Stream: Internet Engineering Task Force (IETF)
RFC: 9584
Category: Standards Track
Published: June 2024
ISSN: 2070-1721
Authors: S. Zhao *Intel* S. Wenger *Tencent* Y. Lim *Samsung Electronics*

RFC 9584
RTP Payload Format for Essential Video Coding (EVC)

Abstract

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

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9584.

Copyright Notice

Table of Contents

1. Introduction

The Essential Video Coding [EVC] standard, which is formally designated as ISO/IEC International Standard 23094-1 [EVC], was published in 2020. One of MPEG's goals is to keep EVC's Baseline profile essentially royalty-free by using technologies published more than 20 years ago or otherwise known to be available for use without a requirement for paying royalties, whereas more advanced profiles follow a reasonable and non-discriminatory licensing terms policy. Both the Baseline profile and higher profiles of EVC [EVC] are reported to provide coding efficiency gains over High Efficiency Video Coding [HEVC] and Advanced Video Coding [AVC] under certain configurations.

This document 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) [RFC9328]. 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 the RTP Payload Format for H.264 video as described in [RFC6184] is widely deployed and generally known in the relevant implementer communities. Certain mechanisms described in [RFC6190] were incorporated, as EVC supports temporal scalability. EVC currently does not offer higher forms of scalability.

1.1. Overview of the EVC Codec

The codings described in [EVC], [AVC], [HEVC], and [VVC] share a similar hybrid video codec design. In this document, 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 standard pertaining to EVC [EVC] to arrive at interoperable, well-performing implementations. The EVC standard has a Baseline profile and a Main profile, the latter being a superset of the Baseline profile but including more advanced features. EVC also includes still image variants of both Baseline and Main profiles, in each of which the bitstream is restricted to a single IDR picture. EVC facilitates certain walled garden implementations under commercial constraints imposed by intellectual property rights by including syntax elements that 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_info of [VVC].

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

1.1.1. Coding-Tool Features (Informative)

Coding blocks and transform structure

EVC uses a traditional block-based coding structure, which divides the encoded image into blocks of up to 64x64 luma samples for the Baseline profile and 128x128 luma samples for the Main profile that can be recursively divided into smaller blocks. The Baseline profiles utilize HEVC-like quad-tree-blocks partitioning that allows a block to be divided horizontally and vertically into four smaller square blocks. The Main profile adds two advanced coding structure tools: 1) Binary Ternary Tree (BTT) partitioning that allows non-square coding units and 2) Split Unit Coding Order segmentation that changes the processing order of the blocks from traditional left-to-right and top-to-bottom scanning order processing to an alternative right-to-left and bottom-to-top scanning order. In the Main profile, the picture can be divided into slices and tiles, which can be independently encoded and/or decoded in parallel.

EVC also uses a traditional video codecs prediction model assuming two general types of predictions: Intra (spatial) and Inter (temporal) predictions. A residue block is calculated by subtracting predicted data from the original (encoded) one. The Baseline profile allows only discrete cosine transform (DCT-2) and scalar quantization to transform and quantize residue data, wherein the Main profile additionally has options to use discrete sine transform (DST-7) and another type of discrete cosine transform (DCT-8). In addition, for the Main profile, Improved Quantization and Transform (IQT) uses a different mapping or 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. The Baseline profile uses a straightforward RLE-based approach to encode the quantized coefficients.

Entropy coding

EVC uses a similar binary arithmetic coding mechanism as HEVC CABAC (context adaptive binary arithmetic coding) and VVC. 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 [VIDEO-CODING]. In the Main profile, an Advanced Deblocking Filter (ADDB) can be used as an alternative, which can further reduce undesirable compression 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 Hadamard Transform Domain Filter (HTDF) is applied to the luma samples before deblocking, and a lookup table is used to determine four adjacent samples for filtering. An adaptive Loop Filter (ALF) allows signals of up to 25 different filters to be sent for the luma components; the best filter can be selected through the classification process for each 4x4 block. Similarly to VVC, the filter parameters of ALF are signaled in the Adaptation Parameter Set (APS).

Inter prediction

The basis of EVC's Inter prediction is motion compensation using interpolation filters with a quarter sample resolution. In the Baseline profile, a motion vector is transmitted using one of three spatially neighboring motion vectors and a temporally collocated motion vector as a predictor. A motion vector difference may be signaled relative to the selected predictor, but there is a case where no motion vector difference is signaled, and there is no remaining data in the block. This mode is called a "skip" mode. The Main profile includes six additional tools to provide improved Inter prediction. With Advanced Motion Vectors Prediction (ADMVP), 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 an 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 without additional signaling.

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, when all coding units are square, 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 28 additional direction modes are available in the Enhanced Intra Prediction Directions (EIPDs). In the Main profile, an encoder can also use Intra Block Copy (IBC), where previously decoded sample blocks of the same picture are 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.

Reference frames management

In EVC, decoded pictures can be stored in a decoded picture buffer (DPB) for predicting pictures that follow them in the decoding order. In the Baseline profile, the management of the DPB (i.e., the process of adding and deleting reference pictures) is controlled by a straightforward AVC-like sliding window approach with very few parameters from the sequence parameter set (SPS). For the Main profile, DPB management can be handled much more flexibly using explicitly signaled Reference Picture Lists (RPLs) in the SPS or slice level.

1.1.2. Systems and Transport Interfaces

EVC inherits the basic systems and transport interface designs from AVC and HEVC. These include the NAL-unit-based syntax, hierarchical syntax and data unit structure, and Supplemental Enhancement Information (SEI) message mechanism. The hierarchical syntax and data unit structure consists of a sequence-level parameter set (i.e., SPS), two picture-level parameter sets (i.e., 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 NAL design of EVC as well as this document 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 picture and followed by zero or more pictures that may depend on each other and the random access point picture. In MPEG-2, the equivalent of a CVS is a Group of Pictures (GOP), which generally starts with an I frame and is 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 Multipoint Control Units (MCUs) involved), a CVS can be as long in duration as the whole session.

Picture and adaptation parameter set

The Picture Parameter Set (PPS) and the Adaptation Parameter Set (APS) 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 of 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 known from AVC, HEVC, and video codecs as old as MPEG-1 Video. The profile defines a set of tools (not to be confused with the "toolset" discussed below) that a decoder compliant with this profile has to support. In EVC, profiles are defined in Annex A of [EVC]. 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 the Main profile, reducing implementation complexity. Levels relate to bitstream complexity in dimensions such as maximum sample decoding rate, maximum picture size, and similar parameters directly related to computational complexity and/or memory demands.

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 elements. These syntax elements, toolset_idc_h and toolset_idc_l (located in the SPS), are bitmasks that allow 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 was using a coding tool that is indicated as not being used in the toolset syntax element would 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 intellectual property-related risks.

Bitstream and elementary stream

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

Random access support

EVC supports random access mechanisms based on IDR and clean random access (CRA) access units.

Temporal scalability support

EVC supports 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 document), in the nuh_temporal_id field.

Reference picture management

EVC's reference picture management is POC-based, similar to HEVC. In the Main profile, substantially all reference picture list manipulations available in HEVC are specified, including explicit transmissions or updates of reference picture lists. Although for reference pictures management purposes, EVC uses a modern VVC-like RPL approach, which is conceptually simpler than the HEVC one. In the Baseline profile, reference picture management is more restricted, allowing for a comparatively simple group of picture structures only.

SEI Message

EVC inherits many of HEVC's SEI messages, occasionally with syntax and/or semantics changes, making them applicable to EVC. In addition, some of the codec-agnostic SEI messages of the VSEI specification [VSEI] are also mapped.

1.1.3. Parallel Processing Support (Informative)

EVC's Baseline profile includes no tools specifically addressing parallel-processing support. The Main profile includes independently decodable slices for parallel processing. The slices are defined as any rectangular region within a picture. They can be encoded to have coding dependencies with other slices from the previous picture but not with other slices in the same picture. No specific support for parallel processing is specified in this RTP payload format.

1.1.4. NAL Unit Header

EVC maintains the NAL unit concept of [VVC] 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.

```
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|F|   Type    | TID | Reserve |E|
+-------------+-----------------+
```

Figure 1: The Structure of the EVC NAL Unit Header

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 this document, the value 1 may be used to indicate a syntax violation, e.g., for a NAL unit resulting from aggregating a number of fragmented units of a NAL unit but missing the last fragment, as described in Section 4.3.3.

Type: 6 bits

nal_unit_type_plus1: This field allows the NAL Unit Type to be computed. The NAL Unit Type (NalUnitType) is equal to the value found in this field, minus 1; in other words:

NalUnitType = nal_unit_type_plus1 - 1.

The NAL unit type is detailed in Table 4 of [EVC]. If the value of NalUnitType is less than or 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 of [EVC]. Note that nal_unit_type_plus1 **MUST NOT** be zero.

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 an 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 standard. Values of nuh_reserved_zero_5bits greater than 0 are reserved for future use by ISO/IEC. Decoders conforming to a profile specified in Annex A of [EVC] 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 to the version of the EVC standard. The value of nuh_extension_flag equal to 1 is reserved for future use by ISO/IEC. Decoders conforming to a profile specified in Annex A of [EVC] 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
- transmission of EVC NAL units of the same bitstream within a single RTP stream
- usage of 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 EVC. Section 3.1.1 lists relevant definitions from [EVC] for convenience. Section 3.1.2 provides definitions specific to this document.

3.1.1. Definitions from the EVC Standard

Access Unit (AU):
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.

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 coded pictures and associated data forming one or more 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 NxN 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 color 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 the [EVC] standard.

Dynamic Range Adjustment (DRA):
A mapping process that is applied to the decoded picture prior to cropping and output as part of the decoding process; it 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.

IDR Access Unit:
An access unit in which the coded picture is an IDR picture.

IDR Picture:
The 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 DPB) 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 DPB.

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.

Slice:
An integer number of tiles of a picture in the tile scan of the picture, exclusively contained in a single NAL unit.

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 width specified by syntax elements in the PPS.

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 Document

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 7.4.2.3 of [EVC] and follows the order of NAL units in the bitstream.

NALU-time:
The value that the RTP timestamp would have if the NAL unit would be transported in its own RTP packet.

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 document, one RTP stream is utilized to transport an EVC bitstream, which may contain 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 (AP), 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
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	Red, Green, and Blue
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] (included 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 APs and Fragmentation Units (FUs) are specified in Sections 4.3.2 and 4.3.3, respectively.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |V=2|P|X|  CC   |M|     PT      |       sequence number         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                           timestamp                           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |           synchronization source (SSRC) identifier            |
   +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
   |            contributing source (CSRC) identifiers             |
   |                             ....                              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: RTP Header According to RFC 3550

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 and 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.

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 is included. For SEI messages, this information is specified in Annex D of [EVC]. 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. According to this document, 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 with larger TID values are not used to decode the ones with smaller TID values. 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 as defined in Table 4 of [EVC] (denoted as PayloadHdr), followed by 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.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |           PayloadHdr          |      DONL (conditional)       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   |                  NAL unit payload data                        |
   |                                                               |
   |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                               :...OPTIONAL RTP padding        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: The Structure of a Single NAL Unit Packet

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 (defined in Section 7.2) is greater than 0, 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 (where sprop-max-don-diff is equal to 0), 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 of one access unit, and it **MUST NOT** contain NAL units from more than one AU. Each NAL unit to be carried in an AP is encapsulated in an aggregation unit. NAL units aggregated in one AP are included in NAL-unit-decoding order.

An AP consists of a payload header, as defined in Table 4 of [EVC] (denoted here as PayloadHdr with Type=56), followed by two or more aggregation units, as shown in Figure 4.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    PayloadHdr (Type=56)       |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
|                                                               |
|             two or more aggregation units                     |
|                                                               |
|                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               :...OPTIONAL RTP padding        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: The Structure of an Aggregation Packet

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 56.

The value of TID **MUST** be the smallest value of TID of all the aggregated NAL units. The value of Reserve and E **MUST** be equal to 0 for this 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 from 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 cannot contain another AP.

> Informative note: If a receiver encounters nested APs, which is against the aforementioned requirement, it has several options, listed in order of ease of implementation: 1) ignore the nested AP; 2) ignore the nested AP and report a "packet loss" to the decoder, if such functionality exists in the API; and 3) implement support for nested APs and extract the NAL units from these nested APs.

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.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |               :       DONL (conditional)      |   NALU size   |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   NALU size   |                                               |
   +-+-+-+-+-+-+-+-+         NAL unit                              |
   |                                                               |
   |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                               :
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: The Structure of the First Aggregation Unit in an AP

> Informative note: The first octet of Figure 5 (indicated by the first colon) belongs to a previous aggregation unit. It is depicted to emphasize that aggregation units are octet aligned only. Similarly, the NAL unit carried in the aggregation unit can terminate at the octet boundary.

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. The variable DON for the aggregated NAL unit is derived as equal to the value of the DONL field, and the variable Decoding Order Number (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 (where 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.

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               :       NALU size               |   NAL unit    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
|                                                               |
|                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: The Structure of an Aggregation Unit That Is Not the First Aggregation Unit in an AP

> Informative note: The first octet of Figure 6 (indicated by the first colon) belongs to a previous aggregation unit. It is depicted to emphasize that aggregation units are octet aligned only. Similarly, the NAL unit carried in the aggregation unit can terminate at the octet boundary.

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

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                          RTP Header                           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   PayloadHdr (Type=56)        |         NALU 1 Size           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |          NALU 1 HDR           |                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+         NALU 1 Data           |
   |                   . . .                                       |
   |                                                               |
   +               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  . . .        | NALU 2 Size                   | NALU 2 HDR    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | NALU 2 HDR    |                                               |
   +-+-+-+-+-+-+-+-+              NALU 2 Data                      |
   |                   . . .                                       |
   |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                               :...OPTIONAL RTP padding        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7: An Example of an AP Packet Containing Two Aggregation Units without the DONL Field

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

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                          RTP Header                           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   PayloadHdr (Type=56)        |        NALU 1 DONL            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |          NALU 1 Size          |            NALU 1 HDR         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   |                 NALU 1 Data   . . .                           |
   |                                                               |
   +        . . .                 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                               :          NALU 2 Size          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |          NALU 2 HDR           |                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+          NALU 2 Data          |
   |                                                               |
   |        . . .                 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                               :...OPTIONAL RTP padding        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 8: An Example of an AP Containing Two Aggregation Units with the DONL Field

4.3.3. Fragmentation Units (FUs)

FUs are introduced to enable fragmenting a single NAL unit into multiple RTP packets, possibly without cooperation or knowledge of the 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 as defined in Table 4 of [EVC] (denoted as PayloadHdr with Type=57), 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.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |    PayloadHdr (Type=57)       |   FU header   | DONL (cond)   |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
   | DONL (cond)   |                                               |
   |-+-+-+-+-+-+-+-+                                               |
   |                         FU payload                            |
   |                                                               |
   |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                               :...OPTIONAL RTP padding        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 9: The Structure of an FU

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.

```
                         0 1 2 3 4 5 6 7
                        +-+-+-+-+-+-+-+-+
                        |S|E|  FuType   |
                        +---------------+
```

Figure 10: The Structure of FU Header

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 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 (where sprop-max-don-diff is equal to 0, or where 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 S-bit and E-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. Instead, 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 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; it represents 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, then AbsDon[n] (the value of AbsDon for NAL unit n) is derived as equal to n.

Otherwise (where 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 (where 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:

- When AbsDon[n] is greater than AbsDon[m], the 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.
- When AbsDon[n] is less than AbsDon[m], the NAL unit n precedes NAL unit m in decoding order.

> Informative note: When two consecutive NAL units in the NAL unit decoding order has 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 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. When transmitting the first intra-coded picture, the originator still determines 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 the values of AbsDon may occur.

5. Packetization Rules

The following packetization rules apply:

- If sprop-max-don-diff is greater than 0, the transmission order of NAL units carried in the RTP stream **MAY** be different from the NAL unit decoding order. Otherwise (where sprop-max-don-diff equals 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 small size **SHOULD** be encapsulated in an AP together with one or more other NAL units 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 AP with its associated VCL NAL unit as, typically, a non-VCL NAL unit would be meaningless without the associated VCL NAL unit being available.
- A single NAL unit packet **MUST** be used for carrying precisely one NAL unit in an RTP packet.

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 also be used 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 considered.

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 62, 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 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 clarify the distinction from a practical receiver buffer, which is also used to compensate for transmission delay jitter, the buffer in this section will henceforth be referred to as the "de-packetization" buffer. 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 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 AP, 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 and TID set to equal the values of the fields F 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.

The receiver has two buffering states: 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 the Session Description Protocol (SDP) [RFC8866] is also provided for applications that use SDP.

Parameters starting with the string "sprop" for stream properties can be used by a sender to provide a receiver with the properties of the stream that is or will be sent. The media sender (and not the receiver) selects whether, and with what values, "sprop" parameters are being sent. This uncommon characteristic of the "sprop" parameters may not be intuitive in the context of some signaling protocol concepts, especially with Offer/Answer. Please see Section 7.3.2 for guidance specific to the use of sprop parameters in the Offer/Answer case.

7.1. Media Type Registration

The receiver **MUST** ignore any parameter unspecified in this document.

Type name: video

Subtype name: evc

Required parameters: N/A

Optional parameters: profile-id, level-id, toolset-id, max-recv-level-id, sprop-sps, sprop-pps, sprop-sei, 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 [RFC3550].

Security considerations: See Section 9 of RFC 9584.

Interoperability considerations: N/A

Published specification: Please refer to RFC 9584 and EVC standard [EVC].

Applications that use this media type: Any application that relies on EVC-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 9584.

Change controller: IETF <avtcore@ietf.org>

7.2. Optional Parameters Definition

profile-id, level-id, toolset-id:

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

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

profile-id:

When profile-id is not present, a value of 0 (i.e., the Baseline profile) **MUST** be inferred.

When used to indicate properties of a bitstream, profile-id **MUST** be derived from the profile_idc in the SPS.

EVC bitstreams transported over RTP using the technologies of this document **SHOULD** refer only to SPSs that have the same value in profile_idc, unless the sender has a priori knowledge that a receiver can correctly decode the EVC bitstream with different profile_idc values (for example, in walled garden scenarios). As exceptions to this rule, if the receiver is known to support a Baseline profile, a bitstream could safely end with CVS referring to an SPS wherein profile_idc indicates the Baseline Still picture profile. A similar exception can be made for Main profile and Main Still picture profile.

level-id:

When level-id is not present, a value of 90 (corresponding to level 3, which allows for approximately standard-definition television (SD TV) resolution and frame rates; see Annex A of [EVC]) **MUST** be inferred.

When used to indicate properties of a bitstream, level-id **MUST** be derived from the level_idc in the SPS.

If the level-id parameter is 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.

toolset-id:

This parameter is a base64-encoding representation (Section 4 of [RFC4648]) of a 64-bit unsigned integer bit mask derived from the concatenation, in network byte order, of the syntax elements toolset_idc_h and toolset_idc_l. When used to indicate properties of a bitstream, its value **MUST** be derived from toolset_idh_h and toolset_idc_l in the sequence parameter set.

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-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-encoding representations (Section 4 of [RFC4648]) of the sequence parameter set NAL units as specified in Section 7.3.2.1 of [EVC].

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-encoding representations (Section 4 of [RFC4648]) of the picture parameter set NAL units as specified in Section 7.3.2.2 of [EVC].

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-encoding representations (Section 4 of [RFC4648]) 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 couple of 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 allow 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).

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: The value of depack-buf-cap indicates the maximum possible size of the de-packetization buffer of the receiver only, without allowing for network jitter. When network jitter occurs, an appropriately sized jitter buffer has to be available as well.

7.3. SDP Parameters

The receiver **MUST** ignore any parameter unspecified in this document.

7.3.1. Mapping of Payload Type Parameters to SDP

The media type video/evc 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 evc (the media subtype).
- The clock rate in the "a=rtpmap" line **MUST** be 90000.
- The **OPTIONAL** parameters profile-id, level-id, toolset-id, max-recv-level-id, 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 "a=fmtp" line is expressed as a media type string, in the form of a semicolon-separated list of parameter=value pairs.
- The **OPTIONAL** parameters sprop-sps, sprop-pps, and sprop-sei, 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-sps, sprop-pps, or sprop-sei **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-sps, sprop-pps, and sprop-sei **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-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].

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

```
m=video 49170 RTP/AVP 98
a=rtpmap:98 evc/90000
a=fmtp:98 profile-id=1;
  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.

```
Offerer->Answerer:
      m=video 49170 RTP/AVP 98
      a=rtpmap:98 evc/90000
      a=fmtp:98 profile-id=1; level_id=90;
```

The above represents an offer for symmetric video communication using [EVC] and its payload specification at the main profile and level 3. Informally speaking, this offer tells the receiver of the offer that the sender is willing to receive up to xKpxx resolution at the maximum bitrates specified in [EVC]. 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 EVC media up to and including level 3.

```
Answerer->Offerer:
      m=video 49170 RTP/AVP 98
      a=rtpmap:98 evc/90000
      a=fmtp:98 profile-id=1; level_id=60
```

> Informative note: level_id shall be set equal to a value of 30 times the level number specified in Table A.1 of [EVC].

With this answer to the offer above, the system receiving the offer advises the Offerer that it is incapable of handling evc at level 3 but is capable of decoding level 2. As EVC 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 level 2. However, nothing prevents an encoder from further downgrading its sending to, for example, level 1 if it were short of cycles or 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 described in [RFC3264] and its updates.

This section applies to all profiles defined in [EVC], specifically to Baseline, Main, and the associated still image profiles.

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

The parameters identifying a media format configuration for EVC are profile-id and level-id. Profile_id **MUST** be used symmetrically.

The Answerer **MUST** structure its answer according to one of the following two options:

- 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; or
- 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 7.3.2.1 ("Payload Format Configuration").

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 answer **MUST NOT** contain a payload type number used in the offer for the media subtype unless the configuration is 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.

7.3.2.1. Payload Format Configuration

The following limitations and rules pertain to the configuration of the payload format buffer management.

- 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 EVC, 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.

- 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 parameters 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 (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 sets in the Offerer-to-Answerer direction.
  - An offer **MAY** include sprop-sps and/or sprop-pps. If none of these parameters are 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-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 sets in the Answerer-to-Offerer direction.
  - An answer **MAY** include sprop-sps and/or sprop-pps. If none of these parameters are 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-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-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-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].

Figure 11 lists the interpretation of all the parameters that **MAY** be used for the various combinations of offer, answer, and direction attributes.

```
                                 sendonly --+
                              recvonly --+  |
                           sendrecv --+  |  |
                                      |  |  |
   profile-id                         C  C  P
   level-id                           D  D  P
   toolset-id                         C  C  P
   max-recv-level-id                  R  R  -
   sprop-max-don-diff                 P  -  P
   sprop-depack-buf-bytes             P  -  P
   depack-buf-cap                     R  R  -
   sprop-sei                          P  -  P
   sprop-sps                          P  -  P
   sprop-pps                          P  -  P


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 level-id parameter on these parameters being
    consistent -- basically, level down-grading is allowed)

 P: properties of the bitstream to be sent
 R: receiver capabilities
 -: not usable; when present MUST be ignored
```

Figure 11: Interpretation of Parameters for Various Combinations of Offers, Answers, and Direction Attributes

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 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.

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. This leads 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 various H.26x video coding payload formats including those for [AVC] and [HEVC], it was occasionally observed that implementations were incapable of parsing most (or all) of the optional parameters and hence rejected offers other than the most basic offers. 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 predetermined 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, and not only to send it.

7.3.3. Multicast

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

- The media format configuration is identified by profile-id and level-id. 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.

7.3.4. Usage in Declarative Session Descriptions

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

- All parameters capable of indicating both bitstream properties and receiver capabilities are used to indicate only bitstream properties. For example, in this case, the parameters profile-id and level-id declare 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:
    - profile-id
    - level-id
    - sprop-sps
    - sprop-pps
    - sprop-max-don-diff
    - sprop-depack-buf-bytes
    - sprop-sei
  - Not usable (when present, they **MUST** be ignored):
    - depack-buf-cap
    - recv-sublayer-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.

7.3.5. 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. An 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) [RFC4585] and Full Intra Request (FIR) [RFC5104] feedback messages with [EVC].

In accordance with this document, a sender **MUST NOT** send Slice Loss Indication (SLI) or Reference Picture Selection Indication (RPSI); and a receiver **MUST** ignore RPSI and **MUST** treat a received SLI as a received PLI, ignoring the "First", "Number", and "PictureID" fields of the PLI.

8.1. Picture Loss Indication (PLI)

As specified in Section 6.3.1 of [RFC4585], 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, IDR picture locations, picture structures, and so forth), a reaction to the reception of a PLI by an EVC sender **SHOULD** be to send an IDR picture and relevant parameter sets, potentially with sufficient redundancy so as to ensure correct reception. However, sometimes information about the bitstream structure is known. For example, such information can be parameter sets that have been conveyed out of band through mechanisms not defined in this document and that are known to 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 [RFC4585] **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 the 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 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 in Section 14 of [RFC3550]) and the security of the call-control stack chosen (that may make use of the media type registration of this document). 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 document seem 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 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. 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** based on [RFC7202].

Like HEVC [RFC7798] and VVC [VVC], EVC [EVC] 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.

EVC [EVC] 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, forwarding all received JavaScript code detected by a decoder implementation to a web browser unchecked would be a bad and insecure implementation practice. 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] or AVPF [RFC4585]. 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. This condition can be satisfied by implementing congestion-control mechanisms to adapt the transmission rate by implementing 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 large value of TID) until the sending bitrate drops to an acceptable range.

The mechanisms mentioned above generally work within a defined profile and level; therefore, no renegotiation of the channel is required. Only when non-downgradable parameters (such as the profile) are required to be changed does it become necessary to terminate and restart the RTP streams. 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, because the trailing FUs 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

The media type specified in Section 7.1 has been registered with IANA.

12. References

12.1. Normative References

[EVC] "Information technology -- General video coding -- Part 1: Essential video coding", ISO/IEC 23094-1:2020, October 2020, <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>.

[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>.

[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>.

[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>.

[RFC9328] Zhao, S., Wenger, S., Sanchez, Y., Wang, Y.-K., and M. M. Hannuksela, "RTP Payload Format for Versatile Video Coding (VVC)", RFC 9328, DOI 10.17487/RFC9328, December 2022, <https://www.rfc-editor.org/info/rfc9328>.

[VSEI] ITU-T, "Versatile supplemental enhancement information messages for coded video bitstreams", ITU-T Recommendation H.274, March 2024, <https://www.itu.int/rec/T-REC-H.274>.

12.2. Informative References

[AVC] ITU-T, "Part 10: Advanced video coding", ITU-T Recommendation H.264, October 2014, <https://www.iso.org/standard/66069.html>.

[HEVC] ITU-T, "High efficiency video coding", ITU-T Recommendation H.265, November 2019, <https://www.itu.int/rec/T-REC-H.265>.

[MPEG2S] IS0/IEC, "Information technology - Generic coding of moving pictures and associated audio information - Part 1: Systems", ISO/IEC 13818-1:2013, June 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>.

[VIDEO-CODING] ITU-T, "Video coding for low bit rate communication", ITU-T Recommendation H.263, January 2005, <https://www.itu.int/rec/T-REC-H.263>.

[VVC] ITU-T, "Versatile video coding", ITU-T Recommendation H.266, August 2020, <http://www.itu.int/rec/T-REC-H.266>.

Acknowledgements

Large parts of this specification share text with the RTP payload format for VVC [RFC9328]. Roman Chernyak is thanked for his valuable review comments. We thank the authors of that specification for their excellent work.

Authors' Addresses

Shuai Zhao
Intel
2200 Mission College Blvd
Santa Clara, California 95054
United States of America
Email: shuai.zhao@ieee.org

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

Youngkwon Lim
Samsung Electronics
6625 Excellence Way
Plano, Texas 75013
United States of America
Email: yklwhite@gmail.com