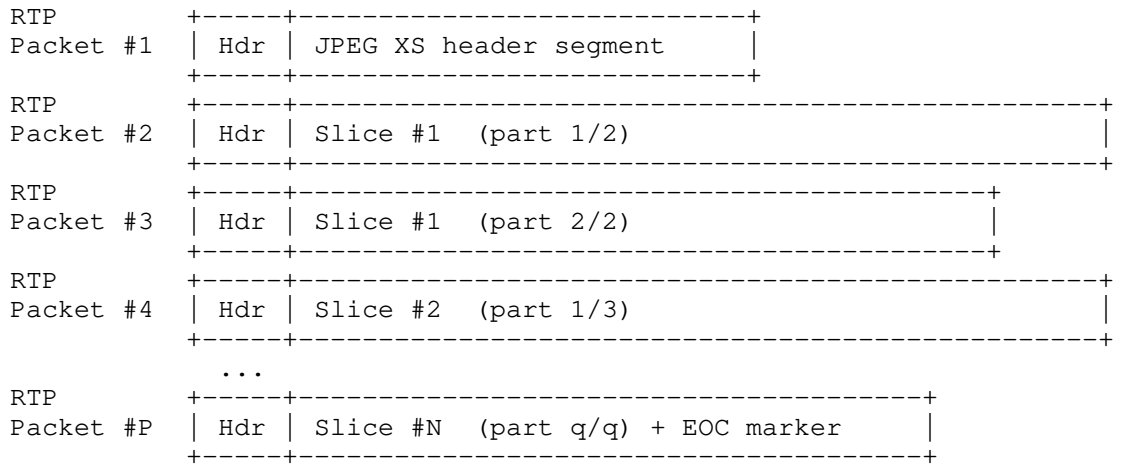


Figure 3: Example of slice packetization mode

Due to the constant bit-rate of JPEG XS, the codestream packetization mode guarantees that a JPEG XS RTP stream will produce a constant number of bytes per frame, and a constant number of RTP packets per frame. To reach the same guarantee with the slice packetization mode, an additional mechanism is required. This can involve a constraint at the rate allocation stage in the JPEG XS encoder to impose a constant bit-rate at the slice level, the usage of padding data, or the insertion of empty RTP packets (i.e. a RTP packet whose payload data is empty).

4.2. RTP Header Usage

The format of the RTP header is specified in RFC 3550 [RFC3550] and reprinted in Figure 4 for convenience. This RTP payload format uses the fields of the header in a manner consistent with that specification.

The RTP payload (and the settings for some RTP header bits) for packetization units are specified in Section 4.3.

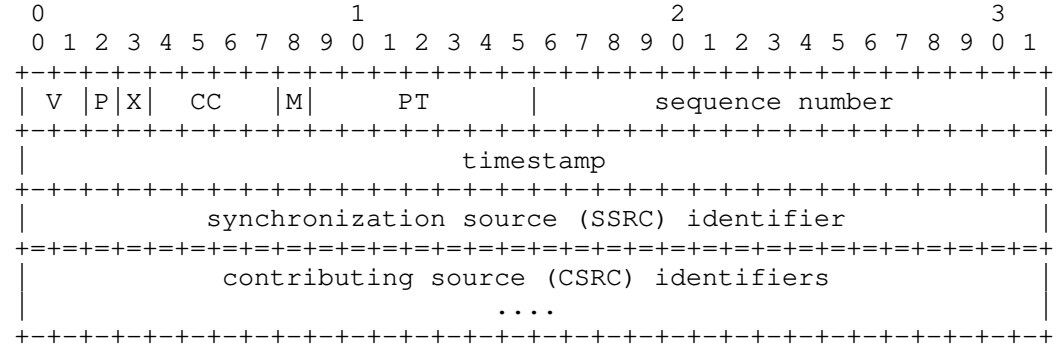


Figure 4: RTP header according to RFC 3550

The version (V), padding (P), extension (X), CSRC count (CC), sequence number, synchronization source (SSRC) and contributing source (CSRC) fields follow their respective definitions in RFC 3550 [RFC3550].

The remaining RTP header information to be set according to this RTP payload format is set as follows:

Marker (M) [1 bit]:

If progressive scan video is being transmitted, the marker bit denotes the end of a video frame. If interlaced video is being transmitted, it denotes the end of the field. The marker bit SHALL be set to 1 for the last packet of the video frame/field. It SHALL be set to 0 for all other packets.

Payload Type (PT) [7 bits]:

A dynamically allocated payload type field that designates the payload as JPEG XS video.

Timestamp [32 bits]:

The RTP timestamp is set to the sampling timestamp of the content. A 90 kHz clock rate SHOULD be used.

As per specified in RFC 3550 [RFC3550] and RFC 4175 [RFC4175], the RTP timestamp designates the sampling instant of the first octet of the frame to which the RTP packet belongs. Packets SHALL not include data from multiple frames, and all packets belonging to the same frame SHALL have the same timestamp. Several successive RTP packets will consequently have equal timestamps if they belong to the same frame (that is until the marker bit is set to 1, marking the last packet of the frame), and the timestamp is only increased when a new frame begins.

If the sampling instant does not correspond to an integer value of the clock, the value SHALL be truncated to the next lowest integer, with no ambiguity.

4.3. Payload Header Usage

The first four bytes of the payload of an RTP packet in this RTP payload format are referred to as the payload header. Figure 5 illustrates the structure of this payload header.

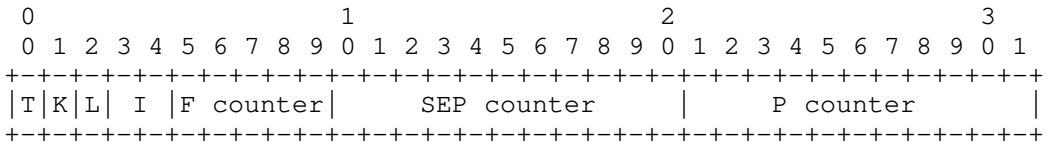


Figure 5: Payload header

The payload header consists of the following fields:

Transmission mode (T) [1 bit]:

The T bit is set to indicate that packets are sent sequentially by the transmitter. This information allows a receiver to dimension its input buffer(s) accordingly. If T=0, nothing can be assumed about the transmission order and packets may be sent out-of-order by the transmitter. If T=1, packets SHALL be sent sequentially by the transmitter.

pacKetization mode (K) [1 bit]:

The K bit is set to indicate which packetization mode is used. K=0 indicates codestream packetization mode, while K=1 indicates slice packetization mode. In the case that the Transmission mode

(T) is set to 0, the slice packetization mode SHALL be used and K SHALL be set to 1.

Last (L) [1 bit]:

The L bit is set to indicate the last packet of a packetization unit. As the end of the frame also ends the packet containing the last unit of the frame, the L bit is set whenever the M bit is set. If codestream packetization mode is used, L bit and M bit are equivalent.

Interlaced information (I) [2 bit]:

These 2 bits are used to indicate how the JPEG XS frame is scanned (progressive or interlaced). In case of an interlaced frame, they also indicate which JPEG XS picture segment the payload is part of (first or second).

00: The payload is progressively scanned.

01: Reserved for future use.

10: The payload is part of the first JPEG XS picture segment of an interlaced video frame. The height specified in the included JPEG XS codestream header is half of the height of the entire displayed image.

11: The payload is part of the second JPEG XS picture segment of an interlaced video frame. The height specified in the included JPEG XS codestream header is half of the height of the entire displayed image.

F counter [5 bits]:

The frame (F) counter identifies the frame number modulo 32 to which a packet belongs. Frame numbers are incremented by 1 for each frame transmitted. The frame number, in addition to the timestamp, may help the decoder manage its input buffer and bring packets back into their natural order.

SEP counter [11 bits]:

The Slice and Extended Packet (SEP) counter is used differently depending on the packetization mode.

* In the case of codestream packetization mode (K=0), this counter resets whenever the Packet counter resets (see

hereunder), and increments by 1 whenever the Packet counter overruns.

* In the case of slice packetization mode (K=1), this counter identifies the slice modulo 2047 to which the packet contributes. If the data belongs to the JPEG XS header segment, this field SHALL have its maximal value, namely 2047 = 0x07ff. Otherwise, it is the slice index modulo 2047. Slice indices are counted from 0 (corresponding to the top of the frame).

P counter [11 bits]:

The packet (P) counter identifies the packet number modulo 2048 within the current packetization unit. It is set to 0 at the start of the packetization unit and incremented by 1 for every subsequent packet (if any) belonging to the same unit. Practically, if codestream packetization mode is enabled, this field counts the packets within a JPEG XS picture segment and is extended by the SEP counter when it overruns. If slice packetization mode is enabled, this field counts the packets within a slice or within the JPEG XS header segment.

4.4. Payload Data

The payload data of a JPEG XS RTP stream consists of a concatenation of multiple JPEG XS frames. In the RTP stream, all the JPEG XS frames SHALL contain identical boxes in their picture segment(s) (i.e. concatenation of the video support box and the colour specification box of any picture segment is always byte exact the same for any of the frames).

Each JPEG XS frame is the concatenation of one or more packetization unit(s), as explained in Section 4.1. Figure 6 depicts this layout for a progressive frame in the codestream packetization mode, Figure 7 depicts this layout for an interlaced frame in the codestream packetization mode, Figure 8 depicts this layout for a progressive frame in the slice packetization mode and Figure 9 depicts this layout for an interlaced frame in the slice packetization mode. The Frame counter value is not indicated because the value is constant for all packetization units of a given frame.

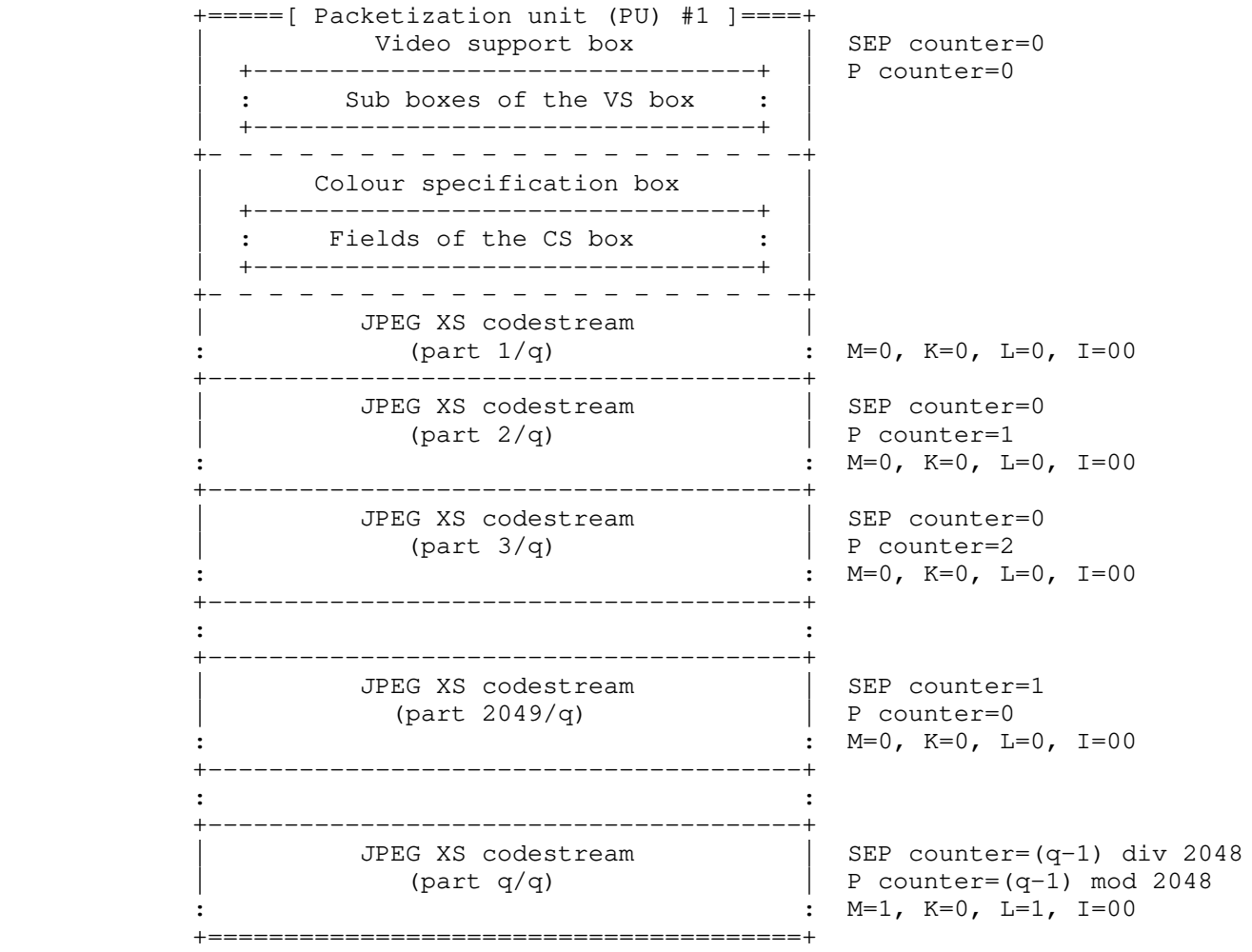


Figure 6: Example of JPEG XS Payload Data (codestream packetization mode, progressive frame)

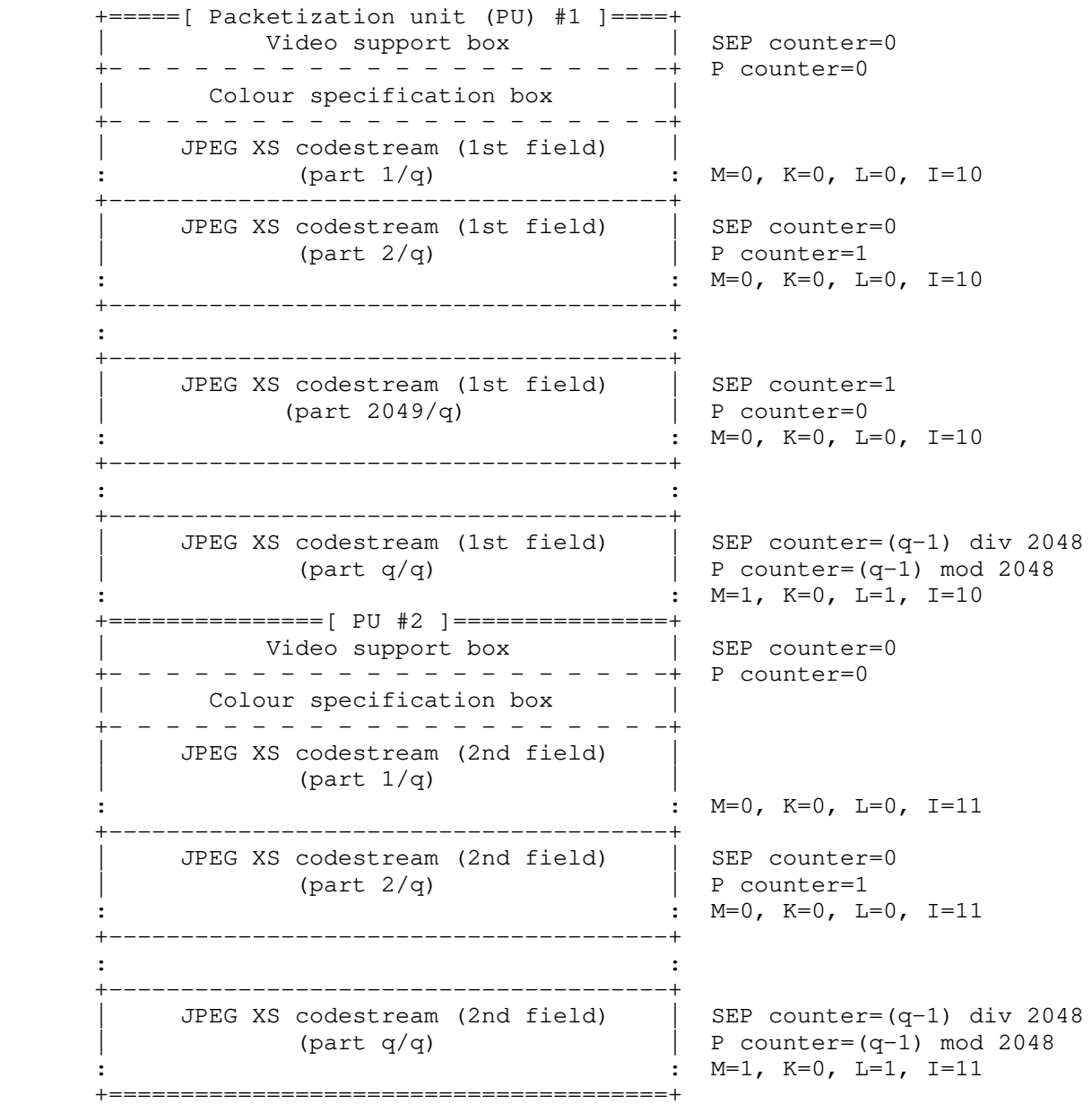


Figure 7: Example of JPEG XS Payload Data (codestream packetization mode, interlaced frame)

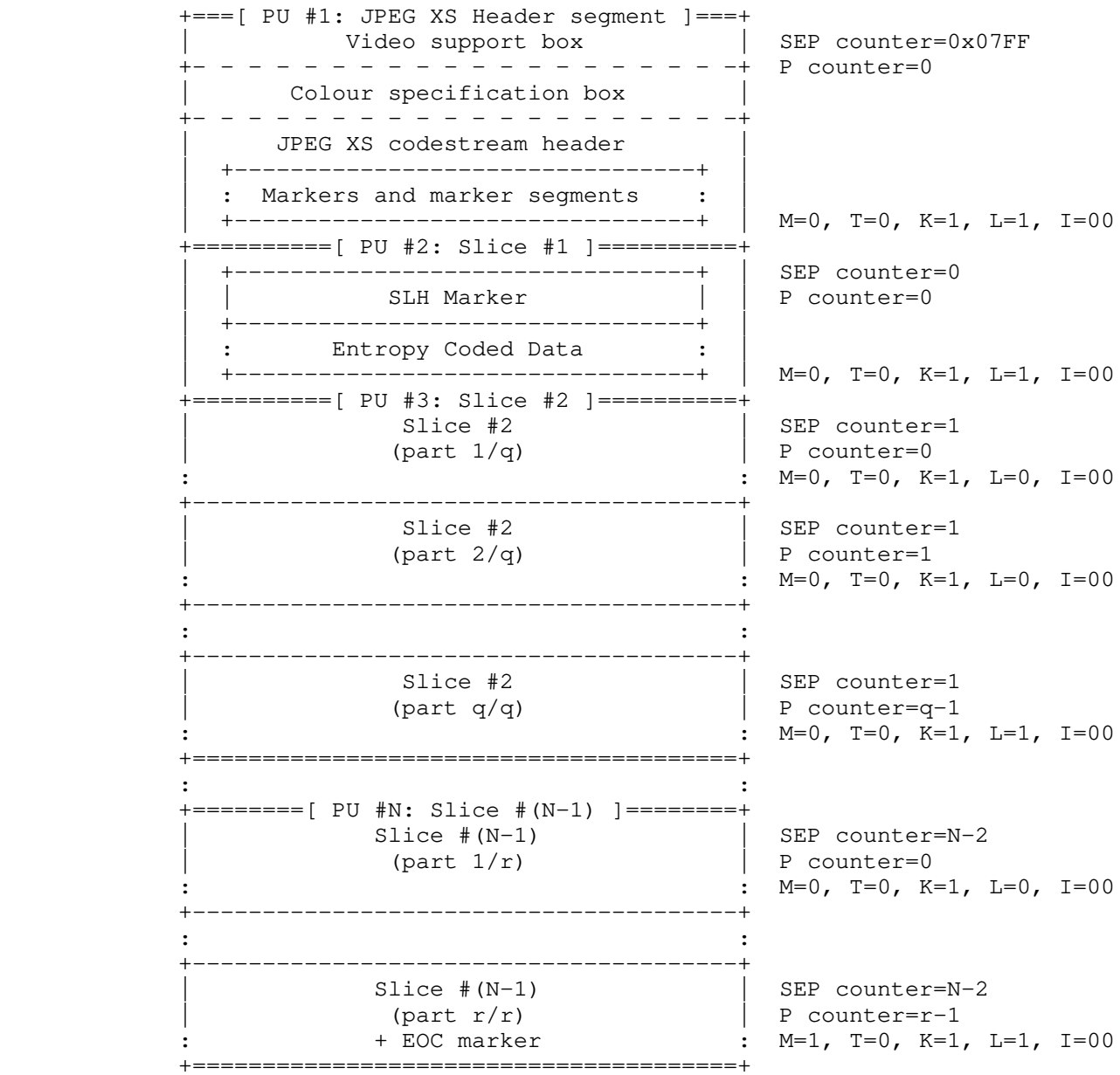
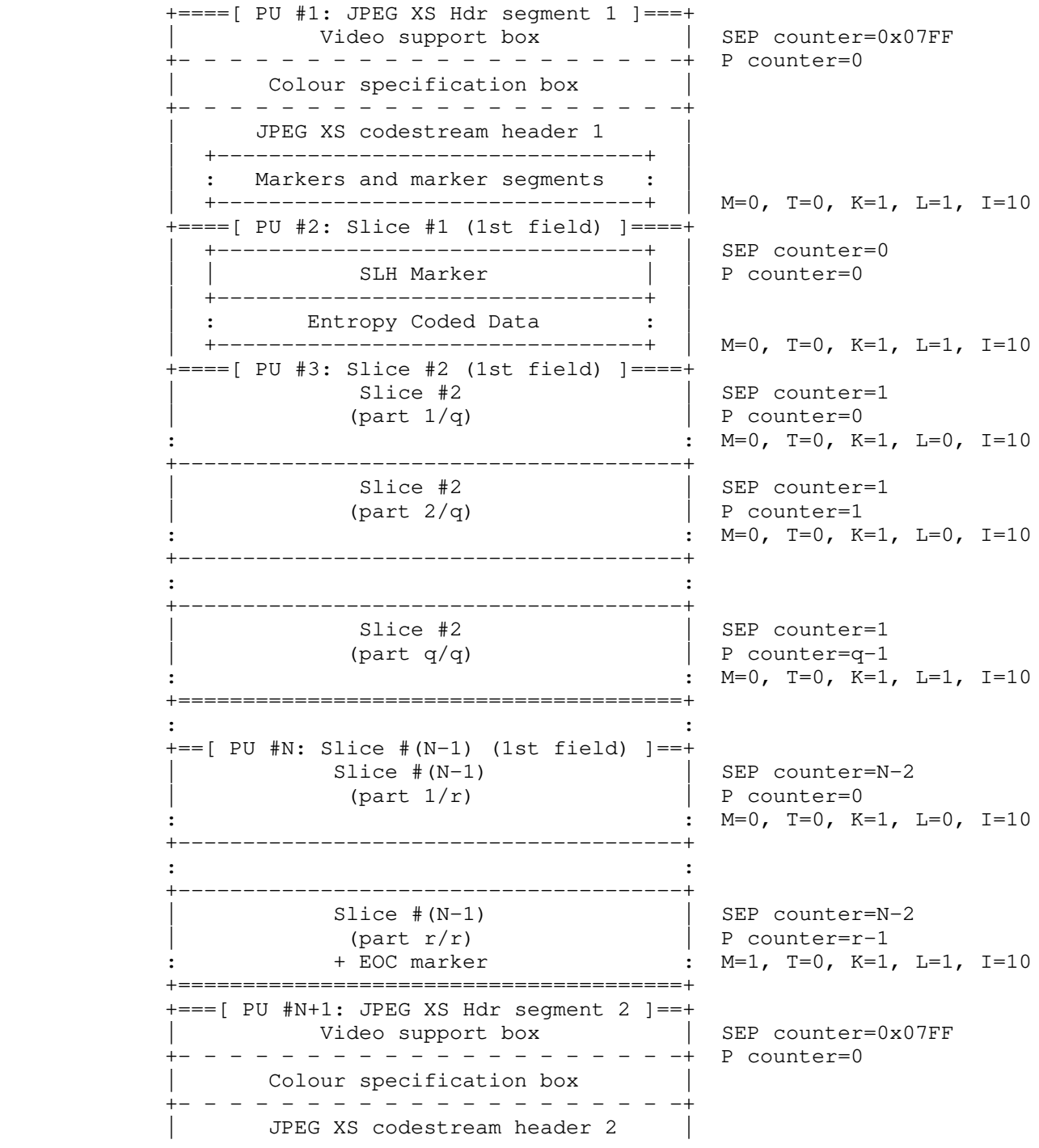


Figure 8: Example of JPEG XS Payload Data (slice packetization mode, progressive frame)


+====[PU #1: JPEG XS Hdr segment 1]===+
Video support box
SEP counter=0x07FF
P counter=0
Colour specification box
JPEG XS codestream header 1
Markers and marker segments
M=0, T=0, K=1, L=1, I=10
+====[PU #2: Slice #1 (1st field)]====+
SLH Marker
SEP counter=0
P counter=0
Entropy Coded Data
M=0, T=0, K=1, L=1, I=10
+====[PU #3: Slice #2 (1st field)]====+
Slice #2
(part 1/q)
SEP counter=1
P counter=0
M=0, T=0, K=1, L=0, I=10
Slice #2
(part 2/q)
SEP counter=1
P counter=1
M=0, T=0, K=1, L=0, I=10
Slice #2
(part q/q)
SEP counter=1
P counter=q-1
M=0, T=0, K=1, L=1, I=10
+==[PU #N: Slice #(N-1) (1st field)]==+
Slice #(N-1)
(part 1/r)
SEP counter=N-2
P counter=0
M=0, T=0, K=1, L=0, I=10
Slice #(N-1)
(part r/r)
+ EOC marker
SEP counter=N-2
P counter=r-1
M=1, T=0, K=1, L=1, I=10
+===[PU #N+1: JPEG XS Hdr segment 2]==+
Video support box
SEP counter=0x07FF
P counter=0
Colour specification box
JPEG XS codestream header 2

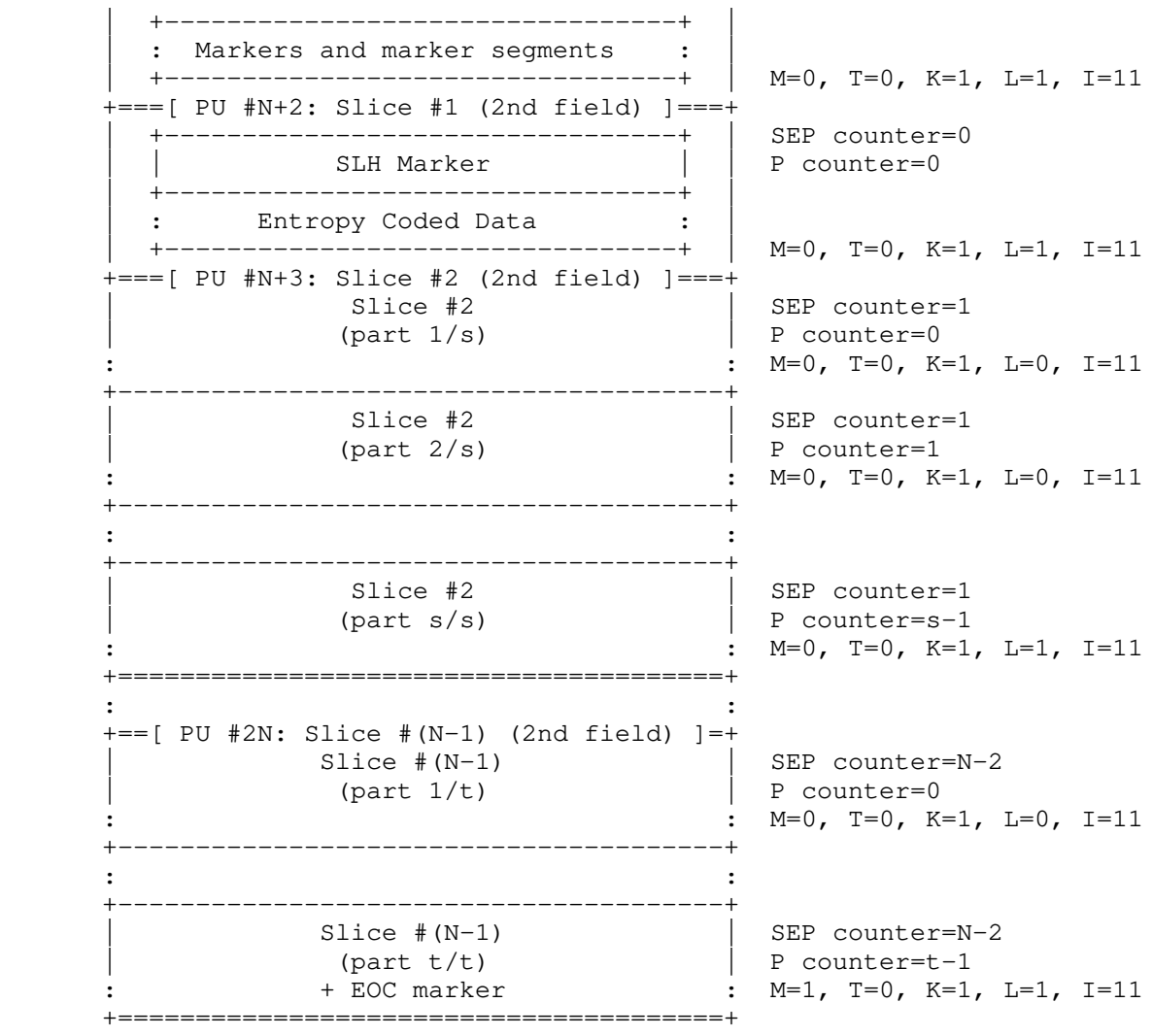


Figure 9: Example of JPEG XS Payload Data (slice packetization mode, interlaced frame)

4.5. Traffic Shaping and Delivery Timing

The traffic shaping and delivery timing SHALL be in accordance with the Network Compatibility Model compliance definitions specified in SMPTE ST 2110-21 [SMPTE-ST2110-21] for either Narrow Linear Senders (Type NL) or Wide Senders (Type W). The session description SHALL include a format-specific parameter of either TP=2110TPNL or

TP=2110TPW to indicate compliance with Type NL or Type W respectively.

NOTE: The Virtual Receiver Buffer Model compliance definitions of ST 2110-21 do not apply.

5. Congestion Control Considerations

Congestion control for RTP SHALL be used in accordance with RFC 3550 [RFC3550], and with any applicable RTP profile: e.g., RFC 3551 [RFC3551]. An additional requirement if best-effort service is being used is users of this payload format SHALL monitor packet loss to ensure that the packet loss rate is within acceptable parameters. Circuit Breakers [RFC8083] is an update to RTP [RFC3550] that defines criteria for when one is required to stop sending RTP Packet Streams and applications implementing this standard SHALL comply with it. RFC 8085 [RFC8085] provides additional information on the best practices for applying congestion control to UDP streams.

6. Payload Format Parameters

6.1. Media Type Definition

Type name: video

Subtype name: jxsv

Required parameters:

rate: The RTP timestamp clock rate. Applications using this payload format SHOULD use a value of 90000.

transmode: This parameter specifies the configured transmission mode as defined by the Transmission mode (T) bit in the payload header of Section 4.3. This value SHALL be equal to the T bit value configured in the RTP stream (i.e. 0 for out-of-order-allowed or 1 for sequential).

Optional parameters:

packetmode: This parameter specifies the configured packetization mode as defined by the packetization mode (K) bit in the payload header of Section 4.3. If specified, this value SHALL be equal to the K bit value configured in the RTP stream (i.e. 0 for codestream or 1 for slice).

profile: The JPEG XS profile in use, as defined in ISO/IEC 21122-2 (JPEG XS Part 2) [ISO21122-2]. Any white space in the

profile name SHALL be replaced by a dash (-). Examples are 'Main-444.12' or 'High-444.12'.

level: The JPEG XS level in use, as defined in ISO/IEC 21122-2 (JPEG XS Part 2) [ISO21122-2]. Any white space in the level name SHALL be replaced by a dash (-). Examples are '2k-1' or '4k-2'.

sublevel: The JPEG XS sublevel in use, as defined in ISO/IEC 21122-2 (JPEG XS Part 2) [ISO21122-2]. Any white space in the sublevel name SHALL be replaced by a dash (-). Examples are 'Sublev3bpp' or 'Sublev6bpp'.

depth: Determines the number of bits per sample. This is an integer with typical values including 8, 10, 12, and 16.

width: Determines the number of pixels per line. This is an integer between 1 and 32767.

height: Determines the number of lines per frame. This is an integer between 1 and 32767.

exactframerate: Signals the frame rate in frames per second. Integer frame rates SHALL be signaled as a single decimal number (e.g. "25") whilst non-integer frame rates SHALL be signaled as a ratio of two integer decimal numbers separated by a "forward-slash" character (e.g. "30000/1001"), utilizing the numerically smallest numerator value possible.

interlaced: If this parameter name is present, it indicates that the video is interlaced, or that the video is Progressive segmented Frame (PsF). If this parameter name is not present, the progressive video format SHALL be assumed.

segmented: If this parameter name is present, and the interlace parameter name is also present, then the video is a Progressive segmented Frame (PsF). Signaling of this parameter without the interlace parameter is forbidden.

sampling: Signals the colour difference signal sub-sampling structure.

Signals utilizing the non-constant luminance Y'C'B C'R signal format of Recommendation ITU-R BT.601-7, Recommendation ITU-R BT.709-6, Recommendation ITU-R BT.2020-2, or Recommendation ITU-R BT.2100 SHALL use the appropriate one of the following values for the Media Type Parameter "sampling":

```
   YCbCr-4:4:4    (4:4:4 sampling)
   YCbCr-4:2:2    (4:2:2 sampling)
   YCbCr-4:2:0    (4:2:0 sampling)
```

Signals utilizing the Constant Luminance Y'C C'BC C'RC signal format of Recommendation ITU-R BT.2020-2 SHALL use the appropriate one of the following values for the Media Type Parameter "sampling":

```
   CLYCbCr-4:4:4  (4:4:4 sampling)
   CLYCbCr-4:2:2  (4:2:2 sampling)
   CLYCbCr-4:2:0  (4:2:0 sampling)
```

Signals utilizing the constant intensity I CT CP signal format of Recommendation ITU-R BT.2100 SHALL use the appropriate one of the following values for the Media Type Parameter "sampling":

```
   ICtCp-4:4:4    (4:4:4 sampling)
   ICtCp-4:2:2    (4:2:2 sampling)
   ICtCp-4:2:0    (4:2:0 sampling)
```

Signals utilizing the 4:4:4 R' G' B' or RGB signal format (such as that of Recommendation ITU-R BT.601, Recommendation ITU-R BT.709, Recommendation ITU-R BT.2020, Recommendation ITU-R BT.2100, SMPTE ST 2065-1 or ST 2065-3) SHALL use the following value for the Media Type Parameter sampling.

```
   RGB            (RGB or R' G' B' samples)
```

Signals utilizing the 4:4:4 X' Y' Z' signal format (such as defined in SMPTE ST 428-1) SHALL use the following value for the Media Type Parameter sampling.

```
   XYZ            (X' Y' Z' samples)
```

Key signals as defined in SMPTE RP 157 SHALL use the value key for the Media Type Parameter sampling. The Key signal is represented as a single component.

```
   KEY            (Samples of the key signal)
```

Signals utilizing a colour sub-sampling other than what is defined here SHALL use the following value for the Media Type Parameter sampling.

```
   UNSPECIFIED    (Sampling signaled by the payload.)
```

colorimetry: Specifies the system colorimetry used by the image samples. Valid values and their specification are:

BT601-5	ITU-R Recommendation BT.601-5.
BT709-2	ITU-R Recommendation BT.709-2.
SMPTE240M	SMPTE ST 240M.
BT601	ITU-R Recommendation BT.601-7.
BT709	ITU-R Recommendation BT.709-6.
BT2020	ITU-R Recommendation BT.2020-2.
BT2100	ITU-R Recommendation BT.2100 Table 2 titled "System colorimetry".
ST2065-1	SMPTE ST 2065-1 Academy Color Encoding Specification (ACES).
ST2065-3	SMPTE ST 2065-3 Academy Density Exchange Encoding (ADX).
XYZ	ISO/IEC 11664-1, section titled "1931 Observer".
UNSPECIFIED	Colorimetry is signaled in the payload by the Color Specification Box of ISO/IEC 21122-3, or it must be manually coordinated between sender and receiver.

Signals utilizing the Recommendation ITU-R BT.2100 colorimetry SHOULD also signal the representational range using the optional parameter RANGE defined below. Signals utilizing the UNSPECIFIED colorimetry might require manual coordination between the sender and the receiver.

TCS: Transfer Characteristic System. This parameter specifies the transfer characteristic system of the image samples. Valid values and their specification are:

SDR	Standard Dynamic Range video streams that utilize the OETF of ITU-R Recommendation BT.709 or ITU-R Recommendation BT.2020. Such streams SHALL be assumed to target the EOTF specified in ITU-R Recommendation BT.1886.
PQ	High dynamic range video streams that utilize the Perceptual Quantization system of ITU-R Recommendation BT.2100.
HLG	High dynamic range video streams that utilize the Hybrid Log-Gamma system of ITU-R Recommendation BT.2100.
UNSPECIFIED	Video streams whose transfer characteristics are signaled by the payload as specified in ISO/IEC 21122-3, or must be manually coordinated between sender and receiver.

RANGE: This parameter SHOULD be used to signal the encoding range of the sample values within the stream. When paired with ITU Rec BT.2100 colorimetry, this parameter has two allowed values NARROW and FULL, corresponding to the ranges specified in table 9 of ITU Rec BT.2100. In any other context, this parameter has three allowed values: NARROW, FULLPROTECT, and FULL, which correspond to the ranges specified in SMPTE RP 2077. In the absence of this parameter, and for all but the UNSPECIFIED colometries, NARROW SHALL be the assumed value. When paired with the UNSPECIFIED colometry, FULL SHALL be the default assumed value.

Encoding considerations:

This media type is framed and binary; see Section 4.8 in RFC 6838 [RFC6838].

Security considerations:

Please see the Security Considerations section in RFC XXXX

6.2. Mapping to SDP

6.2.1. General

A Session Description Protocol (SDP) [RFC4566] object SHALL be created for each RTP stream and it SHALL be in accordance with the provisions of SMPTE ST 2110-10 [SMPTE-ST2110-10].

The information carried in the media type specification has a specific mapping to the SDP fields, used to describe RTP sessions. This information is redundant with the information found in the payload data (namely, in the JPEG XS header segment) and SHALL be consistent with it. In case of discrepancy between parameters values found in the payload data and in the SDP fields, the values from the payload data SHALL prevail.

6.2.2. Media type and subtype

The media type ("video") goes in SDP "m=" as the media name.

The media subtype ("jxsv") goes in SDP "a=rtpmap" as the encoding name, followed by a slash ("/") and the required parameter "rate" corresponding to the RTP timestamp clock rate (which for the payload format defined in this document SHOULD be 90000). The required parameter "transmode" and the additional optional parameters go in the SDP "a=fmtp" attribute by copying them directly from the MIME

media type string as a semicolon-separated list of parameter=value pairs.

A sample SDP mapping for JPEG XS video is as follows:

```
m=video 30000 RTP/AVP 112
a=rtpmap:112 jxsv/90000
a=fmtp:112 transmode=1;sampling=YCbCr-4:2:2;width=1920;
           height=1080;depth=10;colorimetry=BT709;TCS=SDR;
           RANGE=FULL;TP=2110TPNL
```

In this example, a JPEG XS RTP stream is being sent to UDP destination port 30000, with an RTP dynamic payload type of 112 and a media clock rate of 90000 Hz. Note that the "a=fmtp:" line has been wrapped to fit this page, and will be a single long line in the SDP file.

6.2.3. Traffic shaping

The SDP object SHALL include the TP parameter (either 2110TPNL or 2110TPW as specified in Section 4.5) and may include the CMAX parameter as specified in SMPTE ST 2110-21 [SMPTE-ST2110-21].

6.2.4. Offer/Answer Considerations

All parameters are declarative.

7. IANA Considerations

This memo requests that IANA registers video/jxsv 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" [1].

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 such as RTP/AVP [RFC3551], RTP/AVPF [RFC4585], RTP/SAVP [RFC3711], or RTP/SAVPF [RFC5124]. This implies that confidentiality of the media streams is achieved by encryption.

However, as "Securing the RTP Framework: Why RTP Does Not Mandate a Single Media Security Solution" [RFC7202] discusses, it is not an RTP payload format's responsibility to discuss or mandate what solutions are used to meet the basic security goals like confidentiality, integrity, and source authenticity for RTP in general. This responsibility lies on anyone using RTP in an application. They can

find guidance on available security mechanisms and important considerations in "Options for Securing RTP Sessions" [RFC7201]. Applications SHOULD use one or more appropriate strong security mechanisms.

This payload format and the JPEG XS encoding do not exhibit any substantial non-uniformity, either in output or in complexity to perform the decoding operation and thus are unlikely to pose a denial-of-service threat due to the receipt of pathological datagrams.

It is important to note that HD or UHDTV JPEG XS-encoded video can have significant bandwidth requirements (typically more than 1 Gbps for ultra high-definition video, especially if using high framerate). This is sufficient to cause potential for denial-of-service if transmitted onto most currently available Internet paths.

Accordingly, if best-effort service is being used, users of this payload format SHALL monitor packet loss to ensure that the packet loss rate is within acceptable parameters. Packet loss is considered acceptable if a TCP flow across the same network path, and experiencing the same network conditions, would achieve an average throughput, measured on a reasonable timescale, that is not less than the RTP flow is achieving. This condition can be satisfied by implementing congestion control mechanisms to adapt the transmission rate (or the number of layers subscribed for a layered multicast session), or by arranging for a receiver to leave the session if the loss rate is unacceptably high.

This payload format may also be used in networks that provide quality-of-service guarantees. If enhanced service is being used, receivers SHOULD monitor packet loss to ensure that the service that was requested is actually being delivered. If it is not, then they SHOULD assume that they are receiving best-effort service and behave accordingly.

9. Acknowledgments

The authors would like to thank the following people for their valuable contributions to this specification: Alexandre Willeme, Gael Rouvroy, and Jean-Baptise Lorent.

10. RFC Editor Considerations

Note to RFC Editor: This section may be removed after carrying out all the instructions of this section.

RFC XXXX is to be replaced by the RFC number this specification receives when published.

11. References

11.1. Normative References

[ISO21122-1]
International Organization for Standardization (ISO) - International Electrotechnical Commission (IEC), "Information technology - JPEG XS low-latency lightweight image coding system - Part 1: Core coding system", ISO/IEC IS 21122-1, 2019, <https://www.iso.org/standard/74535.html>.

[ISO21122-2]
International Organization for Standardization (ISO) - International Electrotechnical Commission (IEC), "Information technology - JPEG XS low-latency lightweight image coding system - Part 2: Profiles and buffer models", ISO/IEC IS 21122-2, 2019, <https://www.iso.org/standard/74536.html>.

[ISO21122-3]
International Organization for Standardization (ISO) - International Electrotechnical Commission (IEC), "Information technology - JPEG XS low-latency lightweight image coding system - Part 3: Transport and container formats", ISO/IEC IS 21122-3, 2019, <https://www.iso.org/standard/74537.html>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <https://www.rfc-editor.org/info/rfc3550>.

[RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, DOI 10.17487/RFC3551, July 2003, <https://www.rfc-editor.org/info/rfc3551>.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, DOI 10.17487/RFC3711, March 2004, <https://www.rfc-editor.org/info/rfc3711>.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, DOI 10.17487/RFC4566, July 2006, <https://www.rfc-editor.org/info/rfc4566>.

[RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <https://www.rfc-editor.org/info/rfc6838>.

[RFC8083] Perkins, C. and V. Singh, "Multimedia Congestion Control: Circuit Breakers for Unicast RTP Sessions", RFC 8083, DOI 10.17487/RFC8083, March 2017, <https://www.rfc-editor.org/info/rfc8083>.

[RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <https://www.rfc-editor.org/info/rfc8085>.

[SMPTE-ST2110-10] Society of Motion Picture and Television Engineers, "SMPTE Standard - Professional Media Over Managed IP Networks: System Timing and Definitions", SMPTE ST 2110-10:2017, 2017, <https://doi.org/10.5594/SMPTE.ST2110-10.2017>.

[SMPTE-ST2110-21] Society of Motion Picture and Television Engineers, "SMPTE Standard - Professional Media Over Managed IP Networks: Traffic Shaping and Delivery Timing for Video", SMPTE ST 2110-21:2017, 2017, <https://doi.org/10.5594/SMPTE.ST2110-21.2017>.

11.2. Informative References

[RFC4175] Gharai, L. and C. Perkins, "RTP Payload Format for Uncompressed Video", RFC 4175, DOI 10.17487/RFC4175, September 2005, <https://www.rfc-editor.org/info/rfc4175>.

[RFC4585] Ott, J., Wenger, S., Sato, N., Burmeister, C., and J. Rey, "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", RFC 4585, DOI 10.17487/RFC4585, July 2006, <https://www.rfc-editor.org/info/rfc4585>.

[RFC5124] Ott, J. and E. Carrara, "Extended Secure RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/SAVPF)", RFC 5124, DOI 10.17487/RFC5124, February 2008, <https://www.rfc-editor.org/info/rfc5124>.

[RFC7201] Westerlund, M. and C. Perkins, "Options for Securing RTP Sessions", RFC 7201, DOI 10.17487/RFC7201, April 2014, <https://www.rfc-editor.org/info/rfc7201>.

[RFC7202] Perkins, C. and M. Westerlund, "Securing the RTP Framework: Why RTP Does Not Mandate a Single Media Security Solution", RFC 7202, DOI 10.17487/RFC7202, April 2014, <https://www.rfc-editor.org/info/rfc7202>.

11.3. URIs

[1] http://www.iana.org/assignments/rtp-parameters

Authors' Addresses

Sebastien Lugan
intoPIX S.A.
Rue Emile Francqui, 9
1435 Mont-Saint-Guibert
Belgium

Phone: +32 10 23 84 70
Email: rtp@intopix.com
URI: https://www.intopix.com/

Antonin Descampe
Universite catholique de Louvain
Place du Levant, 3 - bte L5.03.02
1348 Louvain-la-Neuve
Belgium

Phone: +32 10 47 25 97
Email: antonin.descampe@uclouvain.be
URI: https://uclouvain.be/en/research-institutes/icteam

Corentin Damman
intoPIX S.A.
Rue Emile Francqui, 9
1435 Mont-Saint-Guibert
Belgium

Phone: +32 10 23 84 70
Email: c.damman@intopix.com
URI: https://www.intopix.com/

Thomas Richter
Fraunhofer IIS
Am Wolfsmantel 33
91048 Erlangen
Germany

Phone: +49 9131 776 5126
Email: thomas.richter@iis.fraunhofer.de
URI: https://www.iis.fraunhofer.de/

Tim Bruylants
intoPIX S.A.
Rue Emile Francqui, 9
1435 Mont-Saint-Guibert
Belgium

Phone: +32 10 23 84 70
Email: t.bruylants@intopix.com
URI: https://www.intopix.com/

AVTCore Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 8, 2021

J. Uberti
S. Holmer
M. Flodman
D. Hong
Google
J. Lennox
8x8 / Jitsi

July 7, 2020

RTP Payload Format for VP9 Video
draft-ietf-payload-vp9-10

Abstract

This memo describes an RTP payload format for the VP9 video codec. The payload format has wide applicability, as it supports applications from low bit-rate peer-to-peer usage, to high bit-rate video conferences. It includes provisions for temporal and spatial scalability.

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 https://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 January 8, 2021.

Copyright Notice

Table of Contents

1. Introduction

This memo describes an RTP payload specification applicable to the transmission of video streams encoded using the VP9 video codec [VP9-BITSTREAM]. The format described in this document can be used both in peer-to-peer and video conferencing applications.

The VP9 video codec was developed by Google, and is the successor to its earlier VP8 [RFC6386] codec. Above the compression improvements and other general enhancements above VP8, VP9 is also designed in a way that allows spatially-scalable video encoding.

2. Conventions, Definitions and Acronyms

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 Description

The VP9 codec can maintain up to eight reference frames, of which up to three can be referenced by any new frame.

VP9 also allows a frame to use another frame of a different resolution as a reference frame. (Specifically, a frame may use any references whose width and height are between 1/16th that of the current frame and twice that of the current frame, inclusive.) This allows internal resolution changes without requiring the use of key frames.

These features together enable an encoder to implement various forms of coarse-grained scalability, including temporal, spatial and quality scalability modes, as well as combinations of these, without the need for explicit scalable coding tools.

Temporal layers define different frame rates of video; spatial and quality layers define different and possibly dependent representations of a single input frame. Spatial layers allow a frame to be encoded at different resolutions, whereas quality layers allow a frame to be encoded at the same resolution but at different qualities (and thus with different amounts of coding error). VP9 supports quality layers as spatial layers without any resolution changes; hereinafter, the term "spatial layer" is used to represent both spatial and quality layers.

This payload format specification defines how such temporal and spatial scalability layers can be described and communicated.

Temporal and spatial scalability layers are associated with non-negative integer IDs. The lowest layer of either type has an ID of 0, and is sometimes referred to as the "base" temporal or spatial layer.

Layers are designed (and MUST be encoded) such that if any layer, and all higher layers, are removed from the bitstream along either of the two dimensions, the remaining bitstream is still correctly decodable.

For terminology, this document uses the term "frame" to refer to a single encoded VP9 frame for a particular resolution/quality, and "picture" to refer to all the representations (frames) at a single

instant in time. A picture thus consists of one or more frames, encoding different spatial layers.

Within a picture, a frame with spatial layer ID equal to SID, where SID > 0, can depend on a frame of the same picture with a lower spatial layer ID. This "inter-layer" dependency can result in additional coding gain compared to the case where only traditional "inter-picture" dependency is used, where a frame depends on previously coded frame in time. For simplicity, this payload format assumes that, within a picture and if inter-layer dependency is used, a spatial layer SID frame can depend only on the immediately previous spatial layer SID-1 frame, when S > 0. Additionally, if inter-picture dependency is used, a spatial layer SID frame is assumed to only depend on a previously coded spatial layer SID frame.

Given above simplifications for inter-layer and inter-picture dependencies, a flag (the D bit described below) is used to indicate whether a spatial layer SID frame depends on the spatial layer SID-1 frame. Given the D bit, a receiver only needs to additionally know the inter-picture dependency structure for a given spatial layer frame in order to determine its decodability. Two modes of describing the inter-picture dependency structure are possible: "flexible mode" and "non-flexible mode". An encoder can only switch between the two on the first packet of a key frame with temporal layer ID equal to 0.

In flexible mode, each packet can contain up to 3 reference indices, which identify all frames referenced by the frame transmitted in the current packet for inter-picture prediction. This (along with the D bit) enables a receiver to identify if a frame is decodable or not and helps it understand the temporal layer structure. Since this is signaled in each packet it makes it possible to have very flexible temporal layer hierarchies and patterns which are changing dynamically.

In non-flexible mode, the inter-picture dependency (the reference indices) of a Picture Group (PG) MUST be pre-specified as part of the scalability structure (SS) data. In this mode, each packet has an index to refer to one of the described pictures in the PG, from which the pictures referenced by the picture transmitted in the current packet for inter-picture prediction can be identified.

(Editor's Note: A "Picture Group", as used in this document, is not the same thing as a the term "Group of Pictures" as it is traditionally used in video coding, i.e. to mean an independently-decoadable run of pictures beginning with a keyframe. Suggestions for better terminology are welcome.)

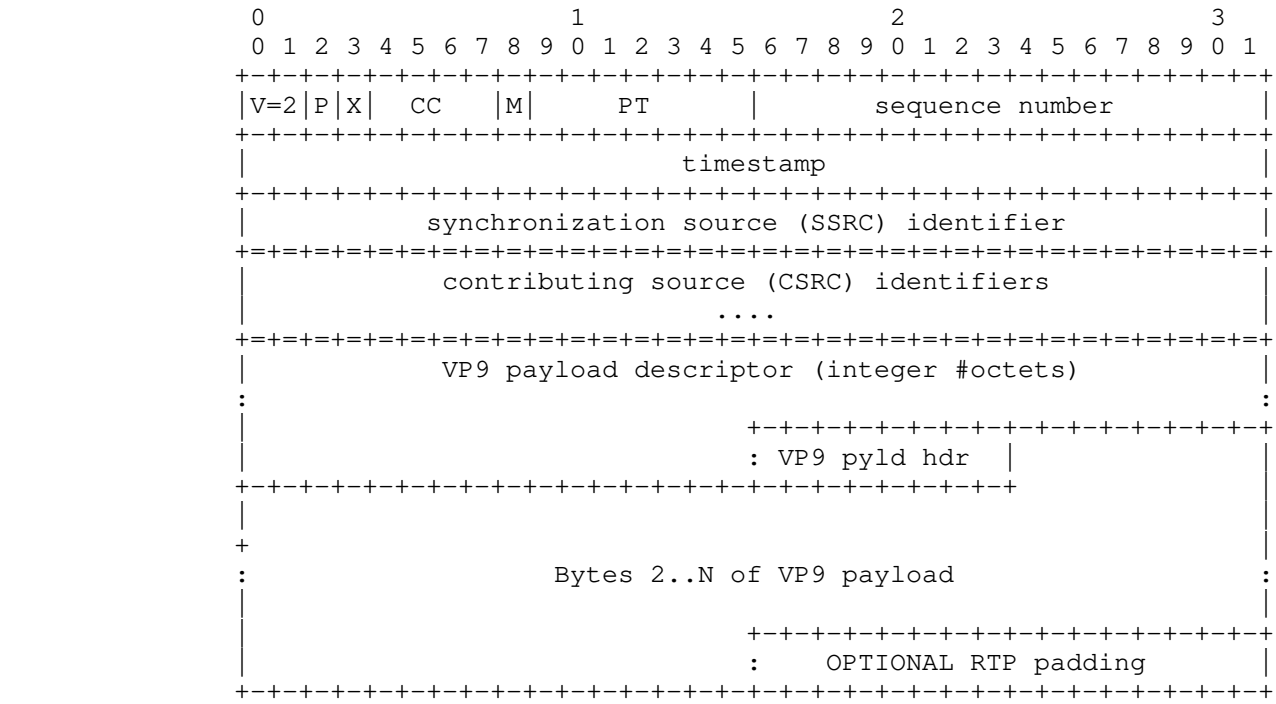
The SS data can also be used to specify the resolution of each spatial layer present in the VP9 stream for both flexible and non-flexible modes.

4. Payload Format

This section describes how the encoded VP9 bitstream is encapsulated in RTP. To handle network losses usage of RTP/AVPF [RFC4585] is RECOMMENDED. All integer fields in the specifications are encoded as unsigned integers in network octet order.

4.1. RTP Header Usage

The general RTP payload format for VP9 is depicted below.



The VP9 payload descriptor will be described in Section 4.2; the VP9 payload header is described in [VP9-BITSTREAM]. OPTIONAL RTP padding MUST NOT be included unless the P bit is set. The figure specifically shows the format for the first packet in a frame. Subsequent packets will not contain the VP9 payload header, and will have later octets in the frame payload.

Figure 1

Marker bit (M): MUST be set to 1 for the final packet of the highest spatial layer frame (the final packet of the picture), and 0 otherwise. Unless spatial scalability is in use for this picture, this will have the same value as the E bit described below. Note this bit MUST be set to 1 for the target spatial layer frame if a stream is being rewritten to remove higher spatial layers.

Payload Type (PT): In line with the policy in Section 3 of [RFC3551], applications using the VP9 RTP payload profile MUST assign a dynamic payload type number to be used in each RTP session and provide a mechanism to indicate the mapping. See Section 6.2 for the mechanism to be used with the Session Description Protocol (SDP) [RFC4566].

Timestamp: The RTP timestamp indicates the time when the input frame was sampled, at a clock rate of 90 kHz. If the input picture is encoded with multiple layer frames, all of the frames of the picture MUST have the same timestamp.

If a frame has the VP9 show_frame field set to 0 (i.e., it is meant only to populate a reference buffer, without being output) its timestamp MAY alternately be set to be the same as the subsequent frame with show_frame equal to 1. (This will be convenient for playing out pre-encoded content packaged with VP9 "superframes", which typically bundle show_frame==0 frames with a subsequent show_frame==1 frame.) Every frame with show_frame==1, however, MUST have a unique timestamp modulo the 2^32 wrap of the field.

The remaining RTP Fixed Header Fields (V, P, X, CC, sequence number, SSRC and CSRC identifiers) are used as specified in Section 5.1 of [RFC3550].

4.2. VP9 Payload Descriptor

In flexible mode (with the F bit below set to 1), The first octets after the RTP header are the VP9 payload descriptor, with the following structure.

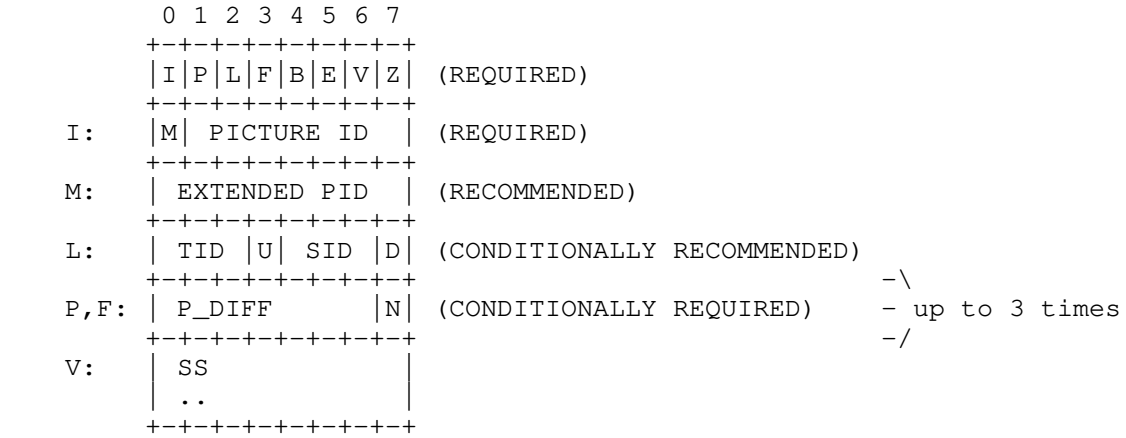


Figure 2

In non-flexible mode (with the F bit below set to 0), The first octets after the RTP header are the VP9 payload descriptor, with the following structure.

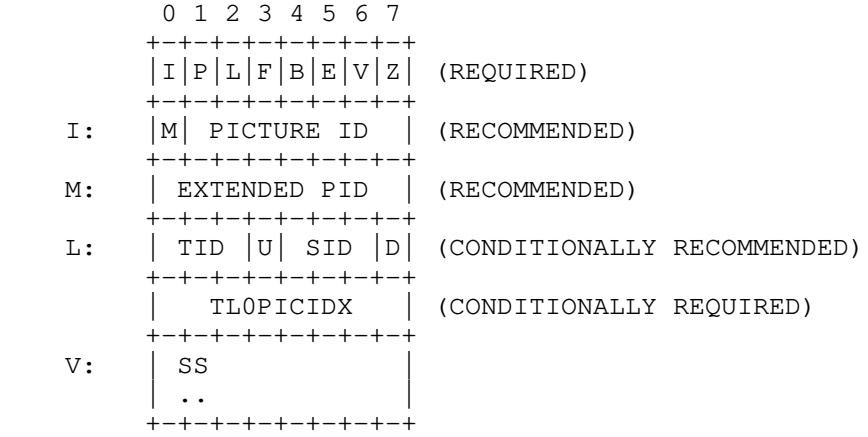


Figure 3

I: Picture ID (PID) present. When set to one, the OPTIONAL PID MUST be present after the mandatory first octet and specified as below.

Otherwise, PID MUST NOT be present. If the SS field was present in the stream's most recent start of a keyframe (i.e., non-flexible scalability mode is in use), then the PID MUST also be present in every packet.

P: Inter-picture predicted frame. When set to zero, the frame does not utilize inter-picture prediction. In this case, up-switching to a current spatial layer's frame is possible from directly lower spatial layer frame. P SHOULD also be set to zero when encoding a layer synchronization frame in response to an LRR [I-D.ietf-avtext-lrr] message (see Section 5.3). When P is set to zero, the TID field (described below) MUST also be set to 0 (if present). Note that the P bit does not forbid intra-picture, inter-layer prediction from earlier frames of the same picture, if any.

L: Layer indices present. When set to one, the one or two octets following the mandatory first octet and the PID (if present) is as described by "Layer indices" below. If the F bit (described below) is set to 1 (indicating flexible mode), then only one octet is present for the layer indices. Otherwise if the F bit is set to 0 (indicating non-flexible mode), then two octets are present for the layer indices.

F: Flexible mode. F set to one indicates flexible mode and if the P bit is also set to one, then the octets following the mandatory first octet, the PID, and layer indices (if present) are as described by "Reference indices" below. This MUST only be set to 1 if the I bit is also set to one; if the I bit is set to zero, then this MUST also be set to zero and ignored by receivers. The value of this F bit MUST only change on the first packet of a key picture. A key picture is a picture whose base spatial layer frame is a key frame, and which thus completely resets the encoder state. This packet will have its P bit equal to zero, SID or D bit (described below) equal to zero, and B bit (described below) equal to 1.

B: Start of a frame. MUST be set to 1 if the first payload octet of the RTP packet is the beginning of a new VP9 frame, and MUST NOT be 1 otherwise. Note that this frame might not be the first frame of a picture.

E: End of a frame. MUST be set to 1 for the final RTP packet of a VP9 frame, and 0 otherwise. This enables a decoder to finish decoding the frame, where it otherwise may need to wait for the next packet to explicitly know that the frame is complete. Note that, if spatial scalability is in use, more frames from the same picture may follow; see the description of the M bit above.

V: Scalability structure (SS) data present. When set to one, the OPTIONAL SS data MUST be present in the payload descriptor. Otherwise, the SS data MUST NOT be present.

Z: Not a reference frame for upper spatial layers. If set to 1, indicates that frames with higher spatial layers SID+1 of the current and following pictures do not depend on the current spatial layer SID frame. This enables a decoder which is targeting a higher spatial layer to know that it can safely discard this packet's frame without processing it, without having to wait for the "D" bit in the higher-layer frame (see below).

The mandatory first octet is followed by the extension data fields that are enabled:

M: The most significant bit of the first octet is an extension flag. The field MUST be present if the I bit is equal to one. If set, the PID field MUST contain 15 bits; otherwise, it MUST contain 7 bits. See PID below.

Picture ID (PID): Picture ID represented in 7 or 15 bits, depending on the M bit. This is a running index of the pictures. The field MUST be present if the I bit is equal to one. If M is set to zero, 7 bits carry the PID; else if M is set to one, 15 bits carry the PID in network byte order. The sender may choose between a 7- or 15-bit index. The PID SHOULD start on a random number, and MUST wrap after reaching the maximum ID. The receiver MUST NOT assume that the number of bits in PID stay the same through the session.

In the non-flexible mode (when the F bit is set to 0), this PID is used as an index to the picture group (PG) specified in the SS data below. In this mode, the PID of the key frame corresponds to the first specified frame in the PG. Then subsequent PIDs are mapped to subsequently specified frames in the PG (modulo N_G, specified in the SS data below), respectively.

All frames of the same picture MUST have the same PID value.

Frames (and their corresponding pictures) with the VP9 show_frame field equal to 0 MUST have distinct PID values from subsequent pictures with show_frame equal to 1. Thus, a Picture as defined in this specification is different than a VP9 Superframe.

All frames of the same picture MUST have the same value for show_frame.

Layer indices: This information is optional but recommended whenever encoding with layers. For both flexible and non-flexible modes, one octet is used to specify a layer frame's temporal layer ID (TID) and spatial layer ID (SID) as shown both in Figure 2 and Figure 3. Additionally, a bit (U) is used to indicate that the current frame is a "switching up point" frame. Another bit (D) is used to indicate whether inter-layer prediction is used for the current frame.

In the non-flexible mode (when the F bit is set to 0), another octet is used to represent temporal layer 0 index (TL0PICIDX), as depicted in Figure 3. The TL0PICIDX is present so that all minimally required frames - the base temporal layer frames - can be tracked.

The TID and SID fields indicate the temporal and spatial layers and can help middleboxes and and endpoints quickly identify which layer a packet belongs to.

TID: The temporal layer ID of current frame. In the case of non-flexible mode, if PID is mapped to a picture in a specified PG, then the value of TID MUST match the corresponding TID value of the mapped picture in the PG.

U: Switching up point. If this bit is set to 1 for the current picture with temporal layer ID equal to TID, then "switch up" to a higher frame rate is possible as subsequent higher temporal layer pictures will not depend on any picture before the current picture (in coding order) with temporal layer ID greater than TID.

SID: The spatial layer ID of current frame. Note that frames with spatial layer SDI > 0 may be dependent on decoded spatial layer SID-1 frame within the same picture. Different frames of the same picture MUST have distinct spatial layer IDs, and frames' spatial layers MUST appear in increasing order within the frame.

D: Inter-layer dependency used. MUST be set to one if current spatial layer SID frame depends on spatial layer SID-1 frame of the same picture. MUST only be set to zero if current spatial layer SID frame does not depend on spatial layer SID-1 frame of the same picture. For the base layer frame (with SID equal to 0), this D bit MUST be set to zero.

TL0PICIDX: 8 bits temporal layer zero index. TL0PICIDX is only present in the non-flexible mode (F = 0). This is a running index for the temporal base layer pictures, i.e., the pictures

with TID set to 0. If TID is larger than 0, TL0PICIDX indicates which temporal base layer picture the current picture depends on. TL0PICIDX MUST be incremented when TID is equal to 0. The index SHOULD start on a random number, and MUST restart at 0 after reaching the maximum number 255.

Reference indices: When P and F are both set to one, indicating a non-key frame in flexible mode, then at least one reference index has to be specified as below. Additional reference indices (total of up to 3 reference indices are allowed) may be specified using the N bit below. When either P or F is set to zero, then no reference index is specified.

P_DIFF: The reference index (in 7 bits) specified as the relative PID from the current picture. For example, when P_DIFF=3 on a packet containing the picture with PID 112 means that the picture refers back to the picture with PID 109. This calculation is done modulo the size of the PID field, i.e., either 7 or 15 bits.

N: 1 if there is additional P_DIFF following the current P_DIFF.

4.2.1. Scalability Structure (SS):

The scalability structure (SS) data describes the resolution of each frame within a picture as well as the inter-picture dependencies for a picture group (PG). If the VP9 payload descriptor's "V" bit is set, the SS data is present in the position indicated in Figure 2 and Figure 3.

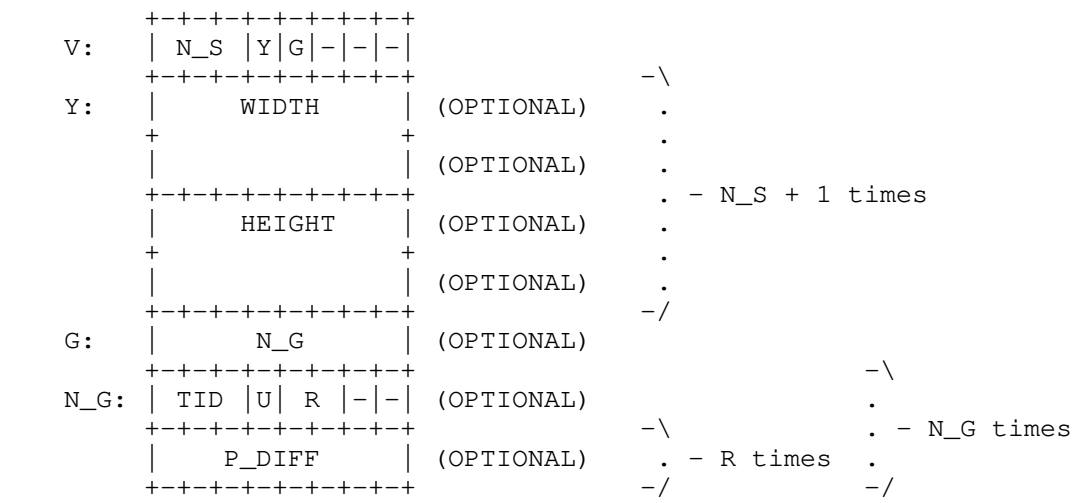


Figure 4

N_S: N_S + 1 indicates the number of spatial layers present in the VP9 stream.

Y: Each spatial layer's frame resolution present. When set to one, the OPTIONAL WIDTH (2 octets) and HEIGHT (2 octets) MUST be present for each layer frame. Otherwise, the resolution MUST NOT be present.

G: PG description present flag.

-: Bit reserved for future use. MUST be set to zero and MUST be ignored by the receiver.

N_G: N_G indicates the number of pictures in a Picture Group (PG). If N_G is greater than 0, then the SS data allows the inter-picture dependency structure of the VP9 stream to be pre-declared, rather than indicating it on the fly with every packet. If N_G is greater than 0, then for N_G pictures in the PG, each picture's temporal layer ID (TID), switch up point (U), and the R reference indices (P_DIFFs) are specified.

The first picture specified in the PG MUST have TID set to 0.

G set to 0 or N_G set to 0 indicates that either there is only one temporal layer or no fixed inter-picture dependency information is present going forward in the bitstream.

Note that for a given picture, all frames follow the same inter-picture dependency structure. However, the frame rate of each spatial layer can be different from each other and this can be controlled with the use of the D bit described above. The specified dependency structure in the SS data MUST be for the highest frame rate layer.

In a scalable stream sent with a fixed pattern, the SS data SHOULD be included in the first packet of every key frame. This is a packet with P bit equal to zero, SID or D bit equal to zero, and B bit equal to 1. The SS data MUST only be changed on the picture that corresponds to the first picture specified in the previous SS data's PG (if the previous SS data's N_G was greater than 0).

4.3. Frame Fragmentation

VP9 frames are fragmented into packets, in RTP sequence number order, beginning with a packet with the B bit set, and ending with a packet with the E bit set. There is no mechanism for finer-grained access to parts of a VP9 frame.

4.4. Scalable encoding considerations

In addition to the use of reference frames, VP9 has several additional forms of inter-frame dependencies, largely involving probability tables for the entropy and tree encoders. In VP9 syntax, the syntax element "error_resilient_mode" resets this additional inter-frame data, allowing a frame's syntax to be decoded independently.

Due to the requirements of scalable streams, a VP9 encoder producing a scalable stream needs to ensure that a frame does not depend on a previous frame (of the same or a previous picture) that can legitimately be removed from the stream. Thus, a frame that follows a removable frame (in full decode order) MUST be encoded with "error_resilient_mode" set to true.

For spatially-scalable streams, this means that "error_resilient_mode" needs to be turned on for the base spatial layer; it can however be turned off for higher spatial layers, assuming they are sent with inter-layer dependency (i.e. with the "D" bit set). For streams that are only temporally-scalable without spatial scalability, "error_resilient_mode" can additionally be turned off for any picture that immediately follows a temporal layer 0 frame.

4.5. Examples of VP9 RTP Stream

4.5.1. Reference picture use for scalable structure

As discussed in Section 3, the VP9 codec can maintain up to eight reference frames, of which up to three can be referenced or updated by any new frame. This section illustrates one way that a scalable structure (with three spatial layers and three temporal layers) can be constructed using these reference frames.

Temporal	Spatial	References	Updates
0	0	0	0
0	1	0,1	1
0	2	1,2	2
2	0	0	6
2	1	1,6	7
2	2	2,7	-
1	0	0	3
1	1	1,3	4
1	2	2,4	5
2	0	3	6
2	1	4,6	7
2	2	5,7	-

Example scalability structure

This structure is constructed such that the "U" bit can always be set.

5. Feedback Messages and Header Extensions

5.1. Reference Picture Selection Indication (RPSI)

The reference picture selection index is a payload-specific feedback message defined within the RTCP-based feedback format. The RPSI message is generated by a receiver and can be used in two ways. Either it can signal a preferred reference picture when a loss has been detected by the decoder -- preferably then a reference that the decoder knows is perfect -- or, it can be used as positive feedback information to acknowledge correct decoding of certain reference pictures. The positive feedback method is useful for VP9 used for point to point (unicast) communication. The use of RPSI for VP9 is preferably combined with a special update pattern of the codec's two special reference frames -- the golden frame and the altref frame -- in which they are updated in an alternating leapfrog fashion. When a receiver has received and correctly decoded a golden or altref frame, and that frame had a PictureID in the payload descriptor, the receiver can acknowledge this simply by sending an RPSI message back to the sender. The message body (i.e., the "native RPSI bit string" in [RFC4585]) is simply the PictureID of the received frame.

Note: because all frames of the same picture must have the same inter-picture reference structure, there is no need for a message to specify which frame is being selected.

5.2. Full Intra Request (FIR)

The Full Intra Request (FIR) [RFC5104] RTCP feedback message allows a receiver to request a full state refresh of an encoded stream.

Upon receipt of an FIR request, a VP9 sender MUST send a picture with a keyframe for its spatial layer 0 layer frame, and then send frames without inter-picture prediction (P=0) for any higher layer frames.

5.3. Layer Refresh Request (LRR)

The Layer Refresh Request [I-D.ietf-avtext-lrr] allows a receiver to request a single layer of a spatially or temporally encoded stream to be refreshed, without necessarily affecting the stream's other layers.

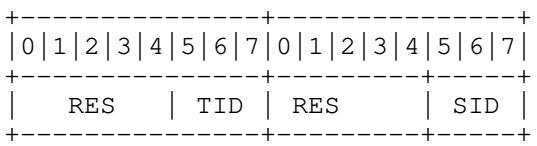


Figure 5

Figure 5 shows the format of LRR's layer index fields for VP9 streams. The two "RES" fields MUST be set to 0 on transmission and ingnored on reception. See Section 4.2 for details on the TID and SID fields.

Identification of a layer refresh frame can be derived from the reference IDs of each frame by backtracking the dependency chain until reaching a point where only decodable frames are being referenced. Therefore it's recommended for both the flexible and the non-flexible mode that, when upgrade frames are being encoded in response to a LRR, those packets should contain layer indices and the reference fields so that the decoder or an MCU can make this derivation.

Example:

LRR {1,0}, {2,1} is sent by an MCU when it is currently relaying {1,0} to a receiver and which wants to upgrade to {2,1}. In response the encoder should encode the next frames in layers {1,1} and {2,1} by only referring to frames in {1,0}, or {0,0}.

In the non-flexible mode, periodic upgrade frames can be defined by the layer structure of the SS, thus periodic upgrade frames can be automatically identified by the picture ID.

5.4. Frame Marking

The Frame Marking RTP header extension [I-D.ietf-avtext-framemarking] is a mechanism to provide information about frames of video streams in a largely codec-independent manner. However, for its extension for scalable codecs, the specific manner in which codec layers are identified needs to be specified specifically for each codec. This section defines how frame marking is used with VP9.

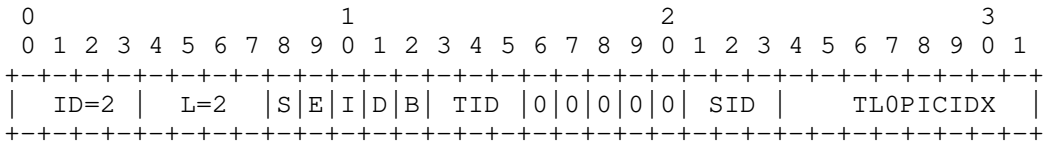


Figure 6

When this header extension is used with VP9, the TID and SID fields MUST match the values in the packet which the header extension is attached to; see Section 4.2 for details on these fields.

See [I-D.ietf-avtext-framemarking] for explanations of the other fields, which are generic.

6. Payload Format Parameters

This payload format has two optional parameters.

6.1. Media Type Definition

This registration is done using the template defined in [RFC6838] and following [RFC4855].

Type name: video

Subtype name: VP9

Required parameters: None.

Optional parameters:
These parameters are used to signal the capabilities of a receiver implementation. If the implementation is willing to receive media, both parameters MUST be provided. These parameters MUST NOT be used for any other purpose.

max-fr: The value of max-fr is an integer indicating the maximum frame rate in units of frames per second that the decoder is capable of decoding.

max-fs: The value of max-fs is an integer indicating the maximum frame size in units of macroblocks that the decoder is capable of decoding.

The decoder is capable of decoding this frame size as long as the width and height of the frame in macroblocks are less than int(sqrt(max-fs * 8)) - for instance, a max-fs of 1200 (capable of supporting 640x480 resolution) will support widths and heights up to 1552 pixels (97 macroblocks).

profile-id: The value of profile-id is an integer indicating the default coding profile, the subset of coding tools that may have been used to generate the stream or that the receiver supports). Table 1 lists all of the profiles defined in section 7.2 of [VP9-BITSTREAM] and the corresponding integer values to be used.

If no profile-id is present, Profile 0 MUST be inferred.

Informative note: See Table 2 for capabilities of coding profiles defined in section 7.2 of [VP9-BITSTREAM].

Encoding considerations:

This media type is framed in RTP and contains binary data; see Section 4.8 of [RFC6838].

Security considerations: See Section 7 of RFC xxxx.
[RFC Editor: Upon publication as an RFC, please replace "XXXX" with the number assigned to this document and remove this note.]

Interoperability considerations: None.

Published specification: VP9 bitstream format [VP9-BITSTREAM] and RFC XXXX.
[RFC Editor: Upon publication as an RFC, please replace "XXXX" with the number assigned to this document and remove this note.]

Applications which use this media type:
For example: Video over IP, video conferencing.

Fragment identifier considerations: N/A.

Additional information: None.

Person & email address to contact for further information:
Jonathan Lennox <jonathan.lennox@8x8.com>

Intended usage: COMMON

Restrictions on usage:
This media type depends on RTP framing, and hence is only defined for transfer via RTP [RFC3550].

Author: Jonathan Lennox <jonathan.lennox@8x8.com>

Change controller:
IETF AVTCore Working Group delegated from the IESG.

Profile	profile-id
0	0
1	1
2	2
3	3

Table 1: Table 1. Table of profile-id integer values representing the VP9 profile corresponding to the set of coding tools supported.

Profile	Bit Depth	SRGB Colorspace	Chroma Subsampling
0	8	No	YUV 4:2:0
1	8	Yes	YUV 4:2:0,4:4:0 or 4:4:4
2	10 or 12	No	YUV 4:2:0
3	10 or 12	Yes	YUV 4:2:0,4:4:0 or 4:4:4

Table 2: Table 2. Table of profile capabilities.

6.2. SDP Parameters

The receiver MUST ignore any fmtp parameter unspecified in this memo.

6.2.1. Mapping of Media Subtype Parameters to SDP

The media type video/VP9 string is mapped to fields in the Session Description Protocol (SDP) [RFC4566] as follows:

- The media name in the "m=" line of SDP MUST be video.
- The encoding name in the "a=rtpmap" line of SDP MUST be VP9 (the media subtype).
- The clock rate in the "a=rtpmap" line MUST be 90000.
- The parameters "max-fs", and "max-fr", MUST be included in the "a=fmtp" line of SDP if SDP is used to declare receiver capabilities. These parameters are expressed as a media subtype

string, in the form of a semicolon separated list of parameter=value pairs.

- The OPTIONAL parameter profile-id, when present, SHOULD be included in the "a=fmtp" line of SDP. This parameter is expressed as a media subtype string, in the form of a parameter=value pair. When the parameter is not present, a value of 0 MUST be used for profile-id.

6.2.1.1. Example

An example of media representation in SDP is as follows:

```
m=video 49170 RTP/AVPF 98
a=rtpmap:98 VP9/90000
a=fmtp:98 max-fr=30; max-fs=3600; profile-id=0;
```

6.2.2. Offer/Answer Considerations

When VP9 is offered over RTP using SDP in an Offer/Answer model [RFC3264] for negotiation for unicast usage, the following limitations and rules apply:

- The parameter identifying a media format configuration for VP9 is profile-id. This media format configuration parameter MUST be used symmetrically; that is, the answerer MUST either maintain all configuration parameters or remove the media format (payload type) completely if one or more of the parameter values are not supported.

- To simplify the handling and matching of these configurations, the same RTP payload type number used in the offer SHOULD also be used in the answer, as specified in [RFC3264]. An answer MUST NOT contain the payload type number used in the offer unless the configuration is exactly the same as in the offer.

7. 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 such as RTP/AVP [RFC3551], RTP/AVPF [RFC4585], RTP/SAVP [RFC3711], or RTP/SAVPF [RFC5124]. SAVPF [RFC5124]. However, as "Securing the RTP Protocol Framework: Why RTP Does Not Mandate a Single Media Security Solution" [RFC7202] discusses, it is not an RTP payload format's responsibility to discuss or mandate what solutions are used to meet the basic security goals like confidentiality, integrity and source authenticity for RTP in general. This responsibility lays on anyone

using RTP in an application. They can find guidance on available security mechanisms in Options for Securing RTP Sessions [RFC7201]. Applications SHOULD use one or more appropriate strong security mechanisms. The rest of this security consideration section discusses the security impacting properties of the payload format itself.

This RTP payload format and its media decoder do not exhibit any significant non-uniformity in the receiver-side computational complexity for packet processing, and thus are unlikely to pose a denial-of-service threat due to the receipt of pathological data. Nor does the RTP payload format contain any active content.

8. Congestion Control

Congestion control for RTP SHALL be used in accordance with RFC 3550 [RFC3550], and with any applicable RTP profile; e.g., RFC 3551 [RFC3551]. The congestion control mechanism can, in a real-time encoding scenario, adapt the transmission rate by instructing the encoder to encode at a certain target rate. Media aware network elements MAY use the information in the VP9 payload descriptor in Section 4.2 to identify non-reference frames and discard them in order to reduce network congestion. Note that discarding of non-reference frames cannot be done if the stream is encrypted (because the non-reference marker is encrypted).

9. IANA Considerations

The IANA is requested to register the media type registration "video/vp9" 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>.

10. Acknowledgments

Alex Eleftheriadis, Yuki Ito, Won Kap Jang, Sergio Garcia Murillo, Roi Sasson, Timothy Terriberry, Emircan Uysaler, and Thomas Volkert commented on the development of this document and provided helpful comments and feedback.

11. References

11.1. Normative References

[I-D.ietf-avtext-framemarking]
Zanaty, M., Berger, E., and S. Nandakumar, "Frame Marking RTP Header Extension", draft-ietf-avtext-framemarking-10 (work in progress), November 2019.

[I-D.ietf-avtext-lrr]
Lennox, J., Hong, D., Uberti, J., Holmer, S., and M. Flodman, "The Layer Refresh Request (LRR) RTCP Feedback Message", draft-ietf-avtext-lrr-07 (work in progress), July 2017.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, DOI 10.17487/RFC3264, June 2002, <https://www.rfc-editor.org/info/rfc3264>.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <https://www.rfc-editor.org/info/rfc3550>.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, DOI 10.17487/RFC4566, July 2006, <https://www.rfc-editor.org/info/rfc4566>.

[RFC4585] Ott, J., Wenger, S., Sato, N., Burmeister, C., and J. Rey, "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", RFC 4585, DOI 10.17487/RFC4585, July 2006, <https://www.rfc-editor.org/info/rfc4585>.

[RFC4855] Casner, S., "Media Type Registration of RTP Payload Formats", RFC 4855, DOI 10.17487/RFC4855, February 2007, <https://www.rfc-editor.org/info/rfc4855>.

[RFC5104] Wenger, S., Chandra, U., Westerlund, M., and B. Burman, "Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)", RFC 5104, DOI 10.17487/RFC5104, February 2008, <https://www.rfc-editor.org/info/rfc5104>.

[RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <https://www.rfc-editor.org/info/rfc6838>.

[VP9-BITSTREAM]
Grange, A., de Rivaz, P., and J. Hunt, "VP9 Bitstream & Decoding Process Specification", Version 0.6, March 2016, <https://storage.googleapis.com/downloads.webmproject.org/docs/vp9/vp9-bitstream-specification-v0.6-20160331-draft.pdf>.

11.2. Informative References

[RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, DOI 10.17487/RFC3551, July 2003, <https://www.rfc-editor.org/info/rfc3551>.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, DOI 10.17487/RFC3711, March 2004, <https://www.rfc-editor.org/info/rfc3711>.

[RFC5124] Ott, J. and E. Carrara, "Extended Secure RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/SAVPF)", RFC 5124, DOI 10.17487/RFC5124, February 2008, <https://www.rfc-editor.org/info/rfc5124>.

[RFC6386] Bankoski, J., Koleszar, J., Quillio, L., Salonen, J., Wilkins, P., and Y. Xu, "VP8 Data Format and Decoding Guide", RFC 6386, DOI 10.17487/RFC6386, November 2011, <https://www.rfc-editor.org/info/rfc6386>.

[RFC7201] Westerlund, M. and C. Perkins, "Options for Securing RTP Sessions", RFC 7201, DOI 10.17487/RFC7201, April 2014, <https://www.rfc-editor.org/info/rfc7201>.

[RFC7202] Perkins, C. and M. Westerlund, "Securing the RTP Framework: Why RTP Does Not Mandate a Single Media Security Solution", RFC 7202, DOI 10.17487/RFC7202, April 2014, <https://www.rfc-editor.org/info/rfc7202>.

Authors' Addresses

Justin Uberti
Google, Inc.
747 6th Street South
Kirkland, WA 98033
USA

Email: justin@uberti.name

Stefan Holmer
Google, Inc.
Kungsbron 2
Stockholm 111 22
Sweden

Email: holmer@google.com

Magnus Flodman
Google, Inc.
Kungsbron 2
Stockholm 111 22
Sweden

Email: mflodman@google.com

Danny Hong
Google, Inc.
1585 Charleston Road
Mountain View, CA 94043
US

Email: dannyhong@google.com

Jonathan Lennox
8x8, Inc. / Jitsi
1350 Broadway
New York, NY 10018
US

Email: jonathan.lennox@8x8.com

AVTCORE
Internet-Draft
Intended status: Standards Track
Expires: 6 May 2021

J. Uberti
Google
C. Jennings
Cisco

2 November 2020

Completely Encrypting RTP Header Extensions and Contributing Sources
draft-uberti-avtcore-cryptex-01

Abstract

While the Secure Real-time Transport Protocol (SRTP) provides confidentiality for the contents of a media packet, a significant amount of metadata is left unprotected, including RTP header extensions and contributing sources (CSRCs). However, this data can be moderately sensitive in many applications. While there have been previous attempts to protect this data, they have had limited deployment, due to complexity as well as technical limitations.

This document proposes a new mechanism to completely encrypt header extensions and CSRCs as well a simpler signaling mechanism intended to facilitate deployment.

Discussion Venues

This note is to be removed before publishing as an RFC.

Source for this draft and an issue tracker can be found at https://github.com/juberti/cryptex.

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 https://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 6 May 2021.

Table of Contents

1. Introduction

1.1. Problem Statement

The Secure Real-time Transport Protocol [RFC3711] mechanism provides message authentication for the entire RTP packet, but only encrypts the RTP payload. This has not historically been a problem, as much of the information carried in the header has minimal sensitivity (e.g., RTP timestamp); in addition, certain fields need to remain as cleartext because they are used for key scheduling (e.g., RTP SSRC and sequence number).

However, as noted in [RFC6904], the security requirements can be different for information carried in RTP header extensions, including the per-packet sound levels defined in [RFC6464] and [RFC6465], which are specifically noted as being sensitive in the Security Considerations section of those RFCs.

In addition to the contents of the header extensions, there are now enough header extensions in active use that the header extension identifiers themselves can provide meaningful information in terms of determining the identity of endpoint and/or application. Accordingly, these identifiers can be considered at least slightly sensitive.

Finally, the CSRCs included in RTP packets can also be sensitive, potentially allowing a network eavesdropper to determine who was speaking and when during an otherwise secure conference call.

1.2. Previous Solutions

[RFC6904] was proposed in 2013 as a solution to the problem of unprotected header extension values. However, it has not seen significant adoption, and has a few technical shortcomings.

First, the mechanism is complicated. Since it allows encryption to be negotiated on a per-extension basis, a fair amount of signaling logic is required. And in the SRTP layer, a somewhat complex transform is required to allow only the selected header extension values to be encrypted. One of the most popular SRTP implementations had a significant bug in this area that was not detected for five years.

Second, it only protects the header extension values, and not their ids or lengths. It also does not protect the CSRCs. As noted above, this leaves a fair amount of potentially sensitive information exposed.

Third, it bloats the header extension space. Because each extension must be offered in both unencrypted and encrypted forms, twice as many header extensions must be offered, which will in many cases push implementations past the 14-extension limit for the use of one-byte extension headers defined in [RFC8285]. Accordingly, implementations will need to use two-byte headers in many cases, which are not supported well by some existing implementations.

Finally, the header extension bloat combined with the need for backwards compatibility results in additional wire overhead. Because two-byte extension headers may not be handled well by existing implementations, one-byte extension identifiers will need to be used for the unencrypted (backwards compatible) forms, and two-byte for the encrypted forms. Thus, deployment of [RFC6904] encryption for header extensions will typically result in multiple extra bytes in each RTP packet, compared to the present situation.

1.3. Goals

From this analysis we can state the desired properties of a solution:

* Build on existing [RFC3711] SRTP framework (simple to understand)

* Build on existing [RFC8285] header extension framework (simple to implement)

* Protection of header extension ids, lengths, and values

* Protection of CSRCs when present

* Simple signaling

* Simple crypto transform and SRTP interactions

* Backward compatible with unencrypted endpoints, if desired

* Backward compatible with existing RTP tooling

The last point deserves further discussion. While we considered possible solutions that would have encrypted more of the RTP header (e.g., the number of CSRCs), we felt the inability to parse the resultant packets with current tools, as well as additional complexity incurred, outweighed the slight improvement in confidentiality.

2. Terminology

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. Design

This specification proposes a mechanism to negotiate encryption of all RTP header extensions (ids, lengths, and values) as well as CSRC values. It reuses the existing SRTP framework, is accordingly simple to implement, and is backward compatible with existing RTP packet parsing code, even when support for this mechanism has been negotiated.

4. Signaling

In order to determine whether this mechanism defined in this specification is supported, this document defines a new "a=extmap-encrypted" Session Description Protocol (SDP) [RFC4566] attribute to indicate support. This attribute takes no value, and can be used at the session level or media level. Offering this attribute indicates that the endpoint is capable of receiving RTP packets encrypted as defined below.

The formal definition of this attribute is:

Name: extmap-encrypted

Value: None

Usage Level: session, media

Charset Dependent: No

Example:

```
a=extmap-encrypted
```

When used with BUNDLE, this attribute is specified as the TRANSPORT category. (todo: REF)

5. RTP Header Processing

[RFC8285] defines two values for the "defined by profile" field for carrying one-byte and two-byte header extensions. In order to allow a receiver to determine if an incoming RTP packet is using the encryption scheme in this specification, two new values are defined:

* 0xC0DE for the encrypted version of the one-byte header extensions (instead of 0xBEDE).

* 0xC2DE for the encrypted versions of the two-byte header extensions (instead of 0x100).

In the case of using two-byte header extensions, the extension id with value 256 MUST NOT be negotiated, as the value of this id is meant to be contained in the "appbits" of the "defined by profile" field, which are not available when using the values above.

If the "a=extmap-allow-mixed" attribute defined in [RFC8285] is negotiated, either one-byte or two-byte header ids can be used (with the values above), as in [RFC8285].

5.1. Sending

When sending an RTP packet that requires any header extensions to a destination that has negotiated header encryption, the header extensions MUST be formatted as [RFC8285] header extensions, as usual.

If one-byte extension ids are in use, the 16-bit RTP header extension tag MUST be set to 0xC0DE to indicate that the encryption defined in this specification has been applied. If two-byte header extension codes are in use, the 16-bit RTP header extension tag MUST be set to 0xC2DE to indicate the same.

The RTP packet MUST then be encrypted as described in Encryption Procedure.

5.2. Receiving

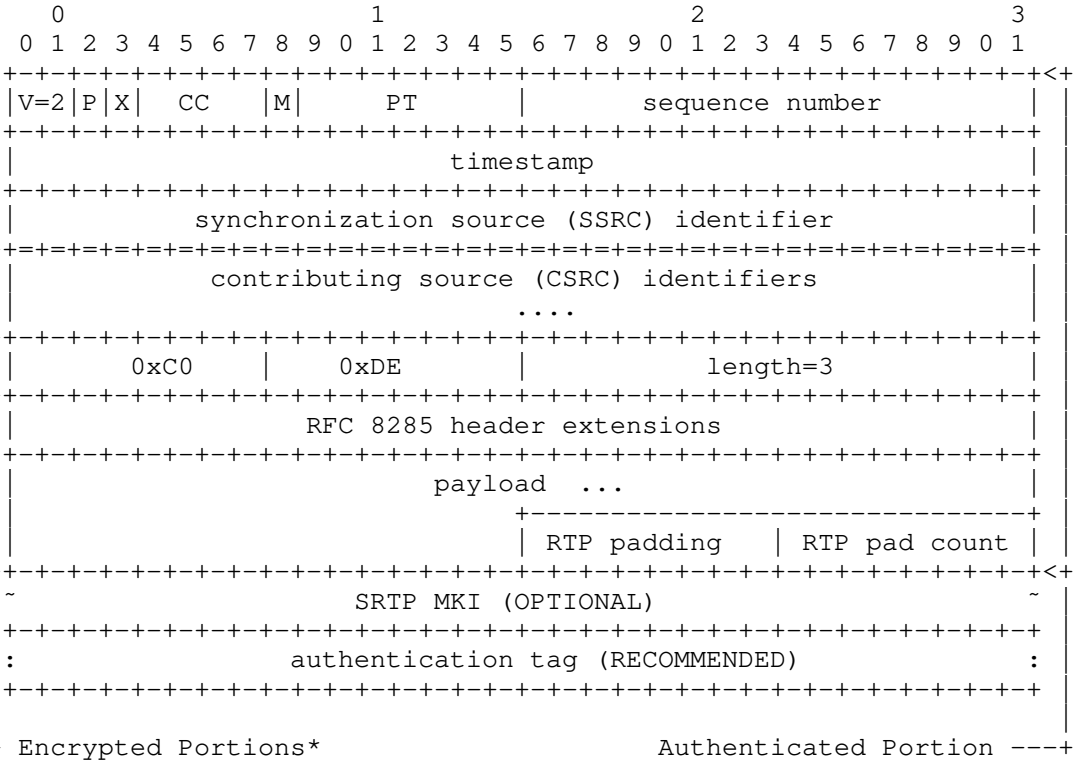
When receiving an RTP packet that contains header extensions, the "defined by profile" field MUST be checked to ensure the payload is formatted according to this specification. If the field does not match one of the values defined above, the implementation MUST instead handle it according to the specification that defines that value. The implemntation MAY stop and report an error if it considers use of this specification mandatory for the RTP stream.

If the RTP packet passes this check, it is then decrypted according to Decryption Procedure, and passed to the the next layer to process the packet and its extensions.

6. Encryption and Decryption

6.1. Packet Structure

When this mechanism is active, the SRTP packet is protected as follows:



* Note that the 4 bytes at the start of the extension block are not encrypted, as required by [RFC8285].

Specifically, the encrypted portion MUST include any CSRC identifiers, any RTP header extension (except for the first 4 bytes), and the RTP payload.

6.2. Encryption Procedure

The encryption procedure is identical to that of [RFC3711] except for the region to encrypt, which is as shown in the section above.

To minimize changes to surrounding code, the encryption mechanism can choose to replace a "defined by profile" field from [RFC8285] with its counterpart defined in RTP Header Processing above and encrypt at the same time.

6.3. Decryption Procedure

The decryption procedure is identical to that of [RFC3711] except for the region to decrypt, which is as shown in the section above.

To minimize changes to surrounding code, the decryption mechanism can choose to replace the "defined by profile" field with its no-encryption counterpart from [RFC8285] and decrypt at the same time.

7. Backwards Compatibility

This specification attempts to encrypt as much as possible without interfering with backwards compatibility for systems that expect a certain structure from an RTPv2 packet, including systems that perform demultiplexing based on packet headers. Accordingly, the first two bytes of the RTP packet are not encrypted.

This specification also attempts to reuse the key scheduling from SRTP, which depends on the RTP packet sequence number and SSRC identifier. Accordingly these values are also not encrypted.

8. Security Considerations

This specification extends SRTP by expanding the portion of the packet that is encrypted, as shown in Packet Structure. It does not change how SRTP authentication works in any way. Given that more of the packet is being encrypted than before, this is necessarily an improvement.

The RTP fields that are left unencrypted (see rationale above) are as follows:

* RTP version
* padding bit
* extension bit
* number of CSRCs
* marker bit
* payload type

* sequence number

* timestamp

* SSRC identifier

* number of [RFC8285] header extensions

These values contain a fixed set (i.e., one that won't be changed by extensions) of information that, at present, is observed to have low sensitivity. In the event any of these values need to be encrypted, SRTP is likely the wrong protocol to use and a fully-encapsulating protocol such as DTLS is preferred (with its attendant per-packet overhead).

9. IANA Considerations

This document defines two new 'defined by profile' attributes, as noted in RTP Header Processing.

10. Acknowledgements

The authors wish to thank Sergio Murillo, Jonathan Lennox, and Inaki Castillo for their review and text suggestions.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, DOI 10.17487/RFC3711, March 2004, <https://www.rfc-editor.org/info/rfc3711>.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, DOI 10.17487/RFC4566, July 2006, <https://www.rfc-editor.org/info/rfc4566>.

[RFC8285] Singer, D., Desineni, H., and R. Even, Ed., "A General Mechanism for RTP Header Extensions", RFC 8285, DOI 10.17487/RFC8285, October 2017, <https://www.rfc-editor.org/info/rfc8285>.

11.2. Informative References

[RFC6464] Lennox, J., Ed., Ivov, E., and E. Marocco, "A Real-time Transport Protocol (RTP) Header Extension for Client-to-Mixer Audio Level Indication", RFC 6464, DOI 10.17487/RFC6464, December 2011, <https://www.rfc-editor.org/info/rfc6464>.

[RFC6465] Ivov, E., Ed., Marocco, E., Ed., and J. Lennox, "A Real-time Transport Protocol (RTP) Header Extension for Mixer-to-Client Audio Level Indication", RFC 6465, DOI 10.17487/RFC6465, December 2011, <https://www.rfc-editor.org/info/rfc6465>.

[RFC6904] Lennox, J., "Encryption of Header Extensions in the Secure Real-time Transport Protocol (SRTP)", RFC 6904, DOI 10.17487/RFC6904, April 2013, <https://www.rfc-editor.org/info/rfc6904>.

Authors' Addresses

Justin Uberti
Google

Email: justin@uberti.name

Cullen Jennings
Cisco

Email: fluffy@iii.ca