

avtcore  
Internet-Draft  
Intended status: Standards Track  
Expires: December 4, 2021

S. Zhao  
S. Wenger  
Tencent  
Y. Sanchez  
Fraunhofer HHI  
Y. Wang  
Bytedance Inc.  
June 02, 2021

RTP Payload Format for Versatile Video Coding (VVC)  
draft-ietf-avtcore-rtp-vvc-09

Abstract

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

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 4, 2021.

Copyright Notice

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

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

## Table of Contents

1.	Introduction	3
1.1.	Overview of the VVC Codec	3
1.1.1.	Coding-Tool Features (informative)	3
1.1.2.	Systems and Transport Interfaces (informative)	6
1.1.3.	High-Level Picture Partitioning (informative)	11
1.1.4.	NAL Unit Header	13
1.2.	Overview of the Payload Format	14
2.	Conventions	15
3.	Definitions and Abbreviations	15
3.1.	Definitions	15
3.1.1.	Definitions from the VVC Specification	15
3.1.2.	Definitions Specific to This Memo	18
3.2.	Abbreviations	19
4.	RTP Payload Format	20
4.1.	RTP Header Usage	20
4.2.	Payload Header Usage	21
4.3.	Payload Structures	22
4.3.1.	Single NAL Unit Packets	22
4.3.2.	Aggregation Packets (APs)	23
4.3.3.	Fragmentation Units	27
4.4.	Decoding Order Number	30
5.	Packetization Rules	32
6.	De-packetization Process	32
7.	Payload Format Parameters	34
7.1.	Media Type Registration	34
7.2.	SDP Parameters	47
7.2.1.	Mapping of Payload Type Parameters to SDP	47
7.2.2.	Usage with SDP Offer/Answer Model	48
8.	Use with Feedback Messages	49
8.1.	Picture Loss Indication (PLI)	49
8.2.	Full Intra Request (FIR)	49
9.	Security Considerations	49
10.	Congestion Control	51
11.	IANA Considerations	52
12.	Acknowledgements	52
13.	References	52

13.1. Normative References . . . . .	52
13.2. Informative References . . . . .	54
Appendix A. Change History . . . . .	55
Authors' Addresses . . . . .	56

## 1. Introduction

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

This memo specifies an RTP payload format for VVC. It shares its basic design with the NAL (Network Abstraction Layer) unit-based RTP payload formats of, H.264 Video Coding [RFC6184], Scalable Video Coding (SVC) [RFC6190], High Efficiency Video Coding (HEVC) [RFC7798] and their respective predecessors. With respect to design philosophy, security, congestion control, and overall implementation complexity, it has similar properties to those earlier payload format specifications. This is a conscious choice, as at least RFC 6184 is widely deployed and generally known in the relevant implementer communities. Certain mechanisms known from [RFC6190] were incorporated in VVC, as VVC version 1 supports temporal, spatial, and signal-to-noise ratio (SNR) scalability.

### 1.1. Overview of the VVC Codec

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

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

#### 1.1.1. Coding-Tool Features (informative)

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

Similar to earlier hybrid-video-coding-based standards, including HEVC, the following basic video coding design is employed by VVC. A

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

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

#### Coding blocks and transform structure

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

#### Entropy coding

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

#### In-loop filtering

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

#### Motion prediction and coding

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

#### Intra prediction and intra-coding

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

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

#### Other coding-tool feature

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

#### 1.1.2. Systems and Transport Interfaces (informative)

VVC inherits the basic systems and transport interfaces designs from HEVC and H.264. These include the NAL-unit-based syntax structure, the hierarchical syntax and data unit structure, the supplemental enhancement information (SEI) message mechanism, and the video buffering model based on the hypothetical reference decoder (HRD). The scalability features of VVC are conceptually similar to the scalable variant of HEVC known as SHVC. The hierarchical syntax and data unit structure consists of parameter sets at various levels (decoder, sequence (pertaining to all), sequence (pertaining to a single), picture), picture-level header parameters, slice-level header parameters, and lower-level parameters.

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

#### Decoding capability information

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

indicated by the values of those fields. With this, a bitstream can be labelled as not using certain tools, which allows among other things for resource allocation in a decoder implementation.

#### Video parameter set

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

#### Sequence parameter set

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

#### Picture and adaptation parameter set

The picture parameter set and the adaptation parameter set (PPS and APS, respectively) carry information pertaining to zero or more pictures and zero or more slices, respectively. The PPS contains information that is likely to stay constant from picture to picture- at least for pictures for a certain type-whereas the APS contains information, such as adaptive loop filter coefficients, that are likely to change from picture to picture or even within a picture. A single APS is referenced by all slices of the same picture if that APS contains information about luma mapping with chroma scaling (LMCS) or scaling list. Different APSs containing ALF parameters can be referenced by slices of the same picture.

## Picture header

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

## Profile, tier, and level

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

## Sub-profiles

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

## General constraint fields

The `profile_tier_level` structure carries a considerable number of constraint fields (most of which are flags), which an encoder can use to indicate to a decoder that it will not use a certain tool or



technology. They were included in reaction to a perceived market need for labelling a bitstream as not exercising a certain tool that has become commercially unviable.

#### Temporal scalability support

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

#### Reference picture resampling (RPR)

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

#### Spatial, SNR, and multiview scalability

VVC includes support for spatial, SNR, and multiview scalability. Scalable video coding is widely considered to have technical benefits and enrich services for various video applications. Until recently, however, the functionality has not been included in the first version of specifications of the video codecs. In VVC, however, all those forms of scalability are supported in the first version of VVC natively through the signaling of the layer\_id in the NAL unit header, the VPS which associates layers with given layer\_ids to each other, reference picture selection, reference picture resampling for spatial scalability, and a number of other mechanisms not relevant for this memo.

#### Spatial scalability

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

#### SNR scalability

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

#### Multiview scalability

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

#### SEI messages

Supplementary enhancement information (SEI) messages are information in the bitstream that do not influence the decoding process as specified in the VVC spec, but address issues of representation/rendering of the decoded bitstream, label the bitstream for certain applications, among other, similar tasks. The overall concept of SEI messages and many of the messages themselves has been inherited from the H.264 and HEVC specs. Except for the SEI messages that affect the specification of the hypothetical reference decoder (HRD), other SEI messages for use in the VVC environment, which are generally useful also in other video coding technologies, are not included in the main VVC specification but in a companion specification [VSEI].

### 1.1.3. High-Level Picture Partitioning (informative)

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

#### Tiles and WPP

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

#### Slices

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

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

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

### Subpictures

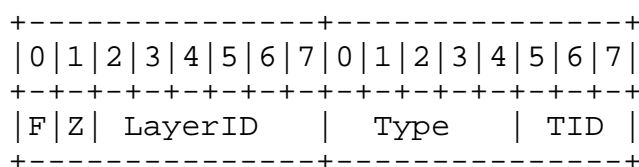
VVC is the first video coding standard that includes the support of subpictures as a feature. Each subpicture consists of one or more complete rectangular slices that collectively cover a rectangular region of the picture. A subpicture may be either specified to be extractable (i.e., coded independently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

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

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

#### 1.1.4. NAL Unit Header

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



The Structure of the VVC NAL Unit Header.

Figure 1

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

F: 1 bit

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

Z: 1 bit

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

LayerId: 6 bits

nuh\_layer\_id. Identifies the layer a NAL unit belongs to, wherein a layer may be, e.g., a spatial scalable layer, a quality scalable layer, a layer containing a different view, etc.

Type: 5 bits

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

TID: 3 bits

nuh\_temporal\_id\_plus1. This field specifies the temporal identifier of the NAL unit plus 1. The value of TemporalId is equal to TID minus 1. A TID value of 0 is illegal to ensure that there is at least one bit in the NAL unit header equal to 1, so to enable independent considerations of start code emulations in the NAL unit header and in the NAL unit payload data.

## 1.2. Overview of the Payload Format

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

- o Usage of RTP header with this payload format

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

## 3. Definitions and Abbreviations

### 3.1. Definitions

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

#### 3.1.1. Definitions from the VVC Specification

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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



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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

### 3.1.2. Definitions Specific to This Memo

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

**Informative note:** The concept of a MANE goes beyond normal routers or gateways in that a MANE has to be aware of the signaling (e.g., to learn about the payload type mappings of the media streams), and in that it has to be trusted when working with Secure RTP (SRTP). The advantage of using MANEs is that they allow packets to be dropped according to the needs of the media coding. For example, if a MANE has to drop packets due to congestion on a certain link, it can identify and remove those packets whose elimination produces the least adverse effect on the user experience. After dropping packets, MANEs must rewrite RTCP packets to match the changes to the RTP stream, as specified in Section 7 of [RFC3550].

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

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

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

### 3.2. Abbreviations

AU	Access Unit
AP	Aggregation Packet
APS	Adaptation Parameter Set
CTU	Coding Tree Unit
CVS	Coded Video Sequence
DPB	Decoded Picture Buffer
DCI	Decoding Capability Information
DON	Decoding Order Number
FIR	Full Intra Request
FU	Fragmentation Unit
GDR	Gradual Decoding Refresh
HRD	Hypothetical Reference Decoder
IDR	Instantaneous Decoding Refresh
MANE	Media-Aware Network Element
MTU	Maximum Transfer Unit
NAL	Network Abstraction Layer

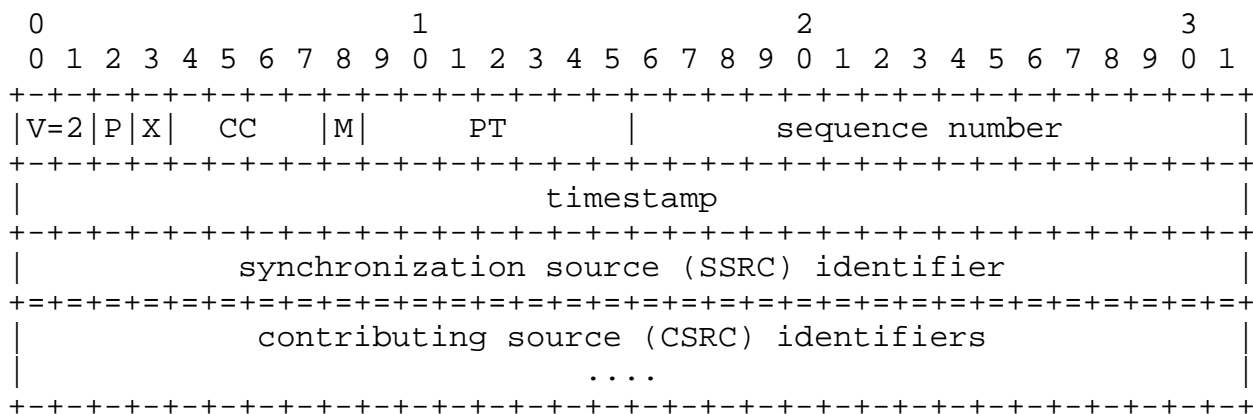
NALU	Network Abstraction Layer Unit
PLI	Picture Loss Indication
PPS	Picture Parameter Set
RPS	Reference Picture Set
RPSI	Reference Picture Selection Indication
SEI	Supplemental Enhancement Information
SLI	Slice Loss Indication
SPS	Sequence Parameter Set
VCL	Video Coding Layer
VPS	Video Parameter Set

4. RTP Payload Format

4.1. RTP Header Usage

The format of the RTP header is specified in [RFC3550] (reprinted as Figure 2 for convenience). This payload format uses the fields of the header in a manner consistent with that specification.

The RTP payload (and the settings for some RTP header bits) for aggregation packets and fragmentation units are specified in Section 4.3.2 and Section 4.3.3, respectively.



RTP Header According to {{RFC3550}}

Figure 2

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

Marker bit (M): 1 bit

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

Payload Type (PT): 7 bits

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

Sequence Number (SN): 16 bits

Set and used in accordance with [RFC3550].

Timestamp: 32 bits

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

Synchronization source (SSRC): 32 bits

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

#### 4.2. Payload Header Usage

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

The TID value indicates (among other things) the relative importance of an RTP packet, for example, because NAL units belonging to higher temporal sublayers are not used for the decoding of lower temporal

sublayers. A lower value of TID indicates a higher importance. More-important NAL units MAY be better protected against transmission losses than less-important NAL units.

For Discussion: quite possibly something similar can be said for the Layer\_id in layered coding, but perhaps not in multiview coding. (The relevant part of the spec is relatively new, therefore the soft language). However, for serious layer pruning, interpretation of the VPS is required. We can add language about the need for stateful interpretation of LayerID vis-a-vis stateless interpretation of TID later.

### 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:

- o Single NAL unit packet: Contains a single NAL unit in the payload, and the NAL unit header of the NAL unit also serves as the payload header. This payload structure is specified in Section 4.4.1.
- o Aggregation Packet (AP): Contains more than one NAL unit within one access unit. This payload structure is specified in Section 4.3.2.
- o Fragmentation Unit (FU): Contains a subset of a single NAL unit. This payload structure is specified in Section 4.3.3.

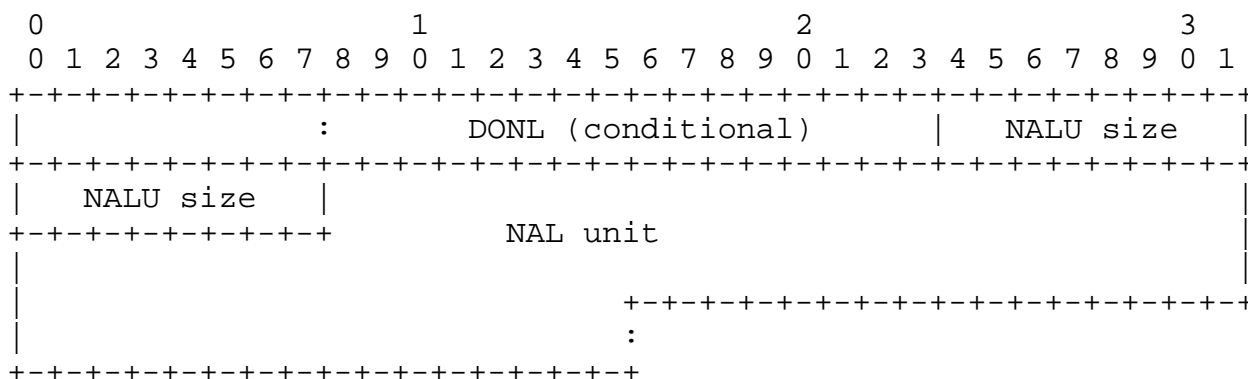
#### 4.3.1. Single NAL Unit Packets

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









The Structure of the First Aggregation Unit in an AP

Figure 5

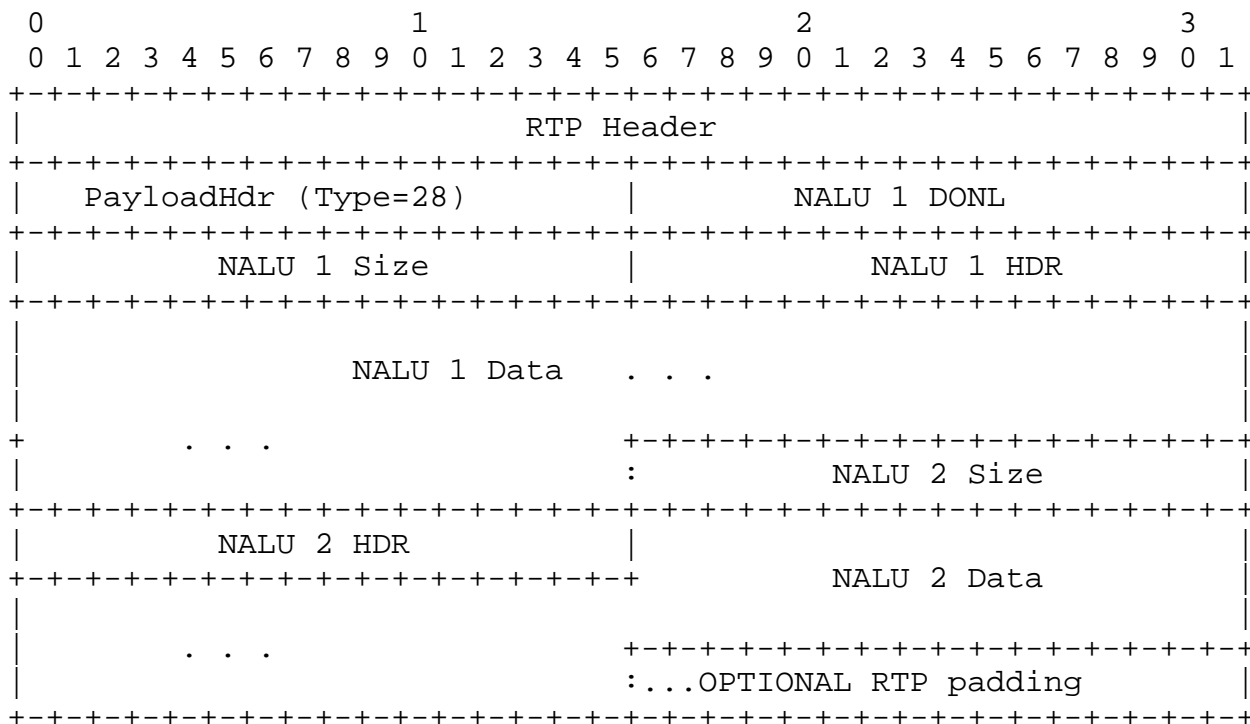
The DONL field, when present, specifies the value of the 16 least significant bits of the decoding order number of the aggregated NAL unit.

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

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



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



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

Figure 8

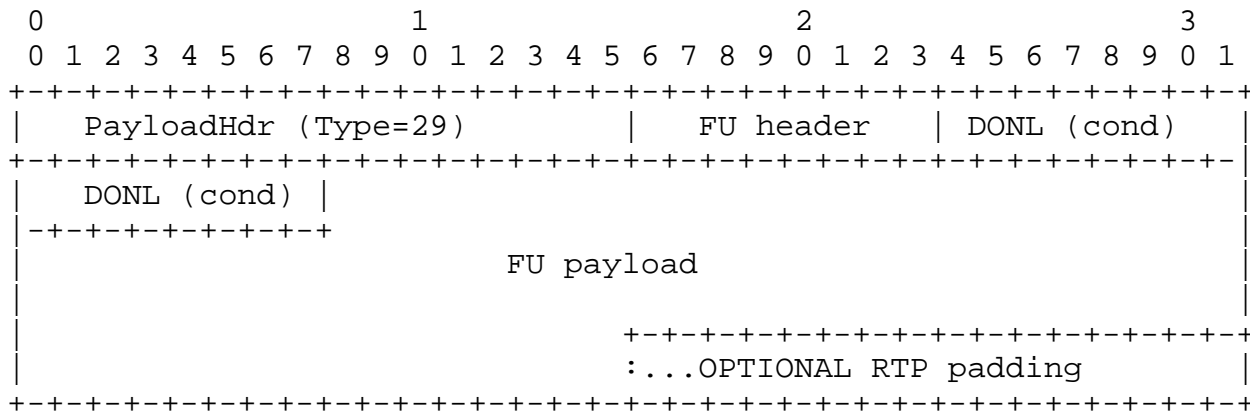
### 4.3.3. Fragmentation Units

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

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

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

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

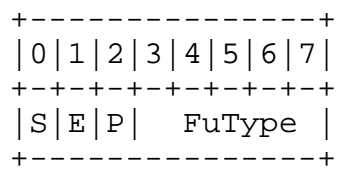


The Structure of an FU

Figure 9

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

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



The Structure of FU Header

Figure 10

The semantics of the FU header fields are as follows:

S: 1 bit

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

E: 1 bit

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

P: 1 bit

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

FuType: 5 bits

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

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

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

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

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

headers of the FUs and the FuType field of the FU header of the FUs. An FU payload MUST NOT be empty.

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

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

#### 4.4. Decoding Order Number

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

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

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

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

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

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

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

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

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

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

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

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

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

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

how many NAL units will be encoded before the first intra-coded picture of the pre-encoded clip follows in decoding order. Thus, the values of AbsDon for the NAL units of the first intra-coded picture of the pre-encoded clip have to be estimated when they are transmitted, and gaps in values of AbsDon may occur.

## 5. Packetization Rules

The following packetization rules apply:

- o If sprop-max-don-diff is greater than 0, the transmission order of NAL units carried in the RTP stream MAY be different than the NAL unit decoding order. Otherwise (sprop-max-don-diff is equal to 0), the transmission order of NAL units carried in the RTP stream MUST be the same as the NAL unit decoding order.
- o A NAL unit of a small size SHOULD be encapsulated in an aggregation packet together one or more other NAL units in order to avoid the unnecessary packetization overhead for small NAL units. For example, non-VCL NAL units such as access unit delimiters, parameter sets, or SEI NAL units are typically small and can often be aggregated with VCL NAL units without violating MTU size constraints.
- o Each non-VCL NAL unit SHOULD, when possible from an MTU size match viewpoint, be encapsulated in an aggregation packet together with its associated VCL NAL unit, as typically a non-VCL NAL unit would be meaningless without the associated VCL NAL unit being available.
- o For carrying exactly one NAL unit in an RTP packet, a single NAL unit packet MUST be used.

## 6. De-packetization Process

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

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



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

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

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

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

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

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

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

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

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

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

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

## 7. Payload Format Parameters

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

### 7.1. Media Type Registration

The receiver MUST ignore any parameter unspecified in this memo.

Type name: video

Subtype name: H266

Required parameters: none

Optional parameters:

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

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

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

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

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

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

- + The tier-flag, profile-id, sub-profile-id, and interop-constraints parameters MUST be used symmetrically, i.e., the value of each of these parameters in the offer MUST be the same as that in the answer, either explicitly signaled or implicitly inferred.

- + The level-id parameter is changeable as long as the highest level indicated by the answer is either equal to or lower than that in the offer. Note that a highest level higher than level-id in the offer for receiving can be included as max-recv-level-id.

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

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

profile-id:

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

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

A `profile_tier_level( )` syntax structure may be contained in an SPS, VPS, or DCI NAL units as specified in [VVC]. One of the following three cases applies to the container NAL unit of the `profile_tier_level( )` syntax structure containing those PTL syntax elements used to derive the values of profile-id, tier-flag, level-id, sub-profile-id, or interop-constraints: 1) The container NAL unit is an SPS, the bitstream is a single-layer bitstream, and the `profile_tier_level( )` syntax structures in

all SPSs referenced by the CVSSs in the bitstream has the same values respectively for those PTL syntax elements; 2) The container NAL unit is a VPS, the `profile_tier_level( )` syntax structure is the one in the VPS that applies to the OLS corresponding to the bitstream, and the `profile_tier_level( )` syntax structures applicable to the OLS corresponding to the bitstream in all VPSs referenced by the CVSSs in the bitstream have the same values respectively for those PTL syntax elements; 3) The container NAL unit is a DCI NAL unit and the `profile_tier_level( )` syntax structures in all DCI NAL units in the bitstream has the same values respectively for those PTL syntax elements.

tier-flag, level-id:

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

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

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

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

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

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

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

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

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

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

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

```
+ tier-flag = general_tier_flag
+ level-id = sub_layer_level_idc[j]
```

`sub-profile-id`:

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

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

`interop-constraints`:

A base16 [RFC4648] (hexadecimal) representation of the data that includes the syntax elements `ptl_frame_only_constraint_flag` and `ptl_multilayer_enabled_flag` and the `general_constraints_info( )` syntax structure that apply to the bitstream in an instance of the `profile_tier_level( )` syntax structure.

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

- + `ptl_frame_only_constraint_flag` = 0
- + `ptl_multilayer_enabled_flag` = 1
- + `gci_present_flag` in the `general_constraints_info( )` syntax structure = 1

editor-note 14: Double check the default values. Currently, no constraints, but actually, with the Main 10 profile as default multi-layer not possible.

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

`sprop-sub-layer-id`:

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

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

`sprop-ols-id`:

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

The value of `sprop-ols-id` MUST be in the range of 0 to 257, inclusive.

`recv-sub-layer-id`:



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

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

`recv-ols-id`:

This parameter MAY be used to signal a receiver's choice of the offered or declared output layer sets in the sprop-vps. The value of `recv-ols-id` indicates the OLS index of the bitstream that a receiver supports. When not present, the value of `recv-ols-id` is inferred to be equal to the value of the `sprop-ols-id` parameter in the SDP offer. When present, the value of `recv-ols-id` must be included only when `sprop-ols-id` was received and must refer to an output layer set in the VPS that is in the same dependency tree as the OLS referred to by `sprop-ols-id`. If this optional parameter is present, `sprop-vps` must have been received or its content must be known a priori at the receiver.

The value of `recv-ols-id` MUST be in the range of 0 to 257, inclusive.

`max-recv-level-id`:

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

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

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

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

`sprop-dci`:

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

`sprop-vps`:

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

The `sprop-vps` parameter MAY contain one or more than one video parameter set NAL unit. However, all other video parameter sets contained in the `sprop-vps` parameter MUST be consistent with the first video parameter set in the `sprop-vps` parameter. A video parameter set `vpsB` is said to be consistent with another video parameter set `vpsA` if any decoder that conforms to the profile, tier, level, and constraints indicated by the 12 bytes of data starting from the syntax element `general_profile_space` to the syntax element `general_level_idc`, inclusive, in the first `profile_tier_level( )` syntax structure in `vpsA` can decode any bitstream that conforms to the profile, tier, level, and constraints indicated by the 12 bytes of data starting from the syntax element `general_profile_space` to the

syntax element `general_level_idc`, inclusive, in the first `profile_tier_level( )` syntax structure in `vpsB`.

`sprop-sei`:

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

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

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

- 1) In an environment where the bitstream was created from film-based source material, and no splicing is going to occur during the lifetime of the session, the film grain characteristics SEI message is likely meaningful, and sending it in `sprop-sei` rather than in the bitstream at each entry point may help with saving bits and allows one to configure the renderer only once, avoiding unwanted artifacts.

2) Examples for SEI messages that would be meaningless to be conveyed in sprop-sei include the decoded picture hash SEI message (it is close to impossible that all decoded pictures have the same hashtag), the display orientation SEI message when the device is a handheld device (as the display orientation may change when the handheld device is turned around), or the filler payload SEI message (as there is no point in just having more bits in SDP).

max-lsr:

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

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

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

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

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

max-fps:

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

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

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

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

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

sprop-max-don-diff:

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

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

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

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

`sprop-depack-buf-bytes`:

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

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

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

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

`depack-buf-cap`:

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

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

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

## 7.2. SDP Parameters

The receiver MUST ignore any parameter unspecified in this memo.

### 7.2.1. Mapping of Payload Type Parameters to SDP

The media type `video/H266` string is mapped to fields in the Session Description Protocol (SDP) [RFC4566] as follows:

- o The media name in the "m=" line of SDP MUST be `video`.
- o The encoding name in the "a=rtpmap" line of SDP MUST be `H266` (the media subtype).
- o The clock rate in the "a=rtpmap" line MUST be `90000`.

- o The OPTIONAL parameters profile-id, tier-flag, sub-profile-id, interop-constraints, level-id, sprop-sub-layer-id, sprop-ols-id, recv-sub-layer-id, recv-ols-id, max-recv-level-id, max-lsr, max-fps, sprop-max-don-diff, sprop-depack-buf-bytes and depack-buf-cap, when present, MUST be included in the "a=fmtp" line of SDP. This parameter is expressed as a media type string, in the form of a semicolon-separated list of parameter=value pairs.
- o The OPTIONAL parameter sprop-vps, when present, MUST be included in the "a=fmtp" line of SDP or conveyed using the "fmtp" source attribute as specified in Section 6.3 of [RFC5576]. For a particular media format (i.e., RTP payload type), sprop-vps 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, sprop-vps is expressed as a media type string, in the form of a parameter=value pair. When conveyed in the "a=fmtp" line of SDP for a particular payload type, the parameter sprop-vps MUST be applied to each SSRC with the payload type. When conveyed using the "fmtp" source attribute, sprop-vps is only associated with the given source and payload type as parts of the "fmtp" source attribute.

An example of media representation in SDP is as follows:

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

#### 7.2.2. Usage with SDP Offer/Answer Model

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

editor-note 21: the following needs to be updated

- o Parameters to identify a media format configuration as VVC:
- o Parameters as bitstream properties:
- o SDP answer for media configurations.
- o capability parameters:
- o others:



## 8. Use with Feedback Messages

The following subsections define the use of the Picture Loss Indication (PLI), Slice Lost Indication (SLI), Reference Picture Selection Indication (RPSI), and Full Intra Request (FIR) feedback messages with HEVC. The PLI, SLI, and RPSI messages are defined in [RFC4585], and the FIR message is defined in [RFC5104].

### 8.1. Picture Loss Indication (PLI)

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

### 8.2. Full Intra Request (FIR)

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

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

## 9. Security Considerations

The scope of this Security Considerations section is limited to the payload format itself and to one feature of [VVC] that may pose a particularly serious security risk if implemented naively. The payload format, in isolation, does not form a complete system.

Implementers are advised to read and understand relevant security-related documents, especially those pertaining to RTP (see the Security Considerations section in [RFC3550] ), and the security of the call-control stack chosen (that may make use of the media type registration of this memo). Implementers should also consider known security vulnerabilities of video coding and decoding implementations in general and avoid those.

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

RTP packets using the payload format defined in this specification are subject to the security considerations discussed in the RTP specification [RFC3550] , and in any applicable RTP profile such as RTP/AVP [RFC3551] , RTP/AVPF [RFC4585] , RTP/SAVP [RFC3711] , or RTP/SAVPF [RFC5124] . However, as "Securing the RTP Framework: Why RTP Does Not Mandate a Single Media Security Solution" [RFC7202] discusses, it is not an RTP payload format's responsibility to discuss or mandate what solutions are used to meet the basic security goals like confidentiality, integrity and source authenticity for RTP in general. This responsibility lays on anyone using RTP in an application. They can find guidance on available security mechanisms and important considerations in "Options for Securing RTP Sessions" [RFC7201] . The rest of this section discusses the security impacting properties of the payload format itself.

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

Like HEVC [RFC7798], [VVC] includes a user data Supplemental Enhancement Information (SEI) message. This SEI message allows inclusion of an arbitrary bitstring into the video bitstream. Such a bitstring could include JavaScript, machine code, and other active content. [VVC] leaves the handling of this SEI message to the

receiving system. In order to avoid harmful side effects the user data SEI message, decoder implementations cannot naively trust its content. For example, it would be a bad and insecure implementation practice to forward any JavaScript a decoder implementation detects to a web browser. The safest way to deal with user data SEI messages is to simply discard them, but that can have negative side effects on the quality of experience by the user.

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

## 10. Congestion Control

Congestion control for RTP SHALL be used in accordance with RTP [RFC3550] and with any applicable RTP profile, e.g., AVP [RFC3551]. If best-effort service is being used, an additional requirement is that users of this payload format MUST monitor packet loss to ensure that the packet loss rate is within an acceptable range. Packet loss is considered acceptable if a TCP flow across the same network path, and experiencing the same network conditions, would achieve an average throughput, measured on a reasonable timescale, that is not less than all RTP streams combined are achieving. This condition can be satisfied by implementing congestion-control mechanisms to adapt the transmission rate, the number of layers subscribed for a layered multicast session, or by arranging for a receiver to leave the session if the loss rate is unacceptably high.

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

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

restart the RTP stream(s). This may be accomplished by using different RTP payload types.

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

## 11. IANA Considerations

Placeholder

## 12. Acknowledgements

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

## 13. References

### 13.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, DOI 10.17487/RFC3264, June 2002, <<https://www.rfc-editor.org/info/rfc3264>>.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <<https://www.rfc-editor.org/info/rfc3550>>.

- [RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, DOI 10.17487/RFC3551, July 2003, <<https://www.rfc-editor.org/info/rfc3551>>.
- [RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, DOI 10.17487/RFC3711, March 2004, <<https://www.rfc-editor.org/info/rfc3711>>.
- [RFC4556] Zhu, L. and B. Tung, "Public Key Cryptography for Initial Authentication in Kerberos (PKINIT)", RFC 4556, DOI 10.17487/RFC4556, June 2006, <<https://www.rfc-editor.org/info/rfc4556>>.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, DOI 10.17487/RFC4566, July 2006, <<https://www.rfc-editor.org/info/rfc4566>>.
- [RFC4585] Ott, J., Wenger, S., Sato, N., Burmeister, C., and J. Rey, "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", RFC 4585, DOI 10.17487/RFC4585, July 2006, <<https://www.rfc-editor.org/info/rfc4585>>.
- [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>>.

- [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>>.
- [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>>.
- [VSEI] "ISO/IEC 23002-7 (ITU-T H.274) Versatile supplemental enhancement information messages for coded video bitstreams", 2020, <<https://www.iso.org/standard/79112.html>>.
- [VVC] "ISO/IEC FDIS 23090-3 Information technology --- Coded representation of immersive media --- Part 3 - Versatile video coding", 2020, <<https://www.iso.org/standard/73022.html>>.

### 13.2. Informative References

- [CABAC] Sole, J, . and . et al, "Transform coefficient coding in HEVC, IEEE Transactions on Circuits and Systems for Video Technology", DOI 10.1109/TCSVT.2012.2223055, December 2012.
- [HEVC] "High efficiency video coding, ITU-T Recommendation H.265", April 2013.
- [MPEG2S] ISO/IEC, ., "Information technology - Generic coding of moving pictures and associated audio information - Part 1: Systems, ISO International Standard 13818-1", 2013.
- [RFC6184] Wang, Y., Even, R., Kristensen, T., and R. Jesup, "RTP Payload Format for H.264 Video", RFC 6184, DOI 10.17487/RFC6184, May 2011, <<https://www.rfc-editor.org/info/rfc6184>>.

- [RFC6190] Wenger, S., Wang, Y., Schierl, T., and A. Eleftheriadis, "RTP Payload Format for Scalable Video Coding", RFC 6190, DOI 10.17487/RFC6190, May 2011, <<https://www.rfc-editor.org/info/rfc6190>>.
- [RFC7201] Westerlund, M. and C. Perkins, "Options for Securing RTP Sessions", RFC 7201, DOI 10.17487/RFC7201, April 2014, <<https://www.rfc-editor.org/info/rfc7201>>.
- [RFC7202] Perkins, C. and M. Westerlund, "Securing the RTP Framework: Why RTP Does Not Mandate a Single Media Security Solution", RFC 7202, DOI 10.17487/RFC7202, April 2014, <<https://www.rfc-editor.org/info/rfc7202>>.
- [RFC7798] Wang, Y., Sanchez, Y., Schierl, T., Wenger, S., and M. Hannuksela, "RTP Payload Format for High Efficiency Video Coding (HEVC)", RFC 7798, DOI 10.17487/RFC7798, March 2016, <<https://www.rfc-editor.org/info/rfc7798>>.

#### Appendix A. Change History

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

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

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

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

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

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

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

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

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

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

Authors' Addresses

Shuai Zhao  
Tencent  
2747 Park Blvd  
Palo Alto 94588  
USA

Email: [shuai.zhao@ieee.org](mailto:shuai.zhao@ieee.org)

Stephan Wenger  
Tencent  
2747 Park Blvd  
Palo Alto 94588  
USA

Email: [stewe@stewe.org](mailto:stewe@stewe.org)

Yago Sanchez  
Fraunhofer HHI  
Einsteinufer 37  
Berlin 10587  
Germany

Email: [yago.sanchez@hhi.fraunhofer.de](mailto:yago.sanchez@hhi.fraunhofer.de)

Ye-Kui Wang  
Bytedance Inc.  
8910 University Center Lane  
San Diego 92122  
USA

Email: [yekui.wang@bytedance.com](mailto:yekui.wang@bytedance.com)