

AVTCORE
Internet-Draft
Intended status: Standards Track
Expires: September 9, 2021

S. Garcia Murillo
CoSMo Software
Y. Fablet
Apple Inc.
A. Gouaillard
CoSMo Software
March 08, 2021

Codec agnostic RTP payload format for video
draft-gouaillard-avtcore-codec-agn-rtp-payload-01

Abstract

RTP Media Chains usually rely on piping encoder output directly to packetizers. Media packetization formats often support a specific codec format and optimize RTP packets generation accordingly.

With the development of Selective Forward Unit (SFU) solutions, that do not process media content server side, the need for media content processing at the origin and at the destination has arised.

RTP Media Chains used e.g. in WebRTC solutions are increasingly relying on application-specific transforms that sit in-between encoder and packetizer on one end and in-between depacketizer and decoder on the other end. This use case has become so important, that the W3C is standardizing the capacity to access encoded content with the [WebRTCInsertableStreams] API proposal. An extremely popular use case is application level end-to-end encryption of media content, using for instance [SFrame].

Whatever the modification applied to the media content, RTP packetizers can no longer expect to use packetization formats that mandate media content to be in a specific codec format.

In the extreme cases like encryption, where the RTP Payload is made completely opaque to the SFUs, some extra mechanism must also be added for them to be able to route the packets without depending on RTP payload or payload headers.

The traditionnal process of creating a new RTP Payload specification per content would not be practical as we would need to make a new one for each codec-transform pair.

This document describes a solution, which provides the following features in the case the encoded content has been modified before reaching the packetizer: - a paylaod agnostic RTP packetization format that can be used on any media content, - a negotiation

mechanism for the above format and the inner payload, Both of the above mechanism are backward compatible with most of (S)RTP/RTCP mechanisms used for bandwidth estimation and congestion control in RTP/SRTP/webrtc, including but not limited to SSRC, RED, FEC, RTX, NACK, SR/RR, REMB, transport-wide-CC, TMBR, It as illustrated by existing implementations in chrome, safari, and Medooze.

This document also describes a solution to allow SFUs to continue performing packet routing on top of this generic RTP packetization format.

This document complements the SFrame (media encryption), and Dependency Descriptor (AV1 payload annex) documents to provide an End-to-End-Encryption solution that would sit on top of SRTP/Webrtc, use SFUs on the media back-end, and leverage W3C APIs in the browser. A high level description of such system will be provided as an informational I-D in the SFrame WG and then cited here.

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 September 9, 2021.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

| | |
|--|----|
| 1. Introduction | 3 |
| 2. Goals | 6 |
| 3. RTP Packetization | 7 |
| 4. Payload Multiplexing | 7 |
| 5. SDP Negotiation | 8 |
| 6. SFU Packet Selection | 9 |
| 7. Redundancy Techniques Considerations | 10 |
| 7.1. Retransmission Techniques | 10 |
| 7.2. Forward Error Correction (FEC) Techniques | 10 |
| 7.3. Redundant Audio Data Techniques | 10 |
| 8. Alternatives | 11 |
| 8.1. Generic Packetization With In-Payload APT | 11 |
| 8.2. A Payload Type for Generic Packetization AND Media Format | 11 |
| 8.3. A RTP Header To Choose Packetization | 13 |
| 9. Security Considerations | 13 |
| 10. IANA Considerations | 14 |
| 10.1. Registration of audio/generic | 14 |
| 11. Registration of video/generic | 14 |
| 12. References | 15 |
| 12.1. Normative References | 15 |
| 12.2. Informative References | 16 |
| Authors' Addresses | 17 |

1. Introduction

As per Figure 1 of [RFC7656], a Media Packetizer transforms a single Encoded Stream into one or several RTP packets. The Encoded Stream is coming straight from the Media Encoder and is expected to follow the format produced by the Media Encoder. A number of Media Packetizer formats have been designed to process a specific format produced by Media Encoder. For instance [RFC6184] is dedicated to the processing of content produced by H.264 Media Encoders, and generates packets following NALUs organization.

WebRTC applications are increasingly deploying end-to-end encryption solutions on top of RTP Media Chains. End-to-end encryption is implemented by inserting application-specific Media Transformers between Media Encoder and Media Packetizer on the sending side, and between Media Depacketizer and Media Decoder on the receiving side, as described in Figure 1 and Figure 2. To support end-to-end encryption, Media Transformers can use the [SFrame] format. In browsers, Media Transformers are implemented using

[WebRTCInsertableStreams], for instance by injecting JavaScript code provided by web pages.

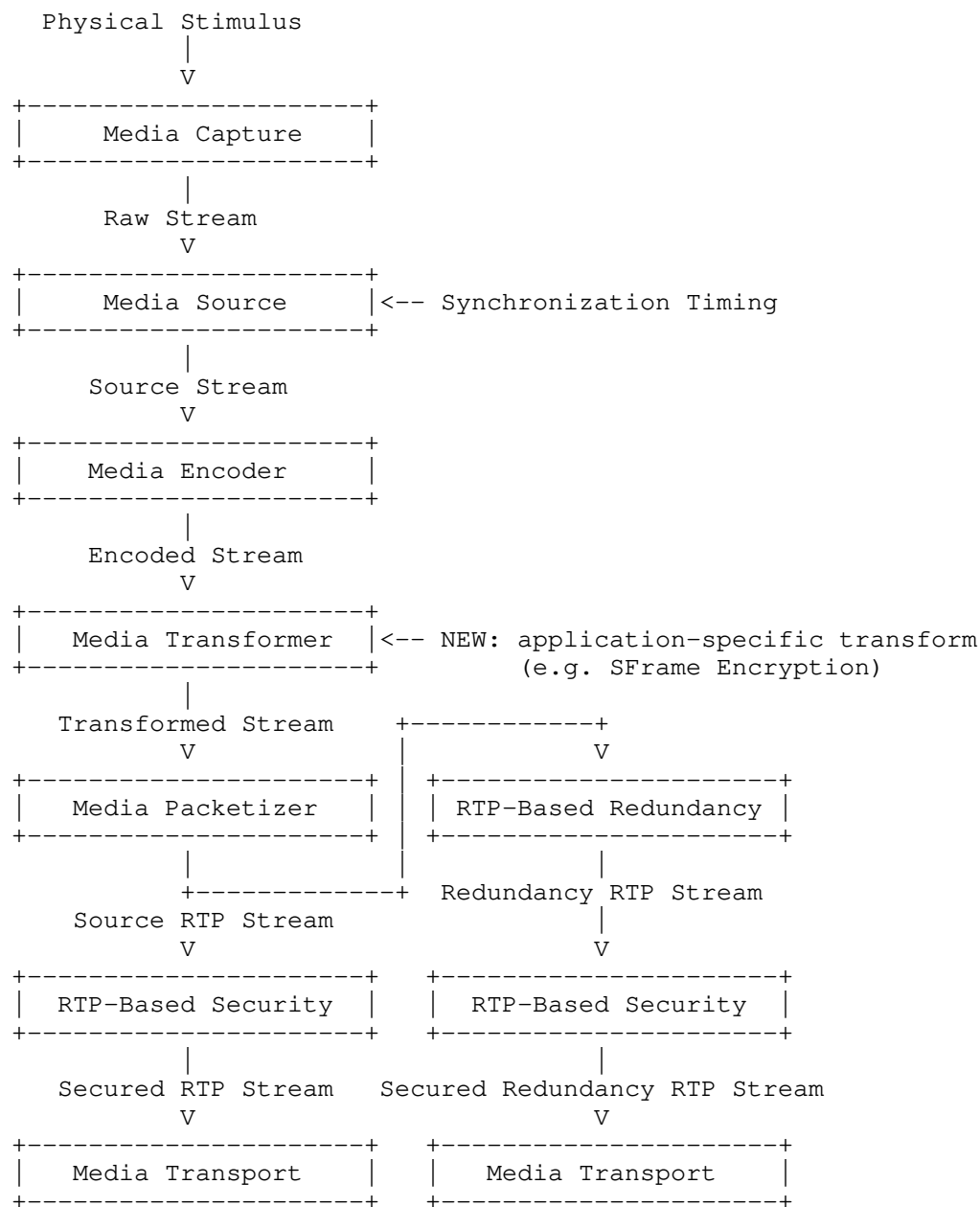
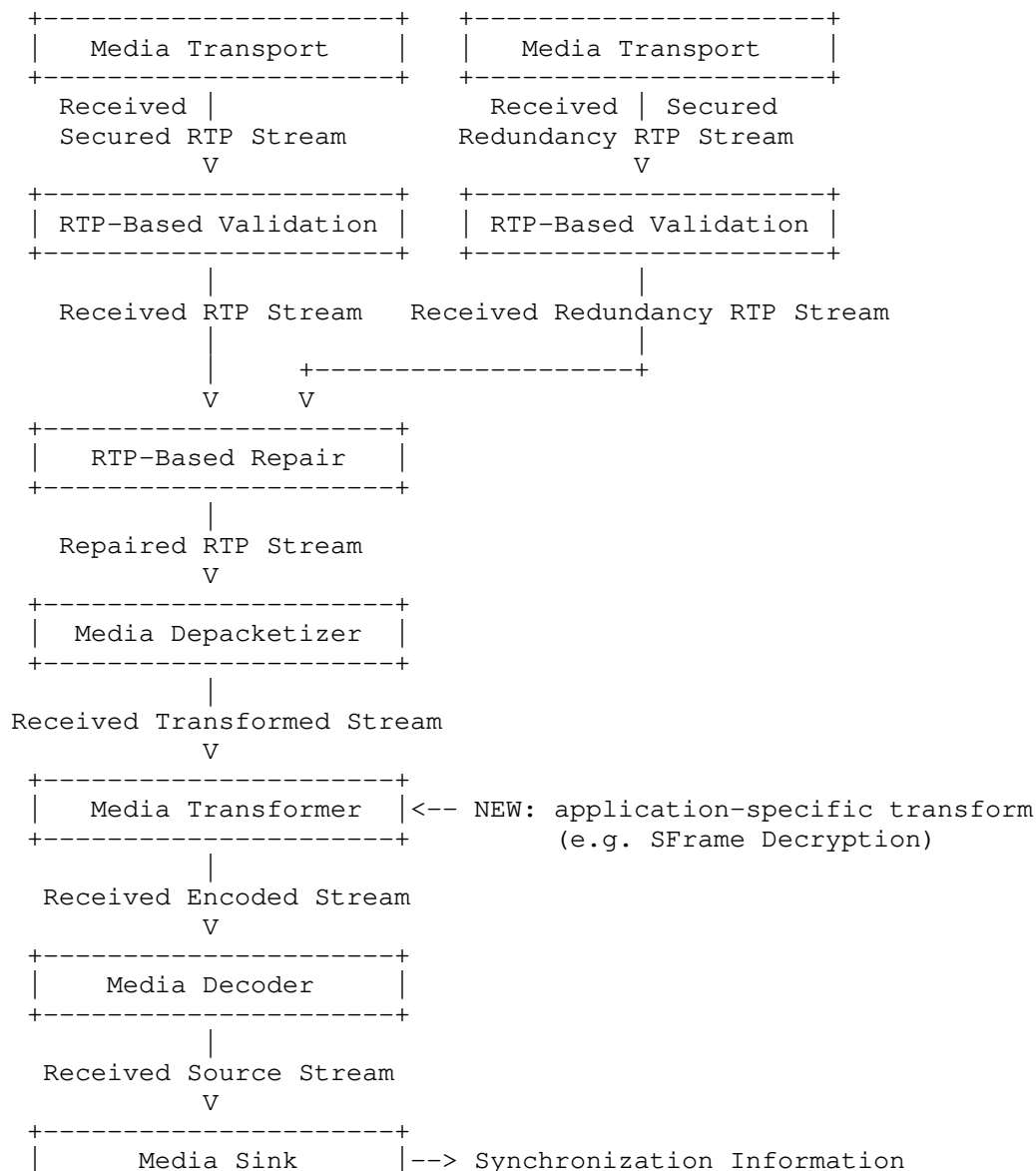


Figure 1: Sender Side Concepts in the Media Chain
With Application-level Media Transform

These RTP packets are sent over the wire to a receiver media chain matching the sender side, reaching the Media Depacketizer that will reconstruct the Encoded Stream before passing it to the Media Decoder.



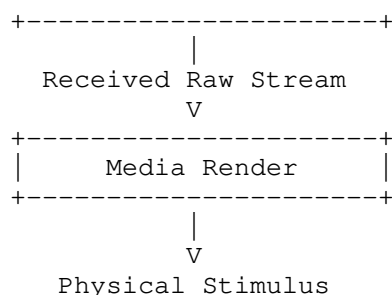


Figure 2: Receiver Side Concepts in the Media Chain
With Application-level Media Transform

This generic packetization does not change how the mapping between one or several encoded or dependant streams are mapped to the RTP streams or how the synchronization sources(s) (SSRC) are assigned.

Given the use of post-encoder application-specific transforms, the whole Media Chain needs to be made aware of it. This includes the sender post-transform Media Chain, Media Transport intermediaries (SFUs typically) and receiver pre-transform Media Chain.

As these transforms can alter Encoded Streams in any possible way, the use of codec-specific Media Packetizers like [RFC6184] on Transformed Stream may be suboptimal on sender side. It may also be problematic on the receiving side in case codec-specific processing is done prior the Media Transformer. Media Transport intermediaries are often looking at the Media Content itself to fuel their packet selection algorithms.

2. Goals

The objective of this document is to support inserting any application-specific transform between encoders and packetizers in the Media Chain. For that purpose, this document will: 1. Provide a generic packetization format that supports any media content (compressed audio, compressed video, encrypted content...) that allows reuse of existing RTP mechanisms in place in WebRTC applications such as RTX, RED or FEC. 2. Provide a way to negotiate use of the generic packetization format between sender and receiver, with minimum impact on existing negotiation approaches. 3. Provide a side-channel information so that network intermediaries (SFU in particular) can do their existing packet routing strategies without inspecting the media content.

3. RTP Packetization

A generic packetizer, by design, is not expected to understand the format of the media to transmit. The unit used by the packetizer to do processing is called a frame in the remainder of the document.

It is the responsibility of the application using the packetizer to group media content in meaningful frames. In the common case of a video codec, the packetizer frame is the frame in byte format (h264 annex b for example) generated by the encoder.

If the application wants to transform encoded content, the application needs to split the encoded content into frames prior the transform. Each frame is then transformed independently, for instance encrypted using [SFrame]. The content of each transformed frame is then processed by the packetizer.

In the case of a video codec supporting spatial scalability, each spatial layer **MUST** be split in its own frame by the application before passing it to the packetizer.

When the packetizer receives a frame from the application, it **MUST** fragment the frame content in multiple RTP packets to ensure packets do not exceed the network maximum transmission unit. The content of the frame will be treated as a binary blob by the packetizer, so the decision about the boundaries of each fragment is decided arbitrarily by the packetizer. The packetizer or any relaying server **MUST NOT** modify the frame content and concatenating the RTP payload of the RTP packets for each frame **MUST** produce the exact binary content of the input frame content.

The marker bit of each RTP packet in a frame **MUST** be set according to the audio and video profiles specified in [RFC3551].

The spatial layer frames are sent in ascending order, with the same RTP timestamp, and only the last RTP packet of the last spatial layer frame will have the marker bit set to 1.

4. Payload Multiplexing

In order to reduce the number of payload type in the SDP exchange, a single payload type code for the generic packetization can be used for all negotiated media formats. That requires to identify the original payload type code of the frame negotiated media format, called the associated payload type (APT) hereunder. The APT value is the payload type code of the associated format passed to the generic Media Packetizer before any transformation is applied.

The APT value is sent in a dedicated header extension. The payload of this header extension can be encoded using either the one-byte or two-byte header defined in [RFC5285]. Figures 3 and 4 show examples with each one of these examples.

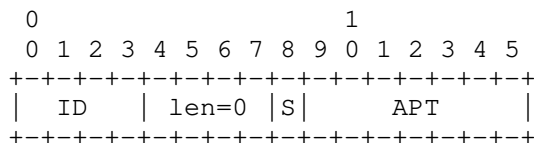


Figure 3: Frame Associated Payload Type Encoding Using the One-Byte Header Format

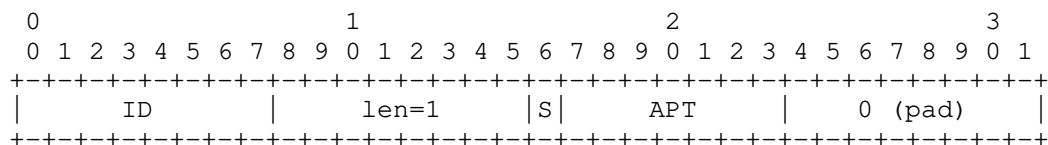


Figure 4: Frame Associated Payload Type Encoding Using the Two-Byte Header Format

The APT value is the associated payload type value. The S bit indicates if the media stream can be forwarded safely starting from this RTP packet. Typically, it will be set to 1 on the first RTP packet of an intra video frame and in all RTP audio packets.

Receivers MUST be ready to receive RTP packets with different associated payload types in the same way they would receive different payload type codes on the RTP packets.

The URI for declaring this header extension in an extmap attribute is "urn:ietf:params:rtp-hdrex:associated-payload-type".

5. SDP Negotiation

To use the RTP generic packetization, the SDP Offer/Answer exchange MUST negotiate: - The payload type of the negotiated codec format - The generic payload type - The associated payload type header extension

Only the negotiated payload types are allowed to be used as associated payload types. Figure 5 illustrates a SDP that negotiates exchange of video using either VP8 or VP9 codecs with the possibility to use the generic packetization. In this example, RTX is also negotiated and will be applied normally on each associated payload type.


```

m=video 9 UDP/TLS/RTP/SAVPF 96 97 98 99 100 101
c=IN IP4 0.0.0.0
a=rtcp:9 IN IP4 0.0.0.0
a=setup:actpass
a=mid:1
a=extmap:1 urn:ietf:params:rtp-hdext:sdes:mid
a=extmap:2 urn:ietf:params:rtp-hdext:sdes:rtp-stream-id
a=extmap:3 urn:ietf:params:rtp-hdext:sdes:repaired-rtp-stream-id
a=extmap:4 urn:ietf:params:rtp-hdext:associated-payload-type
a=sendrecv
a=rtpmap:96 vp9/90000
a=rtpmap:97 vp8/90000
a=rtpmap:98 generic/90000
a=rtpmap:99 rtx/90000
a=fmtp:99 apt=96
a=rtpmap:100 rtx/90000
a=fmtp:100 apt=97
a=rtpmap:101 rtx/90000
a=fmtp:101 apt=98

```

Figure 5: SDP example negotiating the generic payload type and related header extension for video

6. SFU Packet Selection

SFUs need to have a basic understanding of each frame they receive so they can decide to forward it or not and to which endpoint. They might need similar information to support media content recording. This information is either generic to a group of frame (called a stream hereafter) or specific to each frame.

The information is transmitted as a RTP header extension as the RTP packet payload should be treated as opaque by the SFU. This is especially necessary if the payload is end-to-end encrypted. The amount of information should be limited to what is strictly necessary to the SFU task since it is not always as trusted as individual peers.

For audio, configuration information such as Opus TOC might be useful. For video, configuration information might include: - Stream configuration information: resolution, quality, frame rate... - Codec specific configuration information: codec profile like profile_idc... - Frame specific information: whether the stream is decodable when starting from this frame, whether the frame is skippable...

For video content, this information can be sent using a Dependency Descriptor header extension. In that case, the first RTP packet of

the frame will have its `start_of_frame` equal to 1 and the last packet will have its `end_of_frame` equal to 1.

7. Redundancy Techniques Considerations

The solution described in this document is expected to integrate well with the existing RTP ecosystem. This section describes how the generic packetizer can be used jointly with existing techniques that allow to mitigate unreliable transports.

7.1. Retransmission Techniques

[RFC4588] defines a retransmission payload format (RTX) that can be used in case of packet loss. As defined in [RFC4588], RTX is able to handle any payload format, including the format described in this document. Given RTX preserves both RTP packet payload and headers, the receiver will be able to identify the payload type of the recovered packet and whether generic packetization is used. RTX will also allow recovering RTP header extensions that convey information on the media content itself.

7.2. Forward Error Correction (FEC) Techniques

FEC is another technique used in RTP Media Chains to protect media content against packet loss. [RFC5109] defines such a payload format used to transmit FEC for specific packets protection.

FEC may protect some parts of the media content more than others. For instance, intra video frame encoded data or important network abstraction layer units (NALUs) like SPS/PPS may be more protected. With a post-encoder transform and the use of a generic packetization, the granularity of the recovery mechanism is no longer at the NALU level but at the level of the frame generated by the post-encoder transform. In case a SVC codec is used, each spatial layer will be processed as an independent frame. In that case, base layers can be protected more heavily than higher resolution layers.

7.3. Redundant Audio Data Techniques

As defined in [RFC7656] RTP-based redundancy is defined here as a transformation that generates redundant or repair packets sent out as a Redundancy RTP Stream to mitigate Network Transport impairments, like packet loss and delay.

[RFC2198] defines a payload format for sending the same audio data encoded multiple times at different quality levels. This allows to use a lower quality encoding of the audio data, should the higher quality encoding of the audio data is lost during the transmission.

If a Media Transformation is in use, both the primary and redundant encoding must be transformed independently and the redundant packet created normally. As the RTP headers present in the redundant packet are only applicable to the primary encoding, if the payload type for a redundant encoding block is mapped to the generic packetizer, the value of the associated payload type for the primary encoding is applied to the redundant encoding block as well.

8. Alternatives

Various alternatives can be used to implement and negotiate generic packetization. This section describes a few additional alternatives. This section is to be removed before finalization of the document.

8.1. Generic Packetization With In-Payload APT

Instead of using a RTP header extension to convey the APT value, it is prepended in the RTP payload itself. As the value cannot change for a whole frame, its value is prepended to the first packet generated of the frame only. This removes the need to negotiate a dedicated header extension, but may require the SFU to update the payload when sending or recording content.

8.2. A Payload Type for Generic Packetization AND Media Format

The payload type is negotiated in the SDP so as to identify both the negotiated codec format and the generic packetization use. There is no network cost but this increases the number of payload types used in the SDP.

```

m=video 9 UDP/TLS/RTP/SAVPF 96 97 98 99 100 101
c=IN IP4 0.0.0.0
a=rtcp:9 IN IP4 0.0.0.0
a=setup:actpass
a=mid:1
a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:mid
a=extmap:2 urn:ietf:params:rtp-hdrext:sdes:rtp-stream-id
a=extmap:3 urn:ietf:params:rtp-hdrext:sdes:repaired-rtp-stream-id
a=sendrecv
a=rtpmap:96 vp9/90000
a=rtpmap:97 generic/90000
a=fmtp:97 apt=96
a=rtpmap:98 vp8/90000
a=rtpmap:99 generic/90000
a=fmtp:99 apt=98
a=rtpmap:100 rtx/90000
a=fmtp:100 apt=96
a=rtpmap:101 rtx/90000
a=fmtp:101 apt=97
a=rtpmap:102 rtx/90000
a=fmtp:102 apt=98
a=rtpmap:103 rtx/90000
a=fmtp:103 apt=99

```

Figure 6: SDP example negotiating a payload type for format and generic packetization

A variation of this approach is to consider defining generic payload types, each of them having an identified codec format.

```

m=video 9 UDP/TLS/RTP/SAVPF 96 97 98 99 100 101
c=IN IP4 0.0.0.0
a=rtcp:9 IN IP4 0.0.0.0
a=setup:actpass
a=mid:1
a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:mid
a=extmap:2 urn:ietf:params:rtp-hdrext:sdes:rtp-stream-id
a=extmap:3 urn:ietf:params:rtp-hdrext:sdes:repaired-rtp-stream-id
a=sendrecv
a=rtpmap:96 generic/90000
a=fmtp:96 codec=vp9
a=rtpmap:97 generic/90000
a=fmtp:97 codec=vp8
a=rtpmap:98 rtx/90000
a=fmtp:98 apt=96
a=rtpmap:99 rtx/90000
a=fmtp:99 apt=97

```

Figure 7: SDP example negotiating a payload type for format and generic packetization

8.3. A RTP Header To Choose Packetization

A RTP header extension can be used to flag content as opaque so that the receiver knows whether to use or not the generic packetization. As for the API header extension, the RTP header extension may not need to be sent for every packet, it could for instance be sent for the first packet of every intra video frame. The main advantage of this approach is the reduced impact on SDP negotiation.

```
m=video 9 UDP/TLS/RTP/SAVPF 96 97 98 99 100 101
c=IN IP4 0.0.0.0
a=rtcp:9 IN IP4 0.0.0.0
a=setup:actpass
a=mid:1
a=extmap:1 urn:ietf:params:rtp-hdext:sdes:mid
a=extmap:2 urn:ietf:params:rtp-hdext:sdes:rtp-stream-id
a=extmap:3 urn:ietf:params:rtp-hdext:sdes:repaired-rtp-stream-id
a=extmap:4 urn:ietf:params:rtp-hdext:generic-packetization-use
a=sendrecv
a=rtpmap:96 vp9/90000
a=rtpmap:97 vp8/90000
a=rtpmap:98 rtx/90000
a=fmtp:98 apt=96
a=rtpmap:99 rtx/90000
a=fmtp:99 apt=97
```

Figure 8: SDP example negotiating generic packetization as RTP header extension

9. Security Considerations

RTP packets using the payload format defined in this specification are subject to the general security considerations discussed in [RFC3550]. It is not expected that the proposed solutions (generic packetization and header extension) presented in this document can create new security threats. The use and implementation of RTP Media Chains containing Media Transformers needs to be done carerefully. It is important to refer to the security considerations discussed in [SFrame] and [WebRTCInsertableStreams]. In particular Media Transformers on the receiver side need to be prepared to receive arbitrary content, like decoders already do. Similarly, since Media Transformers can be implemented as JavaScript in browsers, RTP Packetizers should be prepared to receive arbitrary content.

10. IANA Considerations

Two new media subtypes have been registered with IANA, as described in this section.

10.1. Registration of audio/generic

Type name: audio

Subtype name: generic

Required parameters: none

Optional parameters: none

Encoding considerations: This format is framed (see Section 4.8 in the template document) and contains binary data.

Security considerations: TBD.

Interoperability considerations: TBD

Published specification: TBD.

Applications that use this media type: TBD.

Additional information: none

Intended usage: COMMON

Restrictions on usage: TBD

Author:

Change controller:

11. Registration of video/generic

Type name: video

Subtype name: generic

Required parameters: none

Optional parameters: none

Encoding considerations: This format is framed (see Section 4.8 in the template document) and contains binary data.

Security considerations: TBD.

Interoperability considerations: TBD

Published specification: TBD.

Applications that use this media type: TBD.

Additional information: none

Intended usage: COMMON

Restrictions on usage: TBD

Author:

Change controller:

12. References

12.1. 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>>.
- [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>>.

- [RFC5285] Singer, D. and H. Desineni, "A General Mechanism for RTP Header Extensions", RFC 5285, DOI 10.17487/RFC5285, July 2008, <<https://www.rfc-editor.org/info/rfc5285>>.
- [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>>.
- [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>>.

12.2. Informative References

- [RFC2198] Perkins, C., Kouvelas, I., Hodson, O., Hardman, V., Handley, M., Bolot, J., Vega-Garcia, A., and S. Fosse-Parisis, "RTP Payload for Redundant Audio Data", RFC 2198, DOI 10.17487/RFC2198, September 1997, <<https://www.rfc-editor.org/info/rfc2198>>.
- [RFC4588] Rey, J., Leon, D., Miyazaki, A., Varsa, V., and R. Hakenberg, "RTP Retransmission Payload Format", RFC 4588, DOI 10.17487/RFC4588, July 2006, <<https://www.rfc-editor.org/info/rfc4588>>.
- [RFC5109] Li, A., Ed., "RTP Payload Format for Generic Forward Error Correction", RFC 5109, DOI 10.17487/RFC5109, December 2007, <<https://www.rfc-editor.org/info/rfc5109>>.
- [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>>.
- [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>>.

[SFrame] "Secure Frame (SFrame)", n.d., <<https://tools.ietf.org/html/draft-omara-sframe>>.

[WebRTCInsertableStreams] "WebRTC Insertable Media using Streams", n.d., <<https://w3c.github.io/webrtc-insertable-streams>>.

Authors' Addresses

Sergio Garcia Murillo
CoSMo Software

Email: sergio.garcia.murillo@cosmosoftware.io

Youenn Fablet
Apple Inc.

Email: youenn@apple.com

Alexandre Gouaillard
CoSMo Software

Email: alex.gouaillard@cosmosoftware.io

AVTCORE
Internet-Draft
Intended status: Standards Track
Expires: 8 September 2022

J. Uberti
Clubhouse
C. Jennings
Cisco
S. Garcia Murillo
CoSMo
7 March 2022

Completely Encrypting RTP Header Extensions and Contributing Sources
draft-ietf-avtcore-cryptex-05

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 defines Cryptex as a new mechanism that completely encrypts header extensions and CSRCs and uses simpler signaling with the goal of facilitating deployment.

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 8 September 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

| | |
|---|----|
| 1. Introduction | 3 |
| 1.1. Problem Statement | 3 |
| 1.2. Previous Solutions | 3 |
| 1.3. Goals | 4 |
| 2. Terminology | 5 |
| 3. Design | 5 |
| 4. Signaling | 5 |
| 5. RTP Header Processing | 6 |
| 5.1. Sending | 6 |
| 5.2. Receiving | 7 |
| 6. Encryption and Decryption | 7 |
| 6.1. Packet Structure | 7 |
| 6.2. Encryption Procedure | 8 |
| 6.3. Decryption Procedure | 9 |
| 7. Backwards Compatibility | 9 |
| 8. Security Considerations | 9 |
| 9. IANA Considerations | 10 |
| 9.1. cryptex SDP Attribute | 10 |
| 10. Acknowledgements | 11 |
| 11. References | 11 |
| 11.1. Normative References | 11 |
| 11.2. Informative References | 12 |
| Appendix A. Test Vectors | 12 |
| A.1. AES-CTR | 12 |
| A.1.1. RTP Packet with 1-byte header extension | 12 |
| A.1.2. RTP Packet with 2-byte header extension | 13 |
| A.1.3. RTP Packet with 1-byte header extension and CSRC fields | 14 |
| A.1.4. RTP Packet with 2-byte header extension and CSRC fields | 15 |
| A.1.5. RTP Packet with empty 1-byte header extension and CSRC fields | 15 |
| A.1.6. RTP Packet with empty 2-byte header extension and CSRC fields | 16 |
| A.2. AES-GCM | 17 |
| A.2.1. RTP Packet with 1-byte header extension | 17 |
| A.2.2. RTP Packet with 2-byte header extension | 17 |

| | | |
|--------------------|---|----|
| A.2.3. | RTP Packet with 1-byte header extension and CSRC fields | 18 |
| A.2.4. | RTP Packet with 2-byte header extension and CSRC fields | 19 |
| A.2.5. | RTP Packet with empty 1-byte header extension and CSRC fields | 20 |
| A.2.6. | RTP Packet with empty 2-byte header extension and CSRC fields | 21 |
| Authors' Addresses | | 21 |

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 the endpoint and/or application. Accordingly, these identifiers can be considered a fingerprinting issue.

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", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [BCP14] RFC2119 RFC8174 when, and only when, they appear in all capitals, as shown here.

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 the mechanism has been negotiated.

4. Signaling

In order to determine whether the mechanism defined in this specification is supported, this document defines a new "a=cryptex" 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.

The presence of this attribute in the SDP (either in an offer or answer) indicates that the endpoint is capable of receiving RTP packets encrypted with Cryptex, as defined below.

Once each peer has verified that the other party supports receiving RTP packets encrypted with Cryptex, senders can unilaterally decide whether to use the Cryptex mechanism or not.

If BUNDLE is in use and the a=cryptex attribute is present for a media line, it MUST be present for all media lines belonging to the same bundle group. This ensures that the encrypted MID header extensions used to demux BUNDLE can be processed correctly. When used with BUNDLE, this attribute is assigned to the TRANSPORT category [RFC8859].

It is possible to signal and negotiate both Encryption of Header Extensions as defined in [RFC6904] and cryptex in the SDP O/A, however if a packet is encrypted with cryptex, it MUST NOT use the [RFC6904] header extension encryption mechanisms.

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 the mechanism defined by this specification has been negotiated, sending a RTP packet that has any CSRCs or contains any {RFC8285} header extensions follows the steps below. This mechanism MUST NOT be used with header extensions other than the [RFC8285] variety.

If the packet contains solely one-byte extension ids, the 16-bit RTP header extension tag MUST be set to 0xC0DE to indicate that the encryption has been applied, and the one-byte framing is being used. If the packet contains only two-byte extension ids, the header extension tag MUST be set to 0xC2DE to indicate encryption has been applied, and the two-byte framing is being used.

If the packet contains CSRCs but no header extensions, an empty extension block consisting of the 0xC0DE tag and a 16-bit length field set to zero (explicitly permitted by [RFC3550]) MUST be appended, and the X bit MUST be set to 1 to indicate an extension block is present. This is necessary to provide the receiver an indication that the CSRCs in the packet are encrypted.

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

5.2. Receiving

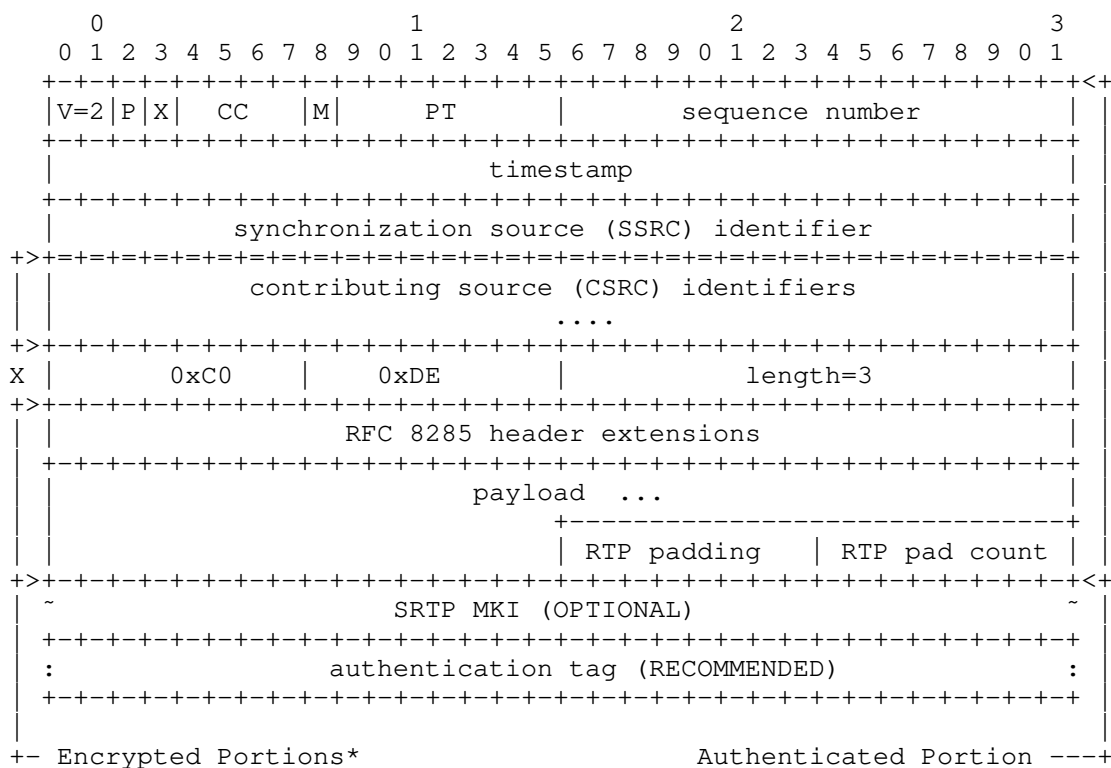
When receiving an RTP packet that contains header extensions, the "defined by profile" field MUST be checked to ensure the payload is formatted according to this specification. If the field does not match one of the values defined above, the implementation MUST instead handle it according to the specification that defines that value. The implementation 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. In the event that a zero-length extension block was added as indicated above, it can be left as-is and will be processed normally.

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.

For AEAD ciphers (e.g., GCM), the 12-byte fixed header and the four-byte header extension header (the "defined by profile" field and the length) are considered AAD, even though they are non-contiguous in the packet if CSRCs are present.

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

9.1. cryptex SDP Attribute

This document updates the "Session Description Protocol Parameters" registry as specified in Section 8.2.4 of [RFC8866]. Specifically, it adds the SDP 'cryptex' attribute to the table for SDP media-level attributes.

Contact name: TBD

Contact email address: TBD

Attribute name: cryptex

Attribute syntax: This attribute takes no values.

Attribute semantics: N/A

Attribute value: N/A

Usage level: media-level

Charset dependent: No

Purpose: The presence of this attribute in the SDP indicates that the endpoint is capable of receiving RTP packets encrypted with Cryptex as described in this document. O/A procedures: SDP O/A procedures are described in Section 4 of this document.

Mux Category: TRANSPORT

10. Acknowledgements

The authors wish to thank Lennart Grahl for pointing out many of the issues with the existing header encryption mechanism, as well as suggestions for this proposal. Thanks also to Jonathan Lennox, Inaki Castillo, and Bernard Aboba for their review and suggestions.

11. References

11.1. Normative References

- [BCP14] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, May 2017.
- <<https://www.rfc-editor.org/info/bcp14>>
- [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>>.
- [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>>.
- [RFC8859] Nandakumar, S., "A Framework for Session Description Protocol (SDP) Attributes When Multiplexing", RFC 8859, DOI 10.17487/RFC8859, January 2021, <<https://www.rfc-editor.org/info/rfc8859>>.
- [RFC8866] Begen, A., Kyzivat, P., Perkins, C., and M. Handley, "SDP: Session Description Protocol", RFC 8866, DOI 10.17487/RFC8866, January 2021, <<https://www.rfc-editor.org/info/rfc8866>>.

11.2. Informative References

- [RFC6464] Lennox, J., Ed., Iovov, 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] Iovov, 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>>.

Appendix A. Test Vectors

All values are in hexadecimal and represented in network order (big endian).

A.1. AES-CTR

Common values are organized as follows:

| | |
|---------------------|--|
| Rollover Counter: | 00000000 |
| Master Key: | e1f97a0d3e018be0d64fa32c06de4139 |
| Master Salt: | 0ec675ad498afeebb6960b3aabe6 |
| Crypto Suite: | AES_CM_128_HMAC_SHA1_80 |
| Session Key: | c61e7a93744f39ee10734afe3ff7a087 |
| Session Salt: | 30cbbc08863d8c85d49db34a9ae1 |
| Authentication Key: | cebe321f6ff7716b6fd4ab49af256a156d38baa4 |

A.1.1. RTP Packet with 1-byte header extension

RTP Packet:

900f1235
decafbad
cafebabe
bede0001
51000200
abababab
abababab
abababab
abababab

Encrypted RTP Packet:

900f1235
decafbad
cafebabe
c0de0001
eb923652
51c3e036
f8de27e9
c27ee3e0
b4651d9f
bc4218a7
0244522f
34a5

A.1.2. RTP Packet with 2-byte header extension

RTP Packet:

900f1236
decafbad
cafebabe
10000001
05020002
abababab
abababab
abababab
abababab

Encrypted RTP Packet:

```
900f1236
decafbad
cafebabe
c2de0001
4ed9cc4e
6a712b30
96c5ca77
339d4204
ce0d7739
6cab6958
5fbce381
94a5
```

A.1.3. RTP Packet with 1-byte header extension and CSRC fields

RTP Packet:

```
920f1238
decafbad
cafebabe
0001e240
0000b26e
bede0001
51000200
abababab
abababab
abababab
abababab
```

Encrypted RTP Packet:

```
920f1238
decafbad
cafebabe
8bb6e12b
5cff16dd
c0de0001
92838c8c
09e58393
e1de3a9a
74734d67
45671338
c3acf11d
a2df8423
bee0
```

A.1.4. RTP Packet with 2-byte header extension and CSRC fields

RTP Packet:

```
920f1239
decafbad
cafebabe
0001e240
0000b26e
10000001
05020002
abababab
abababab
abababab
abababab
```

Encrypted RTP Packet:

```
920f1239
decafbad
cafebabe
f70e513e
b90b9b25
c2de0001
bbed4848
faa64466
5f3d7f34
125914e9
f4d0ae92
3c6f479b
95a0f7b5
3133
```

A.1.5. RTP Packet with empty 1-byte header extension and CSRC fields

RTP Packet:

```
920f123a
decafbad
cafebabe
0001e240
0000b26e
bede0000
abababab
abababab
abababab
abababab
```


Encrypted RTP Packet:

```
920f123a
decafbad
cafebabe
7130b6ab
fe2ab0e3
c0de0000
e3d9f64b
25c9e74c
b4cf8e43
fb92e378
1c2c0cea
b6b3a499
a14c
```

A.1.6. RTP Packet with empty 2-byte header extension and CSRC fields

RTP Packet:

```
920f123b
decafbad
cafebabe
0001e240
0000b26e
10000000
abababab
abababab
abababab
abababab
```

Encrypted RTP Packet:

```
920f123b
decafbad
cafebabe
cbf24c12
4330e1c8
c2de0000
599dd45b
c9d687b6
03e8b59d
771fd38e
88b170e0
cd31e125
eabe
```

A.2. AES-GCM

Common values are organized as follows:

| | |
|-------------------|----------------------------------|
| Rollover Counter: | 00000000 |
| Master Key: | 000102030405060708090a0b0c0d0e0f |
| Master Salt: | a0a1a2a3a4a5a6a7a8a9aaab |
| Crypto Suite: | AEAD_AES_128_GCM |
| Session Key: | 077c6143cb221bc355ff23d5f984a16e |
| Session Salt: | 9af3e95364ebac9c99c5a7c4 |

A.2.1. RTP Packet with 1-byte header extension

RTP Packet:

900f1235
decafbad
cafebabe
bede0001
51000200
abababab
abababab
abababab
abababab

Encrypted RTP Packet:

900f1235
decafbad
cafebabe
c0de0001
39972dc9
572c4d99
e8fc355d
e743fb2e
94f9d8ff
54e72f41
93bbc5c7
4ffab0fa
9fa0fbeb

A.2.2. RTP Packet with 2-byte header extension

RTP Packet:

900f1236
decafbad
cafebabe
10000001
05020002
abababab
abababab
abababab
abababab

Encrypted RTP Packet:

900f1236
decafbad
cafebabe
c2de0001
bb75a4c5
45cd1f41
3bdb7daa
2b1e3263
de313667
c9632490
81b35a65
f5cb6c88
b394235f

A.2.3. RTP Packet with 1-byte header extension and CSRC fields

RTP Packet:

920f1238
decafbad
cafebabe
0001e240
0000b26e
bede0001
51000200
abababab
abababab
abababab
abababab

Encrypted RTP Packet:

920f1238
decafbad
cafebabe
63bbccc4
a7f695c4
c0de0001
8ad7c71f
ac70a80c
92866b4c
6ba98546
ef913586
e95ffaaf
fe956885
bb0647a8
bc094ac8

A.2.4. RTP Packet with 2-byte header extension and CSRC fields

RTP Packet:

920f1239
decafbad
cafebabe
0001e240
0000b26e
10000001
05020002
abababab
abababab
abababab
abababab

Encrypted RTP Packet:

```
920f1239
decafbad
cafebabe
3680524f
8d312b00
c2de0001
c78d1200
38422bc1
11a7187a
18246f98
0c059cc6
bc9df8b6
26394eca
344e4b05
d80fea83
```

A.2.5. RTP Packet with empty 1-byte header extension and CSRC fields

RTP Packet:

```
920f123a
decafbad
cafebabe
0001e240
0000b26e
bede0000
abababab
abababab
abababab
abababab
```

Encrypted RTP Packet:

```
920f123a
decafbad
cafebabe
15b6bb43
37906fff
c0de0000
b7b96453
7a2b03ab
7ba5389c
e9331712
6b5d974d
f30c6884
dcb651c5
e120c1da
```

A.2.6. RTP Packet with empty 2-byte header extension and CSRC fields

RTP Packet:

```
920f123b
decafbad
cafebabe
0001e240
0000b26e
10000000
abababab
abababab
abababab
abababab
```

Encrypted RTP Packet:

```
920f123b
decafbad
cafebabe
dcb38c9e
48bf95f4
c2de0000
61ee432c
f9203170
76613258
d3ce4236
c06ac429
681ad084
13512dc9
8b5207d8
```

Authors' Addresses

Justin Uberti
Clubhouse
Email: justin@uberti.name

Cullen Jennings
Cisco
Email: fluffy@iii.ca

Sergio Garcia Murillo
CoSMo
Email: sergio.garcia.murillo@cosmosoftware.io

avtcore
Internet-Draft
Intended status: Standards Track
Expires: 8 August 2021

S. Zhao
S. Wenger
Tencent
Y. Lim
Samsung Electronics
4 February 2021

RTP Payload Format for Essential Video Coding (EVC)
draft-ietf-avtcore-rtp-evc-01

Abstract

This memo describes an RTP payload format for the video coding standard ISO/IEC International Standard 23094-1 [EVC], also known as Essential Video Coding [EVC] and developed by ISO/IEC JTC1/SC29/WG11 (MPEG). The RTP payload format allows for packetization of one or more Network Abstraction Layer (NAL) units in each RTP packet payload as well as fragmentation of a NAL unit into multiple RTP packets. The payload format has wide applicability in videoconferencing, Internet video streaming, and high-bitrate entertainment-quality video, among other applications.

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 8 August 2021.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document.

Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

| | |
|--|----|
| 1. Introduction | 3 |
| 1.1. Overview of the EVC Codec | 3 |
| 1.1.1. Coding-Tool Features (informative) | 4 |
| 1.1.2. Systems and Transport Interfaces | 6 |
| 1.1.3. Parallel Processing Support (informative) | 8 |
| 1.1.4. NAL Unit Header | 8 |
| 1.2. Overview of the Payload Format | 9 |
| 2. Conventions | 10 |
| 3. Definitions and Abbreviations | 10 |
| 3.1. Definitions | 10 |
| 3.1.1. Definitions from the EVC Specification | 10 |
| 3.1.2. Definitions Specific to This Memo | 12 |
| 3.2. Abbreviations | 13 |
| 4. RTP Payload Format | 14 |
| 4.1. RTP Header Usage | 15 |
| 4.2. Payload Header Usage | 16 |
| 4.3. Payload Structures | 17 |
| 4.3.1. Single NAL Unit Packets | 17 |
| 4.3.2. Aggregation Packets (APs) | 18 |
| 4.3.3. Fragmentation Units | 22 |
| 4.4. Decoding Order Number | 25 |
| 5. Packetization Rules | 26 |
| 6. De-packetization Process | 27 |
| 7. Payload Format Parameters | 29 |
| 7.1. Media Type Registration | 29 |
| 7.2. SDP Parameters | 29 |
| 7.2.1. Mapping of Payload Type Parameters to SDP | 29 |
| 7.2.2. Usage with SDP Offer/Answer Model | 30 |
| 7.2.3. SDP Example | 30 |
| 8. Use with Feedback Messages | 30 |
| 8.1. Picture Loss Indication (PLI) | 30 |
| 8.2. Full Intra Request (FIR) | 30 |
| 9. Security Considerations | 30 |
| 10. Congestion Control | 31 |
| 11. IANA Considerations | 32 |
| 12. Acknowledgements | 32 |
| 13. References | 32 |
| 13.1. Normative References | 32 |
| 13.2. Informative References | 34 |
| Authors' Addresses | 35 |

1. Introduction

The [EVC] specification, which is formally designated as ISO/IEC International Standard 23094-1 [ISO23094-1] has been published in October 2020. One goal of MPEG is to keep [EVC]'s Baseline profile essentially royalty free by using the technologies published more than 20 years or otherwise freely available for use, whereas more advanced profiles follow a reasonable and non-discriminatory licensing terms policy. Both Baseline profile and higher profiles of [EVC] are reported to provide coding efficiency gains over [HEVC] and [AVC] under certain configurations.

This memo describes an RTP payload format for [EVC]. It shares its basic design with the NAL unit-based RTP payload formats of H.264 Video Coding [RFC6184], Scalable Video Coding (SVC) [RFC6190], High Efficiency Video Coding (HEVC) [RFC7798], and Versatile Video Coding (VVC) [I-D.ietf-avtcore-rtp-vvc]. With respect to design philosophy, security, congestion control, and overall implementation complexity, it has similar properties to those earlier payload format specifications. This is a conscious choice, as at least RFC 6184 is widely deployed and generally known in the relevant implementer communities. Certain mechanisms known from [RFC6190] were incorporated as EVC supports temporal scalability. [EVC] currently does not offer higher forms of scalability.

1.1. Overview of the EVC Codec

[EVC], [AVC], [HEVC] and [VVC] share a similar hybrid video codec design. In this memo, we provide a very brief overview of those features of [EVC] that are, in some form, addressed by the payload format specified herein. Implementers have to read, understand, and apply the ISO/IEC specifications pertaining to [EVC] to arrive at interoperable, well-performing implementations. The EVC standard has a Baseline profile and on top of that, a Main profile, the latter including more advanced features. The syntax elements allow encoders to mark a bitstream as to what of the many independent coding tools are exercised in the bitstream, in a spirit similar to the `general_constraint_flags` of [VVC] is provided.

Conceptually, all [EVC], [AVC], [HEVC] and [VVC] include a Video Coding Layer (VCL), which is often used to refer to the coding-tool features, and a Network Abstraction Layer (NAL), which is often used to refer to the systems and transport interface aspects of the codecs.

1.1.1.1. Coding-Tool Features (informative)

Coding blocks and transform structure

[EVC] uses a traditional quad-tree coding structure, which divides the encoded image into blocks of up to 128x128 luma samples, which can be recursively divided into smaller blocks. The Main profile adds two advanced coding structure tools: Binary Ternary Tree (BTT) that allows non-square coding units and segmentation that changes the processing order of the segmentation unit from traditional left-scanning order processing to right-scanning order processing Unit Coding Order (SUCO). In the Main profile, the picture can be divided into slices and tiles, and these slices can be independently encoded and/or decoded in parallel.

When predicting a data block using intra prediction or inter prediction, the remaining data is usually added to the prediction block. The residual data is added to the prediction block. The residual data is obtained by applying an inverse quantization process and an inverse transform. [EVC] includes integer discrete cosine transform (DCT2) and scalar quantization. For the Main profile, Improved Quantization and Transform (IQT) uses a different mapping/clipping function for quantization. An inverse zig-zag scanning order is used for coefficient coding. Advanced Coefficient Coding (ADCC) in the Main profile can code coefficient values more efficiently, for example, indicated by the last non-zero coefficient. In Main profile, Adaptive Transformation Selection (ATS) is also available and can be applied to integer versions of DST7 or DCT8, and not just DCT2.

Entropy coding

[EVC] uses a similar binary arithmetic coding mechanism as [AVC]. The mechanism includes a binarization step and a probability update defined by a lookup table. In the Main profile, the derivation process of syntax elements based on adjacent blocks makes the context modeling and initialization process more efficient.

In-loop filtering

The Baseline profile of [EVC] uses the deblocking filter defined in H.263 Annex J. In the Main profile, compared to the deblocking filter in the Baseline profile, an Advanced Deblocking Filter (ADDB) can be used, which can further reduce artifacts. The Main profile also defines two additional in-loop filters that can be used to improve the quality of decoded pictures before output and/or for inter prediction. A Walsh-Hadamard Transform Domain Filter (HTDF) is applied to the luma samples before deblocking, and the scanning

process is used to determine 4 adjacent samples for filtering. An adaptive Loop Filter (ALF) allows to send signals of up to 25 different filters for the luma components, and the best filter can be selected through the classification process for each 4x4 block. The filter parameters of the ALF filter are signaled in the Adaptation Parameter Set (APS).

Inter-prediction

The basis of [EVC] inter prediction is motion compensation using interpolation filters with a quarter sample resolution. In Baseline profile, a motion vector signal is transmitted using one of three spatially neighboring motion vectors and a temporally collocated motion vector as a predictor. The motion vector difference may be signaled relative to the selected predictor, but for the case where no motion vector difference is signaled and there is no remaining data in the block, there is a specific mode called a skip mode. The Main profile includes six additional tools to provide improved inter prediction. With advanced Motion Interpolation and Signaling (AMIS), adjacent blocks can be conceptually merged to indicate that they use the same motion, but more advanced schemes can also be used to create predictions from the basic model list of candidate predictors. The Merge with Motion Vector Difference (MMVD) tool uses a process similar to the concept of merging neighboring blocks, but also allows the use of expressions that include a starting point, motion amplitude, and direction of motion to send a motion vector signal.

Using Advanced Motion Vector Prediction (AMVP), candidate motion vector predictions for the block can be derived from its neighboring blocks in the same picture and collocated blocks in the reference picture. The Adaptive Motion Vector Resolution (AMVR) tool provides a way to reduce the accuracy of a motion vector from a quarter sample to half sample, full sample, double sample, or quad sample, which provides the efficiency advantage, such as when sending large motion vector differences. The Main profile also includes the Decoder-side Motion Vector Refinement (DMVR), which uses a bilateral template matching process to refine the motion vectors in a bidirectional fashion.

Intra prediction and intra-coding

Intra prediction in [EVC] is performed on adjacent samples of coding units in a partitioned structure. For the Baseline profile, all coding units are square, and there are five different prediction modes: DC (mean value of the neighborhood), horizontal, vertical, and two different diagonal directions. In the Main profile, intra prediction can be applied to any rectangular coding unit, and there are 28 additional direction modes available in the so-called Enhanced

Intra Prediction Directions (EIPD). In the Main profile, an encoder can also use Intra Block Copy (IBC), where a previously decoded sample blocks of the same picture is used as a predictor. A displacement vector in integer sample precision is signaled to indicate where the prediction block in the current picture is used for this mode.

Decoded picture buffer management

In [EVC], decoded pictures can be stored in a decoded picture buffer (DPB) for predicting pictures that follow them in decoding order. In the Baseline profile, the management of the DPB (i.e. the process of adding and deleting reference pictures) is controlled by the information in the SPS. For the Main profile, if a Reference Picture List (RPL) scheme is used, DPB management can be controlled by information that is signaled at the picture level.

1.1.1.2. Systems and Transport Interfaces

[EVC] inherited the basic systems and transport interfaces designs from [AVC] and [HEVC]. These include the NAL-unit-based syntax structure, the hierarchical syntax and data unit structure and the Supplemental Enhancement Information (SEI) message mechanism. The hierarchical syntax and data unit structure consists of a sequence-level parameter set (SPS), two picture-level parameter sets (PPS and APS, each of which can apply to one or more pictures), slice-level header parameters, and lower-level parameters.

A number of key components that influenced the Network Abstraction Layer design of [EVC] as well as this memo are described below

Sequence parameter set

The Sequence Parameter Set (SPS) contains syntax elements pertaining to a coded video sequence (CVS), which is a group of pictures, starting with a random access point, and followed by pictures that may depend on each other and the random access point picture. In MPEG-2, the equivalent of a CVS was a Group of Pictures (GOP), which normally started with an I frame and was followed by P and B frames. While more complex in its options of random access points, EVC retains this basic concept. In many TV-like applications, a CVS contains a few hundred milliseconds to a few seconds of video. In video conferencing (without switching MCUs involved), a CVS can be as long in duration as the whole session.

Picture and adaptation parameter set

The Picture Parameter Set and the Adaptation Parameter Set (PPS and APS, respectively) carry information pertaining to a single picture. The PPS contains information that is likely to stay constant from picture to picture—at least for pictures for a certain type—whereas the APS contains information, such as adaptive loop filter coefficients, that are likely to change from picture to picture.

Profile, level and toolsets

Profiles and levels follow the same design considerations as known from [AVC], [HEVC], and in fact video codecs as old as MPEG-1 visual. A profile defines a set of tools (not to confuse with the "toolset" discussed below) that a decoder compliant with this profile has to support. In [EVC], profiles are defined in Annex A. Formally, they are defined as a set of constraints that a bitstream needs to conform to. In [EVC], the Baseline profile is much more severely constrained than Main profile, reducing implementation complexity. Levels relate to bitstream complexity in dimensions such as maximum sample decoding rate, maximum picture size, and similar parameters that are directly related to computational complexity.

Profiles and levels are signaled in the highest parameter set available, the SPS.

[EVC] contains another mechanism related to the use of coding tools, known as the toolset syntax element. This syntax element, `toolset_idc_h` and `toolset_idc_l` located in the SPS, is a bitmask that allows encoders to indicate which coding tools they are using, within the menu of profiles offered by the profile that is also signaled. No decoder conformance point is associated with the toolset, but a bitstream that were using a coding tool that is indicated as not used in the toolset syntax element would obviously be non-compliant. While MPEG specifically rules out the use of the toolset syntax element as a conformance point, walled garden implementations could do so without incurring the interoperability problems MPEG fears, and create bitstreams and decoders that do not support one or more given tools. That, in turn, may be useful to mitigate certain patent related risks.

Bitstream and elementary stream

Above the Coded Video Sequence (CVS), [EVC] defines a video bitstream that can be used in the MPEG systems context as an elementary stream. For the purpose of this memo, this is not relevant.

Random access support

[EVC] supports random access mechanism solely based on IDR access unit.

Temporal scalability support

[EVC] includes support for temporal scalability through the generalized reference picture selection approach known since [AVC]/SVC. Up to six temporal layers are supported. The temporal layer is signaled in the NAL unit header (which co-serves as the payload header in this memo), in the `nuh_temporal_id` field.

Reference picture management

placeholder

SEI Message

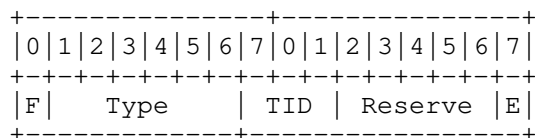
[EVC] inherits many of [HEVC]'s SEI Messages, occasionally with changes in syntax and/or semantics making them applicable to EVC.

1.1.3. Parallel Processing Support (informative)

Placeholder

1.1.4. NAL Unit Header

[EVC] maintains the NAL unit concept of [HEVC] with different parameter options. EVC also uses a two-byte NAL unit header, as shown in Figure 1. The payload of a NAL unit refers to the NAL unit excluding the NAL unit header.



The Structure of the EVC NAL Unit Header

Figure 1

The semantics of the fields in the NAL unit header are as specified in [EVC] and described briefly below for convenience. In addition to the name and size of each field, the corresponding syntax element name in [EVC] is also provided.

F: 1 bit

forbidden_zero_bit. Required to be zero in [EVC]. Note that the inclusion of this bit in the NAL unit header was included to enable transport of EVC video over MPEG-2 transport systems (avoidance of start code emulations) [MPEG2S]. In the context of this memo, the value 1 may be used to indicate a syntax violation, e.g., for a NAL unit resulted from aggregating a number of fragmented units of a NAL unit but missing the last fragment, as described in Section xxx. (section # placeholder)

Type: 6 bits

nal_unit_type_plus1. This field specifies the NAL unit type as defined in Table 4 of [EVC]. If the value of this field is less than and equal to 23, the NAL unit is a VCL NAL unit. Otherwise, the NAL unit is a non-VCL NAL unit. For a reference of all currently defined NAL unit types and their semantics, please refer to Section 7.4.2.2 in [EVC].

TID: 3 bits

nuh_temporal_id. This field specifies the temporal identifier of the NAL unit. The value of TemporalId is equal to TID. TemporalId shall be equal to 0 if it is a IDR NAL unit type (NAL unit type 1).

Reserve: 5 bits

nuh_reserved_zero_5bits. This field shall be equal to the version of the [EVC] specification. Values of nuh_reserved_zero_5bits greater than 0 are reserved for future use by ISO/IEC. Decoders conforming to a profile specified in [EVC] Annex A shall ignore (i.e., remove from the bitstream and discard) all NAL units with values of nuh_reserved_zero_5bits greater than 0.

E: 1 bit

nuh_extension_flag. This field shall be equal the version of the [EVC] specification. Value of nuh_extesion_flag equal to 1 is reserved for future use by ISO/IEC. Decoders conforming to a profile specified in Annex A shall ignore (i.e., remove from the bitstream and discard) all NAL units with values of nuh_extension_flag equal to 1.

1.2. Overview of the Payload Format

This payload format defines the following processes required for transport of [EVC] coded data over RTP [RFC3550]:

- * Usage of RTP header with this payload format
- * Packetization of [EVC] coded NAL units into RTP packets using three types of payload structures: a single NAL unit, aggregation, and fragment unit packet
- * Transmission of [EVC] NAL units of the same bitstream within a single RTP stream.
- * Media type parameters to be used with the Session Description Protocol (SDP) [RFC4566]

2. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown above.

3. Definitions and Abbreviations

3.1. Definitions

This document uses the terms and definitions of EVC. Section 3.1.1 lists relevant definitions from [EVC] for convenience. Section 3.1.2 provides definitions specific to this memo.

3.1.1. Definitions from the EVC Specification

Access Unit: A set of NAL units that are associated with each other according to a specified classification rule, are consecutive in decoding order, and contain exactly one coded picture.

Bitstream: A sequence of bits, in the form of a NAL unit stream or a byte stream, that forms the representation of coded pictures and associated data forming one or more coded video sequences (CVSs).

Coded Picture: A coded representation of a picture containing all CTUs of the picture.

Coded Video Sequence (CVS): A sequence of access units that consists, in decoding order, of an IDR access unit, followed by zero or more access units that are not IDR access units, including all subsequent access units up to but not including any subsequent access unit that is an IDR access unit.

Coding Tree Block (CTB): An $N \times N$ block of samples for some value of N such that the division of a component into CTBs is a partitioning.

Coding Tree Unit (CTU): A CTB of luma samples, two corresponding CTBs of chroma samples of a picture that has three sample arrays, or a CTB of samples of a monochrome picture or a picture that is coded using three separate colour planes and syntax structures used to code the samples.

Decoded Picture: A decoded picture is derived by decoding a coded picture.

Decoded Picture Buffer (DPB): A buffer holding decoded pictures for reference, output reordering, or output delay specified for the hypothetical reference decoder in Annex C of [EVC] specification.

Dynamic Range Adjustment (DRA): A mapping process that is applied to decoded picture prior to cropping and output as part of the decoding process and is controlled by parameters conveyed in an Adaptation Parameter Set (APS).

Hypothetical Reference Decoder (HRD): A hypothetical decoder model that specifies constraints on the variability of conforming NAL unit streams or conforming byte streams that an encoding process may produce.

Instantaneous Decoding Refresh (IDR) access unit: An access unit in which the coded picture is an IDR picture.

Instantaneous Decoding Refresh (IDR) picture: A coded picture for which each VCL NAL unit has `NalUnitType` equal to `IDR_NUT`.

Level: A defined set of constraints on the values that may be taken by the syntax elements and variables of this document, or the value of a transform coefficient prior to scaling.

Network Abstraction Layer (NAL) unit: A syntax structure containing an indication of the type of data to follow and bytes containing that data in the form of an RBSP interspersed as necessary.

Network Abstraction Layer (NAL) Unit Stream: A sequence of NAL units.

Non-IDR Picture: A coded picture that is not an IDR picture.

Non-VCL NAL Unit: A NAL unit that is not a VCL NAL unit.

Picture Parameter Set (PPS): A syntax structure containing syntax elements that apply to zero or more entire coded pictures as determined by a syntax element found in each slice header.

Picture Order Count (POC): A variable that is associated with each picture, uniquely identifies the associated picture among all pictures in the CVS, and, when the associated picture is to be output from the decoded picture buffer, indicates the position of the associated picture in output order relative to the output order positions of the other pictures in the same CVS that are to be output from the decoded picture buffer.

Raw Byte Sequence Payload (RBSP): A syntax structure containing an integer number of bytes that is encapsulated in a NAL unit and that is either empty or has the form of a string of data bits containing syntax elements followed by an RBSP stop bit and zero or more subsequent bits equal to 0.

Sequence Parameter Set (SPS): A syntax structure containing syntax elements that apply to zero or more entire CVSs as determined by the content of a syntax element found in the PPS referred to by a syntax element found in each slice header.

Tile row: A rectangular region of CTUs having a height specified by syntax elements in the PPS and a width equal to the width of the picture.

Tile scan: A specific sequential ordering of CTUs partitioning a picture in which the CTUs are ordered consecutively in CTU raster scan in a tile whereas tiles in a picture are ordered consecutively in a raster scan of the tiles of the picture.

Video coding layer (VCL) NAL unit: A collective term for coded slice NAL units and the subset of NAL units that have reserved values of NalUnitType that are classified as VCL NAL units in this document.

3.1.2. Definitions Specific to This Memo

Media-Aware Network Element (MANE): A network element, such as a middlebox, selective forwarding unit, or application-layer gateway that is capable of parsing certain aspects of the RTP payload headers or the RTP payload and reacting to their contents.

Informative note: The concept of a MANE goes beyond normal routers or gateways in that a MANE has to be aware of the signaling (e.g., to learn about the payload type mappings of the media streams), and in that it has to be trusted when working with Secure RTP (SRTP). The advantage of using MANEs is that they allow packets

to be dropped according to the needs of the media coding. For example, if a MANE has to drop packets due to congestion on a certain link, it can identify and remove those packets whose elimination produces the least adverse effect on the user experience. After dropping packets, MANEs must rewrite RTCP packets to match the changes to the RTP stream, as specified in Section 7 of [RFC3550].

NAL unit decoding order: A NAL unit order that conforms to the constraints on NAL unit order given in Section 8.2 and 8.3 in [EVC], follow the Order of NAL units in the bitstream.

NAL unit output order: A NAL unit order in which NAL units of different access units are in the output order of the decoded pictures corresponding to the access units, as specified in [EVC], and in which NAL units within an access unit are in their decoding order.

RTP stream: See [RFC7656]. Within the scope of this memo, one RTP stream is utilized to transport one or more temporal sub-layers.

Transmission order: The order of packets in ascending RTP sequence number order (in modulo arithmetic). Within an aggregation packet, the NAL unit transmission order is the same as the order of appearance of NAL units in the packet.

3.2. Abbreviations

| | |
|-----|--------------------------------|
| APS | Adaptation Parameter Set |
| ATS | Adaptive Transform Selection |
| B | Bi-predictive |
| CBR | Constant Bit Rate |
| CPB | Coded Picture Buffer |
| CTB | Coding Tree Block |
| CTU | Coding Tree Unit |
| CVS | Coded Video Sequence |
| DPB | Decoded Picture Buffer |
| HRD | Hypothetical Reference Decoder |

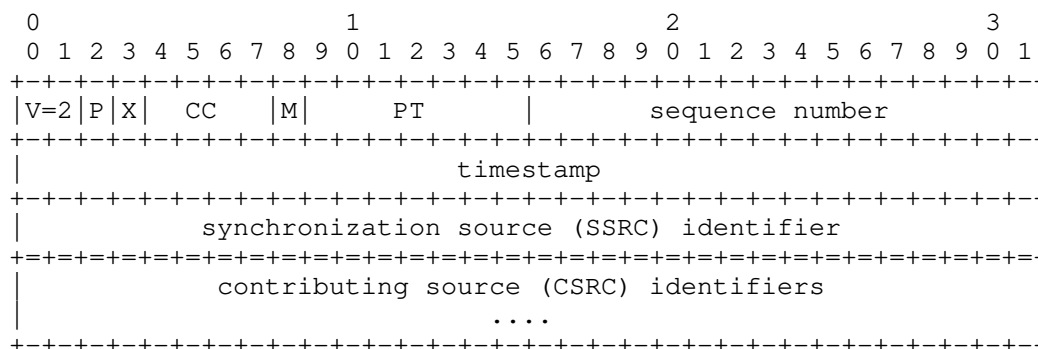
| | |
|------|--------------------------------------|
| HSS | Hypothetical Stream Scheduler |
| I | Intra |
| IDR | Instantaneous Decoding Refresh |
| LSB | Least Significant Bit |
| LTRP | Long-Term Reference Picture |
| MMVD | Merge with Motion Vector Difference |
| MSB | Most Significant Bit |
| NAL | Network Abstraction Layer |
| P | Predictive |
| POC | Picture Order Count |
| PPS | Picture Parameter Set |
| QP | Quantization Parameter |
| RBSP | Raw Byte Sequence Payload |
| RGB | Same as GBR |
| SAR | Sample Aspect Ratio |
| SEI | Supplemental Enhancement Information |
| SODB | String Of Data Bits |
| SPS | Sequence Parameter Set |
| STRP | Short-Term Reference Picture |
| VBR | Variable Bit Rate |
| VCL | Video Coding Layer |

4. RTP Payload Format

4.1. RTP Header Usage

The format of the RTP header is specified in [RFC3550] (reprinted as Figure 2 for convenience). This 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 aggregation packets and fragmentation units are specified in Section 4.3.2 and Section 4.3.3, respectively.



RTP Header According to {{RFC3550}}

Figure 2

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

Marker bit (M): 1 bit

Set for the last packet of the access unit, carried in the current RTP stream. This is in line with the normal use of the M bit in video formats to allow an efficient playout buffer handling.

editor-note 4: The informative note below needs updating once the NAL unit type table is stable in the [EVC] spec.

Informative note: The content of a NAL unit does not tell whether or not the NAL unit is the last NAL unit, in decoding order, of an access unit. An RTP sender implementation may obtain this information from the video encoder. If, however, the implementation cannot obtain this information directly from the encoder, e.g., when the bitstream was pre-encoded, and also there is no timestamp allocated for each NAL unit, then the sender implementation can inspect subsequent NAL units in decoding order to determine whether or not the NAL unit is the

last NAL unit of an access unit as follows. A NAL unit is determined to be the last NAL unit of an access unit if it is the last NAL unit of the bitstream. A NAL unit `nal_uX` is also determined to be the last NAL unit of an access unit if both the following conditions are true: 1) the next VCL NAL unit `nal_uY` in decoding order has the high-order bit of the first byte after its NAL unit header equal to 1 or `nal_unit_type` equal to 27, and 2) all NAL units between `nal_uX` and `nal_uY`, when present, have `nal_unit_type` in the range of 24 to 26, inclusive, equal to 28 or in the range of 29 to 55.

Payload Type (PT): 7 bits

The assignment of an RTP payload type for this new payload format is outside the scope of this document and will not be specified here. The assignment of a payload type has to be performed either through the profile used or in a dynamic way.

Sequence Number (SN): 16 bits

Set and used in accordance with [RFC3550].

Timestamp: 32 bits

The RTP timestamp is set to the sampling timestamp of the content. A 90 kHz clock rate MUST be used. If the NAL unit has no timing properties of its own (e.g., parameter sets or certain SEI NAL units), the RTP timestamp MUST be set to the RTP timestamp of the coded picture of the access unit in which the NAL unit (according to Annex D of [EVC]) is included. Receivers MUST use the RTP timestamp for the display process, even when the bitstream contains picture timing SEI messages or decoding unit information SEI messages as specified in [EVC].

Synchronization source (SSRC): 32 bits

Used to identify the source of the RTP packets. When using SRST, by definition a single SSRC is used for all parts of a single bitstream.

4.2. Payload Header Usage

The first two bytes of the payload of an RTP packet are referred to as the payload header. The payload header consists of the same fields (F, TID, Reserve and E) as the NAL unit header as shown in Section 1.1.4, irrespective of the type of the payload structure.

The TID value indicates (among other things) the relative importance of an RTP packet, for example, because NAL units belonging to higher temporal sub-layers are not used for the decoding of lower temporal sub-layers. A lower value of TID indicates a higher importance. More-important NAL units MAY be better protected against transmission losses than less-important NAL units.

4.3. Payload Structures

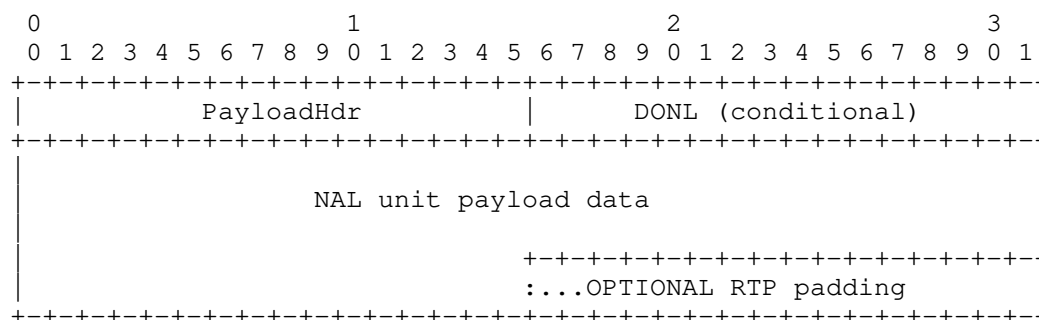
Three different types of RTP packet payload structures are specified. A receiver can identify the type of an RTP packet payload through the Type field in the payload header.

The Three different payload structures are as follows:

- * Single NAL unit packet: Contains a single NAL unit in the payload, and the NAL unit header of the NAL unit also serves as the payload header. This payload structure is specified in Section 4.3.1.
- * Aggregation Packet (AP): Contains more than one NAL unit within one access unit. This payload structure is specified in Section 4.3.2.
- * Fragmentation Unit (FU): Contains a subset of a single NAL unit. This payload structure is specified in Section 4.3.3.

4.3.1. Single NAL Unit Packets

A single NAL unit packet contains exactly one NAL unit, and consists of a payload header (denoted as PayloadHdr), a conditional 16-bit DONL field (in network byte order), and the NAL unit payload data (the NAL unit excluding its NAL unit header) of the contained NAL unit, as shown in Figure 3.



The Structure of a Single NAL Unit Packet

Figure 3

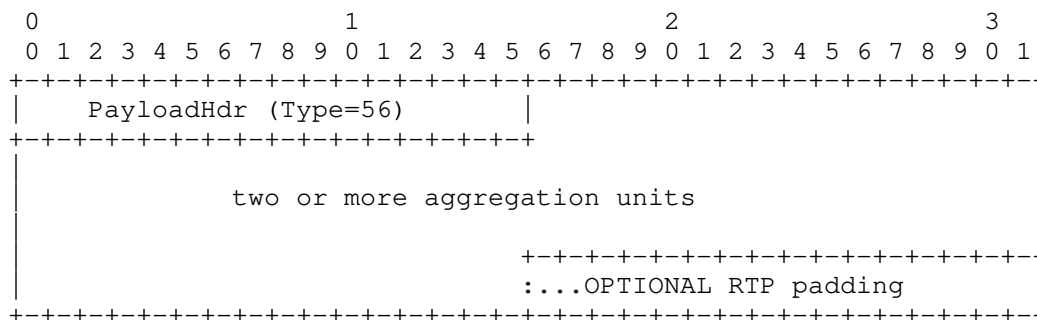
The DONL field, when present, specifies the value of the 16 least significant bits of the decoding order number of the contained NAL unit. If sprop-max-don-diff is greater than 0 for any of the RTP streams, the DONL field MUST be present, and the variable DON for the contained NAL unit is derived as equal to the value of the DONL field. Otherwise (sprop-max-don-diff is equal to 0 for all the RTP streams), the DONL field MUST NOT be present.

4.3.2. Aggregation Packets (APs)

Aggregation Packets (APs) enable the reduction of packetization overhead for small NAL units, such as most of the non-VCL NAL units, which are often only a few octets in size.

An AP aggregates NAL units within one access unit. Each NAL unit to be carried in an AP is encapsulated in an aggregation unit. NAL units aggregated in one AP are in NAL unit decoding order.

An AP consists of a payload header (denoted as PayloadHdr) followed by two or more aggregation units, as shown in Figure 4.



The Structure of an Aggregation Packet

Figure 4

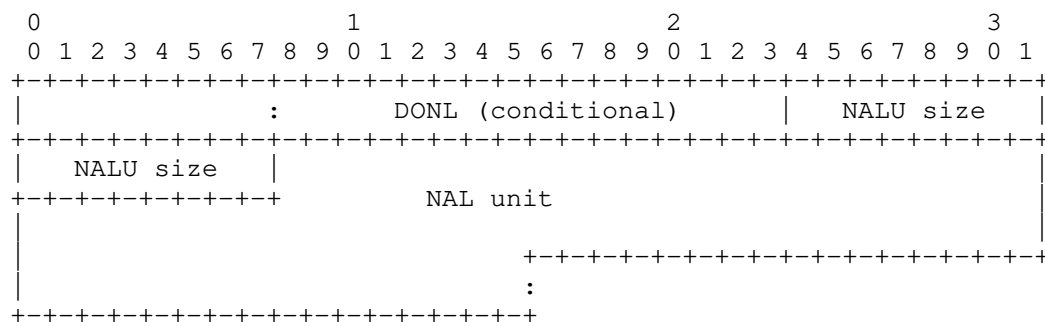
The fields in the payload header are set as follows. The F bit MUST be equal to 0 if the F bit of each aggregated NAL unit is equal to zero; otherwise, it MUST be equal to 1. The Type field MUST be equal to 56.

The value of TID MUST be the lowest value of TID of all the aggregated NAL units. The value of Reserve and E Must match the version of [EVC] specification.

Informative note: All VCL NAL units in an AP have the same TID value since they belong to the same access unit. However, an AP may contain non-VCL NAL units for which the TID value in the NAL unit header may be different than the TID value of the VCL NAL units in the same AP.

An AP MUST carry at least two aggregation units and can carry as many aggregation units as necessary; however, the total amount of data in an AP obviously MUST fit into an IP packet, and the size SHOULD be chosen so that the resulting IP packet is smaller than the path MTU size so to avoid IP layer fragmentation. An AP MUST NOT contain FUs specified in Section 4.3.3. APs MUST NOT be nested; i.e., an AP can not contain another AP.

The first aggregation unit in an AP consists of a conditional 16-bit DONL field (in network byte order) followed by a 16-bit unsigned size information (in network byte order) that indicates the size of the NAL unit in bytes (excluding these two octets, but including the NAL unit header), followed by the NAL unit itself, including its NAL unit header, as shown in Figure 5.



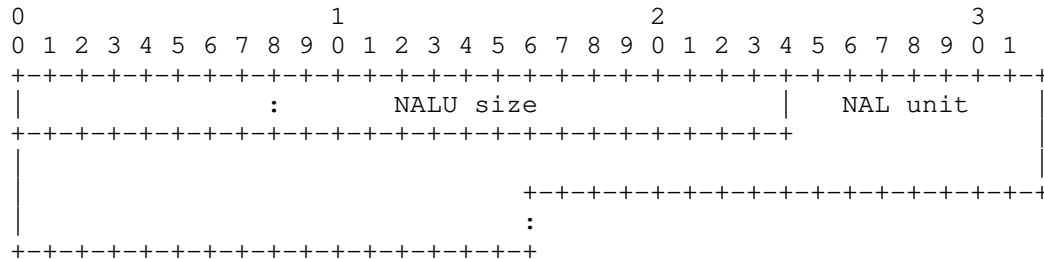
The Structure of the First Aggregation Unit in an AP

Figure 5

The DONL field, when present, specifies the value of the 16 least significant bits of the decoding order number of the aggregated NAL unit.

If `sprop-max-don-diff` is greater than 0 for any of the RTP streams, the DONL field MUST be present in an aggregation unit that is the first aggregation unit in an AP, and the variable DON for the aggregated NAL unit is derived as equal to the value of the DONL field. Otherwise (`sprop-max-don-diff` is equal to 0 for all the RTP streams), the DONL field MUST NOT be present in an aggregation unit that is the first aggregation unit in an AP.

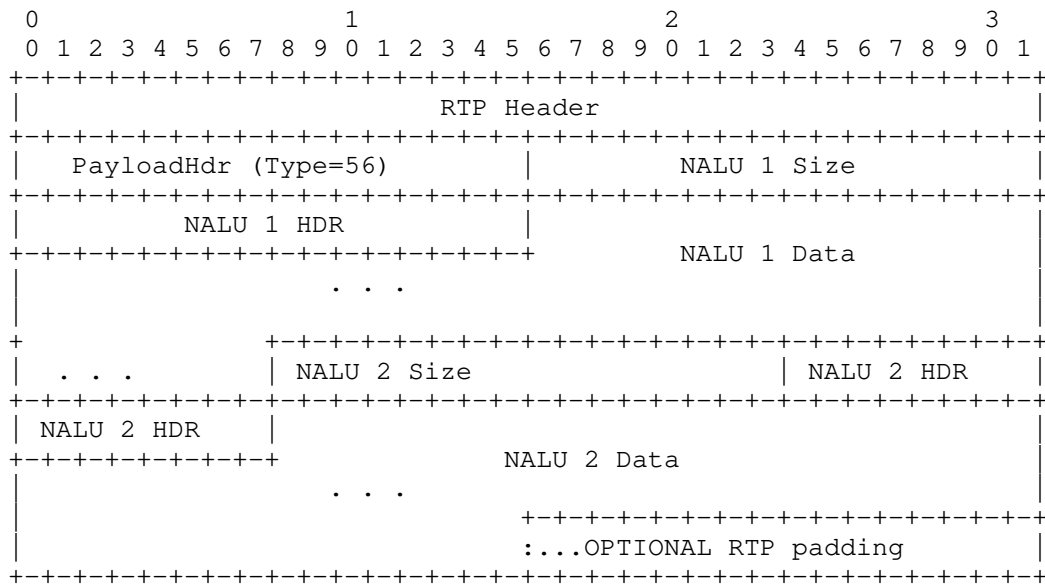
An aggregation unit that is not the first aggregation unit in an AP will be followed immediately by a 16-bit unsigned size information (in network byte order) that indicates the size of the NAL unit in bytes (excluding these two octets, but including the NAL unit header), followed by the NAL unit itself, including its NAL unit header, as shown in Figure 6.



The Structure of an Aggregation Unit That Is Not the First
Aggregation Unit in an AP

Figure 6

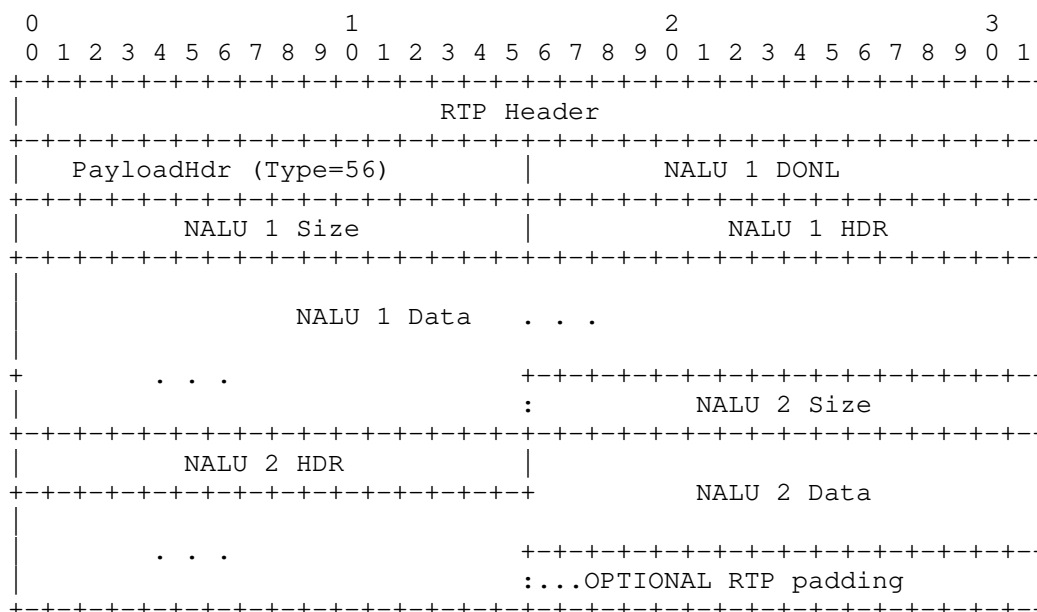
Figure 7 presents an example of an AP that contains two aggregation units, labeled as NALU 1 and NALU 2 in the figure, without the DONL field being present.



An Example of an AP Packet Containing
Two Aggregation Units without the DONL Field

Figure 7

Figure 8 presents an example of an AP that contains two aggregation units, labeled as NALU 1 and NALU 2 in the figure, with the DONL field being present.



An Example of an AP Containing Two Aggregation Units with the DONL Field

Figure 8

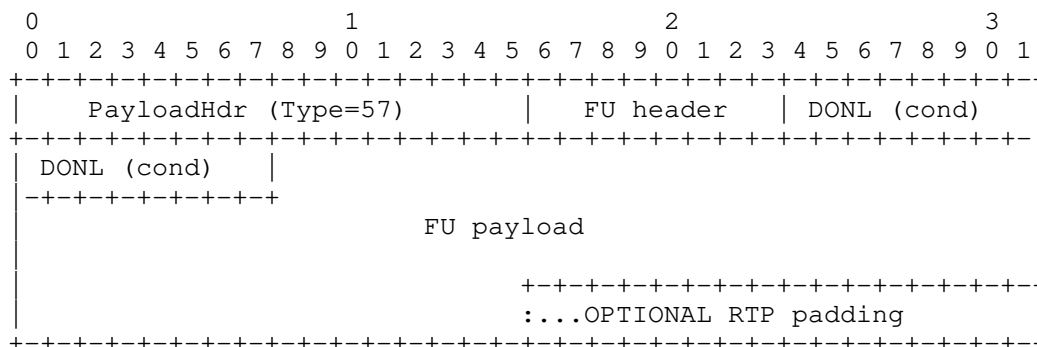
4.3.3. Fragmentation Units

Fragmentation Units (FUs) are introduced to enable fragmenting a single NAL unit into multiple RTP packets, possibly without cooperation or knowledge of the EVC [EVC] encoder. A fragment of a NAL unit consists of an integer number of consecutive octets of that NAL unit. Fragments of the same NAL unit **MUST** be sent in consecutive order with ascending RTP sequence numbers (with no other RTP packets within the same RTP stream being sent between the first and last fragment).

When a NAL unit is fragmented and conveyed within FUs, it is referred to as a fragmented NAL unit. APs MUST NOT be fragmented. FUs MUST NOT be nested; i.e., an FU must not contain a subset of another FU.

The RTP timestamp of an RTP packet carrying an FU is set to the NALU-time of the fragmented NAL unit.

An FU consists of a payload header (denoted as PayloadHdr), an FU header of one octet, a conditional 16-bit DONL field (in network byte order), and an FU payload, as shown in Figure 9.

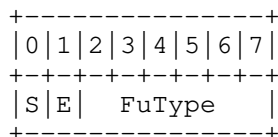


The Structure of an FU

Figure 9

The fields in the payload header are set as follows. The Type field MUST be equal to 57. The fields F, TID, Reserve and E MUST be equal to the fields F, TID, Reserve and E, respectively, of the fragmented NAL unit.

The FU header consists of an S bit, an E bit, and a 6-bit FuType field, as shown in Figure 10.



The Structure of FU Header

Figure 10

The semantics of the FU header fields are as follows:

S: 1 bit

When set to 1, the S bit indicates the start of a fragmented NAL unit, i.e., the first byte of the FU payload is also the first byte of the payload of the fragmented NAL unit. When the FU payload is not the start of the fragmented NAL unit payload, the S bit MUST be set to 0.

E: 1 bit

When set to 1, the E bit indicates the end of a fragmented NAL unit, i.e., the last byte of the payload is also the last byte of the fragmented NAL unit. When the FU payload is not the last fragment of a fragmented NAL unit, the E bit MUST be set to 0.

FuType: 6 bits

The field FuType MUST be equal to the field Type of the fragmented NAL unit.

The DONL field, when present, specifies the value of the 16 least significant bits of the decoding order number of the fragmented NAL unit.

If sprop-max-don-diff is greater than 0 for any of the RTP streams, and the S bit is equal to 1, the DONL field MUST be present in the FU, and the variable DON for the fragmented NAL unit is derived as equal to the value of the DONL field. Otherwise (sprop-max-don-diff is equal to 0 for all the RTP streams, or the S bit is equal to 0), the DONL field MUST NOT be present in the FU.

A non-fragmented NAL unit MUST NOT be transmitted in one FU; i.e., the Start bit and End bit must not both be set to 1 in the same FU header.

The FU payload consists of fragments of the payload of the fragmented NAL unit so that if the FU payloads of consecutive FUs, starting with an FU with the S bit equal to 1 and ending with an FU with the E bit equal to 1, are sequentially concatenated, the payload of the fragmented NAL unit can be reconstructed. The NAL unit header of the fragmented NAL unit is not included as such in the FU payload, but rather the information of the NAL unit header of the fragmented NAL unit is conveyed in F, TID, Reserve and E fields of the FU payload headers of the FUs and the FuType field of the FU header of the FUs. An FU payload MUST NOT be empty.

If an FU is lost, the receiver SHOULD discard all following fragmentation units in transmission order corresponding to the same fragmented NAL unit, unless the decoder in the receiver is known to gracefully handle incomplete NAL units.

A receiver in an endpoint or in a MANE MAY aggregate the first n-1 fragments of a NAL unit to an (incomplete) NAL unit, even if fragment n of that NAL unit is not received. In this case, the `forbidden_zero_bit` of the NAL unit MUST be set to 1 to indicate a syntax violation.

4.4. Decoding Order Number

For each NAL unit, the variable AbsDon is derived, representing the decoding order number that is indicative of the NAL unit decoding order.

Let NAL unit n be the n -th NAL unit in transmission order within an RTP stream.

If `sprop-max-don-diff` is equal to 0 for all the RTP streams carrying the HEVC bitstream, AbsDon[n], the value of AbsDon for NAL unit n , is derived as equal to n .

Otherwise (`sprop-max-don-diff` is greater than 0 for any of the RTP streams), AbsDon[n] is derived as follows, where DON[n] is the value of the variable DON for NAL unit n :

- * If n is equal to 0 (i.e., NAL unit n is the very first NAL unit in transmission order), AbsDon[0] is set equal to DON[0].
- * Otherwise (n is greater than 0), the following applies for derivation of AbsDon[n]:

If DON[n] == DON[$n-1$],
AbsDon[n] = AbsDon[$n-1$]

If (DON[n] > DON[$n-1$] and DON[n] - DON[$n-1$] < 32768),
AbsDon[n] = AbsDon[$n-1$] + DON[n] - DON[$n-1$]

If (DON[n] < DON[$n-1$] and DON[$n-1$] - DON[n] >= 32768),
AbsDon[n] = AbsDon[$n-1$] + 65536 - DON[$n-1$] + DON[n]

If (DON[n] > DON[$n-1$] and DON[n] - DON[$n-1$] >= 32768),
AbsDon[n] = AbsDon[$n-1$] - (DON[$n-1$] + 65536 - DON[n])

If (DON[n] < DON[$n-1$] and DON[$n-1$] - DON[n] < 32768),
AbsDon[n] = AbsDon[$n-1$] - (DON[$n-1$] - DON[n])

For any two NAL units m and n , the following applies:

- * AbsDon[n] greater than AbsDon[m] indicates that NAL unit n follows NAL unit m in NAL unit decoding order.
- * When AbsDon[n] is equal to AbsDon[m], the NAL unit decoding order of the two NAL units can be in either order.

- * AbsDon[n] less than AbsDon[m] indicates that NAL unit n precedes NAL unit m in decoding order.

Informative note: When two consecutive NAL units in the NAL unit decoding order have different values of AbsDon, the absolute difference between the two AbsDon values may be greater than or equal to 1.

Informative note: There are multiple reasons to allow for the absolute difference of the values of AbsDon for two consecutive NAL units in the NAL unit decoding order to be greater than one. An increment by one is not required, as at the time of associating values of AbsDon to NAL units, it may not be known whether all NAL units are to be delivered to the receiver. For example, a gateway might not forward VCL NAL units of higher sub-layers or some SEI NAL units when there is congestion in the network. In another example, the first intra-coded picture of a pre-encoded clip is transmitted in advance to ensure that it is readily available in the receiver, and when transmitting the first intra-coded picture, the originator does not exactly know how many NAL units will be encoded before the first intra-coded picture of the pre-encoded clip follows in decoding order. Thus, the values of AbsDon for the NAL units of the first intra-coded picture of the pre-encoded clip have to be estimated when they are transmitted, and gaps in values of AbsDon may occur.

5. Packetization Rules

The following packetization rules apply:

- * If sprop-max-don-diff is greater than 0 for any of the RTP streams, the transmission order of NAL units carried in the RTP stream MAY be different than the NAL unit decoding order and the NAL unit output order.
- * A NAL unit of a small size SHOULD be encapsulated in an aggregation packet together with one or more other NAL units in order to avoid unnecessary packetization overhead for small NAL units. For example, non-VCL NAL units such as access unit delimiters, parameter sets, or SEI NAL units are typically small and can often be aggregated with VCL NAL units without violating MTU size constraints.

- * Each non-VCL NAL unit SHOULD, when possible from an MTU size match viewpoint, be encapsulated in an aggregation packet together with its associated VCL NAL unit, as typically a non-VCL NAL unit would be meaningless without the associated VCL NAL unit being available.
- * For carrying exactly one NAL unit in an RTP packet, a single NAL unit packet MUST be used.

6. De-packetization Process

The general concept behind de-packetization is to get the NAL units out of the RTP packets in an RTP stream and pass them to the decoder in the NAL unit decoding order.

The de-packetization process is implementation dependent. Therefore, the following description should be seen as an example of a suitable implementation. Other schemes may be used as well, as long as the output for the same input is the same as the process described below. The output is the same when the set of output NAL units and their order are both identical. Optimizations relative to the described algorithms are possible.

All normal RTP mechanisms related to buffer management apply. In particular, duplicated or outdated RTP packets (as indicated by the RTP sequence number and the RTP timestamp) are removed. To determine the exact time for decoding, factors such as a possible intentional delay to allow for proper inter-stream synchronization must be factored in.

NAL units with NAL unit type values in the range of 0 to 55, inclusive, may be passed to the decoder. NAL-unit-like structures with NAL unit type values in the range of 56 to 63, inclusive, MUST NOT be passed to the decoder.

The receiver includes a receiver buffer, which is used to compensate for transmission delay jitter within individual RTP streams and across RTP streams, to reorder NAL units from transmission order to the NAL unit decoding order. In this section, the receiver operation is described under the assumption that there is no transmission delay jitter within an RTP stream. To make a difference from a practical receiver buffer that is also used for compensation of transmission delay jitter, the receiver buffer is hereafter called the de-packetization buffer in this section. Receivers should also prepare for transmission delay jitter; that is, either reserve separate buffers for transmission delay jitter buffering and de-packetization buffering or use a receiver buffer for both transmission delay jitter and de-packetization. Moreover, receivers should take transmission

delay jitter into account in the buffering operation, e.g., by additional initial buffering before starting of decoding and playback.

When `sprop-max-don-diff` is equal to 0 for the received RTP stream, the de-packetization buffer size is zero bytes, and the process described in the remainder of this paragraph applies. The NAL units carried in the RTP stream are directly passed to the decoder in their transmission order, which is identical to their decoding order. When there are several NAL units of the same RTP stream with the same NTP timestamp, the order to pass them to the decoder is their transmission order.

Informative note: The mapping between RTP and NTP timestamps is conveyed in RTCP SR packets. In addition, the mechanisms for faster media timestamp synchronization discussed in [RFC6051] may be used to speed up the acquisition of the RTP-to-wall-clock mapping.

When `sprop-max-don-diff` is greater than 0 for the received RTP stream the process described in the remainder of this section applies.

There are two buffering states in the receiver: initial buffering and buffering while playing. Initial buffering starts when the reception is initialized. After initial buffering, decoding and playback are started, and the buffering-while-playing mode is used.

Regardless of the buffering state, the receiver stores incoming NAL units, in reception order, into the de-packetization buffer. NAL units carried in RTP packets are stored in the de-packetization buffer individually, and the value of `AbsDon` is calculated and stored for each NAL unit.

Initial buffering lasts until condition A (the difference between the greatest and smallest `AbsDon` values of the NAL units in the de-packetization buffer is greater than or equal to the value of `sprop-max-don-diff`) or condition B (the number of NAL units in the de-packetization buffer is greater than the value of `sprop-depack-buf-nalus`) is true.

After initial buffering, whenever condition A or condition B is true, the following operation is repeatedly applied until both condition A and condition B become false:

- * The NAL unit in the de-packetization buffer with the smallest value of `AbsDon` is removed from the de-packetization buffer and passed to the decoder.

When no more NAL units are flowing into the de-packetization buffer, all NAL units remaining in the de-packetization buffer are removed from the buffer and passed to the decoder in the order of increasing AbsDon values.

7. Payload Format Parameters

This section specifies the optional parameters. A mapping of the parameters with Session Description Protocol (SDP) [RFC4556] is also provided for applications that use SDP.

7.1. Media Type Registration

The receiver MUST ignore any parameter unspecified in this memo.

Type name: video

Subtype name: evc

Required parameters: none

Optional parameters:

editor-note 5: To be updated

7.2. SDP Parameters

The receiver MUST ignore any parameter unspecified in this memo.

7.2.1. Mapping of Payload Type Parameters to SDP

The media type video/evc 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 evc (the media subtype).
- * The clock rate in the "a=rtpmap" line MUST be 90000.
- * OPTIONAL PARAMETERS:

editor-note 6: To be updated

7.2.2. Usage with SDP Offer/Answer Model

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

editor-note 7: to be updated

7.2.3. SDP Example

editor-note 8: to be updated

8. Use with Feedback Messages

Placeholder

8.1. Picture Loss Indication (PLI)

Placeholder

8.2. Full Intra Request (FIR)

Placeholder

9. Security Considerations

The scope of this Security Considerations section is limited to the payload format itself and to one feature of [EVC] that may pose a particularly serious security risk if implemented naively. The payload format, in isolation, does not form a complete system. Implementers are advised to read and understand relevant security-related documents, especially those pertaining to RTP (see the Security Considerations section in [RFC3550]), and the security of the call-control stack chosen (that may make use of the media type registration of this memo). Implementers should also consider known security vulnerabilities of video coding and decoding implementations in general and avoid those.

Within this RTP payload format, neither the various media-plane-based mechanisms, nor the signaling part of this memo, seems to pose a security risk beyond those common to all RTP-based systems.

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]. 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 lays 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. The rest of this section discusses the security impacting properties of the payload format itself.

Because the data compression used with this payload format is applied end-to-end, any encryption needs to be performed after compression. A potential denial-of-service threat exists for data encodings using compression techniques that have non-uniform receiver-end computational load. The attacker can inject pathological datagrams into the bitstream that are complex to decode and that cause the receiver to be overloaded. EVC is particularly vulnerable to such attacks, as it is extremely simple to generate datagrams containing NAL units that affect the decoding process of many future NAL units. Therefore, the usage of data origin authentication and data integrity protection of at least the RTP packet is RECOMMENDED, for example, with SRTP [RFC3711].

End-to-end security with authentication, integrity, or confidentiality protection will prevent a MANE from performing media-aware operations other than discarding complete packets. In the case of confidentiality protection, it will even be prevented from discarding packets in a media-aware way. To be allowed to perform such operations, a MANE is required to be a trusted entity that is included in the security context establishment.

10. Congestion Control

Congestion control for RTP SHALL be used in accordance with RTP [RFC3550] and with any applicable RTP profile, e.g., AVP [RFC3551]. If best-effort service is being used, an additional requirement is that users of this payload format MUST monitor packet loss to ensure that the packet loss rate is within an acceptable range. 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 all RTP streams combined is achieving. This condition can be satisfied by implementing congestion-control mechanisms to adapt the transmission rate, 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.

The bitrate adaptation necessary for obeying the congestion control principle is easily achievable when real-time encoding is used, for example, by adequately tuning the quantization parameter. However, when pre-encoded content is being transmitted, bandwidth adaptation requires the pre-coded bitstream to be tailored for such adaptivity. The key mechanism available in [EVC] is temporal scalability. A media sender can remove NAL units belonging to higher temporal sub-layers (i.e., those NAL units with a high value of TID) until the sending bitrate drops to an acceptable range.

The mechanisms mentioned above generally work within a defined profile and level and, therefore, no renegotiation of the channel is required. Only when non-downgradable parameters (such as profile) are required to be changed does it become necessary to terminate and restart the RTP stream(s). This may be accomplished by using different RTP payload types.

MANES MAY remove certain unusable packets from the RTP stream when that RTP stream was damaged due to previous packet losses. This can help reduce the network load in certain special cases. For example, MANES can remove those FUs where the leading FUs belonging to the same NAL unit have been lost or those dependent slice segments when the leading slice segments belonging to the same slice have been lost, because the trailing FUs or dependent slice segments are meaningless to most decoders. MANES can also remove higher temporal scalable layers if the outbound transmission (from the MANE's viewpoint) experiences congestion.

11. IANA Considerations

Placeholder

12. Acknowledgements

Large parts of this specification share text with the RTP payload format for HEVC [RFC7798]. We thank the authors of that specification for their excellent work.

13. References

13.1. Normative References

[EVC] "ISO/IEC FDIS 23094-1 Essential Video Coding", 2020, <<https://www.iso.org/standard/57797.html>>.

- [ISO23094-1] "ISO/IEC DIS Information technology --- General video coding --- Part 1 Essential video coding", n.d., <<https://www.iso.org/standard/57797.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>>.
- [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>>.
- [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>>.
- [RFC4556] Zhu, L. and B. Tung, "Public Key Cryptography for Initial Authentication in Kerberos (PKINIT)", RFC 4556, DOI 10.17487/RFC4556, June 2006, <<https://www.rfc-editor.org/info/rfc4556>>.
- [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>>.

- [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>>.
- [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>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

13.2. Informative References

- [AVC] "ITU-T Recommendation H.264 - Advanced video coding for generic audiovisual services", 2014, <<https://www.iso.org/standard/66069.html>>.
- [HEVC] "High efficiency video coding, ITU-T Recommendation H.265", 2017, <<https://www.iso.org/standard/69668.html>>.
- [I-D.ietf-avtcore-rtp-vvc] Zhao, S., Wenger, S., Sanchez, Y., and Y. Wang, "RTP Payload Format for Versatile Video Coding (VVC)", Work in Progress, Internet-Draft, draft-ietf-avtcore-rtp-vvc-07, 19 January 2021, <<http://www.ietf.org/internet-drafts/draft-ietf-avtcore-rtp-vvc-07.txt>>.
- [MPEG2S] ISO/IEC, ., "Information technology - Generic coding of moving pictures and associated audio information - Part 1: Systems, ISO International Standard 13818-1", 2013.
- [RFC6051] Perkins, C. and T. Schierl, "Rapid Synchronisation of RTP Flows", RFC 6051, DOI 10.17487/RFC6051, November 2010, <<https://www.rfc-editor.org/info/rfc6051>>.
- [RFC6184] Wang, Y.-K., 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.-K., 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>>.

- [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>>.
- [RFC7798] Wang, Y.-K., Sanchez, Y., Schierl, T., Wenger, S., and M. 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>>.
- [VVC] "ISO/IEC FDIS 23090-3 Information technology --- Coded representation of immersive media --- Part 3 - Versatile video coding", 2020, <<https://www.iso.org/standard/73022.html>>.

Authors' Addresses

Shuai Zhao
Tencent
2747 Park Blvd
Palo Alto, 94588
United States of America

Email: shuai.zhao@ieee.org

Stephan Wenger
Tencent
2747 Park Blvd
Palo Alto, 94588
United States of America

Email: stewe@stewe.org

Youngkwon Lim
Samsung Electronics
6625 Excellence Way
Plano, 75013
United States of America

Email: yklwhite@gmail.com

avtcore
Internet-Draft
Intended status: Standards Track
Expires: 6 November 2022

S. Zhao
S. Wenger
Tencent
Y. Sanchez
Fraunhofer HHI
Y. Wang
Bytedance Inc.
M. M Hannuksela
Nokia Technologies
5 May 2022

RTP Payload Format for Versatile Video Coding (VVC)
draft-ietf-avtcore-rtp-vvc-16

Abstract

This memo describes an RTP payload format for the video coding standard ITU-T Recommendation H.266 and ISO/IEC International Standard 23090-3, both also known as Versatile Video Coding (VVC) and developed by the Joint Video Experts Team (JVET). The RTP payload format allows for packetization of one or more Network Abstraction Layer (NAL) units in each RTP packet payload as well as fragmentation of a NAL unit into multiple RTP packets. The payload format has wide applicability in videoconferencing, Internet video streaming, and high-bitrate entertainment-quality video, among other applications.

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 November 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

| | |
|---|----|
| 1. Introduction | 3 |
| 1.1. Overview of the VVC Codec | 3 |
| 1.1.1. Coding-Tool Features (informative) | 3 |
| 1.1.2. Systems and Transport Interfaces (informative) | 6 |
| 1.1.3. High-Level Picture Partitioning (informative) | 11 |
| 1.1.4. NAL Unit Header | 13 |
| 1.2. Overview of the Payload Format | 14 |
| 2. Conventions | 15 |
| 3. Definitions and Abbreviations | 15 |
| 3.1. Definitions | 15 |
| 3.1.1. Definitions from the VVC Specification | 15 |
| 3.1.2. Definitions Specific to This Memo | 18 |
| 3.2. Abbreviations | 19 |
| 4. RTP Payload Format | 20 |
| 4.1. RTP Header Usage | 20 |
| 4.2. Payload Header Usage | 22 |
| 4.3. Payload Structures | 22 |
| 4.3.1. Single NAL Unit Packets | 23 |
| 4.3.2. Aggregation Packets (APs) | 23 |
| 4.3.3. Fragmentation Units | 27 |
| 4.4. Decoding Order Number | 30 |
| 5. Packetization Rules | 31 |
| 6. De-packetization Process | 32 |
| 7. Payload Format Parameters | 34 |
| 7.1. Media Type Registration | 34 |
| 7.2. Optional Parameters Definition | 35 |
| 7.3. SDP Parameters | 45 |
| 7.3.1. Mapping of Payload Type Parameters to SDP | 46 |
| 7.3.2. Usage with SDP Offer/Answer Model | 48 |
| 7.3.3. Usage in Declarative Session Descriptions | 57 |
| 7.3.4. Considerations for Parameter Sets | 59 |
| 8. Use with Feedback Messages | 59 |
| 8.1. Picture Loss Indication (PLI) | 59 |
| 8.2. Full Intra Request (FIR) | 59 |
| 9. Security Considerations | 60 |
| 10. Congestion Control | 61 |
| 11. IANA Considerations | 62 |

| | |
|--|----|
| 12. Acknowledgements | 62 |
| 13. References | 62 |
| 13.1. Normative References | 62 |
| 13.2. Informative References | 64 |
| Appendix A. Change History | 66 |
| Authors' Addresses | 66 |

1. Introduction

The Versatile Video Coding specification was formally published as both ITU-T Recommendation H.266 [VVC] and ISO/IEC International Standard 23090-3 [ISO23090-3]. VVC is reported to provide significant coding efficiency gains over High Efficiency Video Coding [HEVC], also known as H.265, and other earlier video codecs.

This memo specifies an RTP payload format for VVC. It shares its basic design with the NAL (Network Abstraction Layer) unit based RTP payload formats of AVC Video Coding [RFC6184], Scalable Video Coding (SVC) [RFC6190], High Efficiency Video Coding (HEVC) [RFC7798] and their respective predecessors. With respect to design philosophy, security, congestion control, and overall implementation complexity, it has similar properties to those earlier payload format specifications. This is a conscious choice, as at least RFC 6184 is widely deployed and generally known in the relevant implementer communities. Certain scalability-related mechanisms known from [RFC6190] were incorporated into this document, as VVC version 1 supports temporal, spatial, and signal-to-noise ratio (SNR) scalability.

1.1. Overview of the VVC Codec

VVC and HEVC share a similar hybrid video codec design. In this memo, we provide a very brief overview of those features of VVC that are, in some form, addressed by the payload format specified herein. Implementers have to read, understand, and apply the ITU-T/ISO/IEC specifications pertaining to VVC to arrive at interoperable, well-performing implementations.

Conceptually, both VVC and HEVC include a Video Coding Layer (VCL), which is often used to refer to the coding-tool features, and a NAL, which is often used to refer to the systems and transport interface aspects of the codecs.

1.1.1. Coding-Tool Features (informative)

Coding tool features are described below with occasional reference to the coding tool set of HEVC, which is well known in the community.

Similar to earlier hybrid-video-coding-based standards, including HEVC, the following basic video coding design is employed by VVC. A prediction signal is first formed by either intra- or motion-compensated prediction, and the residual (the difference between the original and the prediction) is then coded. The gains in coding efficiency are achieved by redesigning and improving almost all parts of the codec over earlier designs. In addition, VVC includes several tools to make the implementation on parallel architectures easier.

Finally, VVC includes temporal, spatial, and SNR scalability as well as multiview coding support.

Coding blocks and transform structure

Among major coding-tool differences between HEVC and VVC, one of the important improvements is the more flexible coding tree structure in VVC, i.e., multi-type tree. In addition to quadtree, binary and ternary trees are also supported, which contributes significant improvement in coding efficiency. Moreover, the maximum size of a coding tree unit (CTU) is increased from 64x64 to 128x128. To improve the coding efficiency of chroma signal, luma chroma separated trees at CTU level may be employed for intra-slices. The square transforms in HEVC are extended to non-square transforms for rectangular blocks resulting from binary and ternary tree splits. Besides, VVC supports multiple transform sets (MTS), including DCT-2, DST-7, and DCT-8 as well as the non-separable secondary transform. The transforms used in VVC can have different sizes with support for larger transform sizes. For DCT-2, the transform sizes range from 2x2 to 64x64, and for DST-7 and DCT-8, the transform sizes range from 4x4 to 32x32. In addition, VVC also support sub-block transform for both intra and inter coded blocks. For intra coded blocks, intra sub-partitioning (ISP) may be used to allow sub-block based intra prediction and transform. For inter blocks, sub-block transform may be used assuming that only a part of an inter-block has non-zero transform coefficients.

Entropy coding

Similar to HEVC, VVC uses a single entropy-coding engine, which is based on context adaptive binary arithmetic coding [CABAC], but with the support of multi-window sizes. The window sizes can be initialized differently for different context models. Due to such a design, it has more efficient adaptation speed and better coding efficiency. A joint chroma residual coding scheme is applied to further exploit the correlation between the residuals of two color components. In VVC, different residual coding schemes are applied for regular transform coefficients and residual samples generated using transform-skip mode.

In-loop filtering

VVC has more feature support in loop filters than HEVC. The deblocking filter in VVC is similar to HEVC but operates at a smaller grid. After deblocking and sample adaptive offset (SAO), an adaptive loop filter (ALF) may be used. As a Wiener filter, ALF reduces distortion of decoded pictures. Besides, VVC introduces a new module called luma mapping with chroma scaling to fully utilize the dynamic range of signal so that rate-distortion performance of both Standard Dynamic Range (SDR) and High Dynamic Range (HDR) content is improved.

Motion prediction and coding

Compared to HEVC, VVC introduces several improvements in this area. First, there is the adaptive motion vector resolution (AMVR), which can save bit cost for motion vectors by adaptively signaling motion vector resolution. Then the affine motion compensation is included to capture complicated motion like zooming and rotation. Meanwhile, prediction refinement with the optical flow with affine mode (PROF) is further deployed to mimic affine motion at the pixel level. Thirdly the decoder side motion vector refinement (DMVR) is a method to derive MV vector at decoder side based on block matching so that fewer bits may be spent on motion vectors. Bi-directional optical flow (BDOF) is a similar method to PROF. BDOF adds a sample wise offset at 4x4 sub-block level that is derived with equations based on gradients of the prediction samples and a motion difference relative to CU motion vectors. Furthermore, merge with motion vector difference (MMVD) is a special mode, which further signals a limited set of motion vector differences on top of merge mode. In addition to MMVD, there are another three types of special merge modes, i.e., sub-block merge, triangle, and combined intra-/inter-prediction (CIIP). Sub-block merge list includes one candidate of sub-block temporal motion vector prediction (SbTMVP) and up to four candidates of affine motion vectors. Triangle is based on triangular block motion compensation. CIIP combines intra- and inter- predictions with weighting. Adaptive weighting may be employed with a block-level tool called bi-prediction with CU based weighting (BCW) which provides more flexibility than in HEVC.

Intra prediction and intra-coding

To capture the diversified local image texture directions with finer granularity, VVC supports 65 angular directions instead of 33 directions in HEVC. The intra mode coding is based on a 6-most-probable-mode scheme, and the 6 most probable modes are derived using the neighboring intra prediction directions. In addition, to deal with the different distributions of intra prediction angles for different block aspect ratios, a wide-angle intra prediction (WAIP)

scheme is applied in VVC by including intra prediction angles beyond those present in HEVC. Unlike HEVC which only allows using the most adjacent line of reference samples for intra prediction, VVC also allows using two further reference lines, as known as multi-reference-line (MRL) intra prediction. The additional reference lines can be only used for the 6 most probable intra prediction modes. To capture the strong correlation between different colour components, in VVC, a cross-component linear mode (CCLM) is utilized which assumes a linear relationship between the luma sample values and their associated chroma samples. For intra prediction, VVC also applies a position-dependent prediction combination (PDPC) for refining the prediction samples closer to the intra prediction block boundary. Matrix-based intra prediction (MIP) modes are also used in VVC which generates an up to 8x8 intra prediction block using a weighted sum of downsampled neighboring reference samples, and the weights are hardcoded constants.

Other coding-tool features

VVC introduces dependent quantization (DQ) to reduce quantization error by state-based switching between two quantizers.

1.1.2. Systems and Transport Interfaces (informative)

VVC inherits the basic systems and transport interfaces designs from HEVC and AVC. These include the NAL-unit-based syntax structure, the hierarchical syntax and data unit structure, the supplemental enhancement information (SEI) message mechanism, and the video buffering model based on the hypothetical reference decoder (HRD). The scalability features of VVC are conceptually similar to the scalable variant of HEVC known as SHVC. The hierarchical syntax and data unit structure consists of parameter sets at various levels (decoder, sequence (pertaining to all), sequence (pertaining to a single), picture), picture-level header parameters, slice-level header parameters, and lower-level parameters.

A number of key components that influenced the network abstraction layer design of VVC as well as this memo are described below

Decoding capability information

The decoding capability information includes parameters that stay constant for the lifetime of a VVC bitstream, which in IETF terms can translate to a session. Such information includes profile, level, and sub-profile information to determine a maximum capability interop point that is guaranteed to be never exceeded, even if splicing of video sequences occurs within a session. It further includes constraint fields (most of which are flags), which can optionally be

set to indicate that the video bitstream will be constrained in the use of certain features as indicated by the values of those fields. With this, a bitstream can be labeled as not using certain tools, which allows among other things for resource allocation in a decoder implementation.

Video parameter set

The video parameter set (VPS) pertains to one or more coded video sequences (CVSs) of multiple layers covering the same range of access units, and includes, among other information, decoding dependency expressed as information for reference picture list construction of enhancement layers. The VPS provides a "big picture" of a scalable sequence, including what types of operation points are provided, the profile, tier, and level of the operation points, and some other high-level properties of the bitstream that can be used as the basis for session negotiation and content selection, etc. One VPS may be referenced by one or more sequence parameter sets.

Sequence parameter set

The sequence parameter set (SPS) contains syntax elements pertaining to a coded layer video sequence (CLVS), which is a group of pictures belonging to the same layer, starting with a random access point, and followed by pictures that may depend on each other, until the next random access point picture. In MPEG-2, the equivalent of a CVS was a group of pictures (GOP), which normally started with an I frame and was followed by P and B frames. While more complex in its options of random access points, VVC retains this basic concept. One remarkable difference of VVC is that a CLVS may start with a Gradual Decoding Refresh (GDR) picture, without requiring presence of traditional random access points in the bitstream, such as instantaneous decoding refresh (IDR) or clean random access (CRA) pictures. In many TV-like applications, a CVS contains a few hundred milliseconds to a few seconds of video. In video conferencing (without switching MCUs involved), a CVS can be as long in duration as the whole session.

Picture and adaptation parameter set

The picture parameter set and the adaptation parameter set (PPS and APS, respectively) carry information pertaining to zero or more pictures and zero or more slices, respectively. The PPS contains information that is likely to stay constant from picture to picture, at least for pictures for a certain type-whereas the APS contains information, such as adaptive loop filter coefficients, that are likely to change from picture to picture or even within a picture. A single APS is referenced by all slices of the same picture if that

APS contains information about luma mapping with chroma scaling (LMCS) or scaling list. Different APSs containing ALF parameters can be referenced by slices of the same picture.

Picture header

A Picture Header contains information that is common to all slices that belong to the same picture. Being able to send that information as a separate NAL unit when pictures are split into several slices allows for saving bitrate, compared to repeating the same information in all slices. However, there might be scenarios where low-bitrate video is transmitted using a single slice per picture. Having a separate NAL unit to convey that information incurs in an overhead for such scenarios. For such scenarios, the picture header syntax structure is directly included in the slice header, instead of its own NAL unit. The mode of the picture header syntax structure being included in its own NAL unit or not can only be switched on/off for an entire CLVS, and can only be switched off when in the entire CLVS each picture contains only one slice.

Profile, tier, and level

The profile, tier and level syntax structures in DCI, VPS and SPS contain profile, tier, level information for all layers that refer to the DCI, for layers associated with one or more output layer sets specified by the VPS, and for any layer that refers to the SPS, respectively.

Sub-profiles

Within the VVC specification, a sub-profile is a 32-bit number, coded according to ITU-T Rec. T.35, that does not carry a semantics. It is carried in the `profile_tier_level` structure and hence (potentially) present in the DCI, VPS, and SPS. External registration bodies can register a T.35 codepoint with ITU-T registration authorities and associate with their registration a description of bitstream restrictions beyond the profiles defined by ITU-T and ISO/IEC. This would allow encoder manufacturers to label the bitstreams generated by their encoder as complying with such sub-profile. It is expected that upstream standardization organizations (such as: DVB and ATSC), as well as walled-garden video services will take advantage of this labeled system. In contrast to "normal" profiles, it is expected that sub-profiles may indicate encoder choices traditionally left open in the (decoder-centric) video coding specs, such as GOP structures, minimum/maximum QP values, and the mandatory use of certain tools or SEI messages.

General constraint fields

The `profile_tier_level` structure carries a considerable number of constraint fields (most of which are flags), which an encoder can use to indicate to a decoder that it will not use a certain tool or technology. They were included in reaction to a perceived market need for labeled a bitstream as not exercising a certain tool that has become commercially unviable.

Temporal scalability support

VVC includes support of temporal scalability, by inclusion of the signaling of `TemporalId` in the NAL unit header, the restriction that pictures of a particular temporal sublayer cannot be used for inter prediction reference by pictures of a lower temporal sublayer, the sub-bitstream extraction process, and the requirement that each sub-bitstream extraction output be a conforming bitstream. Media-Aware Network Elements (MANEs) can utilize the `TemporalId` in the NAL unit header for stream adaptation purposes based on temporal scalability.

Reference picture resampling (RPR)

In AVC and HEVC, the spatial resolution of pictures cannot change unless a new sequence using a new SPS starts, with an Intra random access point (IRAP) picture. VVC enables picture resolution change within a sequence at a position without encoding an IRAP picture, which is always intra-coded. This feature is sometimes referred to as reference picture resampling (RPR), as the feature needs resampling of a reference picture used for inter prediction when that reference picture has a different resolution than the current picture being decoded. RPR allows resolution change without the need of coding an IRAP picture and hence avoids a momentary bit rate spike caused by an IRAP picture in streaming or video conferencing scenarios, e.g., to cope with network condition changes. RPR can also be used in application scenarios wherein zooming of the entire video region or some region of interest is needed.

Spatial, SNR, and multiview scalability

VVC includes support for spatial, SNR, and multiview scalability. Scalable video coding is widely considered to have technical benefits and enrich services for various video applications. Until recently, however, the functionality has not been included in the first version of specifications of the video codecs. In VVC, however, all those forms of scalability are supported in the first version of VVC natively through the signaling of the `nuh_layer_id` in the NAL unit header, the VPS which associates layers with given `nuh_layer_id` to

each other, reference picture selection, reference picture resampling for spatial scalability, and a number of other mechanisms not relevant for this memo.

Spatial scalability

With the existence of Reference Picture Resampling (RPR), the additional burden for scalability support is just a modification of the high-level syntax (HLS). The inter-layer prediction is employed in a scalable system to improve the coding efficiency of the enhancement layers. In addition to the spatial and temporal motion-compensated predictions that are available in a single-layer codec, the inter-layer prediction in VVC uses the possibly resampled video data of the reconstructed reference picture from a reference layer to predict the current enhancement layer. The resampling process for inter-layer prediction, when used, is performed at the block-level, reusing the existing interpolation process for motion compensation in single-layer coding. It means that no additional resampling process is needed to support spatial scalability.

SNR scalability

SNR scalability is similar to spatial scalability except that the resampling factors are 1:1. In other words, there is no change in resolution, but there is inter-layer prediction.

Multiview scalability

The first version of VVC also supports multiview scalability, wherein a multi-layer bitstream carries layers representing multiple views, and one or more of the represented views can be output at the same time.

SEI messages

Supplemental enhancement information (SEI) messages are information in the bitstream that do not influence the decoding process as specified in the VVC spec, but address issues of representation/rendering of the decoded bitstream, label the bitstream for certain applications, among other, similar tasks. The overall concept of SEI messages and many of the messages themselves has been inherited from the AVC and HEVC specs. Except for the SEI messages that affect the specification of the hypothetical reference decoder (HRD), other SEI messages for use in the VVC environment, which are generally useful also in other video coding technologies, are not included in the main VVC specification but in a companion specification [VSEI].

1.1.1.3. High-Level Picture Partitioning (informative)

VVC inherited the concept of tiles and wavefront parallel processing (WPP) from HEVC, with some minor to moderate differences. The basic concept of slices was kept in VVC but designed in an essentially different form. VVC is the first video coding standard that includes subpictures as a feature, which provides the same functionality as HEVC motion-constrained tile sets (MCTSs) but designed differently to have better coding efficiency and to be friendlier for usage in application systems. More details of these differences are described below.

Tiles and WPP

Same as in HEVC, a picture can be split into tile rows and tile columns in VVC, in-picture prediction across tile boundaries is disallowed, etc. However, the syntax for signaling of tile partitioning has been simplified, by using a unified syntax design for both the uniform and the non-uniform mode. In addition, signaling of entry point offsets for tiles in the slice header is optional in VVC while it is mandatory in HEVC. The WPP design in VVC has two differences compared to HEVC: i) The CTU row delay is reduced from two CTUs to one CTU; ii) signaling of entry point offsets for WPP in the slice header is optional in VVC while it is mandatory in HEVC.

Slices

In VVC, the conventional slices based on CTUs (as in HEVC) or macroblocks (as in AVC) have been removed. The main reasoning behind this architectural change is as follows. The advances in video coding since 2003 (the publication year of AVC v1) have been such that slice-based error concealment has become practically impossible, due to the ever-increasing number and efficiency of in-picture and inter-picture prediction mechanisms. An error-concealed picture is the decoding result of a transmitted coded picture for which there is some data loss (e.g., loss of some slices) of the coded picture or a reference picture for at least some part of the coded picture is not error-free (e.g., that reference picture was an error-concealed picture). For example, when one of the multiple slices of a picture is lost, it may be error-concealed using an interpolation of the neighboring slices. While advanced video coding prediction mechanisms provide significantly higher coding efficiency, they also make it harder for machines to estimate the quality of an error-concealed picture, which was already a hard problem with the use of simpler prediction mechanisms. Advanced in-picture prediction mechanisms also cause the coding efficiency loss due to splitting a picture into multiple slices to be more significant. Furthermore,

network conditions become significantly better while at the same time techniques for dealing with packet losses have become significantly improved. As a result, very few implementations have recently used slices for maximum transmission unit size matching. Instead, substantially all applications where low-delay error resilience is required (e.g., video telephony and video conferencing) rely on system/transport-level error resilience (e.g., retransmission, forward error correction) and/or picture-based error resilience tools (feedback-based error resilience, insertion of IRAPs, scalability with higher protection level of the base layer, and so on). Considering all the above, nowadays it is very rare that a picture that cannot be correctly decoded is passed to the decoder, and when such a rare case occurs, the system can afford to wait for an error-free picture to be decoded and available for display without resulting in frequent and long periods of picture freezing seen by end users.

Slices in VVC have two modes: rectangular slices and raster-scan slices. The rectangular slice, as indicated by its name, covers a rectangular region of the picture. Typically, a rectangular slice consists of several complete tiles. However, it is also possible that a rectangular slice is a subset of a tile and consists of one or more consecutive, complete CTU rows within a tile. A raster-scan slice consists of one or more complete tiles in a tile raster scan order, hence the region covered by a raster-scan slices need not but could have a non-rectangular shape, but it may also happen to have the shape of a rectangle. The concept of slices in VVC is therefore strongly linked to or based on tiles instead of CTUs (as in HEVC) or macroblocks (as in AVC).

Subpictures

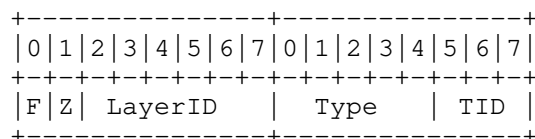
VVC is the first video coding standard that includes the support of subpictures as a feature. Each subpicture consists of one or more complete rectangular slices that collectively cover a rectangular region of the picture. A subpicture may be either specified to be extractable (i.e., coded independently of other subpictures of the same picture and of earlier pictures in decoding order) or not extractable. Regardless of whether a subpicture is extractable or not, the encoder can control whether in-loop filtering (including deblocking, SAO, and ALF) is applied across the subpicture boundaries individually for each subpicture.

Functionally, subpictures are similar to the motion-constrained tile sets (MCTSs) in HEVC. They both allow independent coding and extraction of a rectangular subset of a sequence of coded pictures, for use cases like viewport-dependent 360o video streaming optimization and region of interest (ROI) applications.

There are several important design differences between subpictures and MCTSs. First, the subpictures feature in VVC allows motion vectors of a coding block pointing outside of the subpicture even when the subpicture is extractable by applying sample padding at subpicture boundaries in this case, similarly as at picture boundaries. Second, additional changes were introduced for the selection and derivation of motion vectors in the merge mode and in the decoder side motion vector refinement process of VVC. This allows higher coding efficiency compared to the non-normative motion constraints applied at the encoder-side for MCTSs. Third, rewriting of SHs (and PH NAL units, when present) is not needed when extracting one or more extractable subpictures from a sequence of pictures to create a sub-bitstream that is a conforming bitstream. In sub-bitstream extractions based on HEVC MCTSs, rewriting of SHs is needed. Note that in both HEVC MCTSs extraction and VVC subpictures extraction, rewriting of SPSSs and PPSSs is needed. However, typically there are only a few parameter sets in a bitstream, while each picture has at least one slice, therefore rewriting of SHs can be a significant burden for application systems. Fourth, slices of different subpictures within a picture are allowed to have different NAL unit types. Fifth, VVC specifies HRD and level definitions for subpicture sequences, thus the conformance of the sub-bitstream of each extractable subpicture sequence can be ensured by encoders.

1.1.4. NAL Unit Header

VVC maintains the NAL unit concept of HEVC with modifications. VVC uses a two-byte NAL unit header, as shown in Figure 1. The payload of a NAL unit refers to the NAL unit excluding the NAL unit header.



The Structure of the VVC NAL Unit Header.

Figure 1

The semantics of the fields in the NAL unit header are as specified in VVC and described briefly below for convenience. In addition to the name and size of each field, the corresponding syntax element name in VVC is also provided.

F: 1 bit

`forbidden_zero_bit`. Required to be zero in VVC. Note that the inclusion of this bit in the NAL unit header was to enable transport of VVC video over MPEG-2 transport systems (avoidance of start code emulations) [MPEG2S]. In the context of this memo the value 1 may be used to indicate a syntax violation, e.g., for a NAL unit resulted from aggregating a number of fragmented units of a NAL unit but missing the last fragment, as described in the last sentence of section 4.3.3.

Z: 1 bit

`nuh_reserved_zero_bit`. Required to be zero in VVC, and reserved for future extensions by ITU-T and ISO/IEC. This memo does not overload the "Z" bit for local extensions, as a) overloading the "F" bit is sufficient and b) to preserve the usefulness of this memo to possible future versions of [VVC].

LayerId: 6 bits

`nuh_layer_id`. Identifies the layer a NAL unit belongs to, wherein a layer may be, e.g., a spatial scalable layer, a quality scalable layer, a layer containing a different view, etc.

Type: 5 bits

`nal_unit_type`. This field specifies the NAL unit type as defined in Table 5 of [VVC]. For a reference of all currently defined NAL unit types and their semantics, please refer to Section 7.4.2.2 in [VVC].

TID: 3 bits

`nuh_temporal_id_plus1`. This field specifies the temporal identifier of the NAL unit plus 1. The value of TemporalId is equal to TID minus 1. A TID value of 0 is illegal to ensure that there is at least one bit in the NAL unit header equal to 1, so to enable the consideration of start code emulations in the NAL unit payload data independent of the NAL unit header.

1.2. Overview of the Payload Format

This payload format defines the following processes required for transport of VVC coded data over RTP [RFC3550]:

- * Usage of RTP header with this payload format

- * Packetization of VVC coded NAL units into RTP packets using three types of payload structures: a single NAL unit packet, aggregation packet, and fragment unit
- * Transmission of VVC NAL units of the same bitstream within a single RTP stream
- * Media type parameters to be used with the Session Description Protocol (SDP) [RFC8866]
- * Usage of RTCP feedback messages

2. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Definitions and Abbreviations

3.1. Definitions

This document uses the terms and definitions of VVC. Section 3.1.1 lists relevant definitions from [VVC] for convenience. Section 3.1.2 provides definitions specific to this memo. All the used terms and definitions in this memo are verbatim copies of [VVC] specification.

3.1.1. Definitions from the VVC Specification

Access unit (AU): A set of PUs that belong to different layers and contain coded pictures associated with the same time for output from the DPB.

Adaptation parameter set (APS): A syntax structure containing syntax elements that apply to zero or more slices as determined by zero or more syntax elements found in slice headers.

Bitstream: A sequence of bits, in the form of a NAL unit stream or a byte stream, that forms the representation of a sequence of AUs forming one or more coded video sequences (CVSs).

Coded picture: A coded representation of a picture comprising VCL NAL units with a particular value of `nuh_layer_id` within an AU and containing all CTUs of the picture.

Clean random access (CRA) PU: A PU in which the coded picture is a CRA picture.

Clean random access (CRA) picture: An IRAP picture for which each VCL NAL unit has `nal_unit_type` equal to `CRA_NUT`.

Coded video sequence (CVS): A sequence of AUs that consists, in decoding order, of a CVSS AU, followed by zero or more AUs that are not CVSS AUs, including all subsequent AUs up to but not including any subsequent AU that is a CVSS AU.

Coded video sequence start (CVSS) AU: An AU in which there is a PU for each layer in the CVS and the coded picture in each PU is a CLVSS picture.

Coded layer video sequence (CLVS): A sequence of PUs with the same value of `nuh_layer_id` that consists, in decoding order, of a CLVSS PU, followed by zero or more PUs that are not CLVSS PUs, including all subsequent PUs up to but not including any subsequent PU that is a CLVSS PU.

Coded layer video sequence start (CLVSS) PU: A PU in which the coded picture is a CLVSS picture.

Coded layer video sequence start (CLVSS) picture: A coded picture that is an IRAP picture with `NoOutputBeforeRecoveryFlag` equal to 1 or a GDR picture with `NoOutputBeforeRecoveryFlag` equal to 1.

Coding tree unit (CTU): A CTB of luma samples, two corresponding CTBs of chroma samples of a picture that has three sample arrays, or a CTB of samples of a monochrome picture or a picture that is coded using three separate colour planes and syntax structures used to code the samples.

Decoding Capability Information (DCI): A syntax structure containing syntax elements that apply to the entire bitstream.

Decoded picture buffer (DPB): A buffer holding decoded pictures for reference, output reordering, or output delay specified for the hypothetical reference decoder.

Gradual decoding refresh (GDR) picture: A picture for which each VCL NAL unit has `nal_unit_type` equal to `GDR_NUT`.

Instantaneous decoding refresh (IDR) PU: A PU in which the coded picture is an IDR picture.

Instantaneous decoding refresh (IDR) picture: An IRAP picture for which each VCL NAL unit has `nal_unit_type` equal to `IDR_W_RADL` or `IDR_N_LP`.

Intra random access point (IRAP) AU: An AU in which there is a PU for each layer in the CVS and the coded picture in each PU is an IRAP picture.

Intra random access point (IRAP) PU: A PU in which the coded picture is an IRAP picture.

Intra random access point (IRAP) picture: A coded picture for which all VCL NAL units have the same value of `nal_unit_type` in the range of `IDR_W_RADL` to `CRA_NUT`, inclusive.

Layer: A set of VCL NAL units that all have a particular value of `nuh_layer_id` and the associated non-VCL NAL units.

Network abstraction layer (NAL) unit: A syntax structure containing an indication of the type of data to follow and bytes containing that data in the form of an RBSP interspersed as necessary with emulation prevention bytes.

Network abstraction layer (NAL) unit stream: A sequence of NAL units.

Output Layer Set (OLS): A set of layers for which one or more layers are specified as the output layers.

Operation point (OP): A temporal subset of an OLS, identified by an OLS index and a highest value of `TemporalId`.

Picture parameter set (PPS): A syntax structure containing syntax elements that apply to zero or more entire coded pictures as determined by a syntax element found in each slice header.

Picture unit (PU): A set of NAL units that are associated with each other according to a specified classification rule, are consecutive in decoding order, and contain exactly one coded picture.

Random access: The act of starting the decoding process for a bitstream at a point other than the beginning of the stream.

Sequence parameter set (SPS): A syntax structure containing syntax elements that apply to zero or more entire CLVSs as determined by the content of a syntax element found in the PPS referred to by a syntax element found in each picture header.

Slice: An integer number of complete tiles or an integer number of consecutive complete CTU rows within a tile of a picture that are exclusively contained in a single NAL unit.

Slice header (SH): A part of a coded slice containing the data elements pertaining to all tiles or CTU rows within a tile represented in the slice.

Sublayer: A temporal scalable layer of a temporal scalable bitstream consisting of VCL NAL units with a particular value of the TemporalId variable, and the associated non-VCL NAL units.

Subpicture: A rectangular region of one or more slices within a picture.

Sublayer representation: A subset of the bitstream consisting of NAL units of a particular sublayer and the lower sublayers.

Tile: A rectangular region of CTUs within a particular tile column and a particular tile row in a picture.

Tile column: A rectangular region of CTUs having a height equal to the height of the picture and a width specified by syntax elements in the picture parameter set.

Tile row: A rectangular region of CTUs having a height specified by syntax elements in the picture parameter set and a width equal to the width of the picture.

Video coding layer (VCL) NAL unit: A collective term for coded slice NAL units and the subset of NAL units that have reserved values of nal_unit_type that are classified as VCL NAL units in this Specification.

3.1.2. Definitions Specific to This Memo

Media-Aware Network Element (MANE): A network element, such as a middlebox, selective forwarding unit, or application-layer gateway that is capable of parsing certain aspects of the RTP payload headers or the RTP payload and reacting to their contents.

Informative note: The concept of a MANE goes beyond normal routers or gateways in that a MANE has to be aware of the signaling (e.g., to learn about the payload type mappings of the media streams), and in that it has to be trusted when working with Secure RTP (SRTP). The advantage of using MANEs is that they allow packets to be dropped according to the needs of the media coding. For example, if a MANE has to drop packets due to congestion on a

certain link, it can identify and remove those packets whose elimination produces the least adverse effect on the user experience. After dropping packets, MANEs must rewrite RTCP packets to match the changes to the RTP stream, as specified in Section 7 of [RFC3550].

NAL unit decoding order: A NAL unit order that conforms to the constraints on NAL unit order given in Section 7.4.2.4 in [VVC], follow the Order of NAL units in the bitstream.

RTP stream (See [RFC7656]): Within the scope of this memo, one RTP stream is utilized to transport a VVC bitstream, which may contain one or more layers, and each layer may contain one or more temporal sublayers.

Transmission order: The order of packets in ascending RTP sequence number order (in modulo arithmetic). Within an aggregation packet, the NAL unit transmission order is the same as the order of appearance of NAL units in the packet.

3.2. Abbreviations

| | |
|-----|---------------------------------|
| AU | Access Unit |
| AP | Aggregation Packet |
| APS | Adaptation Parameter Set |
| CTU | Coding Tree Unit |
| CVS | Coded Video Sequence |
| DPB | Decoded Picture Buffer |
| DCI | Decoding Capability Information |
| DON | Decoding Order Number |
| FIR | Full Intra Request |
| FU | Fragmentation Unit |
| GDR | Gradual Decoding Refresh |
| HRD | Hypothetical Reference Decoder |
| IDR | Instantaneous Decoding Refresh |

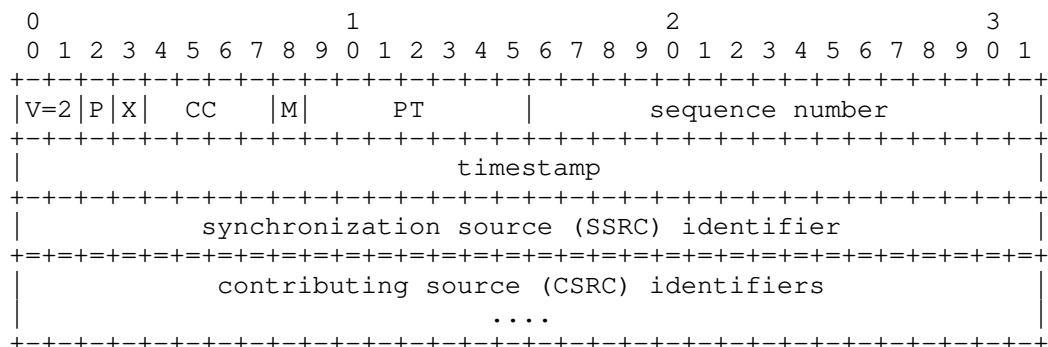
| | |
|------|--|
| IRAP | Intra Random Access Point |
| MANE | Media-Aware Network Element |
| MTU | Maximum Transfer Unit |
| NAL | Network Abstraction Layer |
| NALU | Network Abstraction Layer Unit |
| OLS | Output Layer Set |
| PLI | Picture Loss Indication |
| PPS | Picture Parameter Set |
| RPSI | Reference Picture Selection Indication |
| SEI | Supplemental Enhancement Information |
| SLI | Slice Loss Indication |
| SPS | Sequence Parameter Set |
| VCL | Video Coding Layer |
| VPS | Video Parameter Set |

4. RTP Payload Format

4.1. RTP Header Usage

The format of the RTP header is specified in [RFC3550] (reprinted as Figure 2 for convenience). This 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 aggregation packets and fragmentation units are specified in Section 4.3.2 and Section 4.3.3, respectively.



RTP Header According to [RFC3550]

Figure 2

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

Marker bit (M): 1 bit

Set for the last packet, in transmission order, among each set of packets that contain NAL units of one access unit. This is in line with the normal use of the M bit in video formats to allow an efficient playout buffer handling.

Payload Type (PT): 7 bits

The assignment of an RTP payload type for this new packet format is outside the scope of this document and will not be specified here. The assignment of a payload type has to be performed either through the profile used or in a dynamic way.

Sequence Number (SN): 16 bits

Set and used in accordance with [RFC3550].

Timestamp: 32 bits

The RTP timestamp is set to the sampling timestamp of the content. A 90 kHz clock rate MUST be used. If the NAL unit has no timing properties of its own (e.g., parameter set and SEI NAL units), the RTP timestamp MUST be set to the RTP timestamp of the coded pictures of the access unit in which the NAL unit (according to Section 7.4.2.4 of [VVC]) is included. Receivers MUST use the RTP timestamp for the display process, even when the bitstream contains picture timing SEI messages or decoding unit information SEI messages as specified in [VVC].

Informative note: When picture timing SEI messages are present, the RTP sender is responsible to ensure that the RTP timestamps are consistent with the timing information carried in the picture timing SEI messages.

Synchronization source (SSRC): 32 bits

Used to identify the source of the RTP packets. A single SSRC is used for all parts of a single bitstream.

4.2. Payload Header Usage

The first two bytes of the payload of an RTP packet are referred to as the payload header. The payload header consists of the same fields (F, Z, LayerId, Type, and TID) as the NAL unit header as shown in Section 1.1.4, irrespective of the type of the payload structure.

The TID value indicates (among other things) the relative importance of an RTP packet, for example, because NAL units belonging to higher temporal sublayers are not used for the decoding of lower temporal sublayers. A lower value of TID indicates a higher importance. More-important NAL units MAY be better protected against transmission losses than less-important NAL units.

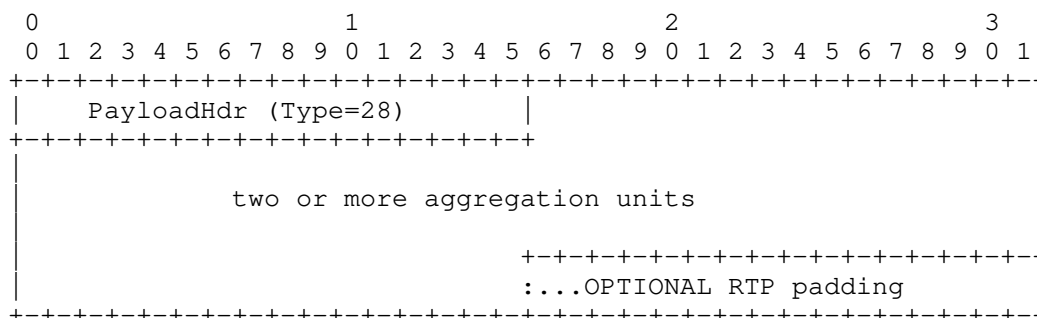
4.3. Payload Structures

Three different types of RTP packet payload structures are specified. A receiver can identify the type of an RTP packet payload through the Type field in the payload header.

The three different payload structures are as follows:

- * Single NAL unit packet: Contains a single NAL unit in the payload, and the NAL unit header of the NAL unit also serves as the payload header. This payload structure is specified in Section 4.4.1.

An AP consists of a payload header (denoted as PayloadHdr) followed by two or more aggregation units, as shown in Figure 4.



The Structure of an Aggregation Packet

Figure 4

The fields in the payload header of an AP are set as follows. The F bit MUST be equal to 0 if the F bit of each aggregated NAL unit is equal to zero; otherwise, it MUST be equal to 1. The Type field MUST be equal to 28.

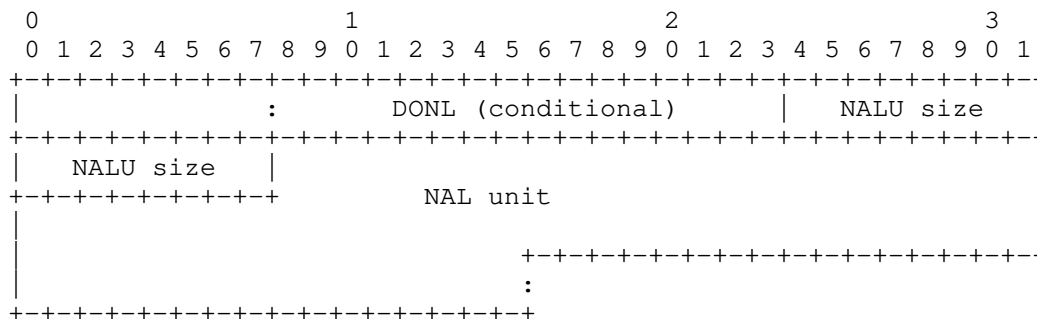
The value of LayerId MUST be equal to the lowest value of LayerId of all the aggregated NAL units. The value of TID MUST be the lowest value of TID of all the aggregated NAL units.

Informative note: All VCL NAL units in an AP have the same TID value since they belong to the same access unit. However, an AP may contain non-VCL NAL units for which the TID value in the NAL unit header may be different than the TID value of the VCL NAL units in the same AP.

Informative Note: If a system envisions sub-picture level or picture level modifications, for example by removing sub-pictures or pictures of a particular layer, a good design choice on the sender's side would be to aggregate NAL units belonging to only the same sub-picture or picture of a particular layer.

An AP MUST carry at least two aggregation units and can carry as many aggregation units as necessary; however, the total amount of data in an AP obviously MUST fit into an IP packet, and the size SHOULD be chosen so that the resulting IP packet is smaller than the MTU size so to avoid IP layer fragmentation. An AP MUST NOT contain FUs specified in Section 4.3.3. APs MUST NOT be nested; i.e., an AP can not contain another AP.

The first aggregation unit in an AP consists of a conditional 16-bit DONL field (in network byte order) followed by a 16-bit unsigned size information (in network byte order) that indicates the size of the NAL unit in bytes (excluding these two octets, but including the NAL unit header), followed by the NAL unit itself, including its NAL unit header, as shown in Figure 5.



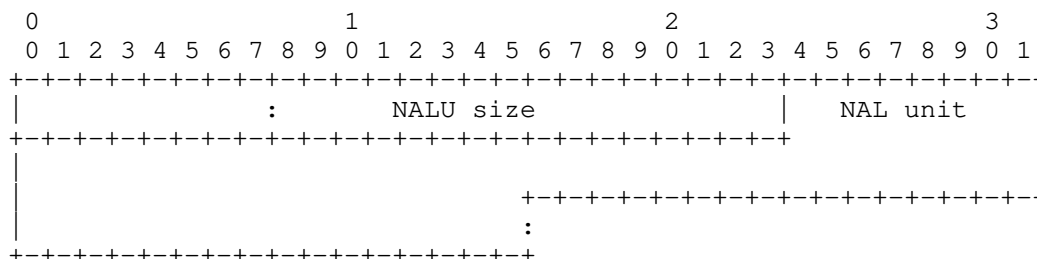
The Structure of the First Aggregation Unit in an AP

Figure 5

The DONL field, when present, specifies the value of the 16 least significant bits of the decoding order number of the aggregated NAL unit.

If `sprop-max-don-diff` is greater than 0, the DONL field MUST be present in an aggregation unit that is the first aggregation unit in an AP, and the variable `DON` for the aggregated NAL unit is derived as equal to the value of the DONL field, and the variable `DON` for an aggregation unit that is not the first aggregation unit in an AP aggregated NAL unit is derived as equal to the `DON` of the preceding aggregated NAL unit in the same AP plus 1 modulo 65536. Otherwise (`sprop-max-don-diff` is equal to 0), the DONL field MUST NOT be present in an aggregation unit that is the first aggregation unit in an AP.

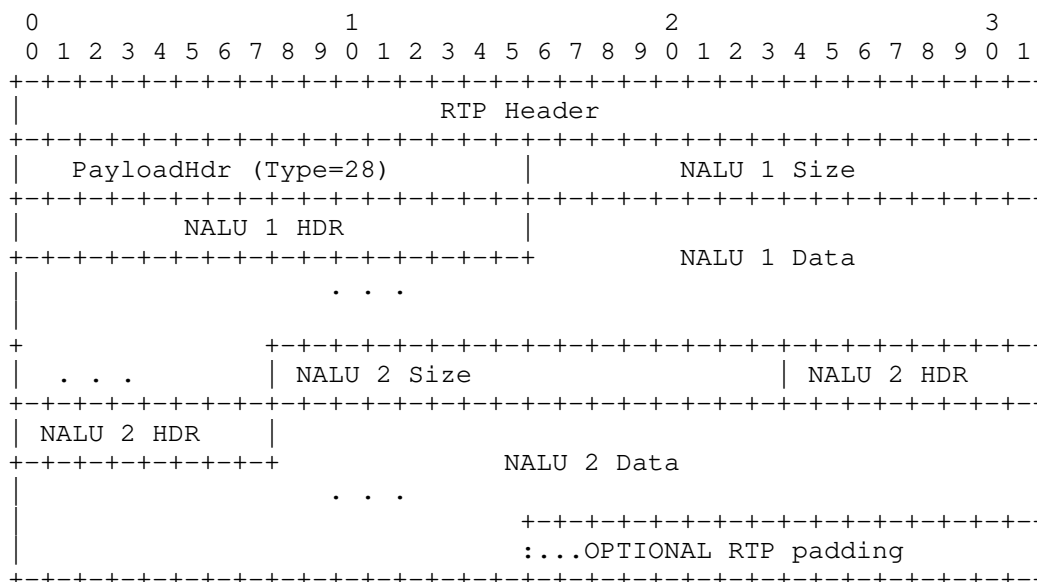
An aggregation unit that is not the first aggregation unit in an AP will be followed immediately by a 16-bit unsigned size information (in network byte order) that indicates the size of the NAL unit in bytes (excluding these two octets, but including the NAL unit header), followed by the NAL unit itself, including its NAL unit header, as shown in Figure 6.



The Structure of an Aggregation Unit That Is Not the First Aggregation Unit in an AP

Figure 6

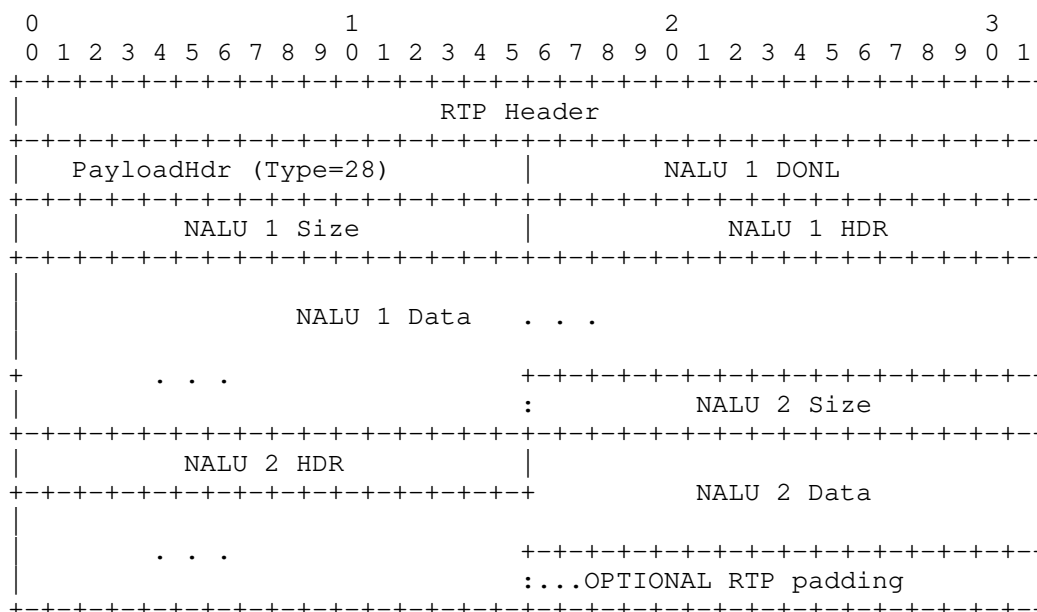
Figure 7 presents an example of an AP that contains two aggregation units, labeled as 1 and 2 in the figure, without the DONL field being present.



An Example of an AP Packet Containing Two Aggregation Units without the DONL Field

Figure 7

Figure 8 presents an example of an AP that contains two aggregation units, labeled as 1 and 2 in the figure, with the DONL field being present.



An Example of an AP Containing Two Aggregation Units with the DONL Field

Figure 8

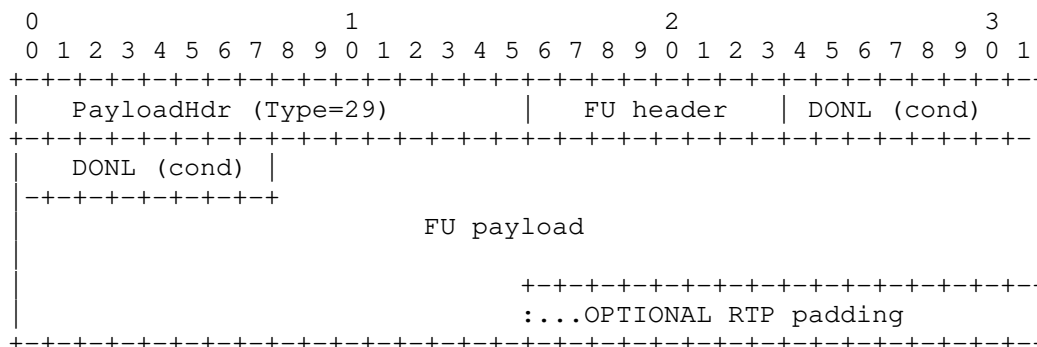
4.3.3. Fragmentation Units

Fragmentation Units (FUs) are introduced to enable fragmenting a single NAL unit into multiple RTP packets, possibly without cooperation or knowledge of the [VVC] encoder. A fragment of a NAL unit consists of an integer number of consecutive octets of that NAL unit. Fragments of the same NAL unit **MUST** be sent in consecutive order with ascending RTP sequence numbers (with no other RTP packets within the same RTP stream being sent between the first and last fragment).

When a NAL unit is fragmented and conveyed within FUs, it is referred to as a fragmented NAL unit. APs MUST NOT be fragmented. FUs MUST NOT be nested; i.e., an FU can not contain a subset of another FU.

The RTP timestamp of an RTP packet carrying an FU is set to the NALU-time of the fragmented NAL unit.

An FU consists of a payload header (denoted as PayloadHdr), an FU header of one octet, a conditional 16-bit DONL field (in network byte order), and an FU payload, as shown in Figure 9.

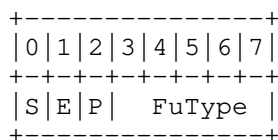


The Structure of an FU

Figure 9

The fields in the payload header are set as follows. The Type field MUST be equal to 29. The fields F, LayerId, and TID MUST be equal to the fields F, LayerId, and TID, respectively, of the fragmented NAL unit.

The FU header consists of an S bit, an E bit, an R bit and a 5-bit FuType field, as shown in Figure 10.



The Structure of FU Header

Figure 10

The semantics of the FU header fields are as follows:

S: 1 bit

When set to 1, the S bit indicates the start of a fragmented NAL unit, i.e., the first byte of the FU payload is also the first byte of the payload of the fragmented NAL unit. When the FU payload is not the start of the fragmented NAL unit payload, the S bit MUST be set to 0.

E: 1 bit

When set to 1, the E bit indicates the end of a fragmented NAL unit, i.e., the last byte of the payload is also the last byte of the fragmented NAL unit. When the FU payload is not the last fragment of a fragmented NAL unit, the E bit MUST be set to 0.

P: 1 bit

When set to 1, the P bit indicates the last FU of the last VCL NAL unit of a coded picture, i.e., the last byte of the FU payload is also the last byte of the last VCL NAL unit of the coded picture. When the FU payload is not the last fragment of the last VCL NAL unit of a coded picture, the P bit MUST be set to 0.

FuType: 5 bits

The field FuType MUST be equal to the field Type of the fragmented NAL unit.

The DONL field, when present, specifies the value of the 16 least significant bits of the decoding order number of the fragmented NAL unit.

If sprop-max-don-diff is greater than 0, and the S bit is equal to 1, the DONL field MUST be present in the FU, and the variable DON for the fragmented NAL unit is derived as equal to the value of the DONL field. Otherwise (sprop-max-don-diff is equal to 0, or the S bit is equal to 0), the DONL field MUST NOT be present in the FU.

A non-fragmented NAL unit MUST NOT be transmitted in one FU; i.e., the Start bit and End bit must not both be set to 1 in the same FU header.

The FU payload consists of fragments of the payload of the fragmented NAL unit so that if the FU payloads of consecutive FUs, starting with an FU with the S bit equal to 1 and ending with an FU with the E bit equal to 1, are sequentially concatenated, the payload of the fragmented NAL unit can be reconstructed. The NAL unit header of the fragmented NAL unit is not included as such in the FU payload, but rather the information of the NAL unit header of the fragmented NAL unit is conveyed in F, LayerId, and TID fields of the FU payload headers of the FUs and the FuType field of the FU header of the FUs. An FU payload MUST NOT be empty.

If an FU is lost, the receiver SHOULD discard all following fragmentation units in transmission order corresponding to the same fragmented NAL unit, unless the decoder in the receiver is known to be prepared to gracefully handle incomplete NAL units.

A receiver in an endpoint or in a MANE MAY aggregate the first $n-1$ fragments of a NAL unit to an (incomplete) NAL unit, even if fragment n of that NAL unit is not received. In this case, the `forbidden_zero_bit` of the NAL unit MUST be set to 1 to indicate a syntax violation.

4.4. Decoding Order Number

For each NAL unit, the variable `AbsDon` is derived, representing the decoding order number that is indicative of the NAL unit decoding order.

Let NAL unit n be the n -th NAL unit in transmission order within an RTP stream.

If `sprop-max-don-diff` is equal to 0, `AbsDon[n]`, the value of `AbsDon` for NAL unit n , is derived as equal to n .

Otherwise (`sprop-max-don-diff` is greater than 0), `AbsDon[n]` is derived as follows, where `DON[n]` is the value of the variable `DON` for NAL unit n :

- * If n is equal to 0 (i.e., NAL unit n is the very first NAL unit in transmission order), `AbsDon[0]` is set equal to `DON[0]`.

- * Otherwise (n is greater than 0), the following applies for derivation of `AbsDon[n]`:

- If `DON[n] == DON[n-1]`,
`AbsDon[n] = AbsDon[n-1]`

- If (`DON[n] > DON[n-1]` and `DON[n] - DON[n-1] < 32768`),
`AbsDon[n] = AbsDon[n-1] + DON[n] - DON[n-1]`

- If (`DON[n] < DON[n-1]` and `DON[n-1] - DON[n] >= 32768`),
`AbsDon[n] = AbsDon[n-1] + 65536 - DON[n-1] + DON[n]`

- If (`DON[n] > DON[n-1]` and `DON[n] - DON[n-1] >= 32768`),
`AbsDon[n] = AbsDon[n-1] - (DON[n-1] + 65536 - DON[n])`

- If (`DON[n] < DON[n-1]` and `DON[n-1] - DON[n] < 32768`),
`AbsDon[n] = AbsDon[n-1] - (DON[n-1] - DON[n])`

For any two NAL units m and n , the following applies:

- * $\text{AbsDon}[n]$ greater than $\text{AbsDon}[m]$ indicates that NAL unit n follows NAL unit m in NAL unit decoding order.
- * When $\text{AbsDon}[n]$ is equal to $\text{AbsDon}[m]$, the NAL unit decoding order of the two NAL units can be in either order.
- * $\text{AbsDon}[n]$ less than $\text{AbsDon}[m]$ indicates that NAL unit n precedes NAL unit m in decoding order.

Informative note: When two consecutive NAL units in the NAL unit decoding order have different values of AbsDon , the absolute difference between the two AbsDon values may be greater than or equal to 1.

Informative note: There are multiple reasons to allow for the absolute difference of the values of AbsDon for two consecutive NAL units in the NAL unit decoding order to be greater than one. An increment by one is not required, as at the time of associating values of AbsDon to NAL units, it may not be known whether all NAL units are to be delivered to the receiver. For example, a gateway might not forward VCL NAL units of higher sublayers or some SEI NAL units when there is congestion in the network. In another example, the first intra-coded picture of a pre-encoded clip is transmitted in advance to ensure that it is readily available in the receiver, and when transmitting the first intra-coded picture, the originator does not exactly know how many NAL units will be encoded before the first intra-coded picture of the pre-encoded clip follows in decoding order. Thus, the values of AbsDon for the NAL units of the first intra-coded picture of the pre-encoded clip have to be estimated when they are transmitted, and gaps in values of AbsDon may occur.

5. Packetization Rules

The following packetization rules apply:

- * If $\text{sprop-max-don-diff}$ is greater than 0, the transmission order of NAL units carried in the RTP stream MAY be different than the NAL unit decoding order. Otherwise ($\text{sprop-max-don-diff}$ is equal to 0), the transmission order of NAL units carried in the RTP stream MUST be the same as the NAL unit decoding order.
- * A NAL unit of a small size SHOULD be encapsulated in an aggregation packet together with one or more other NAL units in order to avoid the unnecessary packetization overhead for small

NAL units. For example, non-VCL NAL units such as access unit delimiters, parameter sets, or SEI NAL units are typically small and can often be aggregated with VCL NAL units without violating MTU size constraints.

- * Each non-VCL NAL unit SHOULD, when possible from an MTU size match viewpoint, be encapsulated in an aggregation packet together with its associated VCL NAL unit, as typically a non-VCL NAL unit would be meaningless without the associated VCL NAL unit being available.
- * For carrying exactly one NAL unit in an RTP packet, a single NAL unit packet MUST be used.

6. De-packetization Process

The general concept behind de-packetization is to get the NAL units out of the RTP packets in an RTP stream and pass them to the decoder in the NAL unit decoding order.

The de-packetization process is implementation dependent. Therefore, the following description should be seen as an example of a suitable implementation. Other schemes may be used as well, as long as the output for the same input is the same as the process described below. The output is the same when the set of output NAL units and their order are both identical. Optimizations relative to the described algorithms are possible.

All normal RTP mechanisms related to buffer management apply. In particular, duplicated or outdated RTP packets (as indicated by the RTP sequence number and the RTP timestamp) are removed. To determine the exact time for decoding, factors such as a possible intentional delay to allow for proper inter-stream synchronization MUST be factored in.

NAL units with NAL unit type values in the range of 0 to 27, inclusive, may be passed to the decoder. NAL-unit-like structures with NAL unit type values in the range of 28 to 31, inclusive, MUST NOT be passed to the decoder.

The receiver includes a receiver buffer, which is used to compensate for transmission delay jitter within individual RTP stream, and to reorder NAL units from transmission order to the NAL unit decoding order. In this section, the receiver operation is described under the assumption that there is no transmission delay jitter within an RTP stream. To make a difference from a practical receiver buffer that is also used for compensation of transmission delay jitter, the receiver buffer is hereafter called the de-packetization buffer in

this section. Receivers should also prepare for transmission delay jitter; that is, either reserve separate buffers for transmission delay jitter buffering and de-packetization buffering or use a receiver buffer for both transmission delay jitter and de-packetization. Moreover, receivers should take transmission delay jitter into account in the buffering operation, e.g., by additional initial buffering before starting of decoding and playback.

The de-packetization process extracts the NAL units from the RTP packets in an RTP stream as follows. When an RTP packet carries a single NAL unit packet, the payload of the RTP packet is extracted as a single NAL unit, excluding the DONL field, i.e., third and fourth bytes, when `sprop-max-don-diff` is greater than 0. When an RTP packet carries an Aggregation Packet, several NAL units are extracted from the payload of the RTP packet. In this case, each NAL unit corresponds to the part of the payload of each aggregation unit that follows the NALU size field as described in Section 4.3.2. When an RTP packet carries a Fragmentation Unit (FU), all RTP packets from the first FU (with the S field equal to 1) of the fragmented NAL unit up to the last FU (with the E field equal to 1) of the fragmented NAL unit are collected. The NAL unit is extracted from these RTP packets by concatenating all FU payloads in the same order as the corresponding RTP packets and appending the NAL unit header with the fields F, LayerId, and TID, set to equal to the values of the fields F, LayerId, and TID in the payload header of the FUs respectively, and with the NAL unit type set equal to the value of the field FuType in the FU header of the FUs, as described in Section 4.3.3.

When `sprop-max-don-diff` is equal to 0, the de-packetization buffer size is zero bytes, and the NAL units carried in the single RTP stream are directly passed to the decoder in their transmission order, which is identical to their decoding order.

When `sprop-max-don-diff` is greater than 0, the process described in the remainder of this section applies.

There are two buffering states in the receiver: initial buffering and buffering while playing. Initial buffering starts when the reception is initialized. After initial buffering, decoding and playback are started, and the buffering-while-playing mode is used.

Regardless of the buffering state, the receiver stores incoming NAL units in reception order into the de-packetization buffer. NAL units carried in RTP packets are stored in the de-packetization buffer individually, and the value of `AbsDon` is calculated and stored for each NAL unit.

Initial buffering lasts until the difference between the greatest and smallest AbsDon values of the NAL units in the de-packetization buffer is greater than or equal to the value of sprop-max-don-diff.

After initial buffering, whenever the difference between the greatest and smallest AbsDon values of the NAL units in the de-packetization buffer is greater than or equal to the value of sprop-max-don-diff, the following operation is repeatedly applied until this difference is smaller than sprop-max-don-diff:

- * The NAL unit in the de-packetization buffer with the smallest value of AbsDon is removed from the de-packetization buffer and passed to the decoder.

When no more NAL units are flowing into the de-packetization buffer, all NAL units remaining in the de-packetization buffer are removed from the buffer and passed to the decoder in the order of increasing AbsDon values.

7. Payload Format Parameters

This section specifies the optional parameters. A mapping of the parameters with Session Description Protocol (SDP) [RFC4556] is also provided for applications that use SDP.

7.1. Media Type Registration

The receiver MUST ignore any parameter unspecified in this memo.

Type name: video

Subtype name: H266

Required parameters: N/A

Optional parameters:

profile-id, tier-flag, sub-profile-id, interop-constraints, level-id, sprop-sublayer-id, sprop-ols-id, recv-sublayer-id, recv-ols-id, max-recv-level-id, sprop-dci, sprop-vps, sprop-sps, sprop-pps, sprop-sei, max-lsr, max-fps, sprop-max-don-diff, sprop-depack-buf-bytes, depack-buf-cap (Refer to Section 7.2 for definitions).

Encoding considerations:

This type is only defined for transfer via RTP (RFC 3550).

Security considerations:

See Section 9 of RFC XXXX.

Interoperability considerations: N/A

Published specification:

Please refer to RFC XXXX and its Section 13.

Applications that use this media type:

Any application that relies on VVC-based video services over RTP

Fragment identifier considerations: N/A

Additional information: N/A

Person & email address to contact for further information:

Stephan Wenger (stewe@stewe.org)

Intended usage: COMMON

Restrictions on usage: N/A

Author: See Authors' Addresses section of RFC XXXX.

Change controller:

IETF Audio/Video Transport Core Maintenance Working Group
delegated from the IESG.

7.2. Optional Parameters Definition

profile-id, tier-flag, sub-profile-id, interop-constraints, and
level-id:

These parameters indicate the profile, tier, default level, sub-profile, and some constraints of the bitstream carried by the RTP stream, or a specific set of the profile, tier, default level, sub-profile and some constraints the receiver supports.

The subset of coding tools that may have been used to generate the bitstream or that the receiver supports, as well as some additional constraints are indicated collectively by profile-id, sub-profile-id, and interop-constraints.

Informative note: There are 128 values of profile-id. The subset of coding tools identified by the profile-id can be further constrained with up to 255 instances of sub-profile-id. In addition, 68 bits included in interop-constraints, which can be extended up to 324 bits provide means to further restrict tools from existing profiles. To be able to support this fine-granular signaling of coding tool subsets with profile-id, sub-profile-id and interop-constraints, it would be safe to require symmetric use of these parameters in SDP offer/answer unless recv-ols-id is included in the SDP answer for choosing one of the layers offered.

The tier is indicated by tier-flag. The default level is indicated by level-id. The tier and the default level specify the limits on values of syntax elements or arithmetic combinations of values of syntax elements that are followed when generating the bitstream or that the receiver supports.

In SDP offer/answer, when the SDP answer does not include the recv-ols-id parameter that is less than the sprop-ols-id parameter in the SDP offer, the following applies:

- The tier-flag, profile-id, sub-profile-id, and interop-constraints parameters MUST be used symmetrically, i.e., the value of each of these parameters in the offer MUST be the same as that in the answer, either explicitly signaled or implicitly inferred.
- The level-id parameter is changeable as long as the highest level indicated by the answer is either equal to or lower than that in the offer. Note that a highest level higher than level-id in the offer for receiving can be included as max-recv-level-id.

In SDP offer/answer, when the SDP answer does include the recv-ols-id parameter that is less than the sprop-ols-id parameter in the SDP offer, the set of tier-flag, profile-id, sub-profile-id, interop-constraints, and level-id parameters included in the answer MUST be consistent with that for the chosen output layer set as indicated in the SDP offer, with the exception that the level-id parameter in the SDP answer is changeable as long as the highest level indicated by the answer is either lower than or equal to that in the offer.

More specifications of these parameters, including how they relate to syntax elements specified in [VVC] are provided below.

profile-id:

When profile-id is not present, a value of 1 (i.e., the Main 10 profile) MUST be inferred.

When used to indicate properties of a bitstream, profile-id is derived from the general_profile_idc syntax element that applies to the bitstream in an instance of the profile_tier_level() syntax structure.

VVC bitstreams transported over RTP using the technologies of this memo SHOULD contain only a single profile_tier_level() structure in the DCI, unless the sender can assure that a receiver can correctly decode the VVC bitstream regardless of which profile_tier_level() structure contained in the DCI was used for deriving profile-id and other parameters for the SDP O/A exchange.

As specified in [VVC], a profile_tier_level() syntax structure may be contained in an SPS NAL unit, and one or more profile_tier_level() syntax structures may be contained in a VPS NAL unit and in a DCI NAL unit. One of the following three cases applies to the container NAL unit of the profile_tier_level() syntax structure containing syntax elements used to derive the values of profile-id, tier-flag, level-id, sub-profile-id, or interop-constraints: 1) The container NAL unit is an SPS, the bitstream is a single-layer bitstream, and the profile_tier_level() syntax structures in all SPSS referenced by the CVSSs in the bitstream has the same values respectively for those profile_tier_level() syntax elements; 2) The container NAL unit is a VPS, the profile_tier_level() syntax structure is the one in the VPS that applies to the OLS corresponding to the bitstream, and the profile_tier_level() syntax structures applicable to the OLS corresponding to the bitstream in all VPSSs referenced by the CVSSs in the bitstream have the same values respectively for those profile_tier_level() syntax elements; 3) The container NAL unit is a DCI NAL unit and the profile_tier_level() syntax structures in all DCI NAL units in the bitstream has the same values respectively for those profile_tier_level() syntax elements.

[VVC] allows for multiple profile_tier_level() structures in a DCI NAL unit, which may contain different values for the syntax elements used to derive the values of profile-id, tier-flag, level-id, sub-profile-id, or interop-constraints in the different entries. However, herein defined is only a single profile-id, tier-flag, level-id, sub-profile-id, or interop-constraints. When signaling these parameters and a DCI NAL unit is present with multiple profile_tier_level() structures, these values SHOULD be the same as the first profile_tier_level structure in the DCI, unless the sender has ensured that the receiver can decode the bitstream when a different value is chosen.

tier-flag, level-id:

The value of tier-flag MUST be in the range of 0 to 1, inclusive.
The value of level-id MUST be in the range of 0 to 255, inclusive.

If the tier-flag and level-id parameters are used to indicate properties of a bitstream, they indicate the tier and the highest level the bitstream complies with.

If the tier-flag and level-id parameters are used for capability exchange, the following applies. If max-recv-level-id is not present, the default level defined by level-id indicates the highest level the codec wishes to support. Otherwise, max-recv-level-id indicates the highest level the codec supports for receiving. For either receiving or sending, all levels that are lower than the highest level supported MUST also be supported.

If no tier-flag is present, a value of 0 MUST be inferred; if no level-id is present, a value of 51 (i.e., level 3.1) MUST be inferred.

Informative note: The level values currently defined in the VVC specification are in the form of "majorNum.minorNum", and the value of the level-id for each of the levels is equal to $\text{majorNum} * 16 + \text{minorNum} * 3$. It is expected that if any levels are defined in the future, the same convention will be used, but this cannot be guaranteed.

When used to indicate properties of a bitstream, the tier-flag and level-id parameters are derived respectively from the syntax element `general_tier_flag`, and the syntax element `general_level_idc` or `sub_layer_level_idc[j]`, that apply to the bitstream, in an instance of the `profile_tier_level()` syntax structure.

If the tier-flag and level-id are derived from the `profile_tier_level()` syntax structure in a DCI NAL unit, the following applies:

- tier-flag = `general_tier_flag`
- level-id = `general_level_idc`

Otherwise, if the tier-flag and level-id are derived from the `profile_tier_level()` syntax structure in an SPS or VPS NAL unit, and the bitstream contains the highest sublayer representation in the OLS corresponding to the bitstream, the following applies:

- tier-flag = general_tier_flag

- level-id = general_level_idc

Otherwise, if the tier-flag and level-id are derived from the profile_tier_level() syntax structure in an SPS or VPS NAL unit, and the bitstream does not contain the highest sublayer representation in the OLS corresponding to the bitstream, the following applies, with j being the value of the sprop-sublayer-id parameter:

- tier-flag = general_tier_flag

- level-id = sub_layer_level_idc[j]

sub-profile-id:

The value of the parameter is a comma-separated (',') list of data using base64 [RFC4648] representation.

When used to indicate properties of a bitstream, sub-profile-id is derived from each of the ptl_num_sub_profiles general_sub_profile_idc[i] syntax elements that apply to the bitstream in a profile_tier_level() syntax structure.

interop-constraints:

A base64 [RFC4648] representation of the data that includes the syntax elements ptl_frame_only_constraint_flag and ptl_multilayer_enabled_flag and the general_constraints_info() syntax structure that apply to the bitstream in an instance of the profile_tier_level() syntax structure.

If the interop-constraints parameter is not present, the following MUST be inferred:

- ptl_frame_only_constraint_flag = 1

- ptl_multilayer_enabled_flag = 0

- gci_present_flag in the general_constraints_info() syntax structure = 0

Using interop-constraints for capability exchange results in a requirement on any bitstream to be compliant with the interop-constraints.

sprop-sublayer-id:

This parameter MAY be used to indicate the highest allowed value of TID in the bitstream. When not present, the value of sprop-sublayer-id is inferred to be equal to 6.

The value of sprop-sublayer-id MUST be in the range of 0 to 6, inclusive.

sprop-ols-id:

This parameter MAY be used to indicate the OLS that the bitstream applies to. When not present, the value of sprop-ols-id is inferred to be equal to TargetOlsIdx as specified in 8.1.1 in [VVC]. If this optional parameter is present, sprop-vps MUST also be present or its content MUST be known a priori at the receiver.

The value of sprop-ols-id MUST be in the range of 0 to 256, inclusive.

Informative note: VVC allows having up to 257 output layer sets indicated in the VPS as the number of output layer sets minus 2 is indicated with a field of 8 bits.

recv-sublayer-id:

This parameter MAY be used to signal a receiver's choice of the offered or declared sublayer representations in the sprop-vps and sprop-sps. The value of recv-sublayer-id indicates the TID of the highest sublayer that a receiver supports. When not present, the value of recv-sublayer-id is inferred to be equal to the value of the sprop-sublayer-id parameter in the SDP offer.

The value of recv-sublayer-id MUST be in the range of 0 to 6, inclusive.

recv-ols-id:

This parameter MAY be used to signal a receiver's choice of the offered or declared output layer sets in the sprop-vps. The value of recv-ols-id indicates the OLS index of the bitstream that a receiver supports. When not present, the value of recv-ols-id is inferred to be equal to value of the sprop-ols-id parameter inferred from or indicated in the SDP offer. When present, the value of recv-ols-id must be included only when sprop-ols-id was received and must refer to an output layer set in the VPS that includes no layers other than all or a subset of the layers of the OLS referred to by sprop-ols-id. If this optional parameter is present, sprop-vps must have been received or its content must be known a priori at the receiver.

The value of `recv-ols-id` MUST be in the range of 0 to 256, inclusive.

`max-recv-level-id`:

This parameter MAY be used to indicate the highest level a receiver supports.

The value of `max-recv-level-id` MUST be in the range of 0 to 255, inclusive.

When `max-recv-level-id` is not present, the value is inferred to be equal to `level-id`.

`max-recv-level-id` MUST NOT be present when the highest level the receiver supports is not higher than the default level.

`sprop-dci`:

This parameter MAY be used to convey a decoding capability information NAL unit of the bitstream for out-of-band transmission. The parameter MAY also be used for capability exchange. The value of the parameter is a base64 [RFC4648] representations of the decoding capability information NAL unit as specified in Section 7.3.2.1 of [VVC].

`sprop-vps`:

This parameter MAY be used to convey any video parameter set NAL unit of the bitstream for out-of-band transmission of video parameter sets. The parameter MAY also be used for capability exchange and to indicate sub-stream characteristics (i.e., properties of output layer sets and sublayer representations as defined in [VVC]). The value of the parameter is a comma-separated ('(',')') list of base64 [RFC4648] representations of the video parameter set NAL units as specified in Section 7.3.2.3 of [VVC].

The `sprop-vps` parameter MAY contain one or more than one video parameter set NAL units. However, all other video parameter sets contained in the `sprop-vps` parameter MUST be consistent with the first video parameter set in the `sprop-vps` parameter. A video parameter set `vpsB` is said to be consistent with another video parameter set `vpsA` if the number of OLSs in `vpsA` and `vpsB` is the same and any decoder that conforms to the profile, tier, level, and constraints indicated by the data starting from the syntax element `general_profile_idc` to the syntax structure `general_constraints_info()`, inclusive, in the `profile_tier_level()`

) syntax structure corresponding to any OLS with index `olsIdx` in `vpsA` can decode any CVS(s) referencing `vpsB` when `TargetOlsIdx` is equal to `olsIdx` that conforms to the profile, tier, level, and constraints indicated by the data starting from the syntax element `general_profile_idc` to the syntax structure `general_constraints_info()`, inclusive, in the `profile_tier_level()` syntax structure corresponding to the OLS with index `TargetOlsIdx` in `vpsB`.

`sprop-sps:`

This parameter MAY be used to convey sequence parameter set NAL units of the bitstream for out-of-band transmission of sequence parameter sets. The value of the parameter is a comma-separated ('(',')') list of base64 [RFC4648] representations of the sequence parameter set NAL units as specified in Section 7.3.2.4 of [VVC].

A sequence parameter set `spsB` is said to be consistent with another sequence parameter set `spsA` if any decoder that conforms to the profile, tier, level, and constraints indicated by the data starting from the syntax element `general_profile_idc` to the syntax structure `general_constraints_info()`, inclusive, in the `profile_tier_level()` syntax structure in `spsA` can decode any CLVS(s) referencing `spsB` that conforms to the profile, tier, level, and constraints indicated by the data starting from the syntax element `general_profile_idc` to the syntax structure `general_constraints_info()`, inclusive, in the `profile_tier_level()` syntax structure in `spsB`.

`sprop-pps:`

This parameter MAY be used to convey picture parameter set NAL units of the bitstream for out-of-band transmission of picture parameter sets. The value of the parameter is a comma-separated ('(',')') list of base64 [RFC4648] representations of the picture parameter set NAL units as specified in Section 7.3.2.5 of [VVC].

`sprop-sei:`

This parameter MAY be used to convey one or more SEI messages that describe bitstream characteristics. When present, a decoder can rely on the bitstream characteristics that are described in the SEI messages for the entire duration of the session, independently from the persistence scopes of the SEI messages as specified in [VSEI].

The value of the parameter is a comma-separated (',') list of base64 [RFC4648] representations of SEI NAL units as specified in [VSEI].

Informative note: Intentionally, no list of applicable or inapplicable SEI messages is specified here. Conveying certain SEI messages in sprop-sei may be sensible in some application scenarios and meaningless in others. However, a few examples are described below:

- 1) In an environment where the bitstream was created from film-based source material, and no splicing is going to occur during the lifetime of the session, the film grain characteristics SEI message is likely meaningful, and sending it in sprop-sei rather than in the bitstream at each entry point may help with saving bits and allows one to configure the renderer only once, avoiding unwanted artifacts.
- 2) Examples for SEI messages that would be meaningless to be conveyed in sprop-sei include the decoded picture hash SEI message (it is close to impossible that all decoded pictures have the same hashtag) or the filler payload SEI message (as there is no point in just having more bits in SDP).

max-lsr:

The max-lsr MAY be used to signal the capabilities of a receiver implementation and MUST NOT be used for any other purpose. The value of max-lsr is an integer indicating the maximum processing rate in units of luma samples per second. The max-lsr parameter signals that the receiver is capable of decoding video at a higher rate than is required by the highest level.

Informative note: When the OPTIONAL media type parameters are used to signal the properties of a bitstream, and max-lsr is not present, the values of tier-flag, profile-id, sub-profile-id interop-constraints, and level-id must always be such that the bitstream complies fully with the specified profile, tier, and level.

When max-lsr is signaled, the receiver MUST be able to decode bitstreams that conform to the highest level, with the exception that the MaxLumaSr value in Table 136 of [VVC] for the highest level is replaced with the value of max-lsr. Senders MAY use this knowledge to send pictures of a given size at a higher picture rate than is indicated in the highest level.

When not present, the value of max-lsr is inferred to be equal to the value of MaxLumaSr given in Table 136 of [VVC] for the highest level.

The value of max-lsr MUST be in the range of MaxLumaSr to $16 * \text{MaxLumaSr}$, inclusive, where MaxLumaSr is given in Table 136 of [VVC] for the highest level.

max-fps:

The value of max-fps is an integer indicating the maximum picture rate in units of pictures per 100 seconds that can be effectively processed by the receiver. The max-fps parameter MAY be used to signal that the receiver has a constraint in that it is not capable of processing video effectively at the full picture rate that is implied by the highest level and, when present, max-lsr.

The value of max-fps is not necessarily the picture rate at which the maximum picture size can be sent, it constitutes a constraint on maximum picture rate for all resolutions.

Informative note: The max-fps parameter is semantically different from max-lsr in that max-fps is used to signal a constraint, lowering the maximum picture rate from what is implied by other parameters.

The encoder MUST use a picture rate equal to or less than this value. In cases where the max-fps parameter is absent, the encoder is free to choose any picture rate according to the highest level and any signaled optional parameters.

The value of max-fps MUST be smaller than or equal to the full picture rate that is implied by the highest level and, when present, max-lsr.

sprop-max-don-diff:

If there is no NAL unit naluA that is followed in transmission order by any NAL unit preceding naluA in decoding order (i.e., the transmission order of the NAL units is the same as the decoding order), the value of this parameter MUST be equal to 0.

Otherwise, this parameter specifies the maximum absolute difference between the decoding order number (i.e., AbsDon) values of any two NAL units naluA and naluB, where naluA follows naluB in decoding order and precedes naluB in transmission order.

The value of `sprop-max-don-diff` MUST be an integer in the range of 0 to 32767, inclusive.

When not present, the value of `sprop-max-don-diff` is inferred to be equal to 0.

`sprop-depack-buf-bytes`:

This parameter signals the required size of the de-packetization buffer in units of bytes. The value of the parameter MUST be greater than or equal to the maximum buffer occupancy (in units of bytes) of the de-packetization buffer as specified in Section 6.

The value of `sprop-depack-buf-bytes` MUST be an integer in the range of 0 to 4294967295, inclusive.

When `sprop-max-don-diff` is present and greater than 0, this parameter MUST be present and the value MUST be greater than 0. When not present, the value of `sprop-depack-buf-bytes` is inferred to be equal to 0.

Informative note: The value of `sprop-depack-buf-bytes` indicates the required size of the de-packetization buffer only. When network jitter can occur, an appropriately sized jitter buffer has to be available as well.

`depack-buf-cap`:

This parameter signals the capabilities of a receiver implementation and indicates the amount of de-packetization buffer space in units of bytes that the receiver has available for reconstructing the NAL unit decoding order from NAL units carried in the RTP stream. A receiver is able to handle any RTP stream for which the value of the `sprop-depack-buf-bytes` parameter is smaller than or equal to this parameter.

When not present, the value of `depack-buf-cap` is inferred to be equal to 4294967295. The value of `depack-buf-cap` MUST be an integer in the range of 1 to 4294967295, inclusive.

Informative note: `depack-buf-cap` indicates the maximum possible size of the de-packetization buffer of the receiver only, without allowing for network jitter.

7.3. SDP Parameters

The receiver MUST ignore any parameter unspecified in this memo.

7.3.1. Mapping of Payload Type Parameters to SDP

The media type video/H266 string is mapped to fields in the Session Description Protocol (SDP) [RFC8866] 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 H266 (the media subtype).
- * The clock rate in the "a=rtpmap" line MUST be 90000.
- * The OPTIONAL parameters profile-id, tier-flag, sub-profile-id, interop-constraints, level-id, sprop-sublayer-id, sprop-ols-id, recv-sublayer-id, recv-ols-id, max-recv-level-id, max-lsr, max-fps, sprop-max-don-diff, sprop-depack-buf-bytes and depack-buf-cap, when present, MUST be included in the "a=fmtp" line of SDP. The fmtp line is expressed as a media type string, in the form of a semicolon-separated list of parameter=value pairs.
- * The OPTIONAL parameter sprop-vps, sprop-sps, sprop-pps, sprop-sei, and sprop-dci, when present, MUST be included in the "a=fmtp" line of SDP or conveyed using the "fmtp" source attribute as specified in Section 6.3 of [RFC5576]. For a particular media format (i.e., RTP payload type), sprop-vps, sprop-sps, sprop-pps, sprop-sei, or sprop-dci MUST NOT be both included in the "a=fmtp" line of SDP and conveyed using the "fmtp" source attribute. When included in the "a=fmtp" line of SDP, those parameters are expressed as a media type string, in the form of a semicolon-separated list of parameter=value pairs. When conveyed in the "a=fmtp" line of SDP for a particular payload type, the parameters sprop-vps, sprop-sps, sprop-pps, sprop-sei, and sprop-dci MUST be applied to each SSRC with the payload type. When conveyed using the "fmtp" source attribute, these parameters are only associated with the given source and payload type as parts of the "fmtp" source attribute.

Informative note: Conveyance of sprop-vps, sprop-sps, and sprop-pps using the "fmtp" source attribute allows for out-of-band transport of parameter sets in topologies like Topo-Video-switch-MCU as specified in [RFC7667]

An general usage of media representation in SDP is as follows:

```
m=video 49170 RTP/AVP 98
a=rtpmap:98 H266/90000
a=fmtp:98 profile-id=1;
    sprop-vps=<video parameter sets data>;
    sprop-sps=<sequence parameter set data>;
    sprop-pps=<picture parameter set data>;
```

A SIP Offer/Answer exchange wherein both parties are expected to both send and receive could look like the following. Only the media codec-specific parts of the SDP are shown. Some lines are wrapped due to text constraints.

Offerer->Answerer:

```
m=video 49170 RTP/AVP 98
a=rtpmap:98 H266/90000
a=fmtp:98 profile-id=1; level_id=83;
```

The above represents an offer for symmetric video communication using [VVC] and it's payload specification, at the main profile and level 5.1 (and, as the levels are downgradable, all lower levels. Informally speaking, this offer tells the receiver of the offer that the sender is willing to receive up to 4Kp60 resolution at the maximum bitrates specified in [VVC]. At the same time, if this offer were accepted "as is", the offer can expect that the answerer would be able to receive and properly decode H.266 media up to and including level 5.1.

Answerer->Offerer:

```
m=video 49170 RTP/AVP 98
a=rtpmap:98 H266/90000
a=fmtp:98 profile-id=1; level_id=67
```

With this answer to the offer above, the system receiving the offer advises the offerer that it is incapable of handling H.266 at level 5.1 but is capable of decoding 1080p60. As H.266 video codecs must support decoding at all levels below the maximum level they implement, the resulting user experience would likely be that both systems send video at 1080p60. However, nothing prevents an encoder from further downgrading its sending to, for example 720p30 if it were short of cycles, bandwidth, or for other reasons.

7.3.2. Usage with SDP Offer/Answer Model

This section describes the negotiation of unicast messages using the offer-answer model as described in [RFC3264] and its updates. The section is split into subsections, covering a) media format configurations not involving non-temporal scalability; b) scalable media format configurations; c) the description of the use of those parameters not involving the media configuration itself but rather the parameters of the payload format design; and d) multicast.

7.3.2.1. Non-scalable media format configuration

A non-scalable VVC media configuration is such a configuration where no non-temporal scalability mechanisms are allowed. In [VVC] version 1, that implies that `general_profile_idc` indicates one of the following profiles: Main10, Main10 Still Picture, Main 10 4:4:4, Main10 4:4:4 Still Picture, with `general_profile_idc` values of 1, 65, 33, and 97, respectively. Note that non-scalable media configurations includes temporal scalability, inline with VVC's design philosophy and profile structure.

The following limitations and rules pertaining to the media configuration apply:

- * The parameters identifying a media format configuration for VVC are `profile-id`, `tier-flag`, `sub-profile-id`, `level-id`, and `interop-constraints`. These media configuration parameters, except `level-id`, MUST be used symmetrically.

The answerer MUST structure its answer in according to one of the following three options:

1) maintain all configuration parameters with the values remaining the same as in the offer for the media format (payload type), with the exception that the value of `level-id` is changeable as long as the highest level indicated by the answer is not higher than that indicated by the offer;

2) include in the answer the `recv-sublayer-id` parameter, with a value less than the `sprop-sublayer-id` parameter in the offer, for the media format (payload type), and maintain all configuration parameters with the values remaining the same as in the offer for the media format (payload type), with the exception that the value of `level-id` is changeable as long as the highest level indicated by the answer is not higher than the level indicated by the `sprop-sps` or `sprop-vps` in offer for the chosen sublayer representation; or

3) remove the media format (payload type) completely (when one or more of the parameter values are not supported).

Informative note: The above requirement for symmetric use does not apply for level-id, and does not apply for the other bitstream or RTP stream properties and capability parameters as described in Section 7.3.2.3 below.

- * To simplify 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].
- * The same RTP payload type number used in the offer for the media subtype H266 MUST be used in the answer when the answer includes recv-sublayer-id. When the answer does not include recv-sublayer-id, the answer MUST NOT contain a payload type number used in the offer for the media subtype H266 unless the configuration is exactly the same as in the offer or the configuration in the answer only differs from that in the offer with a different value of level-id. The answer MAY contain the recv-sublayer-id parameter if an VVC bitstream contains multiple operation points (using temporal scalability and sublayers) and sprop-sps or sprop-vps is included in the offer where information of sublayers are present in the first sequence parameter set or video parameter set contained in sprop-sps or sprop-vps respectively. If the sprop-sps or sprop-vps is provided in an offer, an answerer MAY select a particular operation point indicated in the first sequence parameter set or video parameter set contained in sprop-sps or sprop-vps respectively. When the answer includes a recv-sublayer-id that is less than a sprop-sublayer-id in the offer, the following applies:
 - 1) When sprop-sps parameter is present, all sequence parameter sets contained in the sprop-sps parameter in the SDP answer and all sequence parameter sets sent in-band for either the offerer-to-answerer direction or the answerer-to-offerer direction MUST be consistent with the first sequence parameter set in the sprop-sps parameter of the offer (see the semantics of sprop-sps in Section 7.1 of this document on one sequence parameter set being consistent with another sequence parameter set).

2) When sprop-vps parameter is present, all video parameter sets contained in the sprop-vps parameter in the SDP answer and all video parameter sets sent in-band for either the offerer-to-answerer direction or the answerer-to-offerer direction MUST be consistent with the first video parameter set in the sprop-vps parameter of the offer (see the semantics of sprop-vps in Section 7.1 of this document on one video parameter set being consistent with another video parameter set).

3) The bitstream sent in either direction MUST conform to the profile, tier, level, and constraints of the chosen sublayer representation as indicated by the profile_tier_level() syntax structure in the first sequence parameter set in the sprop-sps parameter or by the first profile_tier_level() syntax structure in the first video parameter set in the sprop-vps parameter of the offer.

Informative note: When an offerer receives an answer that does not include recv-sublayer-id, it has to compare payload types not declared in the offer based on the media type (i.e., video/H266) and the above media configuration parameters with any payload types it has already declared. This will enable it to determine whether the configuration in question is new or if it is equivalent to configuration already offered, since a different payload type number may be used in the answer. The ability to perform operation point selection enables a receiver to utilize the temporal scalable nature of an VVC bitstream.

7.3.2.2. Scalable media format configuration

A scalable VVC media configuration is such a configuration where non-temporal scalability mechanisms are allowed. In [VVC] version 1, that implies that general_profile_idc indicates one of the following profiles: Multilayer Main 10, and Multilayer Main 10 4:4:4, with general_profile_idc values of 17 and 49, respectively.

The following limitations and rules pertaining to the media configuration apply. They are listed in an order that would be logical for an implementation to follow:

- * The parameters identifying a media format configuration for scalable VVC are profile-id, tier-flag, sub-profile-id, level-id, interop-constraints, and sprop-vps. These media configuration parameters, except level-id, MUST be used symmetrically, except as noted below.

- * The answerer MAY include a level-id that MUST be lower than or equal to the level-id indicated in the offer (either expressed by level-id in the offer, or implied by the default level as specific in Section 7.1).
- * When sprop-ols-id is present in an offer, sprop-vps MUST also be present in the same offer and including at least one valid VPS, so to allow the answerer to meaningfully interpret sprop-ols-id and select recv-ols-id (see below).
- * The answerer MUST NOT include recv-ols-id unless the offer includes sprop-ols-id. When present, recv-ols-id MUST indicate a supported output layer set in the VPS that includes no layers other than all or a subset of the layers of the OLS referred to by sprop-ols-id. If unable, the answerer MUST remove the media format.

Informative note: if an offerer wants to offer more than one output layer set, it can do so by offering multiple VVC media with different payload types.

- * The offerer MAY include sprop-sublayer-id which indicates the highest allowed value of TID in the bitstream. The answerer MAY include recv-sublayer-id which can be used to reduce the number of sublayers from the value of sprop-sublayer-id.
- * When the answerer includes recv-ols-id and configuration parameters profile-id, tier-flag, sub-profile-id, level-id, and interop-constraints, it MUST use the configuration parameter values as signaled in the sprop-vps for the operating point with the largest number of sublayers for the chosen output layer set, with the exception that the value of level-id is changeable as long as the highest level indicated by the answer is not higher than the level indicated by the sprop-vps in offer for the operating point with the largest number of sublayers for the chosen output layer set.

7.3.2.3. Payload format configuration

The following limitations and rules pertain to the configuration of the payload format buffer management mostly and apply to both scalable and non-scalable VVC.

- * The parameters sprop-max-don-diff, and sprop-depack-buf-bytes describe the properties of an RTP stream that the offerer or the answerer is sending for the media format configuration. This differs from the normal usage of the offer/answer parameters: normally such parameters declare the properties of the bitstream

or RTP stream that the offerer or the answerer is able to receive. When dealing with VVC, the offerer assumes that the answerer will be able to receive media encoded using the configuration being offered.

Informative note: The above parameters apply for any RTP stream, when present, sent by a declaring entity with the same configuration. In other words, the applicability of the above parameters to RTP streams depends on the source endpoint. Rather than being bound to the payload type, the values may have to be applied to another payload type when being sent, as they apply for the configuration.

- * The capability parameter max-lsr MAY be used to declare further capabilities of the offerer or answerer for receiving. It MUST NOT be present when the direction attribute is sendonly.
- * The capability parameter max-fps MAY be used to declare lower capabilities of the offerer or answerer for receiving. It MUST NOT be present when the direction attribute is sendonly.
- * When an offerer offers an interleaved stream, indicated by the presence of sprop-max-don-diff with a value larger than zero, the offerer MUST include the size of the de-packetization buffer sprop-depack-buf-bytes.
- * To enable the offerer and answerer to inform each other about their capabilities for de-packetization buffering in receiving RTP streams, both parties are RECOMMENDED to include depack-buf-cap.
- * The sprop-dci, sprop-vps, sprop-sps, or sprop-pps, when present (included in the "a=fmtp" line of SDP or conveyed using the "fmtp" source attribute as specified in Section 6.3 of [RFC5576]), are used for out-of-band transport of the parameter sets (DCI, VPS, SPS, or PPS, respectively).
- * The answerer MAY use either out-of-band or in-band transport of parameter sets for the bitstream it is sending, regardless of whether out-of-band parameter sets transport has been used in the offerer-to-answerer direction. Parameter sets included in an answer are independent of those parameter sets included in the offer, as they are used for decoding two different bitstreams, one from the answerer to the offerer and the other in the opposite direction. In case some RTP packets are sent before the SDP offer/answer settles down, in-band parameter sets MUST be used for those RTP stream parts sent before the SDP offer/answer.

- * The following rules apply to transport of parameter set in the offerer-to-answerer direction.
 - An offer MAY include sprop-dci, sprop-vps, sprop-sps, and/or sprop-pps. If none of these parameters is present in the offer, then only in-band transport of parameter sets is used.
 - If the level to use in the offerer-to-answerer direction is equal to the default level in the offer, the answerer MUST be prepared to use the parameter sets included in sprop-vps, sprop-sps, and sprop-pps (either included in the "a=fmtp" line of SDP or conveyed using the "fmtp" source attribute) for decoding the incoming bitstream, e.g., by passing these parameter set NAL units to the video decoder before passing any NAL units carried in the RTP streams. Otherwise, the answerer MUST ignore sprop-vps, sprop-sps, and sprop-pps (either included in the "a=fmtp" line of SDP or conveyed using the "fmtp" source attribute) and the offerer MUST transmit parameter sets in-band.
- * The following rules apply to transport of parameter set in the answerer-to-offerer direction.
 - An answer MAY include sprop-dci, sprop-vps, sprop-sps, and/or sprop-pps. If none of these parameters is present in the answer, then only in-band transport of parameter sets is used.
 - The offerer MUST be prepared to use the parameter sets included in sprop-vps, sprop-sps, and sprop-pps (either included in the "a=fmtp" line of SDP or conveyed using the "fmtp" source attribute) for decoding the incoming bitstream, e.g., by passing these parameter set NAL units to the video decoder before passing any NAL units carried in the RTP streams.
- * When sprop-dci, sprop-vps, sprop-sps, and/or sprop-pps are conveyed using the "fmtp" source attribute as specified in Section 6.3 of [RFC5576], the receiver of the parameters MUST store the parameter sets included in sprop-dci, sprop-vps, sprop-sps, and/or sprop-pps and associate them with the source given as part of the "fmtp" source attribute. Parameter sets associated with one source (given as part of the "fmtp" source attribute) MUST only be used to decode NAL units conveyed in RTP packets from the same source (given as part of the "fmtp" source attribute). When this mechanism is in use, SSRC collision detection and resolution MUST be performed as specified in [RFC5576].

Table 1 lists the interpretation of all the parameters that MAY be used for the various combinations of offer, answer, and direction attributes. Note that the two columns wherein the `recv-ols-id` parameter is used only apply to answers, whereas the other columns apply to both offers and answers.

| | sendonly --+ | | | |
|-----------------------------------|--------------|---|---|---|
| answer: recvonly, recv-ols-id --+ | | | | |
| recvonly w/o recv-ols-id --+ | | | | |
| answer: sendrecv, recv-ols-id --+ | | | | |
| sendrecv w/o recv-ols-id --+ | | | | |
| profile-id | C | D | C | D |
| tier-flag | C | D | C | D |
| level-id | D | D | D | D |
| sub-profile-id | C | D | C | D |
| interop-constraints | C | D | C | D |
| max-recv-level-id | R | R | R | R |
| sprop-max-don-diff | P | P | - | - |
| sprop-depack-buf-bytes | P | P | - | - |
| depack-buf-cap | R | R | R | R |
| max-lsr | R | R | R | R |
| max-fps | R | R | R | R |
| sprop-dci | P | P | - | - |
| sprop-sei | P | P | - | - |
| sprop-vps | P | P | - | - |
| sprop-sps | P | P | - | - |
| sprop-pps | P | P | - | - |
| sprop-sublayer-id | P | P | - | - |
| recv-sublayer-id | O | O | O | O |
| sprop-ols-id | P | P | - | - |
| recv-ols-id | X | O | X | O |

Table 1. Interpretation of parameters for various combinations of offers, answers, direction attributes, with and without recv-ols-id. Columns that do not indicate offer or answer apply to both.

Legend:

- C: configuration for sending and receiving bitstreams
- D: changeable configuration, same as C except possible to answer with a different but consistent value (see the semantics of the six parameters related to profile, tier, and level on these parameters being consistent)
- P: properties of the bitstream to be sent
- R: receiver capabilities
- O: operation point selection
- X: MUST NOT be present
- : not usable, when present MUST be ignored

Parameters used for declaring receiver capabilities are, in general, downgradable; i.e., they express the upper limit for a sender's possible behavior. Thus, a sender MAY select to set its encoder using only lower/lesser or equal values of these parameters.

When the answer does not include a `recv-ols-id` that is less than the `sprop-ols-id` in the offer, parameters declaring a configuration point are not changeable, with the exception of the `level-id` parameter for unicast usage, and these parameters express values a receiver expects to be used and MUST be used verbatim in the answer as in the offer.

When a sender's capabilities are declared with the configuration parameters, these parameters express a configuration that is acceptable for the sender to receive bitstreams. In order to achieve high interoperability levels, it is often advisable to offer multiple alternative configurations. It is impossible to offer multiple configurations in a single payload type. Thus, when multiple configuration offers are made, each offer requires its own RTP payload type associated with the offer. However, it is possible to offer multiple operation points using one configuration in a single payload type by including `sprop-vps` in the offer and `recv-ols-id` in the answer.

An implementation SHOULD be able to understand all media type parameters (including all optional media type parameters), even if it doesn't support the functionality related to the parameter. This, in conjunction with proper application logic in the implementation allows the implementation, after having received an offer, to create an answer by potentially downgrading one or more of the optional parameters to the point where the implementation can cope, leading to higher chances of interoperability beyond the most basic interop points (for which, as described above, no optional parameters are necessary).

Informative note: in implementations of previous H.26x payload formats it was occasionally observed that implementations were incapable of parsing most (or all) of the optional parameters. As a result, the offer-answer exchange resulted in a baseline performance (using the default values for the optional parameters) with the resulting suboptimal user experience. However, there are valid reasons to forego the implementation complexity of implementing the parsing of some or all of the optional parameters, for example, when there is pre-determined knowledge, not negotiated by an SDP-based offer/answer process, of the capabilities of the involved systems (walled gardens, baseline requirements defined in application standards higher up in the stack, and similar).

An answerer MAY extend the offer with additional media format configurations. However, to enable their usage, in most cases a second offer is required from the offerer to provide the bitstream property parameters that the media sender will use. This also has the effect that the offerer has to be able to receive this media format configuration, not only to send it.

7.3.2.4. Multicast

For bitstreams being delivered over multicast, the following rules apply:

- * The media format configuration is identified by profile-id, tier-flag, sub-profile-id, level-id, and interop-constraints. These media format configuration parameters, including level-id, MUST be used symmetrically; that is, the answerer MUST either maintain all configuration parameters or remove the media format (payload type) completely. Note that this implies that the level-id for offer/answer in multicast is not changeable.
- * 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 a payload type number used in the offer unless the configuration is the same as in the offer.
- * Parameter sets received MUST be associated with the originating source and MUST only be used in decoding the incoming bitstream from the same source.
- * The rules for other parameters are the same as above for unicast as long as the three above rules are obeyed.

7.3.3. Usage in Declarative Session Descriptions

When VVC over RTP is offered with SDP in a declarative style, as in Real Time Streaming Protocol (RTSP) [RFC7826] or Session Announcement Protocol (SAP) [RFC2974], the following considerations are necessary.

- * All parameters capable of indicating both bitstream properties and receiver capabilities are used to indicate only bitstream properties. For example, in this case, the parameter profile-id, tier-id, level-id declares the values used by the bitstream, not the capabilities for receiving bitstreams. As a result, the following interpretation of the parameters MUST be used:
 - Declaring actual configuration or bitstream properties:

- o profile-id
 - o tier-flag
 - o level-id
 - o interop-constraints
 - o sub-profile-id
 - o sprop-dci
 - o sprop-vps
 - o sprop-sps
 - o sprop-pps
 - o sprop-max-don-diff
 - o sprop-depack-buf-bytes
 - o sprop-sublayer-id
 - o sprop-ols-id
 - o sprop-sei
- Not usable (when present, they MUST be ignored):
- o max-lsr
 - o max-fps
 - o max-recv-level-id
 - o depack-buf-cap
 - o recv-sublayer-id
 - o recv-ols-id
- A receiver of the SDP is required to support all parameters and values of the parameters provided; otherwise, the receiver MUST reject (RTSP) or not participate in (SAP) the session. It falls on the creator of the session to use values that are expected to be supported by the receiving application.

7.3.4. Considerations for Parameter Sets

When out-of-band transport of parameter sets is used, parameter sets MAY still be additionally transported in-band unless explicitly disallowed by an application, and some of these additional parameter sets may update some of the out-of-band transported parameter sets. Update of a parameter set refers to the sending of a parameter set of the same type using the same parameter set ID but with different values for at least one other parameter of the parameter set.

8. Use with Feedback Messages

The following subsections define the use of the Picture Loss Indication (PLI) and Full Intra Request (FIR) feedback messages with [VVC]. The PLI is defined in [RFC4585], and the FIR message is defined in [RFC5104]. In accordance with this memo, unlike [HEVC], a sender MUST NOT send Slice Loss Indication (SLI) or Reference Picture Selection Indication (RPSI), and a receiver SHOULD ignore RPSI and treat a received SLI as a PLI.

8.1. Picture Loss Indication (PLI)

As specified in RFC 4585, Section 6.3.1, the reception of a PLI by a media sender indicates "the loss of an undefined amount of coded video data belonging to one or more pictures". Without having any specific knowledge of the setup of the bitstream (such as use and location of in-band parameter sets, non-IRAP decoder refresh points, picture structures, and so forth), a reaction to the reception of an PLI by a VVC sender SHOULD be to send an IRAP picture and relevant parameter sets; potentially with sufficient redundancy so to ensure correct reception. However, sometimes information about the bitstream structure is known. For example, state could have been established outside of the mechanisms defined in this document that parameter sets are conveyed out of band only, and stay static for the duration of the session. In that case, it is obviously unnecessary to send them in-band as a result of the reception of a PLI. Other examples could be devised based on a priori knowledge of different aspects of the bitstream structure. In all cases, the timing and congestion control mechanisms of RFC 4585 MUST be observed.

8.2. Full Intra Request (FIR)

The purpose of the FIR message is to force an encoder to send an independent decoder refresh point as soon as possible, while observing applicable congestion-control-related constraints, such as those set out in [RFC8082]).

Upon reception of a FIR, a sender MUST send an IDR picture. Parameter sets MUST also be sent, except when there is a priori knowledge that the parameter sets have been correctly established. A typical example for that is an understanding between sender and receiver, established by means outside this document, that parameter sets are exclusively sent out-of-band.

9. Security Considerations

The scope of this Security Considerations section is limited to the payload format itself and to one feature of [VVC] that may pose a particularly serious security risk if implemented naively. The payload format, in isolation, does not form a complete system. Implementers are advised to read and understand relevant security-related documents, especially those pertaining to RTP (see the Security Considerations section in [RFC3550]), and the security of the call-control stack chosen (that may make use of the media type registration of this memo). Implementers should also consider known security vulnerabilities of video coding and decoding implementations in general and avoid those.

Within this RTP payload format, and with the exception of the user data SEI message as described below, no security threats other than those common to RTP payload formats are known. In other words, neither the various media-plane-based mechanisms, nor the signaling part of this memo, seems to pose a security risk beyond those common to all RTP-based systems.

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]. 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 lays 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]. The rest of this section discusses the security impacting properties of the payload format itself.

Because the data compression used with this payload format is applied end-to-end, any encryption needs to be performed after compression. A potential denial-of-service threat exists for data encodings using compression techniques that have non-uniform receiver-end computational load. The attacker can inject pathological datagrams

into the bitstream that are complex to decode and that cause the receiver to be overloaded. [VVC] is particularly vulnerable to such attacks, as it is extremely simple to generate datagrams containing NAL units that affect the decoding process of many future NAL units. Therefore, the usage of data origin authentication and data integrity protection of at least the RTP packet is RECOMMENDED, for example, with SRTP [RFC3711].

Like HEVC [RFC7798], [VVC] includes a user data Supplemental Enhancement Information (SEI) message. This SEI message allows inclusion of an arbitrary bitstring into the video bitstream. Such a bitstring could include JavaScript, machine code, and other active content. [VVC] leaves the handling of this SEI message to the receiving system. In order to avoid harmful side effects of the user data SEI message, decoder implementations cannot naively trust its content. For example, it would be a bad and insecure implementation practice to forward any JavaScript a decoder implementation detects to a web browser. The safest way to deal with user data SEI messages is to simply discard them, but that can have negative side effects on the quality of experience by the user.

End-to-end security with authentication, integrity, or confidentiality protection will prevent a MANE from performing media-aware operations other than discarding complete packets. In the case of confidentiality protection, it will even be prevented from discarding packets in a media-aware way. To be allowed to perform such operations, a MANE is required to be a trusted entity that is included in the security context establishment.

10. Congestion Control

Congestion control for RTP SHALL be used in accordance with RTP [RFC3550] and with any applicable RTP profile, e.g., AVP [RFC3551]. If best-effort service is being used, an additional requirement is that users of this payload format MUST monitor packet loss to ensure that the packet loss rate is within an acceptable range. 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 all RTP streams combined are achieved. This condition can be satisfied by implementing congestion-control mechanisms to adapt the transmission rate, 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.

The bitrate adaptation necessary for obeying the congestion control principle is easily achievable when real-time encoding is used, for example, by adequately tuning the quantization parameter. However,

when pre-encoded content is being transmitted, bandwidth adaptation requires the pre-coded bitstream to be tailored for such adaptivity. The key mechanisms available in [VVC] are temporal scalability, and spatial/SNR scalability. A media sender can remove NAL units belonging to higher temporal sublayers (i.e., those NAL units with a high value of TID) or higher spatio-SNR layers until the sending bitrate drops to an acceptable range.

The mechanisms mentioned above generally work within a defined profile and level and, therefore, no renegotiation of the channel is required. Only when non-downgradable parameters (such as profile) are required to be changed does it become necessary to terminate and restart the RTP stream(s). This may be accomplished by using different RTP payload types.

MANEs MAY remove certain unusable packets from the RTP stream when that RTP stream was damaged due to previous packet losses. This can help reduce the network load in certain special cases. For example, MANEs can remove those FUs where the leading FUs belonging to the same NAL unit have been lost or those dependent slice segments when the leading slice segments belonging to the same slice have been lost, because the trailing FUs or dependent slice segments are meaningless to most decoders. MANE can also remove higher temporal scalable layers if the outbound transmission (from the MANE's viewpoint) experiences congestion.

11. IANA Considerations

A new media type, as specified in Section 7.1 of this memo, has been registered with IANA.

12. Acknowledgements

Dr. Byeongdoo Choi is thanked for the video codec related technical discussion and other aspects in this memo. Xin Zhao and Dr. Xiang Li are thanked for their contributions on [VVC] specification descriptive content. Spencer Dawkins is thanked for his valuable review comments that led to great improvements of this memo. Some parts of this specification share text with the RTP payload format for HEVC [RFC7798]. We thank the authors of that specification for their excellent work.

13. References

13.1. Normative References

- [ISO23090-3] ISO/IEC 23090-3, "Information technology - Coded representation of immersive media Part 3 Versatile Video Coding", 2021, <<https://www.iso.org/standard/73022.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>>.
- [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>>.
- [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>>.
- [RFC4556] Zhu, L. and B. Tung, "Public Key Cryptography for Initial Authentication in Kerberos (PKINIT)", RFC 4556, DOI 10.17487/RFC4556, June 2006, <<https://www.rfc-editor.org/info/rfc4556>>.
- [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>>.
- [RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <<https://www.rfc-editor.org/info/rfc4648>>.

- [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>>.
- [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>>.
- [RFC5576] Lennox, J., Ott, J., and T. Schierl, "Source-Specific Media Attributes in the Session Description Protocol (SDP)", RFC 5576, DOI 10.17487/RFC5576, June 2009, <<https://www.rfc-editor.org/info/rfc5576>>.
- [RFC8082] Wenger, S., Lennox, J., Burman, B., and M. Westerlund, "Using Codec Control Messages in the RTP Audio-Visual Profile with Feedback with Layered Codecs", RFC 8082, DOI 10.17487/RFC8082, March 2017, <<https://www.rfc-editor.org/info/rfc8082>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8866] Begen, A., Kyzivat, P., Perkins, C., and M. Handley, "SDP: Session Description Protocol", RFC 8866, DOI 10.17487/RFC8866, January 2021, <<https://www.rfc-editor.org/info/rfc8866>>.
- [VSEI] "Versatile supplemental enhancement information messages for coded video bitstreams", 2020, <<https://www.itu.int/rec/T-REC-H.274>>.
- [VVC] "Versatile Video Coding, ITU-T Recommendation H.266", 2020, <<http://www.itu.int/rec/T-REC-H.266>>.

13.2. Informative References

- [CABAC] and et al, "Transform coefficient coding in HEVC, IEEE Transactions on Circuits and Systems for Video Technology", DOI 10.1109/TCSVT.2012.2223055, December 2012, <<https://doi.org/10.1109/TCSVT.2012.2223055>>.
- [HEVC] "High efficiency video coding, ITU-T Recommendation H.265", 2019, <<https://www.itu.int/rec/T-REC-H.265>>.

- [MPEG2S] ISO/IEC, "Information technology - Generic coding of moving pictures and associated audio information - Part 1: Systems, ISO International Standard 13818-1", 2013.
- [RFC2974] Handley, M., Perkins, C., and E. Whelan, "Session Announcement Protocol", RFC 2974, DOI 10.17487/RFC2974, October 2000, <<https://www.rfc-editor.org/info/rfc2974>>.
- [RFC6184] Wang, Y.-K., 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.-K., 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>>.
- [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>>.
- [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>>.
- [RFC7798] Wang, Y.-K., Sanchez, Y., Schierl, T., Wenger, S., and M. 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>>.
- [RFC7826] Schulzrinne, H., Rao, A., Lanphier, R., Westerlund, M., and M. Stiemerling, Ed., "Real-Time Streaming Protocol Version 2.0", RFC 7826, DOI 10.17487/RFC7826, December 2016, <<https://www.rfc-editor.org/info/rfc7826>>.

Appendix A. Change History

To RFC Editor: PLEASE REMOVE THIS SECTION BEFORE PUBLICATION

draft-zhao-payload-rtp-vvc-00 initial version

draft-zhao-payload-rtp-vvc-01 editorial clarifications and
corrections

draft-ietf-payload-rtp-vvc-00 initial WG draft

draft-ietf-payload-rtp-vvc-01 VVC specification update

draft-ietf-payload-rtp-vvc-02 VVC specification update

draft-ietf-payload-rtp-vvc-03 VVC coding tool introduction
update

draft-ietf-payload-rtp-vvc-04 VVC coding tool introduction
update

draft-ietf-payload-rtp-vvc-05 reference update and adding
placement for open issues

draft-ietf-payload-rtp-vvc-06 address editor's note

draft-ietf-payload-rtp-vvc-07 address editor's notes

draft-ietf-payload-rtp-vvc-08 address editor's notes

draft-ietf-payload-rtp-vvc-09 address editor's notes

draft-ietf-payload-rtp-vvc-10 address editor's notes

draft-ietf-payload-rtp-vvc-11 address editor's notes

draft-ietf-payload-rtp-vvc-12 address editor's notes

draft-ietf-payload-rtp-vvc-13 address editor's notes

draft-ietf-payload-rtp-vvc-14 address 2nd WGLC comments

Authors' Addresses

Shuai Zhao
Tencent
2747 Park Blvd
Palo Alto, 94588
United States of America
Email: shuai.zhao@ieee.org

Stephan Wenger
Tencent
2747 Park Blvd
Palo Alto, 94588
United States of America
Email: stewe@stewe.org

Yago Sanchez
Fraunhofer HHI
Einsteinufer 37
10587 Berlin
Germany
Email: yago.sanchez@hhi.fraunhofer.de

Ye-Kui Wang
Bytedance Inc.
8910 University Center Lane
San Diego, 92122
United States of America
Email: yekui.wang@bytedance.com

Miska M. Hannuksela
Nokia Technologies
Hatanpään valtatie 30
FI-33100 Tampere
Finland
Email: miska.hannuksela@nokia.com

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: 15 May 2022

M. Zanaty
E. Berger
S. Nandakumar
Cisco Systems
November 2021

Frame Marking RTP Header Extension
draft-ietf-avtext-framemarking-13

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 5 May 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components

extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

| | |
|--|----|
| 1. Introduction | 2 |
| 2. Key Words for Normative Requirements | 4 |
| 3. Frame Marking RTP Header Extension | 4 |
| 3.1. Long Extension for Scalable Streams | 5 |
| 3.2. Short Extension for Non-Scalable Streams | 7 |
| 3.3. Layer ID Mappings for Scalable Streams | 7 |
| 3.3.1. VP9 LID Mapping | 7 |
| 3.3.2. H265 LID Mapping | 8 |
| 3.3.3. H264-SVC LID Mapping | 9 |
| 3.3.4. H264 (AVC) LID Mapping | 9 |
| 3.3.5. VP8 LID Mapping | 10 |
| 3.3.6. Future Codec LID Mapping | 11 |
| 3.4. Signaling Information | 11 |
| 3.5. Usage Considerations | 11 |
| 3.5.1. Relation to Layer Refresh Request (LRR) | 12 |
| 3.5.2. Scalability Structures | 12 |
| 4. Security Considerations | 12 |
| 5. Acknowledgements | 12 |
| 6. IANA Considerations | 12 |
| 7. References | 13 |
| 7.1. Normative References | 13 |
| 7.2. Informative References | 13 |
| Authors' Addresses | 14 |

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 [RFC8871] 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].

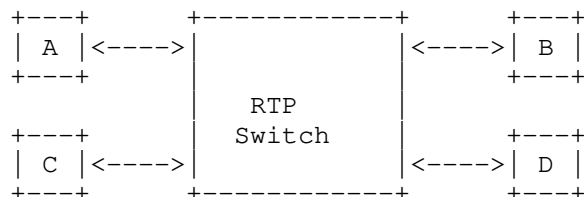


Figure 1: RTP switch

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 [RFC8871].

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

```

      0                               1                               2                               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
| ID=? | L=2 | S|E|I|D|B| TID | LID | TLOPICIDX |
+-----+-----+-----+-----+-----+-----+-----+-----+
      or
+-----+-----+-----+-----+-----+-----+-----+-----+
| ID=? | L=1 | S|E|I|D|B| TID | LID | (TLOPICIDX omitted)
+-----+-----+-----+-----+-----+-----+-----+-----+
      or
+-----+-----+-----+-----+-----+-----+-----+-----+
| ID=? | L=0 | S|E|I|D|B| TID | (LID and TLOPICIDX 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, TLOPICIDX MUST also be omitted. It is implicitly 0 in the short extension format or when omitted in the long extension format.
- * TLOPICIDX: 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 TLOPICIDX.

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.

| | | | | | | | | | | | | | | | |
|---|------|---|---|---|-----|---|---------|---|---|---------|---|---|---|---|---|
| 0 | | | | | | | | | | 1 | | | | | |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 |
| + | + | + | + | + | + | + | + | + | + | + | + | + | + | + | + |
| | ID=? | | | | L=0 | | S E I D | | | 0 0 0 0 | | | | | |

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; otherwise MUST be 0.
- * 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.
- * 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.
- * 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.
- * 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. VP9 LID Mapping

The following shows the VP9 [I-D.ietf-payload-vp9] Spatial Layer ID (SID, 3 bits) and Temporal Layer ID (TID, 3 bits) from the VP9 payload descriptor mapped to the generic LID and TID fields.

The S bit MUST match the B bit in the VP9 payload descriptor.

The E bit MUST match the E bit in the VP9 payload descriptor.

The I bit MUST match the inverse of the P bit in the VP9 payload descriptor.

The D bit MUST be 1 if the refresh_frame_flags in the VP9 payload uncompressed header are all 0, otherwise it MUST be 0.

The B bit MUST be 0 if TID is 0; otherwise, if TID is not 0, it MUST match the U bit in the VP9 payload descriptor. Note: When using temporally nested scalability structures as recommended in Section 3.5.2, the B bit and VP9 U bit will always be 1 if TID is not 0, since it is always possible to switch up to a higher temporal layer in such nested structures.

TID and TLOPICIDX MUST match the correspondingly named fields in the VP9 payload descriptor.

[illegible]

3.3.2. 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.

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+
| ID=? | L=2 | S|E|I|D|B | TID | 0 | 0 | LayerID | TLOPICIDX |
+-----+-----+-----+-----+

```

3.3.3. 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.

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+
| ID=? | L=2 | S|E|I|D|B | TID | 0 | DID | QID | TLOPICIDX |
+-----+-----+-----+-----+

```

3.3.4. 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.

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
| ID=? | L=2 | S|E|I|D|B| TID | 0|0|0|0|0|0|0|0|0| TL0PICIDX |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

3.3.5. 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. Note: When using temporally nested scalability structures as recommended in Section 3.5.2, the B bit and VP8 Y bit will always be 1 if TID is not 0, since it is always possible to switch up to a higher temporal layer in such nested structures.

TID and TLOPICIDX MUST match the correspondingly named fields in the VP8 payload descriptor.

```

      0                               1                               2                               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| ID=? | L=2 | S|E|I|D|B| TID | 0|0|0|0|0|0|0|0|0|    TLOPICIDX |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

3.3.6. 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.

3.4. Signaling Information

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

An example attribute line in SDP:

```
a=extmap:3 urn:ietf:params:rtp-hdext: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 dependencies 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:frame-marking-info
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>>.
- [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>>.
- [RFC6184] Wang, Y.-K., 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.-K., 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.-K., 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>>.

7.2. Informative References

- [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>>.
- [RFC6464] Lennox, J., Ed., Iovov, 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>>.
- [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>>.
- [RFC8871] Jones, P., Benham, D., and C. Groves, "A Solution Framework for Private Media in Privacy-Enhanced RTP Conferencing (PERC)", RFC 8871, DOI 10.17487/RFC8871, January 2021, <<https://www.rfc-editor.org/info/rfc8871>>.
- [I-D.ietf-avtext-lrr]
Lennox, J., Hong, D., Uberti, J., Holmer, S., and M. Flodman, "The Layer Refresh Request (LRR) RTCP Feedback Message", Work in Progress, Internet-Draft, draft-ietf-avtext-lrr-07, 2 July 2017, <<https://www.ietf.org/archive/id/draft-ietf-avtext-lrr-07.txt>>.
- [I-D.ietf-payload-vp9]
Uberti, J., Holmer, S., Flodman, M., Hong, D., and J. Lennox, "RTP Payload Format for VP9 Video", Work in Progress, Internet-Draft, draft-ietf-payload-vp9-16, 10 June 2021, <<https://www.ietf.org/internet-drafts/draft-ietf-payload-vp9-16.txt>>.

Authors' Addresses

Mo Zanaty
Cisco Systems
170 West Tasman Drive
San Jose, CA 95134
United States of America

Email: mzanaty@cisco.com

Espen Berger
Cisco Systems

Email: espeberg@cisco.com

Suhas Nandakumar
Cisco Systems
170 West Tasman Drive
San Jose, CA 95134
United States of America

Email: snandaku@cisco.com

avtcore
Internet-Draft
Intended status: Standards Track
Expires: January 29, 2022

S. Lugan
intoPIX
A. Descampe
UCL
C. Damman
intoPIX
T. Richter
IIS
T. Bruylants
intoPIX
July 28, 2021

RTP Payload Format for ISO/IEC 21122 (JPEG XS)
draft-ietf-payload-rtp-jpegxs-18

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 video frame rates, while offering visually lossless quality, reduced power consumption, and encoding-decoding latency confined to a fraction of a video 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 January 29, 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

| | |
|--|----|
| 1. Introduction | 2 |
| 2. Conventions, Definitions, and Abbreviations | 3 |
| 3. Media Format Description | 5 |
| 3.1. Image Data Structures | 5 |
| 3.2. Codestream | 5 |
| 3.3. Video support box and color specification box | 6 |
| 3.4. JPEG XS Frame | 6 |
| 4. RTP Payload Format | 7 |
| 4.1. RTP packetization | 7 |
| 4.2. RTP Header Usage | 10 |
| 4.3. Payload Header Usage | 11 |
| 4.4. Payload Data | 13 |
| 5. Traffic Shaping and Delivery Timing | 18 |
| 6. Congestion Control Considerations | 19 |
| 7. Payload Format Parameters | 19 |
| 7.1. Media Type Registration | 19 |
| 8. SDP Parameters | 24 |
| 8.1. Mapping of Payload Type Parameters to SDP | 24 |
| 8.2. Usage with SDP Offer/Answer Model | 25 |
| 9. IANA Considerations | 26 |
| 10. Security Considerations | 26 |
| 11. Acknowledgments | 27 |
| 12. RFC Editor Considerations | 27 |
| 13. References | 27 |
| 13.1. Normative References | 27 |
| 13.2. Informative References | 29 |
| Authors' Addresses | 30 |

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 latency that is introduced by the encoding-decoding process can be confined to a fraction of a video 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", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

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 video frame.

Color specification box (CS box)

An ISO color specification box defined in JPEG XS Part 3 [ISO21122-3] that includes color related metadata required to correctly display JPEG XS video frames, such as color 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 video frame dimension and component precision.

JPEG XS frame

In the case of progressive video, a single JPEG XS picture segment. In the case of interlaced video, the concatenation of two JPEG XS picture segments.

JPEG XS header segment

The concatenation of a video support box [ISO21122-3], a color specification box [ISO21122-3], and a JPEG XS codestream header.

JPEG XS picture segment

The concatenation of a video support box [ISO21122-3], a color specification box [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 an RTP packet payload (excluding its payload header).

SLH marker

A marker that represents a slice header, as defined in [ISO21122-1].

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. The SOC marker is considered an integral part of the JPEG XS codestream header.

Video support box (VS box)

An ISO video support box, as defined in [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

This section explains the terminology and concepts used in this memo that are specific to JPEG XS as specified in [ISO21122-1], [ISO21122-2], and [ISO21122-3].

3.1. Image Data Structures

JPEG XS is a low-latency lightweight image coding system for coding continuous-tone grayscale or continuous-tone color 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 of 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 is formed by (in the given order):

- o a JPEG XS codestream header, which starts with an SOC marker,
- o one or more slices,
- o an EOC marker to signal the end of the codestream.

The JPEG XS codestream format, including the definition of all markers, is further defined in [ISO21122-1]. It represents sample values of a single image, without any interpretation relative to a color space.

3.3. Video support box and color 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 color 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 [ISO21122-3].

The video support box provides information on the maximum bitrate, the frame rate, the interlaced mode (progressive or interlaced), the subsampling image format, the informative timecode of the current JPEG XS frame, the profile, level/sublevel used, and optionally on the buffer model and the mastering display metadata.

Note that the profile and level/sublevel, specified by respectively the Ppih and Plev fields [ISO21122-2], specify limits on the capabilities needed to decode the codestream and handle the output. Profiles represent a limit on the required algorithmic features and parameter ranges used in the codestream. The combination of level and sublevel defines a lower bound on the required throughput for a decoder in respectively the image (or decoded) domain and the codestream (or coded) domain. The actual defined profiles and levels/sublevels, along with the associated values for the Ppih and Plev fields, are defined in [ISO21122-2].

The color specification box indicates the color primaries, transfer characteristics, matrix coefficients and video full range flag needed to specify the color space of the video stream.

3.4. JPEG XS Frame

The concatenation of a video support box, a color 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 picture 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 color specification box is byte exact the same for both picture segments of the frame).

Note that the interlaced mode, as signaled by the `frat` field [ISO21122-3] in the video support box, indicates either progressive, interlaced top-field first, or interlaced bottom-field first mode. Thus, in the case of interlaced content, its value SHALL also 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 Synchronization source (SSRC) [RFC3550].

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 an 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).

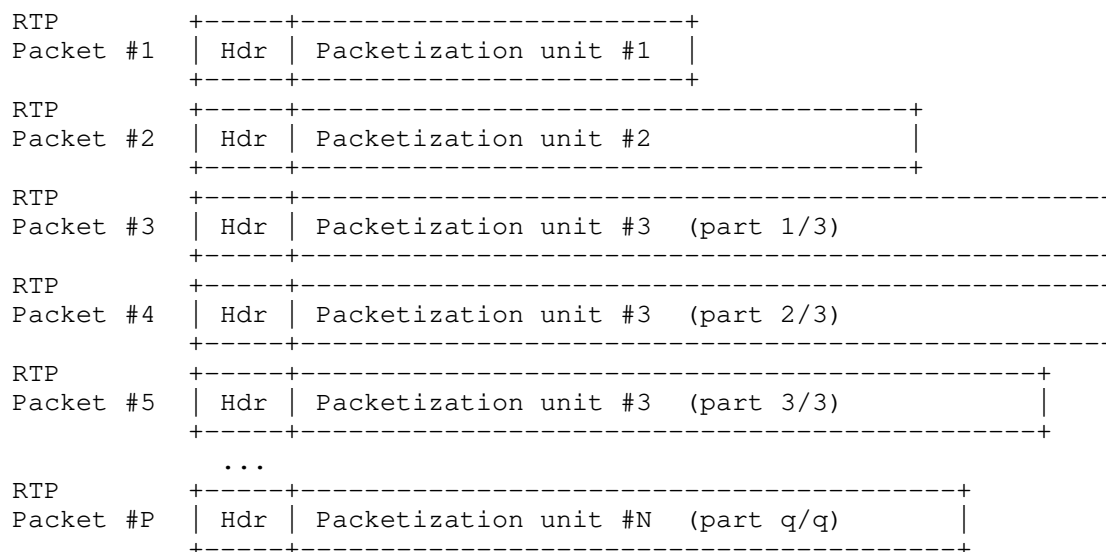


Figure 1: Example of ADU packetization

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 preceded 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.

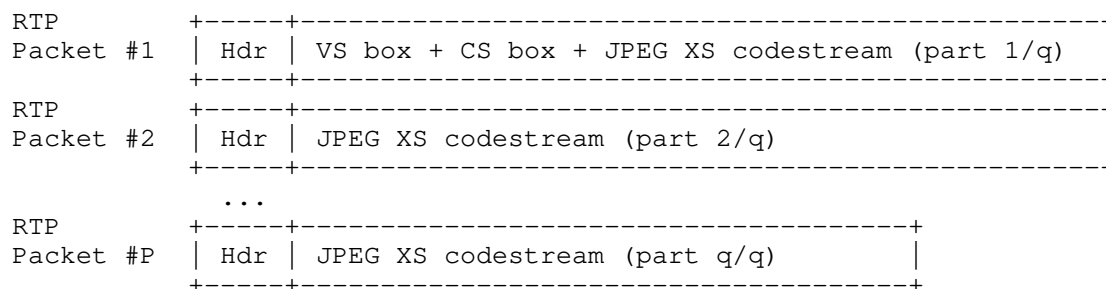


Figure 2: Example of codestream packetization mode

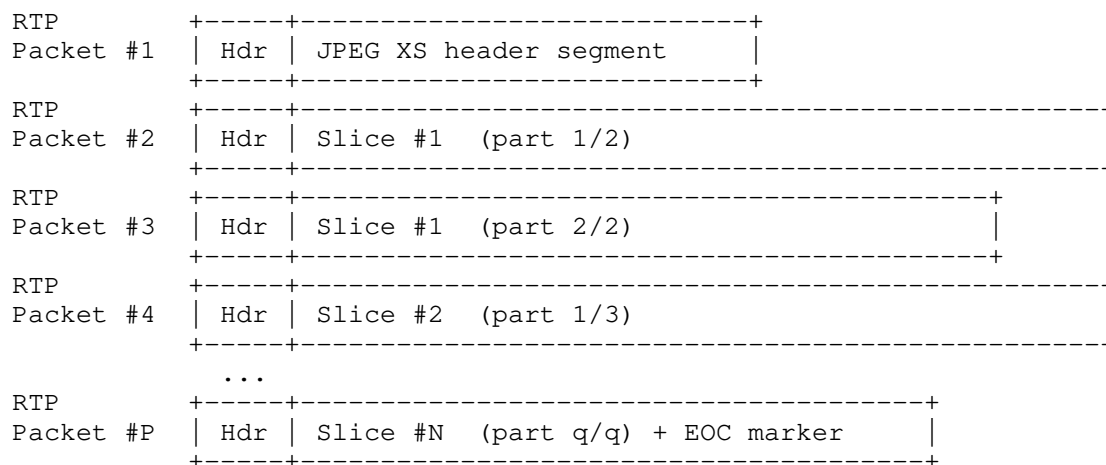


Figure 3: Example of slice packetization mode

In a constant bit-rate (CBR) scenario of JPEG XS, the codestream packetization mode guarantees that a JPEG XS RTP stream will produce a constant number of bytes per video frame, and a constant number of RTP packets per video frame. However, to provide similar guarantees with JPEG XS in a variable bit-rate (VBR) mode or when using the slice packetization mode (for either CBR or VBR), additional mechanisms are needed. 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. an RTP packet whose payload data is empty). But, management of the amount of produced packets per video frame is application dependent and not a strict requirement of this RTP payload specification.

4.2. RTP Header Usage

The format of the RTP header is specified in [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.

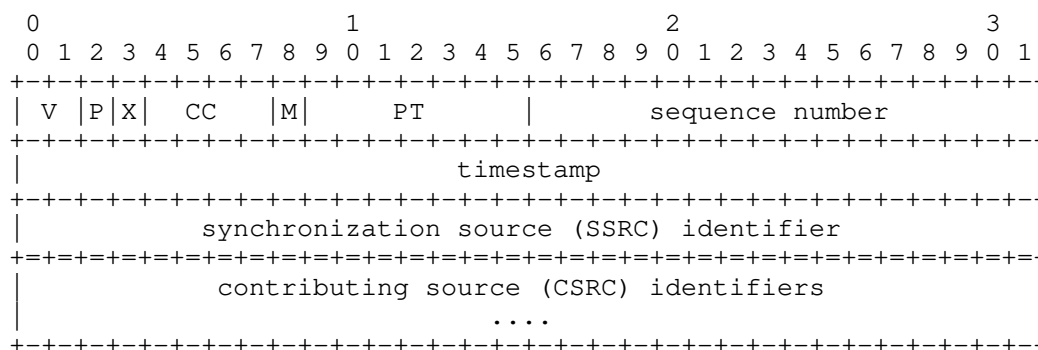


Figure 4: RTP header according to RFC 3550

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 [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 SHALL be used.

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

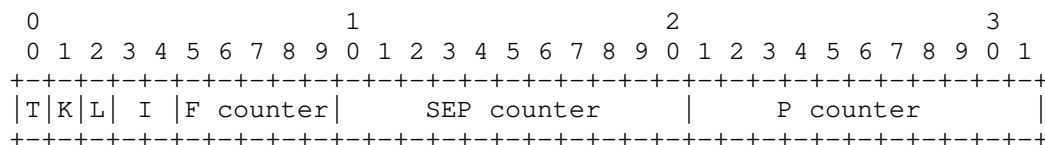


Figure 5: Payload header

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. The T bit value SHALL be identical for all packets of the RTP stream.

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 (out-of-order), the slice packetization mode SHALL be used and K SHALL be set to 1. This is required, because only the slice packetization mode supports out-of-order packet transmission. The K bit value SHALL be identical for all packets of the RTP stream.

Last (L) [1 bit]:

The L bit is set to indicate the last packet of a packetization unit. As the end of the video frame also ends the packet containing the last unit of the video frame, the L bit is set whenever the M bit is set. In the codestream packetization mode the L bit and M bit get an equivalent meaning, so they SHALL have identical values in each packet.

Interlaced information (I) [2 bit]:

These two I 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 video frame number modulo 32 to which a packet belongs. Frame numbers are incremented by 1 for each video 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 Section 4.4), 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 video 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. Within the RTP stream, all of the video support boxes and all of the color specification boxes SHALL retain their respective layouts for each JPEG XS frame. Thus, each video support box in the RTP stream SHALL define the same sub boxes. The effective values in the boxes are allowed to change under the condition that their relative byte offsets SHALL NOT change.

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 video frame in the codestream packetization mode, Figure 7 depicts this layout for an interlaced video frame in the codestream packetization mode, Figure 8 depicts this layout for a progressive video frame in the slice packetization mode and Figure 9 depicts this layout for an interlaced video 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 video frame.

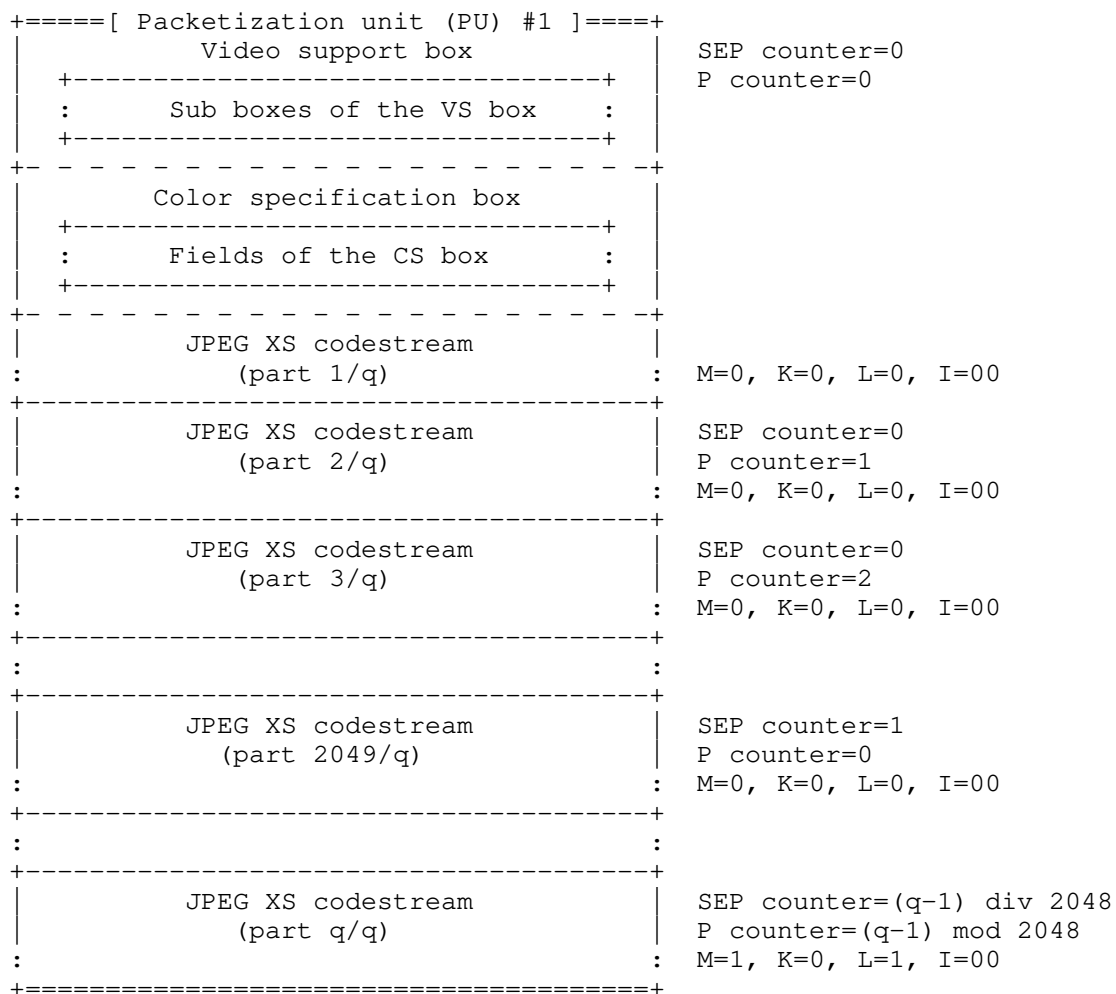


Figure 6: Example of JPEG XS Payload Data (codestream packetization mode, progressive video frame)

| | | |
|--|--------------------------------|----------------------------|
| +===== [Packetization unit (PU) #1] =====+ | | |
| | Video support box | SEP counter=0 |
| + - - - - - | | + P counter=0 |
| | Color specification box | |
| + - - - - - | | + - - - - - |
| | JPEG XS codestream (1st field) | |
| : | (part 1/q) | : M=0, K=0, L=0, I=10 |
| + - - - - - | | + - - - - - |
| | JPEG XS codestream (1st field) | SEP counter=0 |
| : | (part 2/q) | : P counter=1 |
| : | | : M=0, K=0, L=0, I=10 |
| + - - - - - | | + - - - - - |
| : | | : |
| + - - - - - | | + - - - - - |
| | JPEG XS codestream (1st field) | SEP counter=1 |
| : | (part 2049/q) | : P counter=0 |
| : | | : M=0, K=0, L=0, I=10 |
| + - - - - - | | + - - - - - |
| : | | : |
| + - - - - - | | + - - - - - |
| | JPEG XS codestream (1st field) | SEP counter=(q-1) div 2048 |
| : | (part q/q) | : P counter=(q-1) mod 2048 |
| : | | : M=1, K=0, L=1, I=10 |
| +===== | [PU #2]===== | +===== |
| | Video support box | SEP counter=0 |
| + - - - - - | | + P counter=0 |
| | Color specification box | |
| + - - - - - | | + - - - - - |
| | JPEG XS codestream (2nd field) | |
| : | (part 1/q) | : M=0, K=0, L=0, I=11 |
| + - - - - - | | + - - - - - |
| | JPEG XS codestream (2nd field) | SEP counter=0 |
| : | (part 2/q) | : P counter=1 |
| : | | : M=0, K=0, L=0, I=11 |
| + - - - - - | | + - - - - - |
| : | | : |
| + - - - - - | | + - - - - - |
| | JPEG XS codestream (2nd field) | SEP counter=(q-1) div 2048 |
| : | (part q/q) | : P counter=(q-1) mod 2048 |
| : | | : M=1, K=0, L=1, I=11 |
| +===== | | +===== |

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

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

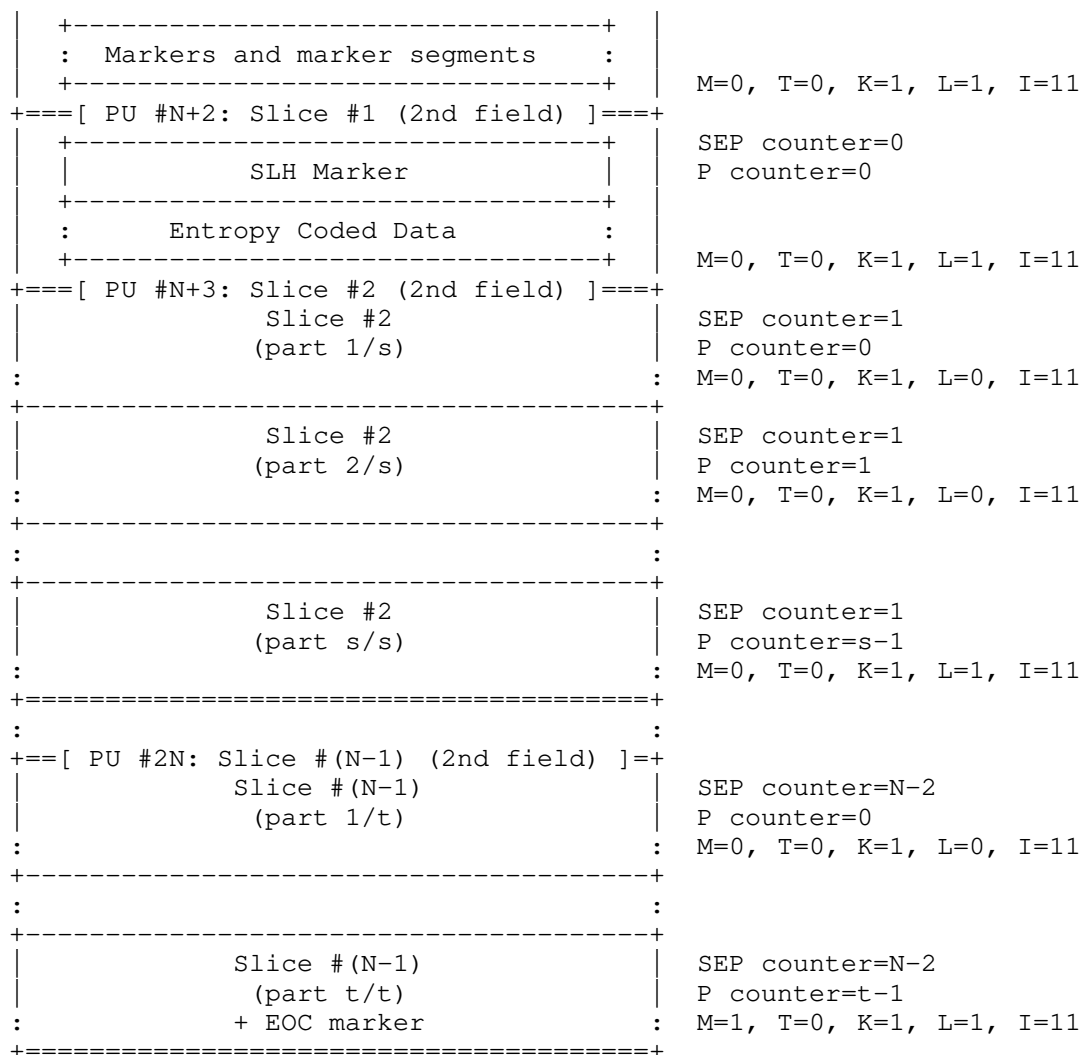


Figure 9: Example of JPEG XS Payload Data (slice packetization mode, interlaced video frame)

5. Traffic Shaping and Delivery Timing

In order to facilitate proper synchronization between senders and receivers it is RECOMMENDED to implement traffic shaping and delivery timing in accordance with the Network Compatibility Model compliance definitions specified in [SMPTE-ST2110-21]. In such case, the session description SHALL signal the compliance with the media type parameter TP. The actual applied traffic shaping and timing delivery

mechanism is outside the scope of this memo and does not influence the payload packetization.

6. Congestion Control Considerations

Congestion control for RTP SHALL be used in accordance with [RFC3550], and with any applicable RTP profile: e.g. RTP/AVP [RFC3551] or RTP/AVPF [RFC4585].

While JPEG XS is mainly designed to be used in controlled network environments, it can also be employed in best-effort network environments, like the Internet. However, in this case the users of this payload format SHALL monitor packet loss to ensure that the packet loss rate is within acceptable parameters. This can be achieved for example by means of the RTP Control Protocol (RTCP) Feedback for Congestion Control [RFC8888].

In addition, 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.

[RFC8085] provides additional information on the best practices for applying congestion control to UDP streams.

7. Payload Format Parameters

This section specifies the required and optional parameters of the payload format and/or the RTP stream. All parameters are declarative, meaning that the information signaled by the parameters is also present in the payload data, namely in the payload header (see Section 4.3) or in the JPEG XS header segment [ISO21122-1] [ISO21122-3]. When provided, their respective values SHALL be consistent with the payload.

7.1. Media Type Registration

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

The receiver SHALL ignore any unrecognized parameter.

Type name: video

Subtype name: jxsv

Clock rate: 90000

Required parameters:

- rate: The RTP timestamp clock rate. Applications using this payload format SHALL use a value of 90000.
- packetmode: This parameter specifies the configured packetization mode as defined by the packetization mode (K) bit in the payload header of Section 4.3. This value SHALL be equal to the K bit value configured in the RTP stream (i.e. 0 for codestream or 1 for slice).

Optional parameters:

- transmode: This parameter specifies the configured transmission mode as defined by the Transmission mode (T) bit in the payload header of Section 4.3. If specified, 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-only). If not specified, a value 1 (sequential-only) SHALL be assumed and the T bit SHALL be set to 1.
- profile: The JPEG XS profile [ISO21122-2] in use. Any white space in the profile name SHALL be omitted. Examples of valid profile names are 'Main444.12' or 'High444.12'.
- level: The JPEG XS level [ISO21122-2] in use. Any white space in the level name SHALL be omitted. Examples of valid levels are '2k-1' or '4k-2'.
- sublevel: The JPEG XS sublevel [ISO21122-2] in use. Any white space in the sublevel name SHALL be omitted. Examples of valid sublevels 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 inclusive.
- height: Determines the number of lines per video frame. This is an integer between 1 and 32767 inclusive.
- exactframerate: Signals the video 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.

interlace: 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 color 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 color 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 [ISO21122-3], 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 [ISO21122-3], 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 colorimetry, NARROW SHALL be the assumed value. When paired with the UNSPECIFIED colorimetry, FULL SHALL be the default assumed value.

Encoding considerations:

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

Security considerations:

Please see the Security Considerations (Section 10) of RFC XXXX.

Interoperability considerations:

None.

Published specification:

See RFC XXXX and its References section.

Applications that use this media type:

Any application that transmits video over RTP (like SMPTE ST 2110).

Fragment identifier considerations:
N/A.

Additional information:
None.

Person & email address to contact for further information:
S. Lugan <rtp@intopix.com> and Th. Richter <jpeg-xs-techsupport@iis.fraunhofer.de>.

Intended usage:
COMMON

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

Author:
See the Authors' Addresses section of RFC XXXX.

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

8. SDP Parameters

A mapping of the parameters into the Session Description Protocol (SDP) [RFC8866] is provided for applications that use SDP.

8.1. Mapping of Payload Type Parameters to SDP

The media type video/jxsv string is mapped to fields in the Session Description Protocol (SDP) [RFC8866] as follows:

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 SHALL be 90000).

The required parameter "packetmode", and any of the additional optional parameters, as described in Section 7.1, go in the SDP media format description, being the "a=fmtp" attribute (Format Parameters), by copying them directly from the MIME media type string as a semicolon-separated list of parameter=value pairs.

All parameters of the media format SHALL correspond to the parameters of the payload. In case of discrepancies between payload parameter

values and SDP fields, the values from the payload data SHALL prevail.

The receiver SHALL ignore any parameter that is not defined in Section 7.1.

An example SDP mapping for JPEG XS video is as follows:

```
m=video 30000 RTP/AVP 112
a=rtpmap:112 jxsv/90000
a=fmtp:112 packetmode=0;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 to be 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. This example includes the TP parameter (as specified in Section 5).

8.2. Usage with SDP Offer/Answer Model

When JPEG XS is offered over RTP using SDP in an offer/answer model [RFC3264] for negotiation for unicast usage, the following limitations and rules apply:

The "a=fmtp" attribute SHALL be present specifying the required parameter "packetmode", and MAY specify any of the optional parameters, as described in Section 7.1.

All parameters in the "a=fmtp" attribute indicate sending capabilities (i.e. properties of the payload).

An answerer of the SDP is required to support all parameters and values of the parameters provided by the offerer; otherwise, the answerer SHALL reject the session. It falls on the offerer to use values that are expected to be supported by the answerer. If the answerer accepts the session, it SHALL reply with the exact same parameters values in the "a=fmtp" attribute as it was offered.

The same RTP payload type number used in the offer SHOULD be used in the answer, as specified in [RFC3264].

9. IANA Considerations

The IANA is requested to register the media type registration "video/jxsv" as specified in Section 7.1. The media type is also requested to be added to the IANA registry for "RTP Payload Format MIME types" <<https://www.iana.org/assignments/rtp-parameters>>.

10. Security Considerations

RTP packets using the payload format defined in this memo are subject to the security considerations discussed in [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.

Implementations of this RTP payload format need to take appropriate security considerations into account. It is important for the decoder to be robust against malicious or malformed payloads and ensure that they do not cause the decoder to overrun its allocated memory or otherwise misbehave. An overrun in allocated memory could lead to arbitrary code execution by an attacker. The same applies to the encoder, even though problems in encoders are typically rarer.

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.

This payload format and the JPEG XS encoding do not contain code that is executable.

It is important to note that HD or UHD TV 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.

11. Acknowledgments

The authors would like to thank the following people for their valuable contributions to this memo: Arnaud Germain, Alexandre Willeme, Gael Rouvroy, Siegfried Foessel, and Jean-Baptiste Lorent.

12. 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.

13. References

13.1. 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.

- [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.
- [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.
- [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>>.
- [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>>.
- [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>>.
- [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>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8866] Begen, A., Kyzivat, P., Perkins, C., and M. Handley, "SDP: Session Description Protocol", RFC 8866, DOI 10.17487/RFC8866, January 2021, <<https://www.rfc-editor.org/info/rfc8866>>.

13.2. Informative References

- [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>>.
- [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>>.
- [RFC8888] Sarker, Z., Perkins, C., Singh, V., and M. Ramalho, "RTP Control Protocol (RTCP) Feedback for Congestion Control", RFC 8888, DOI 10.17487/RFC8888, January 2021, <<https://www.rfc-editor.org/info/rfc8888>>.

[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>>.

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: 12 December 2021

J. Uberti
S. Holmer
M. Flodman
D. Hong
Google
J. Lennox
8x8 / Jitsi
10 June 2021

RTP Payload Format for VP9 Video
draft-ietf-payload-vp9-16

Abstract

This specification 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 12 December 2021.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components

extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

| | |
|---|----|
| 1. Introduction | 2 |
| 2. Conventions, Definitions and Acronyms | 3 |
| 3. Media Format Description | 3 |
| 4. Payload Format | 5 |
| 4.1. RTP Header Usage | 5 |
| 4.2. VP9 Payload Descriptor | 6 |
| 4.2.1. Scalability Structure (SS): | 11 |
| 4.3. Frame Fragmentation | 13 |
| 4.4. Scalable encoding considerations | 13 |
| 4.5. Examples of VP9 RTP Stream | 13 |
| 4.5.1. Reference picture use for scalable structure | 14 |
| 5. Feedback Messages and Header Extensions | 14 |
| 5.1. Reference Picture Selection Indication (RPSI) | 15 |
| 5.2. Full Intra Request (FIR) | 15 |
| 5.3. Layer Refresh Request (LRR) | 15 |
| 6. Payload Format Parameters | 16 |
| 6.1. SDP Parameters | 18 |
| 6.1.1. Mapping of Media Subtype Parameters to SDP | 18 |
| 6.1.2. Offer/Answer Considerations | 19 |
| 7. Media Type Definition | 19 |
| 8. Security Considerations | 21 |
| 9. Congestion Control | 21 |
| 10. IANA Considerations | 22 |
| 11. Acknowledgments | 22 |
| 12. References | 22 |
| 12.1. Normative References | 22 |
| 12.2. Informative References | 23 |
| Authors' Addresses | 24 |

1. Introduction

This specification describes an RTP [RFC3550] 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", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

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 the spatial or temporal dimension, 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 scalability structures which are changing dynamically.

In non-flexible mode, frames are encoded using a fixed, recurring pattern of dependencies; the set of pictures that recur in this pattern is known as a Picture Group (PG). In this mode, the inter-picture dependencies (the reference indices) of the Picture Group MUST be pre-specified as part of the scalability structure (SS) data. 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.

(Note: A "Picture Group", as used in this document, is not the same thing as the term "Group of Pictures" as it is traditionally used in video coding, i.e. to mean an independently-decodable run of pictures beginning with a keyframe.)

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.

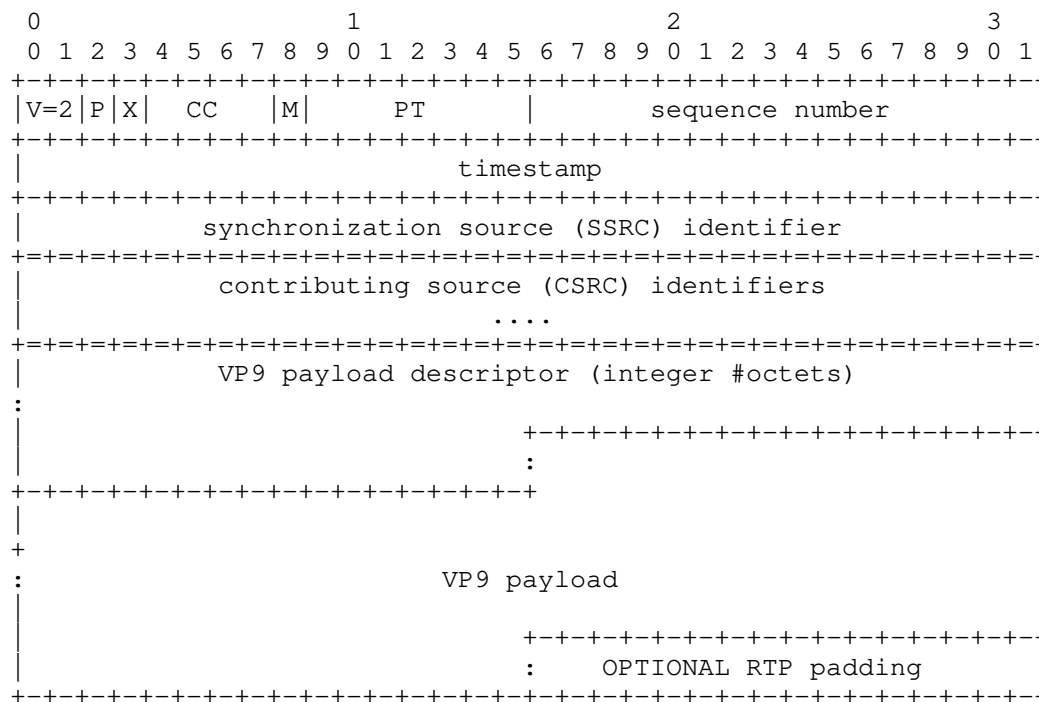


Figure 1

The VP9 payload descriptor will be described in Section 4.2; the VP9 payload is described in [VP9-BITSTREAM]. OPTIONAL RTP padding MUST NOT be included unless the P bit is set.

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 6.1 for the mechanism to be used with the Session Description Protocol (SDP) [RFC8866].

Timestamp: The RTP timestamp [RFC3550] 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 alternatively 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.

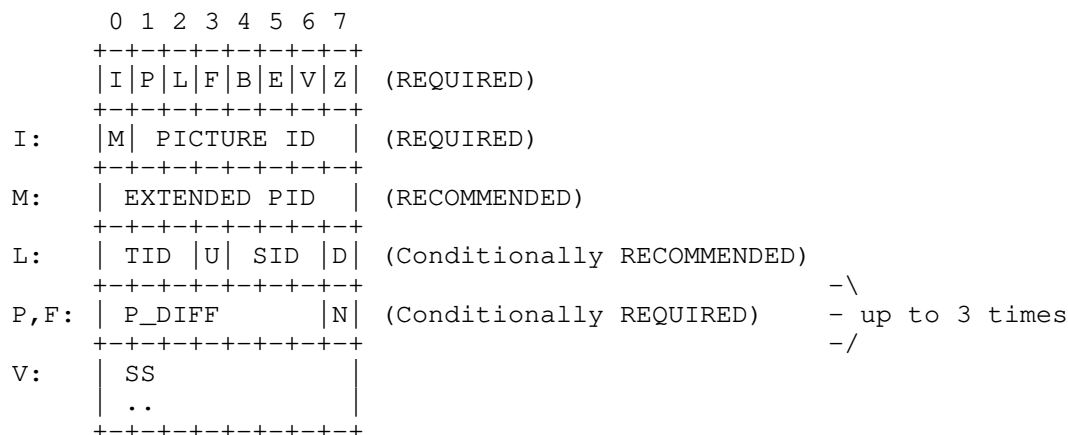


Figure 2

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.

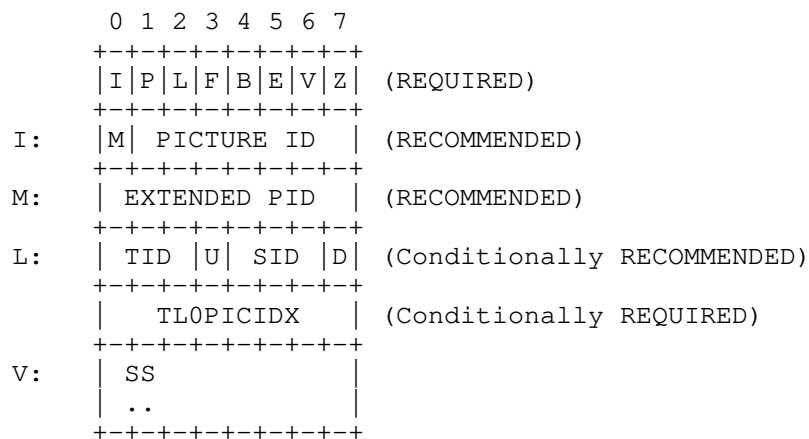


Figure 3

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 V bit was set in the stream's most recent start of a keyframe (i.e. the SS field was present) and the F bit is set to 0 (i.e. non-flexible scalability mode is in use), then this bit MUST be set on 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. (Flexible mode's Reference indices are defined as offsets from the Picture ID field, so they would have no meaning if I were not set.) 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 L 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 B 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 and greater 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 M is 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, where the sender increments the value by 1 for each picture it sends. (Note however that because a middlebox can discard pictures where permitted by the scalability structure, Picture IDs as received by a receiver might not be contiguous.) This 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 (0x7f or 0x7fff depending on the index size chosen). The receiver MUST NOT assume that the number of bits in PID stay the same through the session. If this field transitions from 7-bits to 15-bits, the value is zero-extended (i.e. the value after 0x6e is 0x006f); if the field transitions from 15 bits to 7 bits, it is truncated (i.e. the value after 0x1bbe is 0xbf).

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 (TLOPICIDX), as depicted in Figure 3. The TLOPICIDX 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 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 SID > 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 and only

if the current spatial layer SID frame depends on spatial layer SID-1 frame of the same picture, otherwise MUST be set to zero. 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 by 1 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 MUST 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. A P_DIFF value of 0 is invalid.

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.

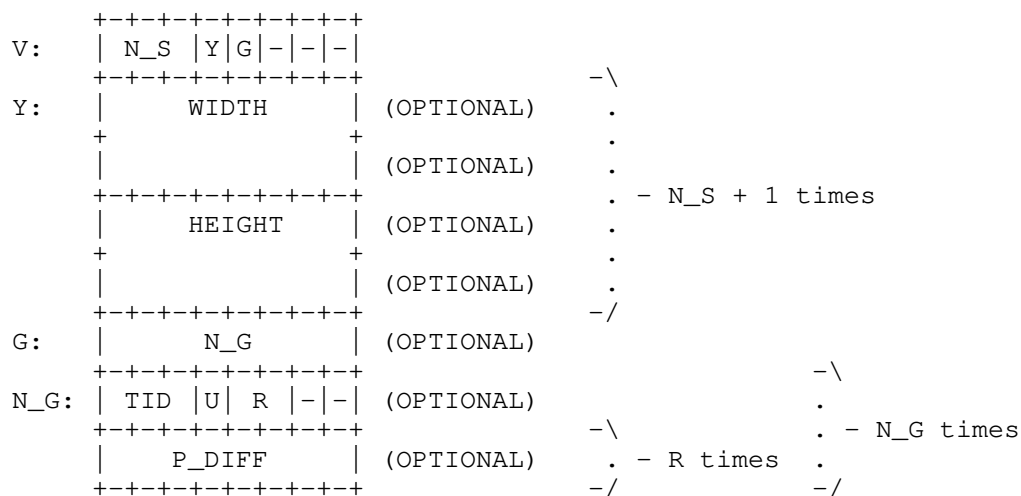


Figure 4

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 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 (for non-flexible mode) or no fixed inter-picture dependency information is present (for flexible mode) 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 described 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 L 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 frame that might be removed (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 | – |

Table 1: 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 Picture ID 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 (7 or 15 bit) Picture ID 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 (LRR) [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.

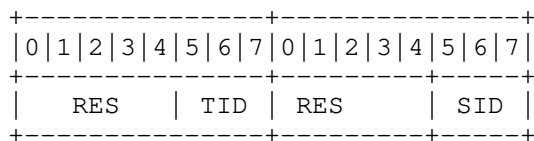


Figure 5

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 ignored 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 switching up points are being encoded in response to a LRR, those packets should contain layer indices and the reference field(s) so that the decoder or a selective forwarding middleboxes [RFC7667] 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.

6. Payload Format Parameters

This payload format has three optional parameters, "max-fr", "max-fs", and "profile-id".

The max-fr and max-fs 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. A media sender SHOULD NOT send media with a frame rate or frame size exceeding the max-fr and max-fs values signaled. (There may be scenarios, such as pre-encoded media or selective forwarding middleboxes [RFC7667], where a media sender does not have media available that fits within a receivers max-fs and max-fr value; in such scenarios, a sender MAY exceed the signaled values.)

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 $\text{int}(\sqrt{\text{max-fs} * 8})$ – 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 2 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. (The profile-id parameter was added relatively late in the development of this specification, so some existing implementations may not send it.)

Informative note: See Table 3 for capabilities of coding profiles defined in section 7.2 of [VP9-BITSTREAM].

A receiver MUST ignore any parameter unspecified in this specification.

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

Table 2: 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 4:2:2, 4:4:0 or 4:4:4 |
| 2 | 10 or 12 | No | YUV 4:2:0 |
| 3 | 10 or 12 | Yes | YUV 4:2:2, 4:4:0 or 4:4:4 |

Table 3: Table of profile capabilities.

6.1. SDP Parameters

6.1.1. Mapping of Media Subtype Parameters to SDP

The media type video/VP9 string is mapped to fields in the Session Description Protocol (SDP) [RFC8866] 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-fr" and "max-fs" MUST be included in the "a=fmtp" line of SDP if the receiver wishes to declare its 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 inferred for profile-id.

6.1.1.1. 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
```

6.1.2. 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 this configuration parameter or remove the media format (payload type) completely if it is not supported.
- * The max-fr and max-fs parameters are used declaratively to describe receiver capabilities, even in the Offer/Answer model. The values in an answer are used to describe the answerer's capabilities, and thus their values are set independently of the values in the offer.
- * 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 and in a subsequent offer, as specified in [RFC3264]. An answer or subsequent offer MUST NOT contain the payload type number used in the offer unless the profile-id value is exactly the same as in the original offer. However, max-fr and max-fs parameters MAY be changed in subsequent offers and answers, with the same payload type number, if an endpoint wishes to change its declared receiver capabilities.

7. Media Type Definition

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

Type name:
video

Subtype name:
VP9

Required parameters:
N/A.

Optional parameters:
There are three optional parameters, "max-fr", "max-fs", and "profile-id". See Section 6 for their definition.

Encoding considerations:

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

Security considerations:

See Section 8 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.

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]. 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.

Implementations of this RTP payload format need to take appropriate security considerations into account. It is extremely important for the decoder to be robust against malicious or malformed payloads and ensure that they do not cause the decoder to overrun its allocated memory or otherwise mis-behave. An overrun in allocated memory could lead to arbitrary code execution by an attacker. The same applies to the encoder, even though problems in encoders are typically rarer.

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.

9. 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).

10. IANA Considerations

The IANA is requested to register the media type registration "video/vp9" as specified in Section 7. 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>>.

11. 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.

12. References

12.1. Normative 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", Work in Progress, Internet-Draft, draft-ietf-avtext-lrr-07, 2 July 2017, <<https://www.ietf.org/archive/id/draft-ietf-avtext-lrr-07.txt>>.
- [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>>.
- [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>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8866] Begen, A., Kyzivat, P., Perkins, C., and M. Handley, "SDP: Session Description Protocol", RFC 8866, DOI 10.17487/RFC8866, January 2021, <<https://www.rfc-editor.org/info/rfc8866>>.
- [VP9-BITSTREAM]
Grange, A., de Rivaz, P., and J. Hunt, "VP9 Bitstream & Decoding Process Specification", Version 0.6, 31 March 2016,
<<https://storage.googleapis.com/downloads.webmproject.org/docs/vp9/vp9-bitstream-specification-v0.6-20160331-draft.pdf>>.

12.2. 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>>.
- [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

Justin Uberti
Google, Inc.
747 6th Street South
Kirkland, WA 98033
United States of America

Email: justin@uberti.name

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

Email: holmer@google.com

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

Email: mflodman@google.com

Danny Hong
Google, Inc.
1585 Charleston Road
Mountain View, CA 94043
United States of America

Email: dannyhong@google.com

Jonathan Lennox
8x8, Inc. / Jitsi
Jersey City, NJ 07302
United States of America

Email: jonathan.lennox@8x8.com