Network Working Group
Internet Draft
Intended status: Standards track
Expires: August 2014

Y.-K. Wang
Qualcomm
Y. Sanchez
T. Schierl
Fraunhofer HHI
S. Wenger
Vidyo
M. M. Hannuksela
Nokia

February 12, 2014

RTP Payload Format for High Efficiency Video Coding
draft-ietf-payload-rtp-h265-02.txt

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on August 12, 2014.

Copyright and License Notice

Copyright (c) 2014 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 (http://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.

Abstract

This memo describes an RTP payload format for the video coding standard ITU-T Recommendation H.265 and ISO/IEC International Standard 23008-2, both also known as High Efficiency Video Coding (HEVC) [HEVC], developed by the Joint Collaborative Team on Video Coding (JCT-VC). 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. Furthermore, it supports transmission of an HEVC stream over a single as well as multiple RTP flows. The payload format has wide applicability in videoconferencing, Internet video streaming, and high bit-rate entertainment-quality video, among others.

Table of Contents

Status of this Memo.....1
Abstract.....3
Table of Contents.....3
1 . Introduction.....5
1.1 . Overview of the HEVC Codec.....5
1.1.1 Coding-Tool Features.....5
1.1.2 Systems and Transport Interfaces.....7
1.1.3 Parallel Processing Support.....14
1.1.4 NAL Unit Header.....16
1.2 . Overview of the Payload Format.....17
2 . Conventions.....18
3 . Definitions and Abbreviations.....18
3.1 Definitions.....18
3.1.1 Definitions from the HEVC Specification.....18
3.1.2 Definitions Specific to This Memo.....20
3.2 Abbreviations.....21
4 . RTP Payload Format.....23
4.1 RTP Header Usage.....23
4.2 Payload Header Usage.....25
4.3 Payload Structures.....25
4.4 Transmission Modes.....26
4.5 Decoding Order Number.....27
4.6 Single NAL Unit Packets.....28

4.7 Aggregation Packets (APs).....29
4.8 Fragmentation Units (FUs).....34
4.9 PACI packets.....37
4.9.1 Reasons for the PACI rules (informative).....40
4.10 Payload Header Extensions.....41
5 . Packetization Rules.....43
6 . De-packetization Process.....43
7 . Payload Format Parameters.....45
7.1 Media Type Registration.....45
7.2 SDP Parameters.....64
7.2.1 Mapping of Payload Type Parameters to SDP.....64
7.2.2 Usage with SDP Offer/Answer Model.....65
7.2.3 Usage in Declarative Session Descriptions.....73
7.2.4 Parameter Sets Considerations.....74
7.2.5 Dependency Signaling in Multi-Session Transmission...74
8 . Use with Feedback Messages.....75
8.1 Use of HEVC with the RPSI Feedback Message.....76
9 . Security Considerations.....76
10 . Congestion Control.....78
11 . IANA Consideration.....79
12 . Acknowledgements.....79
13 . References.....79
13.1 Normative References.....79
13.2 Informative References.....81
14 . Authors' Addresses.....82

1. Introduction

1.1. Overview of the HEVC Codec

High Efficiency Video Coding [HEVC], formally known as ITU-T Recommendation H.265 and ISO/IEC International Standard 23008-2 was ratified by ITU-T in April 2013 and reportedly provides significant coding efficiency gains over H.264 [H.264].

As both H.264 [H.264] and its RTP payload format [RFC6184] are widely deployed and generally known in the relevant implementer communities, frequently only the differences between those two specifications are highlighted in non-normative, explanatory parts of this memo. Basic familiarity with both specifications is assumed for those parts. However, the normative parts of this memo do not require study of H.264 or its RTP payload format.

H.264 and HEVC share a similar hybrid video codec design. Conceptually, both technologies include a video coding layer (VCL), which is often used to refer to the coding-tool features, and a network abstraction layer (NAL), which is often used to refer to the systems and transport interface aspects of the codecs.

1.1.1 Coding-Tool Features

Similarly to earlier hybrid-video-coding-based standards, including H.264, the following basic video coding design is employed by HEVC. A prediction signal is first formed either by 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, HEVC includes several tools to make the implementation on parallel architectures easier. Below is a summary of HEVC coding-tool features.

Quad-tree block and transform structure

One of the major tools that contribute significantly to the coding efficiency of HEVC is the usage of flexible coding blocks and transforms, which are defined in a hierarchical quad-tree manner. Unlike H.264, where the basic coding block is a macroblock of fixed

size 16x16, HEVC defines a Coding Tree Unit (CTU) of a maximum size of 64x64. Each CTU can be divided into smaller units in a hierarchical quad-tree manner and can represent smaller blocks down to size 4x4. Similarly, the transforms used in HEVC can have different sizes, starting from 4x4 and going up to 32x32. Utilizing large blocks and transforms contribute to the major gain of HEVC, especially at high resolutions.

Entropy coding

HEVC uses a single entropy coding engine, which is based on Context Adaptive Binary Arithmetic Coding (CABAC), whereas H.264 uses two distinct entropy coding engines. CABAC in HEVC shares many similarities with CABAC of H.264, but contains several improvements. Those include improvements in coding efficiency and lowered implementation complexity, especially for parallel architectures.

In-loop filtering

H.264 includes an in-loop adaptive deblocking filter, where the blocking artifacts around the transform edges in the reconstructed picture are smoothed to improve the picture quality and compression efficiency. In HEVC, a similar deblocking filter is employed but with somewhat lower complexity. In addition, pictures undergo a subsequent filtering operation called Sample Adaptive Offset (SAO), which is a new design element in HEVC. SAO basically adds a pixel-level offset in an adaptive manner and usually acts as a de-ringing filter. It is observed that SAO improves the picture quality, especially around sharp edges contributing substantially to visual quality improvements of HEVC.

Motion prediction and coding

There have been a number of improvements in this area that are summarized as follows. The first category is motion merge and advanced motion vector prediction (AMVP) modes. The motion information of a prediction block can be inferred from the spatially or temporally neighboring blocks. This is similar to the DIRECT mode in H.264 but includes new aspects to incorporate the flexible quad-tree structure and methods to improve the parallel implementations. In addition, the motion vector predictor can be

signaled for improved efficiency. The second category is high-precision interpolation. The interpolation filter length is increased to 8-tap from 6-tap, which improves the coding efficiency but also comes with increased complexity. In addition, the interpolation filter is defined with higher precision without any intermediate rounding operations to further improve the coding efficiency.

Intra prediction and intra coding

Compared to 8 intra prediction modes in H.264, HEVC supports angular intra prediction with 33 directions. This increased flexibility improves both objective coding efficiency and visual quality as the edges can be better predicted and ringing artifacts around the edges can be reduced. In addition, the reference samples are adaptively smoothed based on the prediction direction. To avoid contouring artifacts a new interpolative prediction generation is included to improve the visual quality. Furthermore, discrete sine transform (DST) is utilized instead of traditional discrete cosine transform (DCT) for 4x4 intra transform blocks.

Other coding-tool features

HEVC includes some tools for lossless coding and efficient screen content coding, such as skipping the transform for certain blocks. These tools are particularly useful for example when streaming the user-interface of a mobile device to a large display.

1.1.2 Systems and Transport Interfaces

HEVC inherited the basic systems and transport interfaces designs, such as the NAL-unit-based syntax structure, the hierarchical syntax and data unit structure from sequence-level parameter sets, multi-picture-level or picture-level parameter sets, slice-level header parameters, lower-level parameters, the supplemental enhancement information (SEI) message mechanism, the hypothetical reference decoder (HRD) based video buffering model, and so on. In the following, a list of differences in these aspects compared to H.264 is summarized.

Video parameter set

A new type of parameter set, called video parameter set (VPS), was introduced. For the first (2013) version of [HEVC], the video parameter set NAL unit is required to be available prior to its activation, while the information contained in the video parameter set is not necessary for operation of the decoding process. For future HEVC extensions, such as the 3D or scalable extensions, the video parameter set is expected to include information necessary for operation of the decoding process, e.g. decoding dependency or information for reference picture set construction of enhancement layers. The VPS provides a "big picture" of a bitstream, 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. (see section 7.1).

Profile, tier and level

The profile, tier and level syntax structure that can be included in both VPS and sequence parameter set (SPS) includes 12 bytes data to describe the entire bitstream (including all temporally scalable layers, which are referred to as sub-layers in the HEVC specification), and can optionally include more profile, tier and level information pertaining to individual temporally scalable layers. The profile indicator indicates the "best viewed as" profile when the bitstream conforms to multiple profiles, similar to the major brand concept in the ISO base media file format (ISOBMFF) [ISOBMFF] and file formats derived based on ISOBMFF, such as the 3GPP file format [3GP]. The profile, tier and level syntax structure also includes the indications of whether the bitstream is free of frame-packed content, whether the bitstream is free of interlaced source content and free of field pictures, i.e. contains only frame pictures of progressive source, such that clients/players with no support of post-processing functionalities for handling of frame-packed or interlaced source content or field pictures can reject those bitstreams.

Bitstream and elementary stream

HEVC includes a definition of an elementary stream, which is new compared to H.264. An elementary stream consists of a sequence of one or more bitstreams. An elementary stream that consists of two or more bitstreams has typically been formed by splicing together two or more bitstreams (or parts thereof). When an elementary stream contains more than one bitstream, the last NAL unit of the last access unit of a bitstream (except the last bitstream in the elementary stream) must contain an end of bitstream NAL unit and the first access unit of the subsequent bitstream must be an intra random access point (IRAP) access unit. This IRAP access unit may be a clean random access (CRA), broken link access (BLA), or instantaneous decoding refresh (IDR) access unit.

Random access support

HEVC includes signaling in NAL unit header, through NAL unit types, of IRAP pictures beyond IDR pictures. Three types of IRAP pictures, namely IDR, CRA and BLA pictures are supported, wherein IDR pictures are conventionally referred to as closed group-of-pictures (closed-GOP) random access points, and CRA and BLA pictures are those conventionally referred to as open-GOP random access points. BLA pictures usually originate from splicing of two bitstreams or part thereof at a CRA picture, e.g. during stream switching. To enable better systems usage of IRAP pictures, altogether six different NAL units are defined to signal the properties of the IRAP pictures, which can be used to better match the stream access point (SAP) types as defined in the ISOBMFF [ISOBMFF], which are utilized for random access support in both 3GP-DASH [3GPDASH] and MPEG DASH [MPEGDASH]. Pictures following an IRAP picture in decoding order and preceding the IRAP picture in output order are referred to as leading pictures associated with the IRAP picture. There are two types of leading pictures, namely random access decodable leading (RADL) pictures and random access skipped leading (RASL) pictures. RADL pictures are decodable when the decoding started at the associated IRAP picture, and RASL pictures are not decodable when the decoding started at the associated IRAP picture and are usually discarded. HEVC provides mechanisms to enable the specification of conformance of bitstreams with RASL pictures being discarded, thus

to provide a standard-compliant way to enable systems components to discard RASL pictures when needed.

Temporal scalability support

HEVC includes an improved support of temporal scalability, by inclusion of the signaling of TemporalId in the NAL unit header, the restriction that pictures of a particular temporal sub-layer cannot be used for inter prediction reference by pictures of a lower temporal sub-layer, 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.

Temporal sub-layer switching support

HEVC specifies, through NAL unit types present in the NAL unit header, the signaling of temporal sub-layer access (TSA) and stepwise temporal sub-layer access (STSA). A TSA picture and pictures following the TSA picture in decoding order do not use pictures prior to the TSA picture in decoding order with TemporalId greater than or equal to that of the TSA picture for inter prediction reference. A TSA picture enables up-switching, at the TSA picture, to the sub-layer containing the TSA picture or any higher sub-layer, from the immediately lower sub-layer. An STSA picture does not use pictures with the same TemporalId as the STSA picture for inter prediction reference. Pictures following an STSA picture in decoding order with the same TemporalId as the STSA picture do not use pictures prior to the STSA picture in decoding order with the same TemporalId as the STSA picture for inter prediction reference. An STSA picture enables up-switching, at the STSA picture, to the sub-layer containing the STSA picture, from the immediately lower sub-layer.

Sub-layer reference or non-reference pictures

The concept and signaling of reference/non-reference pictures in HEVC are different from H.264. In H.264, if a picture may be used by any other picture for inter prediction reference, it is a reference picture; otherwise it is a non-reference picture, and this

is signaled by two bits in the NAL unit header. In HEVC, a picture is called a reference picture only when it is marked as "used for reference". In addition, the concept of sub-layer reference picture was introduced. If a picture may be used by another other picture with the same TemporalId for inter prediction reference, it is a sub-layer reference picture; otherwise it is a sub-layer non-reference picture. Whether a picture is a sub-layer reference picture or sub-layer non-reference picture is signaled through NAL unit type values.

Extensibility

Besides the TemporalId in the NAL unit header, HEVC also includes the signaling of a six-bit layer ID in the NAL unit header, which must be equal to 0 for a single-layer bitstream. Extension mechanisms have been included in VPS, SPS, PPS, SEI NAL unit, slice headers, and so on. All these extension mechanisms enable future extensions in a backward compatible manner, such that bitstreams encoded according to potential future HEVC extensions can be fed to then-legacy decoders (e.g. HEVC version 1 decoders) and the then-legacy decoders can decode and output the base layer bitstream.

Bitstream extraction

HEVC includes a bitstream extraction process as an integral part of the overall decoding process, as well as specification of the use of the bitstream extraction process in description of bitstream conformance tests as part of the hypothetical reference decoder (HRD) specification.

Reference picture management

The reference picture management of HEVC, including reference picture marking and removal from the decoded picture buffer (DPB) as well as reference picture list construction (RPLC), differs from that of H.264. Instead of the sliding window plus adaptive memory management control operation (MMCO) based reference picture marking mechanism in H.264, HEVC specifies a reference picture set (RPS) based reference picture management and marking mechanism, and the RPLC is consequently based on the RPS mechanism. A reference picture set consists of a set of reference pictures associated with

a picture, consisting of all reference pictures that are prior to the associated picture in decoding order, that may be used for inter prediction of the associated picture or any picture following the associated picture in decoding order. The reference picture set consists of five lists of reference pictures; RefPicSetStCurrBefore, RefPicSetStCurrAfter, RefPicSetStFoll, RefPicSetLtCurr and RefPicSetLtFoll. RefPicSetStCurrBefore, RefPicSetStCurrAfter and RefPicSetLtCurr contain all reference pictures that may be used in inter prediction of the current picture and that may be used in inter prediction of one or more of the pictures following the current picture in decoding order. RefPicSetStFoll and RefPicSetLtFoll consist of all reference pictures that are not used in inter prediction of the current picture but may be used in inter prediction of one or more of the pictures following the current picture in decoding order. RPS provides an "intra-coded" signaling of the DPB status, instead of an "inter-coded" signaling, mainly for improved error resilience. The RPLC process in HEVC is based on the RPS, by signaling an index to an RPS subset for each reference index. The RPLC process has been simplified compared to that in H.264, by removal of the reference picture list modification (also referred to as reference picture list reordering) process.

Ultra low delay support

HEVC specifies a sub-picture-level HRD operation, for support of the so-called ultra-low delay. The mechanism specifies a standard-compliant way to enable delay reduction below one picture interval. Sub-picture-level coded picture buffer (CPB) and DPB parameters may be signaled, and utilization of these information for the derivation of CPB timing (wherein the CPB removal time corresponds to decoding time) and DPB output timing (display time) is specified. Decoders are allowed to operate the HRD at the conventional access-unit-level, even when the sub-picture-level HRD parameters are present.

New SEI messages

HEVC inherits many H.264 SEI messages with changes in syntax and/or semantics making them applicable to HEVC. Additionally, there are a few new SEI messages reviewed briefly in the following paragraphs.

The structure of pictures SEI message provides information on the NAL unit types, picture order count values, and prediction dependencies of a sequence of pictures. The SEI message can be used for example for concluding what impact a lost picture has on other pictures.

The decoded picture hash SEI message provides a checksum derived from the sample values of a decoded picture. It can be used for detecting whether a picture was correctly received and decoded.

The active parameter sets SEI message includes the IDs of the active video parameter set and the active sequence parameter set and can be used to activate VPSs and SPSs. In addition, the SEI message includes the following indications: 1) An indication of whether "full random accessibility" is supported (when supported, all parameter sets needed for decoding of the remaining of the bitstream when random accessing from the beginning of the current coded video sequence by completely discarding all access units earlier in decoding order are present in the remaining bitstream and all coded pictures in the remaining bitstream can be correctly decoded); 2) An indication of whether there is no parameter set within the current coded video sequence that updates another parameter set of the same type preceding in decoding order. An update of a parameter set refers to the use of the same parameter set ID but with some other parameters changed. If this property is true for all coded video sequences in the bitstream, then all parameter sets can be sent out-of-band before session start.

The decoding unit information SEI message provides coded picture buffer removal delay information for a decoding unit. The message can be used in very-low-delay buffering operations.

The region refresh information SEI message can be used together with the recovery point SEI message (present in both H.264 and HEVC) for improved support of gradual decoding refresh (GDR). This supports random access from inter-coded pictures, wherein complete pictures can be correctly decoded or recovered after an indicated number of pictures in output/display order.

1.1.3 Parallel Processing Support

The reportedly significantly higher encoding computational demand of HEVC over H.264, in conjunction with the ever increasing video resolution (both spatially and temporally) required by the market, led to the adoption of VCL coding tools specifically targeted to allow for parallelization on the sub-picture level. That is, parallelization occurs, at the minimum, at the granularity of an integer number of CTUs. The targets for this type of high-level parallelization are multicore CPUs and DSPs as well as multiprocessor systems. In a system design, to be useful, these tools require signaling support, which is provided in Section 7 of this memo. This section provides a brief overview of the tools available in [HEVC].

Many of the tools incorporated in HEVC were designed keeping in mind the potential parallel implementations in multi-core/multi-processor architectures. Specifically, for parallelization, four picture partition strategies are available.

Slices are segments of the bitstream that can be reconstructed independently from other slices within the same picture (though there may still be interdependencies through loop filtering operations). Slices are the only tool that can be used for parallelization that is also available, in virtually identical form, in H.264. Slices based parallelization does not require much inter-processor or inter-core communication (except for inter-processor or inter-core data sharing for motion compensation when decoding a predictively coded picture, which is typically much heavier than inter-processor or inter-core data sharing due to in-picture prediction), as slices are designed to be independently decodable. However, for the same reason, slices can require some coding overhead. Further, slices (in contrast to some of the other tools mentioned below) also serve as the key mechanism for bitstream partitioning to match Maximum Transfer Unit (MTU) size requirements, due to the in-picture independence of slices and the fact that each regular slice is encapsulated in its own NAL unit. In many cases, the goal of parallelization and the goal of MTU size matching can place contradicting demands to the slice layout in a picture. The realization of this situation led to the development of the more advanced tools mentioned below. This payload format does not

contain any specific mechanisms aiding parallelization through slices.

Dependent slice segments allow for fragmentation of a coded slice into fragments at CTU boundaries without breaking any in-picture prediction mechanism. They are complementary to the fragmentation mechanism described in this memo in that they need the cooperation of the encoder. As a dependent slice segment necessarily contains an integer number of CTUs, a decoder using multiple cores operating on CTUs can process a dependent slice segment without communicating parts of the slice segment's bitstream to other cores. Fragmentation, as specified in this memo, in contrast, does not guarantee that a fragment contains an integer number of CTUs.

In wavefront parallel processing (WPP), the picture is partitioned into rows of CTUs. Entropy decoding and prediction are allowed to use data from CTUs in other partitions. Parallel processing is possible through parallel decoding of CTU rows, where the start of the decoding of a row is delayed by two CTUs, so to ensure that data related to a CTU above and to the right of the subject CTU is available before the subject CTU is being decoded. Using this staggered start (which appears like a wavefront when represented graphically), parallelization is possible with up to as many processors/cores as the picture contains CTU rows.

Because in-picture prediction between neighboring CTU rows within a picture is allowed, the required inter-processor/inter-core communication to enable in-picture prediction can be substantial. The WPP partitioning does not result in the creation of more NAL units compared to when it is not applied, thus WPP cannot be used for MTU size matching, though slices can be used in combination for that purpose.

Tiles define horizontal and vertical boundaries that partition a picture into tile columns and rows. The scan order of CTUs is changed to be local within a tile (in the order of a CTU raster scan of a tile), before decoding the top-left CTU of the next tile in the order of tile raster scan of a picture. Similar to slices, tiles break in-picture prediction dependencies (including entropy decoding dependencies). However, they do not need to be included into individual NAL units (same as WPP in this regard), hence tiles

cannot be used for MTU size matching, though slices can be used in combination for that purpose. Each tile can be processed by one processor/core, and the inter-processor/inter-core communication required for in-picture prediction between processing units decoding neighboring tiles is limited to conveying the shared slice header in cases a slice is spanning more than one tile, and loop filtering related sharing of reconstructed samples and metadata. Insofar, tiles are less demanding in terms of inter-processor communication bandwidth compared to WPP due to the in-picture independence between two neighboring partitions.

1.1.4 NAL Unit Header

HEVC maintains the NAL unit concept of H.264 with modifications. HEVC 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.

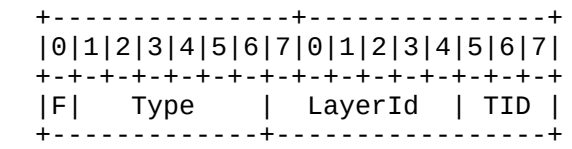


Figure 1 The structure of HEVC NAL unit header

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

F: 1 bit
forbidden_zero_bit. MUST be zero. HEVC declares a value of 1 as a syntax violation. Note that the inclusion of this bit in the NAL unit header is to enable transport of HEVC video over MPEG-2 transport systems (avoidance of start code emulations) [MPEG2S].

Type: 6 bits
nal_unit_type. This field specifies the NAL unit type as defined in Table 7-1 of [HEVC]. If the most significant bit of this

field of a NAL unit is equal to 0 (i.e. the value of this field is less than 32), the NAL unit is a VCL NAL unit. Otherwise, the NAL unit is a non-VCL NAL unit. For a reference of all currently defined NAL unit types and their semantics, please refer to Section 7.4.1 in [HEVC].

LayerId: 6 bits
nuh_layer_id. MUST be equal to zero. It is anticipated that in future scalable or 3D video coding extensions of this specification, this syntax element will be used to identify additional layers that may be present in the coded video sequence, wherein a layer may be, e.g. a spatial scalable layer, a quality scalable layer, a texture view, or a depth view.

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 HEVC coded data over RTP [RFC3550]:

- Usage of RTP header with this payload format
- Packetization of HEVC coded NAL units into RTP packets using three types of payload structures, namely single NAL unit packet, aggregation packet, and fragment unit
- Transmission of HEVC NAL units of the same bitstream within a single RTP stream (note that RTP stream is used equivalently as RTP flow in this memo) or multiple RTP streams
- Media type parameters to be used with the Session Description Protocol (SDP) [RFC4566]

- o A payload header extension mechanism and data structures for enhanced support of temporal scalability based on that extension mechanism.

2. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, RFC 2119 [RFC2119].

In this document, these key words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying the RFC 2119 significance.

This specification uses the notion of setting and clearing a bit when bit fields are handled. Setting a bit is the same as assigning that bit the value of 1 (On). Clearing a bit is the same as assigning that bit the value of 0 (Off).

3. Definitions and Abbreviations

3.1 Definitions

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

3.1.1 Definitions from the HEVC Specification

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

BLA access unit: An access unit in which the coded picture is a BLA picture.

BLA picture: An IRAP picture for which each VCL NAL unit has nal_unit_type equal to BLA_W_LP, BLA_W_RADL, or BLA_N_LP.

coded video sequence: A sequence of access units that consists, in decoding order, of an IRAP access unit with NoRaslOutputFlag equal to 1, followed by zero or more access units that are not IRAP access units with NoRaslOutputFlag equal to 1, including all subsequent access units up to but not including any subsequent access unit that is an IRAP access unit with NoRaslOutputFlag equal to 1.

Informative note: An IRAP access unit may be an IDR access unit, a BLA access unit, or a CRA access unit. The value of NoRaslOutputFlag is equal to 1 for each IDR access unit, each BLA access unit, and each CRA access unit that is the first access unit in the bitstream in decoding order, is the first access unit that follows an end of sequence NAL unit in decoding order, or has HandleCraAsBlaFlag equal to 1.

CRA access unit: An access unit in which the coded picture is a CRA picture.

CRA picture: A RAP picture for which each VCL NAL unit has nal_unit_type equal to CRA_NUT.

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

IDR picture: A RAP picture for which each VCL NAL unit has nal_unit_type equal to IDR_W_RADL or IDR_N_LP.

IRAP access unit: An access unit in which the coded picture is an IRAP picture.

IRAP picture: A coded picture for which each VCL NAL unit has nal_unit_type in the range of BLA_W_LP to RSV_IRAP_VCL23, 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, or one of a set of syntactical structures having a hierarchical relationship.

operation point: bitstream created from another bitstream by operation of the sub-bitstream extraction process with the another bitstream, a target highest TemporalId, and a target layer identifier list as inputs.

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

sub-layer: 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.

tile: A rectangular region of coding tree blocks within a particular tile column and a particular tile row in a picture.

tile column: A rectangular region of coding tree blocks 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 coding tree blocks having a height specified by syntax elements in the picture parameter set and a width equal to the width of the picture.

3.1.2 Definitions Specific to This Memo

dependent RTP stream: An RTP stream in an MST on which another RTP stream depends.

highest RTP stream: The RTP stream in an MST on which no other RTP stream depends.

media aware network element (MANE): A network element, such as a middlebox 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 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].

multi-stream transmission (MST): Transmission of an HEVC bitstream using more than one RTP stream.

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 [HEVC].

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

RTP stream: A sequence of RTP packets with increasing sequence numbers (except for wrap-around), identical PT and identical SSRC (Synchronization Source), carried in one RTP session. Within the scope of this memo, one RTP stream is utilized to transport one or more temporal sub-layers.

single-stream transmission (SST): Transmission of an HEVC bitstream using only one RTP stream.

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

AP	Aggregation Packet
BLA	Broken Link Access
CRA	Clean Random Access
CTB	Coding Tree Block
CTU	Coding Tree Unit
CVS	Coded Video Sequence
FU	Fragmentation Unit

GDR	Gradual Decoding Refresh
HRD	Hypothetical Reference Decoder
IDR	Instantaneous Decoding Refresh
IRAP	Intra Random Access Point
MANE	Media Aware Network Element
MST	Multi-Stream Transmission
MTU	Maximum Transfer Unit
NAL	Network Abstraction Layer
NALU	Network Abstraction Layer Unit
PACI	PAyload Content Information
PHES	Payload Header Extension Structure
PPS	Picture Parameter Set
RADL	Random Access Decodable Leading (Picture)
RASL	Random Access Skipped Leading (Picture)
RPS	Reference Picture Set
SEI	Supplemental Enhancement Information
SPS	Sequence Parameter Set
SST	Single-Stream Transmission
STSA	Step-wise Temporal Sub-layer Access
TSA	Temporal Sub-layer Access
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] and reprinted in 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 Sections 4.7 and 4.8, respectively.

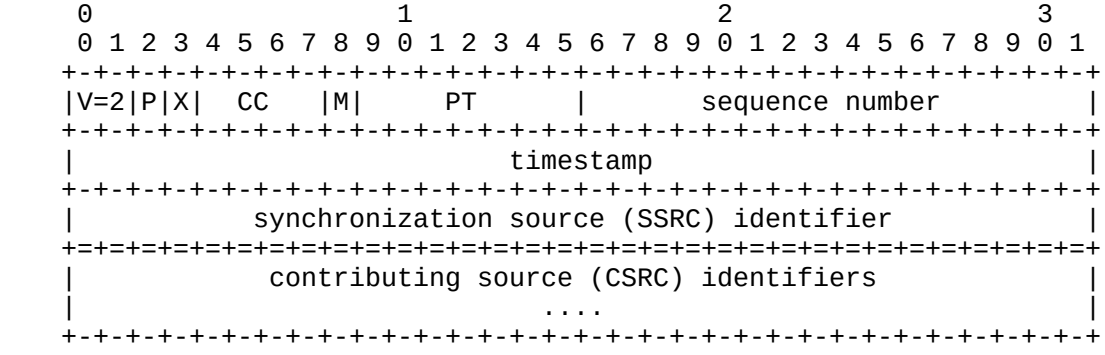


Figure 2 RTP header according to [RFC3550]

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

Marker bit (M): 1 bit

Set for the last packet of the access unit indicated by the RTP timestamp, in line with the normal use of the M bit in video formats, to allow an efficient playout buffer handling. Decoders can use this bit as an early indication of the last packet of an access unit.

Informative note: The content of a NAL unit does not tell whether or not the NAL unit is the last NAL unit, in decoding order, of an access unit. An RTP sender implementation may

obtain this information from the video encoder. If, however, the implementation cannot obtain this information directly from the encoder, e.g. when the stream was pre-encoded, and also there is no timestamp allocated for each NAL unit, then the sender implementation can inspect subsequent NAL units in decoding order to determine whether or not the NAL unit is the last NAL unit of an access unit as follows. A NAL unit naluX is the last NAL unit of an access unit if it is the last NAL unit of the stream or the next VCL NAL unit naluY in decoding order has the high-order bit of the first byte after its NAL unit header equal to 1, and all NAL units between naluX and naluY, when present, have nal_unit_type in the range of 32 to 35, inclusive, equal to 39, or in the ranges of 41 to 44, inclusive, or 48 to 55, inclusive.

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 RFC 3550.

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 is set to the RTP timestamp of the coded picture of the access unit in which the NAL unit is included, according to Section 7.4.2.4.4 of [HEVC].

Receivers SHOULD ignore the picture output timing information in any picture timing SEI messages or decoding unit information SEI messages as specified in [HEVC]. Instead, receivers SHOULD use the RTP timestamp for the display process. Receivers MUST pass

picture timing SEI messages and decoding unit information SEI messages to the decoder and MAY use the field/frame related information for the display process e.g. when frame doubling or frame tripling is indicated by the field/frame related information.

4.2 Payload Header Usage

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

4.3 Payload Structures

The first two bytes of the payload of an RTP packet are referred to as the payload header. In most cases, the payload header consists of the same fields (F, Type, LayerId, and TID) as the NAL unit header as shown in section 1.1.4, irrespective of the type of the payload structure. The single exception is an RTP packet carrying a Payload Content Information (PACI) NAL-unit like structure.

Four 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 four different payload structures are as follows:

- Single NAL unit packet: Contains a single NAL unit in the payload, and the NAL unit header of the NAL unit also serves as the payload header. This payload structure is specified in section 4.6.

- Aggregation packet (AP): Contains more than one NAL unit within one access unit. This payload structure is specified in section 4.7.

- Fragmentation unit (FU): Contains a subset of a single NAL unit. This payload structure is specified in section 4.8.

- o PACI carrying RTP packet: Contains a payload header (that differs from other payload headers for efficiency), a Payload Header Extension Structure (PHES), and a PACI payload. This payload structure is specified in section 4.9.

4.4 Transmission Modes

This memo enables transmission of an HEVC bitstream over a single RTP stream or multiple RTP streams. The concept and working principle is inherited from the design of single and multiple session transmission in [RFC6190] and follows a similar design. If only one RTP stream is used for transmission of the HEVC bitstream, the transmission mode is referred to as single-stream transmission (SST); otherwise (more than one RTP stream is used for transmission of the HEVC bitstream), the transmission mode is referred to as multi-stream transmission (MST).

Dependency of one RTP stream on another RTP stream is indicated as specified in [RFC5583]. In MST, the RTP stream on which on other RTP stream depends is referred to as the highest RTP stream. When an RTP stream A depends on another RTP stream B, the RTP stream B is referred to as a dependent RTP stream of the RTP stream A.

> Informative note: An MST may involve one or more RTP sessions. For example, each RTP stream in an MST may be in its own RTP session. For another example, a set of multiple RTP streams in an MST may belong to the same RTP session, e.g. as indicated by the mechanism specified in [I-D.ietf-avtcore-rtp-multi-stream] or [I-D.ietf-mmusic-sdp-bundle-negotiation].

SST SHOULD be used for point-to-point unicast scenarios, while MST SHOULD be used for point-to-multipoint multicast scenarios where different receivers require different operation points of the same HEVC bitstream, to improve bandwidth utilizing efficiency.

> Informative note: A multicast may degrade to a unicast after all but one receivers have left (this is a justification of the first "SHOULD" instead of "MUST"), and there might be scenarios where MST is desirable but not possible e.g. when IP multicast is not deployed in certain network (this is a justification of the second "SHOULD" instead of "MUST").

Receivers MUST support both SST and MST.

4.5 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-depack-buf-nalus is equal to 0, AbsDon[n], the value of AbsDon for NAL unit n, is derived as equal to n.

Otherwise (sprop-depack-buf-nalus is greater than 0), AbsDon[n] is derived as follows, where DON[n] is the value of the variable DON for NAL unit n:

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

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

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

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

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

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

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

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

- AbsDon[n] greater than AbsDon[m] indicates that NAL unit n follows NAL unit m in NAL unit decoding order.

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

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

When two consecutive NAL units in the NAL unit decoding order have different values of AbsDon, the value of AbsDon for the second NAL unit in decoding order MUST be greater than the value of AbsDon for the first NAL unit, and 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 may not forward VCL NAL units of higher sub-layers or some SEI NAL units when there is congestion in the network. In another example, the first intra 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 picture, the originator does not exactly know how many NAL units will be encoded before the first intra picture of the pre-encoded clip follows in decoding order. Thus, the values of AbsDon for the NAL units of the first intra picture of the pre-encoded clip have to be estimated when they are transmitted, and gaps in values of AbsDon may occur. Another example is MST where the AbsDon values must indicate cross-layer decoding order for NAL units conveyed in all the RTP streams.

4.6 Single NAL Unit Packets

A single NAL unit packet contains exactly one NAL unit, and consists of a payload header (denoted as PayloadHdr), an optional 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.

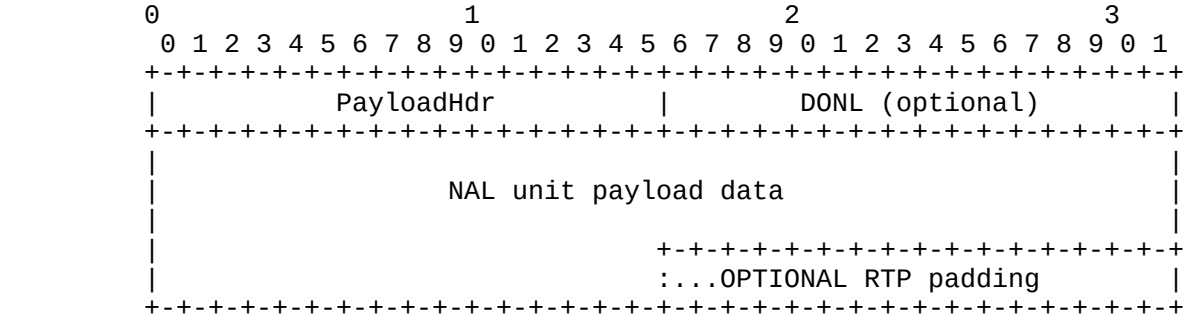


Figure 3 The structure a single NAL unit packet

The payload header SHOULD be an exact copy of the NAL unit header of the contained NAL unit. However, the Type (i.e. nal_unit_type) field MAY be changed, e.g. when it is desirable to handle a CRA picture to be a BLA picture [JCTVC-J0107].

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

If sprop-depack-buf-nalus is greater than 0, the DONL field MUST be present, and the variable DON for the contained NAL unit is derived as equal to the value of the DONL field. Otherwise (sprop-depack-buf-nalus is equal to 0), the DONL field MUST NOT be present.

4.7 Aggregation Packets (APs)

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

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

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

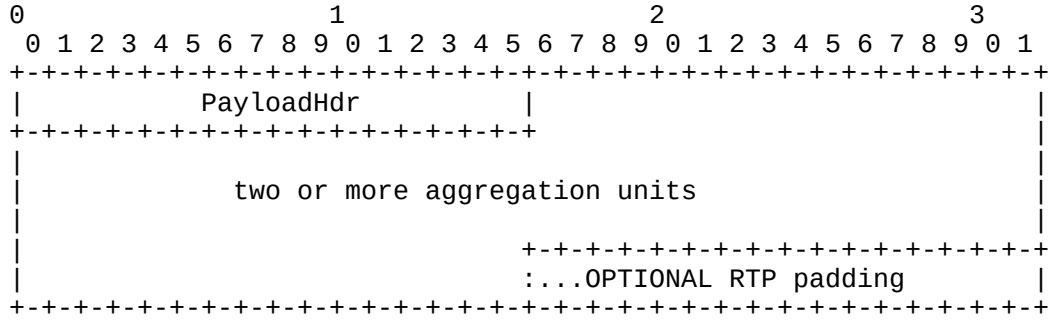


Figure 4 The structure of an aggregation packet

The fields in the payload header are set as follows. The F bit MUST be equal to 0 if the F bit of each aggregated NAL unit is equal to zero; otherwise, it MUST be equal to 1. The Type field MUST be equal to 48. The value of LayerId MUST be equal to the lowest value of LayerId of all the aggregated NAL units. The value of TID MUST be the lowest value of TID of all the aggregated NAL units.

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

An AP MUST carry at least two aggregation units and can carry as many aggregation units as necessary; however, the total amount of data in an AP obviously MUST fit into an IP packet, and the size SHOULD be chosen so that the resulting IP packet is smaller than the MTU size so to avoid IP layer fragmentation. An AP MUST NOT contain Fragmentation Units (FUs) specified in section 4.8. APs MUST NOT be nested; i.e. an AP MUST NOT contain another AP.

The first aggregation unit in an AP consists of an optional 16-bit DONL field (in network byte order) followed by a 16-bit unsigned size information (in network byte order) that indicates the size of

the NAL unit in bytes (excluding these two octets, but including the NAL unit header), followed by the NAL unit itself, including its NAL unit header, as shown in Figure 5.

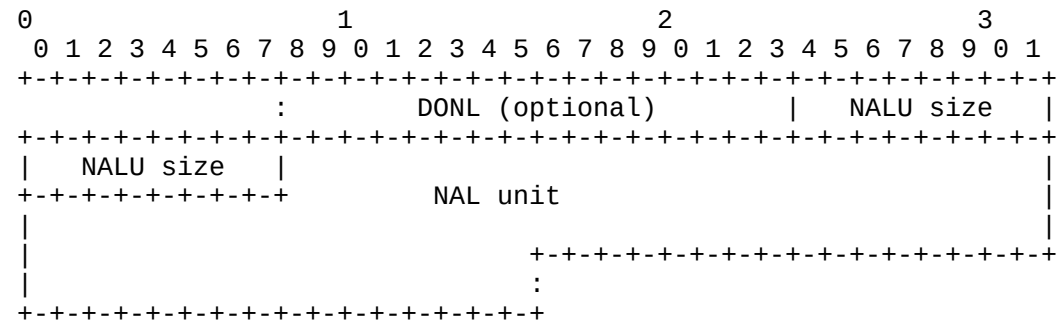


Figure 5 The structure of the first aggregation unit in an AP

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-depack-buf-nalus 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. Otherwise (sprop-depack-buf-nalus 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 consists of an optional 8-bit DOND field followed by a 16-bit unsigned size information (in network byte order) that indicates the size of the NAL unit in bytes (excluding these two octets, but including the NAL unit header), followed by the NAL unit itself, including its NAL unit header, as shown in Figure 6.

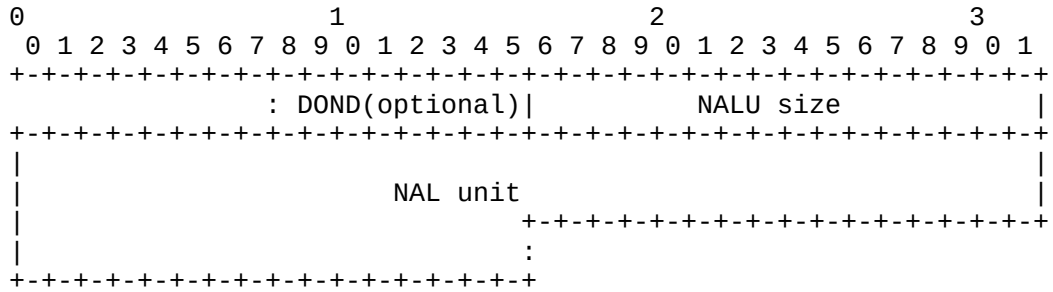


Figure 6 The structure of an aggregation unit that is not the first aggregation unit in an AP

When present, the DOND field plus 1 specifies the difference between the decoding order number values of the current aggregated NAL unit and the preceding aggregated NAL unit in the same AP.

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

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

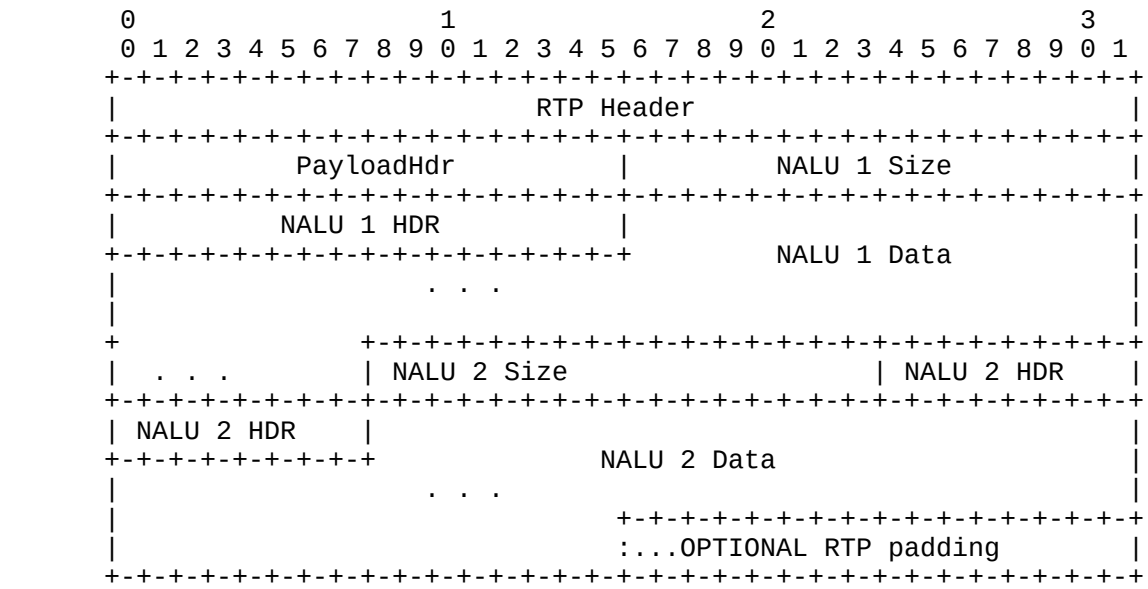


Figure 7 An example of an AP packet containing two aggregation units without the DONL and DOND fields

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

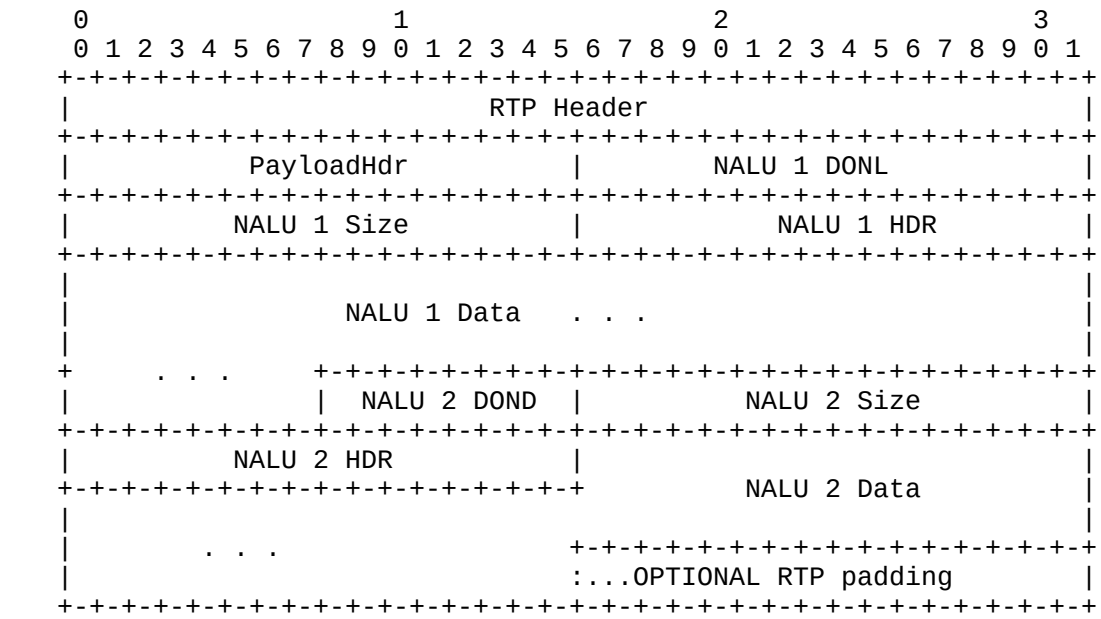


Figure 8 An example of an AP containing two aggregation units with the DONL and DOND fields

4.8 Fragmentation Units (FUs)

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

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

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

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

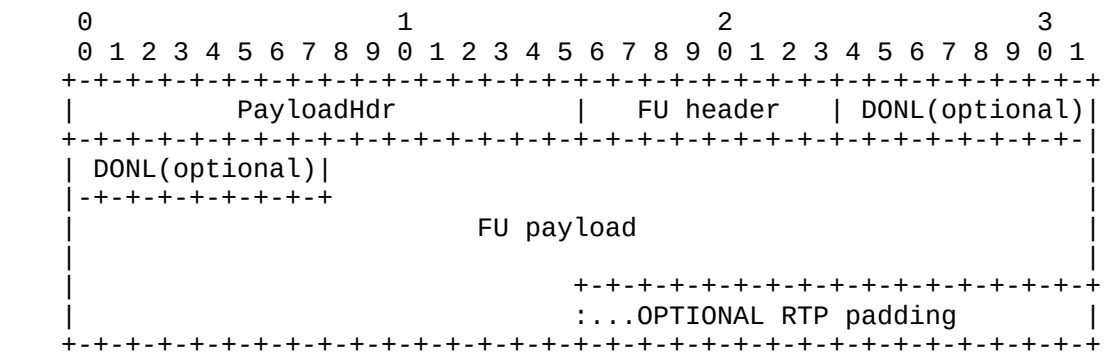


Figure 9 The structure of an FU

The fields in the payload header are set as follows. The Type field MUST be equal to 49. 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, and a 6-bit FuType field, as shown in Figure 10.

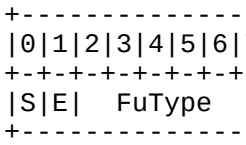


Figure 10 The structure of FU header

The semantics of the FU header fields are as follows:

S: 1 bit
When set to one, 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 zero.

E: 1 bit
When set to one, 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 zero.

FuType: 6 bits
The field FuType MUST be equal to the field Type of the fragmented NAL unit.

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

If sprop-depack-buf-nalus 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-depack-buf-nalus 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 one 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 MAY have any number of octets and MAY be empty.

Informative note: Empty FU payloads are allowed to reduce the latency of a certain class of senders in nearly lossless environments. These senders can be characterized in that they packetize fragments of a NAL unit before the NAL unit is completely generated and, hence, before the NAL unit size is known. If zero-length FU payloads were not allowed, the sender would have to generate at least one bit of data of the following fragment of the NAL unit before the current FU could be sent. Due to the characteristics of HEVC, where sometimes several CTUs occupy zero bits, this is undesirable and can add delay. However, the (potential) use of zero-length FU payloads should be carefully weighted against the increased risk of the loss of at least a part of the fragmented NAL unit because of the additional packets employed for its transmission.

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 one to indicate a syntax violation.

4.9 PACI packets

This section specifies the PACI packet structure, based on a payload header extension mechanism that is generic and extensible to carry payload header extensions.

The structure of an RTP packet carrying a Payload Header Extension Structure (PHES) and a PACI payload is as follows:

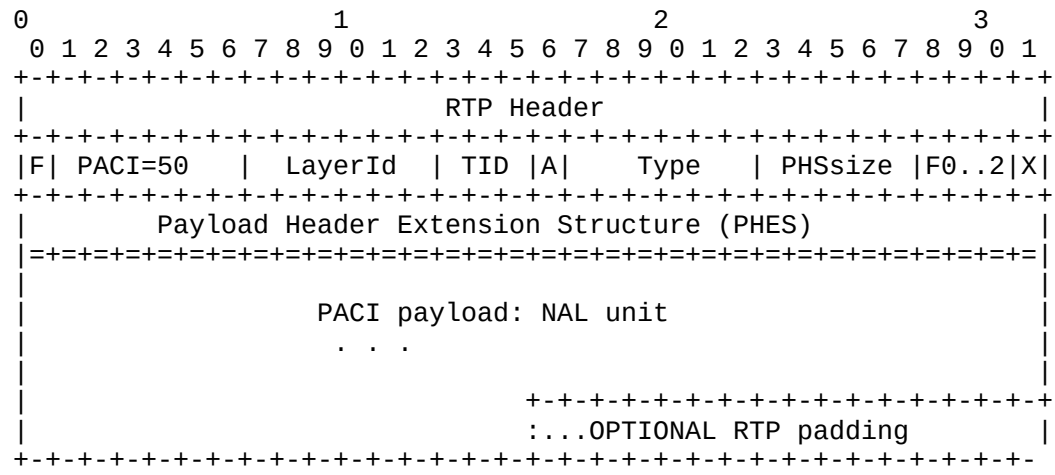


Figure 11 The structure of a PACI

The semantics of the fields are as follows:

F: 1 bit
Forbidden_zero-bit. MUST be zero.

PACI: 6 bits
Indicates a PACI, and must be 50.

LayerId: 6 bits
Copy of the LayerId field of the PACI payload NAL unit or NAL unit like structure

TID: 3 bits
Copy of the TID field of the PACI payload NAL unit or NAL unit like structure

A: 1 bit
Copy of the F bit of the PACI payload NAL unit or NAL unit like structure

Type: 6 bits
Copy of the Type field of the PACI payload NAL unit or NAL unit like structure

PHSsize: 5 bits
Indicates the total length of the PHES. The value is limited to be less than or equal to 32 octets, to simplify encoder design for MTU size matching.

F0..2: 3 bits
Each of the three bits indicate, when set, the presence of an optional field (or set of fields) in the PHES.

X: 1 bit
The X bit, when set, indicates the presence of another octet consisting of seven flags and another X bit, each of the seven flags indicating the presence of more PHES fields (for future extensions).

PHES: variable number of octets
A variable number of octets as indicated by the value of PHSsize.

PACI Payload
The NAL unit or NAL unit like structure (such as: FU or AP) to be carried, not including the first two octets.

Informative note: The first two octets of the NAL unit or NAL unit like structure carried in the PACI payload are not included in the PACI payload. Rather, the respective values are copied in locations of the PayloadHdr of the RTP packet. This design offers two advantages: first, the overall structure of the payload header is preserved, i.e. there is no special case of payload header structure that needs to be implemented for PACI. Second, no additional overhead is introduced.

A PACI payload MAY be a single NAL unit, an FU, or an AP. PACIs MUST NOT be fragmented or aggregated. The following subsection documents the reasons for these design choices.

4.9.1 Reasons for the PACI rules (informative)

A PACI cannot be fragmented. If a PACI could be fragmented, and a fragment other than the first fragment would get lost, access to the information in the PACI would not be possible. Therefore, a PACI must not be fragmented. In other words, an FU must not carry (fragments of) a PACI.

A PACI cannot be aggregated. Aggregation of PACIs is inadvisable from a compression viewpoint, as, in many cases, several to be aggregated NAL units would share identical PACI fields and values which would be carried redundantly for no reason. Most, if not all the practical effects of PACI aggregation can be achieved by aggregating NAL units and bundling them with a PACI (see below). Therefore, a PACI must not be aggregated. In other words, an AP must not contain a PACI.

The payload of a PACI can be a fragment. Both middleboxes and sending systems with inflexible (often hardware-based) encoders occasionally find themselves in situations where a PACI and its headers, combined, are larger than the MTU size. In such a scenario, the middlebox or sender can fragment the NAL unit and encapsulate the fragment in a PACI. Doing so preserves the payload header extension information for all fragments, allowing downstream middleboxes and the receiver to take advantage of that information. Therefore, a sender may place a fragment into a PACI, and a receiver must be able to handle such a PACI.

The payload of a PACI can be an aggregation NAL unit. HEVC bitstreams can contain unevenly sized and/or small (when compared to the MTU size) NAL units. In order to efficiently packetize such small NAL units, AP were introduced. The benefits of APs are independent from the need for a payload header extension. Therefore, a sender may place an AP into a PACI, and a receiver must be able to handle such a PACI.

4.10 Payload Header Extensions

This section describes the single payload header extension defined in this specification. If, in the future, additional payload header extensions become necessary, they could be specified in this section of an updated version of this document, or in their own documents.

When bit 0 of the field F0..2 is set to 1 in a PACI, this indicates the presence of the temporal scalability information fields TL0REFIDX, IrapPicID, S, and E as follows:

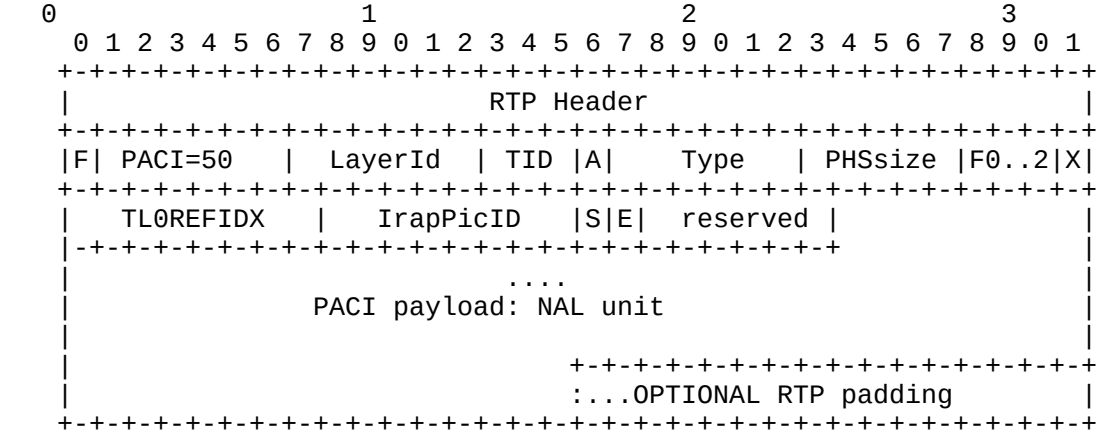


Figure 12 The structure of a PACI with a PHES containing some temporal scalability information

TL0PICIDX (8 bits)
When present, the TL0PICIDX field MUST be set to equal to temporal_sub_layer_zero_idx as specified in Section D.3.32 of [H.265] for the access unit containing the NAL unit in the PACI.

IrapPicID (8 bits)
When present, the IrapPicID field MUST be set to equal to irap_pic_id as specified in Section D.3.32 of [H.265] for the access unit containing the NAL unit in the PACI.

S (1 bit)
The S bit MUST be set to 1 if any of the following conditions is true and MUST be set to 0 otherwise:

- The NAL unit in the payload of the PACI is the first VCL NAL unit, in decoding order, of a picture.
- The NAL unit in the payload of the PACI is an AP and the NAL unit in the first contained aggregation unit is the first VCL NAL unit, in decoding order, of a picture.
- The NAL unit in the payload of the PACI is an FU with its S bit equal to 1 and the FU payload containing a fragment of the first VCL NAL unit, in decoding order of a picture.

E (1 bit)
The E bit MUST be set to 1 if any of the following conditions is true and MUST be set to 0 otherwise:

- The NAL unit in the payload of the PACI is the last VCL NAL unit, in decoding order, of a picture.
- The NAL unit in the payload of the PACI is an AP and the NAL unit in the last contained aggregation unit is the last VCL NAL unit, in decoding order, of a picture.
- The NAL unit in the payload of the PACI is an FU with its E bit equal to 1 and the FU payload containing a fragment of the last VCL NAL unit, in decoding order of a picture.

The values of bits 1 and 2 of the field F0..2 MUST be set to 0, the value of the X bit MUST be set to 0, and the value of PHSsize MUST be set to 3. Receivers SHALL allow other values of the fields F0..2, X, and PHSsize, and SHALL any ignore additional fields, when present, than specified above in the PHES.

5. Packetization Rules

The following packetization rules apply:

- If sprop-depack-buf-nalus is greater than 0 for an RTP stream, the transmission order of NAL units carried in the RTP stream MAY be different than the NAL unit decoding order. Otherwise (sprop-depack-buf-nalus is equal to 0 for an RTP stream), the transmission order of NAL units carried in the RTP stream MUST be the same as the NAL unit decoding order.

- A NAL unit of a small size SHOULD be encapsulated in an aggregation packet together with one or more other NAL units in order to avoid the unnecessary packetization overhead for small NAL units. For example, non-VCL NAL units such as access unit delimiters, parameter sets, or SEI NAL units are typically small and can often be aggregated with VCL NAL units without violating MTU size constraints.

- Each non-VCL NAL unit SHOULD be encapsulated in an aggregation packet together with its associated VCL NAL unit, as typically a non-VCL NAL unit would be meaningless without the associated VCL NAL unit being available.

- For carrying exactly one NAL unit in an RTP packet, a single NAL unit packet MUST be used.

6. De-packetization Process

The general concept behind de-packetization is to get the NAL units out of the RTP packets in an RTP stream and all the dependent RTP streams, if any, 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 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 47, inclusive may be passed to the decoder. NAL-unit-like structures with NAL unit type values in the range of 48 to 63, inclusive, MUST NOT be passed to the decoder.

The receiver includes a receiver buffer, which is used to compensate for transmission delay jitter, to reorder NAL units from transmission order to the NAL unit decoding order, and to recover the NAL unit decoding order in MST, when applicable. In this section, the receiver operation is described under the assumption that there is no transmission delay jitter. To make a difference from a practical receiver buffer that is also used for compensation of transmission delay jitter, the receiver buffer is here after called the de-packetization buffer in this section. Receivers SHOULD also prepare for transmission delay jitter; i.e. 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.

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. When MST is in use, NAL units of all RTP streams are stored in the same de-packetization buffer.

Initial buffering lasts until condition A (the number of NAL units in the de-packetization buffer is greater than the value of sprop-depack-buf-nalus of the highest RTP stream) is true.

After initial buffering, whenever condition A is true, the following operation is repeatedly applied until condition A becomes false:

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

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

7. Payload Format Parameters

This section specifies the parameters that MAY be used to select optional features of the payload format and certain features or properties of the bitstream. The parameters are specified here as part of the media type registration for the HEVC codec. A mapping of the parameters into the Session Description Protocol (SDP) [RFC4566] is also provided for applications that use SDP. Equivalent parameters could be defined elsewhere for use with control protocols that do not use SDP.

7.1 Media Type Registration

The media subtype for the HEVC codec is allocated from the IETF tree.

The receiver MUST ignore any unspecified parameter.

Media Type name: video

Media subtype name: H265

Required parameters: none

OPTIONAL parameters:

In the following definitions of parameters, "the stream" or "the NAL unit stream" refers to all NAL units conveyed in the current RTP stream in SST, and all NAL units conveyed in the current RTP stream and all NAL units conveyed in other RTP streams that the current RTP stream depends on in MST.

profile-space, profile-id:

The profile-space parameter indicates the context for interpretation of the profile-id parameter value. The profile, which specifies the subset of coding tools that may have been used to generate the stream or that the receiver supports, as specified in [HEVC], is defined by the combination of profile-space and profile-id. Note that profile-space is required to be equal to 0 in [HEVC], but other values for it may be specified in the future by ITU-T or ISO/IEC.

If the profile-space and profile-id parameters are used to indicate properties of a NAL unit stream, it indicates that, to decode the stream, the minimum subset of coding tools a decoder has to support is the profile specified by both parameters.

If the profile-space and profile-id parameters are used for capability exchange or session setup, it indicates the subset of coding tools, which is equal to the profile, that the codec supports for both receiving and sending.

If no profile-space is present, a value of 0 MUST be inferred and if no profile-id is present the Main profile (i.e. a value of 1) MUST be inferred.

When used to indicate properties of a NAL unit stream, the profile-space and profile-id parameters are derived from the sequence parameter set or video parameter set NAL units, as specified in [HEVC], as follows.

If the RTP stream is not a dependent RTP stream, the following applies:

- o profile_space = general_profile_space
- o profile_id = general_profile_idc

Otherwise (the RTP stream is a dependent RTP stream), the following applies, with j being the value of the sub-layer-id parameter:

- o profile_space = sub_layer_profile_space[j]
- o profile_id = sub_layer_profile_idc[j]

tier-flag, level-id:

The tier-flag parameter indicates the context for interpretation of the level-id value. The default level, which limits values of syntax elements or on arithmetic combinations of values of syntax elements, as specified in [HEVC], is defined by the combination of tier-flag and level-id.

If the tier-flag and level-id parameters are used to indicate properties of a NAL unit stream, it indicates that, to decode the stream the lowest level the decoder has to support is the default level.

If the tier-flag and level-id parameters are used for capability exchange or session setup, the following applies. If max-recv-level-id is not present, the default level defined by tier-flag and level-id indicates the highest level the codec wishes to support. Otherwise, tier-flag and max-recv-level-id indicate 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 and if no level-id is present, a value of 93 (i.e. level 3.1) MUST be inferred.

When used to indicate properties of a NAL unit stream, the tier-flag and level-id parameters are derived from the

sequence parameter set or video parameter set NAL units, as specified in [HEVC], as follows.

If the RTP stream is not a dependent RTP stream, the following applies:

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

Otherwise (the RTP stream is a dependent RTP stream), the following applies, with j being the value of the sub-layer-id parameter:

- o tier-flag = sub_layer_tier_flag[j]
- o level-id = sub_layer_level_idc[j]

interop-constraints:

A base16 [RFC4648] (hexadecimal) representation of the six bytes derived from the sequence parameter set or video parameter set NAL units as specified in [HEVC] consisting of progressive_source_flag, interlaced_source_flag, non_packed_constraint_flag, frame_only_constraint_flag, and reserved_zero_44bits. Note that reserved_zero_44bits is required to be equal to 0 in [HEVC], but other values for it may be specified in the future by ITU-T or ISO/IEC.

If no interop-constraints are present, the following MUST be inferred:

- o progressive_source_flag = 1
- o interlaced_source_flag = 0
- o non_packed_constraint_flag = 1
- o frame_only_constraint_flag = 1
- o reserved_zero_44bits = 0

When used to indicate properties of a NAL unit stream, the following applies.

If the RTP stream is not a dependent RTP stream, the following applies:

- o progressive_source_flag = general_progressive_source_flag
- o interlaced_source_flag = general_interlaced_source_flag
- o non_packed_constraint_flag = general_non_packed_constraint_flag
- o frame_only_constraint_flag = general_frame_only_constraint_flag
- o reserved_zero_44bits = general_reserved_zero_44bits

Otherwise (the RTP stream is a dependent RTP stream), the following applies, with j being the value of the sub-layer-id parameter:

- o progressive_source_flag = sub_layer_progressive_source_flag[j]
- o interlaced_source_flag = sub_layer_interlaced_source_flag[j]
- o non_packed_constraint_flag = sub_layer_non_packed_constraint_flag[j]
- o frame_only_constraint_flag = sub_layer_frame_only_constraint_flag[j]
- o reserved_zero_44bits = sub_layer_reserved_zero_44bits[j]

profile-compatibility-indicator:

A base16 [RFC4648] representation of the four bytes representing the 32 profile compatibility flags in the sequence parameter set or video parameter set NAL units. A decoder conforming to a certain profile may be able to decode bitstreams conforming to other profiles. The profile-compatibility-indicator provides exact information of the ability of a decoder conforming to a certain profile to decode bitstreams conforming to another profile. More concretely, if the profile compatibility flag corresponding to the profile a decoder conforms to is set, then the decoder is able to decode any bitstream with the flag set, irrespective of the profile the bitstream conforms to (provided that the decoder supports the highest level of the bitstream).

When used to indicate properties of a NAL unit stream, the following applies.

If the RTP stream is not a dependent RTP stream, the following applies with j = 0..31:

o The 32 flags = general_profile_compatibility_flag[j]

Otherwise (the RTP stream is a dependent RTP stream), the following applies with i being the value of the sub-layer-id parameter and j = 0..31:

o The 32 flags = sub_layer_profile_compatibility_flag[i][j]

sub-layer-id:

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

recv-sub-layer-id:

This parameter MAY be used to signal a receiver’s choice of the offered or declared sub-layers in the sprop-vps. The value of recv-sub-layer-id indicates the TID of the highest sub-layer of the stream that a receiver supports. When not present, the value of recv-sub-layer-id is inferred to be equal to sub-layer-id.

max-recv-level-id:

This parameter MAY be used, together with tier-flag, to indicate the highest level a receiver supports. The highest level the receiver supports is equal to the value of max-recv-level-id divided by 30 for the Main or High tier (as determined by tier-flag equal to 0 or 1, respectively).

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-vps:

This parameter MAY be used to convey any video parameter set NAL unit of the stream. When present, the parameter MAY be used to indicate codec capability and sub-stream characteristics (i.e. properties of sub-layer representations as defined in [HEVC]) as well as for out-of-band transmission of video parameter sets. 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 7.3.2.1 of [HEVC].

sprop-sps:

This parameter MAY be used to convey sequence parameter set NAL units of the stream for out-of-band transmission of sequence parameter sets. The value of the parameter is a comma-separated (',') list of base64 [RFC4648] representations of the sequence parameter set NAL units as specified in Section 7.3.2.2 of [HEVC].

sprop-pps:

This parameter MAY be used to convey picture parameter set NAL units of the stream for out-of-band transmission of picture parameter sets. The value of the parameter is a comma-separated (',') list of base64 [RFC4648] representations of the picture parameter set NAL units as specified in Section 7.3.2.3 of [HEVC].

max-lsr, max-lps, max-cpb, max-dpb, max-br, max-tr, max-tc:

These parameters MAY be used to signal the capabilities of a receiver implementation. These parameters MUST NOT be used for any other purpose. The highest level (specified by tier-flag and max-recv-level-id) MUST be such that the receiver is fully capable of supporting. max-lsr, max-lps, max-cpb, max-dpb, max-br, max-tr, and max-tc MAY be used to indicate capabilities of the receiver that extend the required capabilities of the highest level, as specified below.

When more than one parameter from the set (max-lsr, max-lps, max-cpb, max-dpb, max-br, max-tr, max-tc) is present, the

receiver MUST support all signaled capabilities simultaneously. For example, if both max-lsr and max-br are present, the highest level with the extension of both the picture rate and bitrate is supported. That is, the receiver is able to decode NAL unit streams in which the luma sample rate is up to max-lsr (inclusive), the bitrate is up to max-br (inclusive), the coded picture buffer size is derived as specified in the semantics of the max-br parameter below, and the other properties comply with the highest level specified by tier-flag and max-recv-level-id.

Informative note: When the OPTIONAL media type parameters are used to signal the properties of a NAL unit stream, and max-lsr, max-lps, max-cpb, max-dpb, max-br, max-tr, and max-tc are not present, the values of profile-space, profile-id, tier-flag, and level-id must always be such that the NAL unit stream complies fully with the specified profile and level.

max-lsr:
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.

When max-lsr is signaled, the receiver MUST be able to decode NAL unit streams that conform to the highest level, with the exception that the MaxLumaSR value in Table A-2 of [HEVC] for the highest level is replaced with the value of max-lsr. The value of max-lsr MUST be greater than or equal to the value of MaxLumaSR given in Table A-2 of [HEVC] for the highest level. 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 A-2 of [HEVC] for the highest level.

max-lps:
The value of max-lps is an integer indicating the maximum

picture size in units of luma samples. The max-lps parameter signals that the receiver is capable of decoding larger picture sizes than are required by the highest level. When max-lps is signaled, the receiver MUST be able to decode NAL unit streams that conform to the highest level, with the exception that the MaxLumaPS value in Table A-1 of [HEVC] for the highest level is replaced with the value of max-lps. The value of max-lps MUST be greater than or equal to the value of MaxLumaPS given in Table A-1 of [HEVC] for the highest level. Senders MAY use this knowledge to send larger pictures at a proportionally lower picture rate than is indicated in the highest level.

When not present, the value of max-lps is inferred to be equal to the value of MaxLumaPS given in Table A-1 of [HEVC] for the highest level.

max-cpb:
The value of max-cpb is an integer indicating the maximum coded picture buffer size in units of CpbBrVclFactor bits for the VCL HRD parameters and in units of CpbBrNalFactor bits for the NAL HRD parameters, where CpbBrVclFactor and CpbBrNalFactor are defined in Section A.4 of [HEVC]. The max-cpb parameter signals that the receiver has more memory than the minimum amount of coded picture buffer memory required by the highest level. When max-cpb is signaled, the receiver MUST be able to decode NAL unit streams that conform to the highest level, with the exception that the MaxCPB value in Table A-1 of [HEVC] for the highest level is replaced with the value of max-cpb. The value of max-cpb MUST be greater than or equal to the value of MaxCPB given in Table A-1 of [HEVC] for the highest level. Senders MAY use this knowledge to construct coded video streams with greater variation of bitrate than can be achieved with the MaxCPB value in Table A-1 of [HEVC].

When not present, the value of max-cpb is inferred to be equal to the value of MaxCPB given in Table A-1 of [HEVC] for the highest level.

Informative note: The coded picture buffer is used in the hypothetical reference decoder (Annex C of HEVC). The use of the hypothetical reference decoder is recommended in HEVC encoders to verify that the produced bitstream conforms to the standard and to control the output bitrate. Thus, the coded picture buffer is conceptually independent of any other potential buffers in the receiver, including de-packetization and de-jitter buffers. The coded picture buffer need not be implemented in decoders as specified in Annex C of HEVC, but rather standard-compliant decoders can have any buffering arrangements provided that they can decode standard-compliant bitstreams. Thus, in practice, the input buffer for a video decoder can be integrated with de-packetization and de-jitter buffers of the receiver.

max-dpb:

The value of max-dpb is an integer indicating the maximum decoded picture buffer size in units decoded pictures at the MaxLumaPS for the highest level, i.e. number of decoded pictures at the maximum picture size defined by the highest level. The value of max-dpb MUST be smaller than or equal to 16. The max-dpb parameter signals that the receiver has more memory than the minimum amount of decoded picture buffer memory required by default, which is MaxDpbPicBuf as defined in [HEVC] (equal to 6). When max-dpb is signaled, the receiver MUST be able to decode NAL unit streams that conform to the highest level, with the exception that the MaxDpbPicBuff value defined in [HEVC] as 6 is replaced with the value of max-dpb. Consequently, a receiver that signals max-dpb MUST be capable of storing the following number of decoded pictures (MaxDpbSize) in its decoded picture buffer:

```
                  if( PicSizeInSamplesY <= ( MaxLumaPS >> 2 ) )
     MaxDpbSize = Min( 4 * max-dpb, 16 )
   else if ( PicSizeInSamplesY <= ( MaxLumaPS >> 1 ) )
     MaxDpbSize = Min( 2 * max-dpb, 16 )
   else if ( PicSizeInSamplesY <= ( ( 3 * MaxLumaPS ) >> 2 ) )
     MaxDpbSize = Min( (4 * max-dpb) / 3, 16 )
   else
     MaxDpbSize = max-dpb
```

Wherein MaxLumaPS given in Table A-1 of [HEVC] for the highest level and PicSizeInSamplesY is the current size of each decoded picture in units of luma samples as defined in [HEVC].

The value of max-dpb MUST be greater than or equal to the value of MaxDpbPicBuf (i.e. 6) as defined in [HEVC]. Senders MAY use this knowledge to construct coded video streams with improved compression.

When not present, the value of max-dpb is inferred to be equal to the value of MaxDpbPicBuf (i.e. 6) as defined in [HEVC].

Informative note: This parameter was added primarily to complement a similar codepoint in the ITU-T Recommendation H.245, so as to facilitate signaling gateway designs. The decoded picture buffer stores reconstructed samples. There is no relationship between the size of the decoded picture buffer and the buffers used in RTP, especially de-packetization and de-jitter buffers.

max-br:
The value of max-br is an integer indicating the maximum video bitrate in units of CpbBrVclFactor bits per second for the VCL HRD parameters and in units of CpbBrNalFactor bits per second for the NAL HRD parameters, where CpbBrVclFactor and CpbBrNalFactor are defined in Section A.4 of [HEVC].

The max-br parameter signals that the video decoder of the receiver is capable of decoding video at a higher bitrate than is required by the highest level.

When max-br is signaled, the video codec of the receiver MUST be able to decode NAL unit streams that conform to the highest level, with the following exceptions in the limits specified by the highest level:

- The value of max-br replaces the MaxBR value in Table A-2 of [HEVC] for the highest level.
- When the max-cpb parameter is not present, the result of the following formula replaces the value of MaxCPB in Table A-1 of [HEVC]:

(MaxCPB of the highest level) * max-br / (MaxBR of the highest level)

For example, if a receiver signals capability for Main profile Level 2 with max-br equal to 2000, this indicates a maximum video bitrate of 2000 kbits/sec for VCL HRD parameters, a maximum video bitrate of 2200 kbits/sec for NAL HRD parameters, and a CPB size of 2000000 bits (2000000 / 1500000 * 1500000).

The value of max-br MUST be greater than or equal to the value MaxBR given in Table A-2 of [HEVC] for the highest level.

Senders MAY use this knowledge to send higher bitrate video as allowed in the level definition of Annex A of HEVC to achieve improved video quality.

When not present, the value of max-br is inferred to be equal to the value of MaxBR given in Table A-2 of [HEVC] for the highest level.

Informative note: This parameter was added primarily to complement a similar codepoint in the ITU-T Recommendation H.245, so as to facilitate signaling gateway designs. The assumption that the network is capable of handling such bitrates at any given time cannot be made from the value of this parameter. In particular, no conclusion can be drawn that the signaled bitrate is possible under congestion control constraints.

max-tr:
The value of max-tr is an integer indication the maximum number of tile rows. The max-tr parameter signals that the receiver is capable of decoding video with a larger number of tile rows than the value allowed by the highest level.

When max-tr is signaled, the receiver MUST be able to decode NAL unit streams that conform to the highest level, with the exception that the MaxTileRows value in Table A-1 of [HEVC] for the highest level is replaced with the value of max-tr.

The value of max-tr MUST be greater than or equal to the value of MaxTileRows given in Table A-1 of [HEVC] for the highest level. Senders MAY use this knowledge to send pictures utilizing a larger number of tile rows than the value allowed by the highest level.

When not present, the value of max-tr is inferred to be equal to the value of MaxTileRows given in Table A-1 of [HEVC] for the highest level.

max-tc:

The value of max-tc is an integer indication the maximum number of tile columns. The max-tc parameter signals that the receiver is capable of decoding video with a larger number of tile columns than the value allowed by the highest level.

When max-tc is signaled, the receiver MUST be able to decode NAL unit streams that conform to the highest level, with the exception that the MaxTileCols value in Table A-1 of [HEVC] for the highest level is replaced with the value of max-tc.

The value of max-tc MUST be greater than or equal to the value of MaxTileCols given in Table A-1 of [HEVC] for the highest level. Senders MAY use this knowledge to send pictures utilizing a larger number of tile columns than the value allowed by the highest level.

When not present, the value of max-tc is inferred to be equal to the value of MaxTileCols given in Table A-1 of [HEVC] for the highest level.

max-fps:

The value of max-fps is an integer indicating the maximum picture rate in units of hundreds of pictures per second that can be efficiently received. The max-fps parameter MAY be used to signal that the receiver has a constraint in that it is not capable of decoding video efficiently at the full picture rate that is implied by the highest level and, when present, one or more of the parameters max-lsr, max-lps, and max-br.

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, max-lps, max-cpb, max-dpb, max-br, max-tr, and max-tc 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.

sprop-depack-buf-nalus:

This parameter specifies the maximum number of NAL units that precede a NAL unit in the de-packetization buffer in reception order and follow the NAL unit in decoding order.

The value of sprop-depack-buf-nalus MUST be an integer in the range of 0 to 32767, inclusive.

When not present, the value of sprop-depack-buf-nalus is inferred to be equal to 0.

When the RTP stream depends on one or more other RTP streams (in this case MST is in use), this parameter MUST be present and the value MUST be greater than 0.

Informative note: When the RTP stream does not depends on other RTP streams, either MST or SST may be in use.

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 the RTP stream depends on one or more other RTP streams (in this case MST is in use) or sprop-depack-buf-nalus is present and is greater than 0, this parameter MUST be present and the value MUST be greater than 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. A receiver is able to handle any 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. When network jitter can occur, an appropriately sized jitter buffer has to be available as well.

sprop-segmentation-id:

This parameter MAY be used to signal the segmentation tools present in the stream and that can be used for parallelization. The value of sprop-segmentation-id MUST be an integer in the range of 0 to 3, inclusive. When not present, the value of sprop-segmentation-id is inferred to be equal to 0.

When sprop-segmentation-id is equal to 0, no information about the segmentation tools is provided. When sprop-segmentation-id is equal to 1, it indicates that slices are present in the stream. When sprop-segmentation-id is equal to 2, it indicates that tiles are present in the stream. When sprop-segmentation-id is equal to 3, it indicates that WPP is used in the stream.

sprop-spatial-segmentation-idc:

A base16 [RFC4648] representation of the syntax element min_spatial_segmentation_idc as specified in [HEVC]. This parameter MAY be used to describe parallelization capabilities of the stream.

dec-parallel-cap:

This parameter MAY be used to indicate the decoder's additional decoding capabilities given the presence of tools enabling parallel decoding, such as slices, tiles, and WPP, in the video stream. The decoding capability of the decoder may vary with the setting of the parallel decoding tools present in the stream, e.g. the size of the tiles that are present in a stream. Therefore, multiple capability points may be provided, each indicating the minimum required decoding capability that is associated with a parallelism requirement, which is a requirement on the video stream that enables parallel decoding.

Each capability point is defined as a combination of 1) a parallelism requirement, 2) a profile (determined by profile-space and profile-id), 3) a highest level, and 4) a maximum processing rate, a maximum picture size, and a maximum video bitrate that may be equal to or greater than that determined by the highest level. The parameter's syntax in ABNF [RFC5234] is as follows:

```
dec-parallel-cap = "dec-parallel-cap={" cap-point *("," cap-point) "}"
```

```
cap-point = ("w" / "t") ":" spatial-seg-idc 1*(";"
            cap-parameter)

spatial-seg-idc = 1*4DIGIT ; 1-4095

cap-parameter = tier-flag / level-id / max-lsr
              / max-lps / max-br
```

The set of capability points expressed by the dec-parallel-cap parameter is enclosed in a pair of curly braces ("{}"). Each set of two consecutive capability points is separated by a comma (','). Within each capability point, each set of two consecutive parameters, and when present, their values, is separated by a semicolon (';').

The profile of all capability points is determined by profile-space and profile-id that are outside the dec-parallel-cap parameter.

Each capability point starts with an indication of the parallelism requirement, which consists of a parallel tool type, which may be equal to 'w' or 't', and a decimal value of the spatial-seg-idc parameter. When the type is 'w', the capability point is valid only for H.265 bitstreams with WPP in use, i.e. entropy_coding_sync_enabled_flag equal to 1. When the type is 't', the capability point is valid only for H.265 bitstreams with WPP not in use (i.e. entropy_coding_sync_enabled_flag equal to 0). The capability-point is valid only for H.265 bitstreams with min_spatial_segmentation_idc equal to or greater than spatial-seg-idc.

The value of spatial-seg-idc MUST be greater than 0.

After the parallelism requirement indication, each capability point continues with one or more pairs of parameter and value in any order for any of the following parameters:

- o tier-flag
- o level-id
- o max-lsr

- o max-lps
- o max-br

At most one occurrence of each of the above five parameters is allowed within each capability point.

The values of dec-parallel-cap.tier-flag and dec-parallel-cap.level-id for a capability point indicate the highest level of the capability point. The values of dec-parallel-cap.max-lsr, dec-parallel-cap.max-lps, and dec-parallel-cap.max-br for a capability point indicate the maximum processing rate in units of luma samples per second, the maximum picture size in units of luma samples, and the maximum video bitrate (in units of CpbBrVclFactor bits per second for the VCL HRD parameters and in units of CpbBrNalFactor bits per second for the NAL HRD parameters where CpbBrVclFactor and CpbBrNalFactor are defined in Section A.4 of [HEVC]).

When not present, the value of dec-parallel-cap.tier-flag is inferred to be equal to the value of tier-flag outside the dec-parallel-cap parameter. When not present, the value of dec-parallel-cap.level-id is inferred to be equal to the value of max-recv-level-id outside the dec-parallel-cap parameter. When not present, the value of dec-parallel-cap.max-lsr, dec-parallel-cap.max-lps, or dec-parallel-cap.max-br is inferred to be equal to the value of max-lsr, max-lps, or max-br, respectively, outside the dec-parallel-cap parameter.

The general decoding capability, expressed by the set of parameters outside of dec-parallel-cap, is defined as the capability point that is determined by the following combination of parameters: 1) the parallelism requirement corresponding to the value of sprop-segmentation-id equal to 0 for a stream, 2) the profile determined by profile-space and profile-id, 3) the highest level determined by tier-flag and max-recv-level-id, and 4) the maximum processing rate, the maximum picture size, and the maximum video bitrate determined by the highest level. The general decoding capability MUST NOT be included as one of the set of capability points in the dec-parallel-cap parameter.

For example, the following parameters express the general decoding capability of 720p30 (Level 3.1) plus an additional decoding capability of 1080p30 (Level 4) given that the spatially largest tile or slice used in the bitstream is equal to or less than 1/3 of the picture size:

a=fmtp:98 level-id=93;dec-parallel-cap={t:8;level-id=120}

For another example, the following parameters express an additional decoding capability of 1080p30, using dec-parallel-cap.max-lsr and dec-parallel-cap.max-lps, given that WPP is used in the stream:

a=fmtp:98 level-id=93;dec-parallel-cap={w:8;
max-lsr=62668800;max-lps=2088960}

Informative note: When min_spatial_segmentation_idc is present in a stream and WPP is not used, [HEVC] specifies that there is no slice or no tile in the stream containing more than 4 * PicSizeInSamplesY / (min_spatial_segmentation_idc + 4) luma samples.

Encoding considerations:

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

Security considerations:

See Section 9 of RFC XXXX.

Public specification:

Please refer to Section 13 of RFC XXXX.

Additional information: None

File extensions: none

Macintosh file type code: none

Object identifier or OID: none

Person & email address to contact for further information:

Intended usage: COMMON

Author: See Section 14 of RFC XXXX.

Change controller:

IETF Audio/Video Transport Payloads working group delegated from the IESG.

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/H265 string is mapped to fields in the Session Description Protocol (SDP) [RFC4566] as follows:

- The media name in the "m=" line of SDP MUST be video.
- The encoding name in the "a=rtpmap" line of SDP MUST be H265 (the media subtype).
- The clock rate in the "a=rtpmap" line MUST be 90000.
- The OPTIONAL parameters "profile-space", "profile-id", "tier-flag", "level-id", "interop-constraints", "profile-compatibility-indicator", "sub-layer-id", "recv-sub-layer-id", "max-recv-level-id", "max-lsr", "max-lps", "max-cpb", "max-dpb", "max-br", "max-tr", "max-tc", "max-fps", "sprop-depack-buf-nalus", "sprop-depack-buf-bytes", "depack-buf-cap", "sprop-segmentation-id", "sprop-spatial-segmentation-idc", and "dec-parallel-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 parameters "sprop-vps", "sprop-sps", and "sprop-pps", when present, MUST be included in the "a=fmtp" line of SDP or conveyed using the "fmtp" source attribute as specified in section 6.3 of [RFC5576]. For a particular media format (i.e. RTP payload type), "sprop-vps" "sprop-sps", or "sprop-pps" 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, these parameters are expressed as a media type string, in the form of a semicolon separated list of parameter=value pairs. When conveyed using the "fmtp" source attribute, these parameters are only associated with the given source and payload type as parts of the "fmtp" source attribute.

Informative note: Conveyance of "sprop-vps", "sprop-sps", and "sprop-pps" using the "fmtp" source attribute allows for out-of-band transport of parameter sets in topologies like Topo-Video-switch-MCU as specified in [RFC5117].

An example of media representation in SDP is as follows:

```
m=video 49170 RTP/AVP 98
a=rtpmap:98 H265/90000
a=fmtp:98 profile-id=1;
          sprop-vps=<video parameter sets data>
```

7.2.2 Usage with SDP Offer/Answer Model

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

- The parameters identifying a media format configuration for HEVC are profile-space, profile-id, tier-flag, level-id, interop-constraints, and profile-compatibility-indicator. These media configuration parameters, except for level-id, MUST be used symmetrically when the answerer does not include recv-sub-layer-id in the answer for the media format (payload type). In other words, the answerer MUST 1) maintain all configuration parameters for the media format (payload type), 2) include recv-sub-layer-id in the answer for the media format (payload type), or 3) remove the media format (payload type) completely (when one or more of the parameter values are not supported). The value of level-id is changeable.

  Informative note: The requirement for symmetric use does not apply for level-id, and does not apply for the other stream properties and capability parameters.

- To simplify handling and matching of these configurations, the same RTP payload type number used in the offer SHOULD also be used in the answer, as specified in [RFC3264]. The same RTP payload type number used in the offer MUST also be used in the answer when the answer includes recv-sub-layer-id. When the answer does not include recv-sub-layer-id, the answer MUST NOT contain a payload type number used in the offer unless the configuration is exactly the same as in the offer or the configuration in the answer only differs from that in the offer with a different value of level-id. The answer MAY contain the recv-sub-layer-id parameter if an HEVC stream contains multiple operation points (using temporal scalability and sub-layers) and sprop-vps is included in the offer where sub-layers are present in the video parameter set. If the sprop-vps is provided in an offer, an answerer MAY select a particular operation point in the received and/or in the sent stream. When recv-sub-layer-id is present in the answer, the media configuration parameters MUST NOT be present in the answer. Rather, the media configuration that the answerer will use for receiving and/or sending is the one used for the selected operation point as indicated in the offer.

  Informative note: When an offerer receives an answer that does not include recv-sub-layer-id, it has to compare payload

types not declared in the offer based on the media type (i.e. video/H265) and the above media configuration parameters with any payload types it has already declared. This will enable it to determine whether the configuration in question is new or if it is equivalent to configuration already offered, since a different payload type number may be used in the answer. The ability to perform operation point selection enables a receiver to utilize the temporal scalable nature of an HEVC stream.

- The parameters sprop-depack-buf-nalus and sprop-depack-buf-bytes describe the properties of the RTP stream that the offerer or the answerer is sending for the media format configuration. This differs from the normal usage of the Offer/Answer parameters: normally such parameters declare the properties of the stream that the offerer or the answerer is able to receive. When dealing with HEVC, the offerer assumes that the answerer will be able to receive media encoded using the configuration being offered.

  Informative note: The above parameters apply for any stream sent by a declaring entity with the same configuration; i.e. they are dependent on their source. Rather than being bound to the payload type, the values may have to be applied to another payload type when being sent, as they apply for the configuration.

- The capability parameters max-lsr, max-lps, max-cpb, max-dpb, max-br, max-tr, and max-tc MAY be used to declare further capabilities of the offerer or answerer for receiving. These parameters MUST NOT be present when the direction attribute is "sendonly".

- The capability parameter max-fps MAY be used to declare lower capabilities of the offerer or answerer for receiving. The parameters MUST NOT be present when the direction attribute is "sendonly".

- The capability parameter dec-parallel-cap MAY be used to declare additional decoding capabilities of the offerer or answerer for receiving. Upon receiving such a declaration of a receiver, a sender MAY send a stream to the receiver utilizing those capabilities under the assumption that the stream fulfills the parallelism requirement. A stream that is sent based on choosing a capability point with parallel tool type 'w' from dec-parallel-cap MUST have entropy_coding_sync_enabled_flag equal to 1 and min_spatial_segmentation_idc equal to or larger than dec-parallel-cap.spatial-seg-idc of the capability point. A stream that is sent based on choosing a capability point with parallel tool type 't' from dec-parallel-cap MUST have entropy_coding_sync_enabled_flag equal to 0 and min_spatial_segmentation_idc equal to or larger than dec-parallel-cap.spatial-seg-idc of the capability point.

- An offerer has to include the size of the de-packetization buffer, sprop-depack-buf-bytes, and sprop-depack-buf-nalus, in the offer for an interleaved HEVC stream or for the MST transmission mode. To enable the offerer and answerer to inform each other about their capabilities for de-packetization buffering in receiving streams, both parties are RECOMMENDED to include depack-buf-cap. For interleaved streams or in MST, it is also RECOMMENDED to consider offering multiple payload types with different buffering requirements when the capabilities of the receiver are unknown.

- The sprop-vps, sprop-sps, or sprop-pps, when present (included in the "a=fmtp" line of SDP or conveyed using the "fmtp" source attribute as specified in section 6.3 of [RFC5576]), are used for out-of-band transport of the parameter sets (VPS, SPS, or PPS respectively). However, when out-of-band transport of parameter sets is used, parameter sets MAY still be additionally transported in-band unless explicitly disallowed by an application.

- The answerer MAY use either out-of-band or in-band transport of parameter sets for the stream it is sending, regardless of whether out-of-band parameter sets transport has been used in the offerer-to-answerer direction. Parameter sets included in an answer are independent of those parameter sets included in the offer, as they are used for decoding two different video streams, one from the answerer to the offerer and the other in the opposite direction.

- The following rules apply to transport of parameter set in the offerer-to-answerer direction.

  - An offer MAY include sprop-vps, sprop-sps, and/or sprop-pps. If none of these parameters is present in the offer, then only in-band transport of parameter sets is used.

  - If the level to use in the offerer-to-answerer direction is equal to the default level in the offer, the answerer MUST be prepared to use the parameter sets included in sprop-vps, sprop-sps, and sprop-pps (either included in the "a=fmtp" line of SDP or conveyed using the "fmtp" source attribute) for decoding the incoming NAL unit stream. Otherwise, the answerer MUST ignore sprop-vps, sprop-sps, and sprop-pps (either included in the "a=fmtp" line of SDP or conveyed using the "fmtp" source attribute) and the offerer MUST transmit parameter sets in-band.

  - In MST, the answerer MUST be prepared to use the parameter sets included in sprop-vps, sprop-sps, and sprop-pps of all RTP streams that a particular RTP stream depends on, when present (either included in the "a=fmtp" line of SDP or conveyed using the "fmtp" source attribute), for decoding the incoming NAL unit stream.

- The following rules apply to transport of parameter set in the answerer-to-offerer direction.

  - An answer MAY include sprop-vps, sprop-sps, and/or sprop-pps. If none of these parameters is present in the answer, then only in-band transport of parameter sets is used.

- If the level to use in the answerer-to-offerer direction is equal to the default level in the answer, the offerer MUST be prepared to use the parameter sets included in sprop-vps, sprop-sps, and sprop-pps (either included in the "a=fmtp" line of SDP or conveyed using the "fmtp" source attribute) for decoding the incoming NAL unit stream. Otherwise, the offerer MUST ignore sprop-vps, sprop-sps, and sprop-pps (either included in the "a=fmtp" line of SDP or conveyed using the "fmtp" source attribute) and the answerer MUST transmit parameter sets in-band.

- In MST, the offerer MUST be prepared to use the parameter sets included in sprop-vps, sprop-sps, and sprop-pps of all RTP streams that a particular RTP stream depends on, when present (either included in the "a=fmtp" line of SDP or conveyed using the "fmtp" source attribute), for decoding the incoming NAL unit stream.

- When sprop-vps, sprop-sps, and/or sprop-pps are conveyed using the "fmtp" source attribute as specified in section 6.3 of [RFC5576], the receiver of the parameters MUST store the parameter sets included in sprop-vps, sprop-sps, and/or sprop-pps and associate them with the source given as part of the "fmtp" source attribute. Parameter sets associated with one source MUST only be used to decode NAL units conveyed in RTP packets from the same source. When this mechanism is in use, SSRC collision detection and resolution MUST be performed as specified in [RFC5576].

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

- The media format configuration is identified by profile-space, profile-id, tier-flag, level-id, interop-constraints, and profile-compatibility-indicator. These media format configuration parameters, including level-id, MUST be used symmetrically; that is, the answerer MUST either maintain all configuration parameters or remove the media format (payload type) completely. Note that this implies that the level-id for Offer/Answer in multicast is not changeable.

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

- Parameter sets received MUST be associated with the originating source and MUST only be used in decoding the incoming NAL unit stream from the same source.

- The rules for other parameters are the same as above for unicast as long as the above rules are obeyed.

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

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

	sendonly --+			
answer: recvonly, recv-sub-layer-id --+				
recvonly w/o recv-sub-layer-id --+				
answer: sendrecv, recv-sub-layer-id --+				
sendrecv w/o recv-sub-layer-id --+				
profile-space	C	X	C	X	P
profile-id	C	X	C	X	P
tier-flag	C	X	C	X	P
level-id	C	X	C	X	P
interop-constraints	C	X	C	X	P
profile-compatibility-indicator	C	X	C	X	P
max-recv-level-id	R	R	R	R	-
sprop-depack-buf-nalus	P	P	-	-	P
sprop-depack-buf-bytes	P	P	-	-	P
depack-buf-cap	R	R	R	R	-
sprop-segmentation-id	P	P	P	P	P

sprop-spatial-segmentation-idc	P	P	P	P	P
max-br	R	R	R	R	-
max-cpb	R	R	R	R	-
max-dpb	R	R	R	R	-
max-lsr	R	R	R	R	-
max-lps	R	R	R	R	-
max-tr	R	R	R	R	-
max-tc	R	R	R	R	-
max-fps	R	R	R	R	-
sprop-vps	P	P	-	-	P
sprop-sps	P	P	-	-	P
sprop-pps	P	P	-	-	P
sub-layer-id	P	P	-	-	P
recv-sub-layer-id	X	O	X	O	-
dec-parallel-cap	R	R	R	R	-

Legend:

C: configuration for sending and receiving streams
P: properties of the stream to be sent
R: receiver capabilities
O: operation point selection
X: MUST NOT be present
-: not usable, when present SHOULD be ignored

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

Parameters declaring a configuration point are not changeable, with the exception of the level-id parameter for unicast usage. This expresses values a receiver expects to be used and MUST be used verbatim on the sender side. If level-id is changed, an answerer MUST NOT include the recv-sub-layer-id parameter.

When a sender's capabilities are declared, and non-changeable parameters are used in this declaration, these parameters express a configuration that is acceptable for the sender to receive streams. In order to achieve high interoperability levels, it is often advisable to offer multiple alternative configurations. It is

impossible to offer multiple configurations in a single payload type. Thus, when multiple configuration offers are made, each offer requires its own RTP payload type associated with the offer.

A receiver SHOULD understand all media type parameters, even if it only supports a subset of the payload format's functionality. This ensures that a receiver is capable of understanding when an offer to receive media can be downgraded to what is supported by the receiver of the offer.

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

7.2.3 Usage in Declarative Session Descriptions

When HEVC over RTP is offered with SDP in a declarative style, as in Real Time Streaming Protocol (RTSP) [RFC2326] or Session Announcement Protocol (SAP) [RFC2974], the following considerations are necessary.

- o All parameters capable of indicating both stream properties and receiver capabilities are used to indicate only stream properties. For example, in this case, the parameter profile-tier-level-id declares the values used by the stream, not the capabilities for receiving streams. This results in that the following interpretation of the parameters MUST be used:

Declaring actual configuration or stream properties:

- profile-space
- profile-id
- tier-flag
- level-id
- interop-constraints
- sprop-vps
- sprop-sps
- sprop-pps

- sprop-depack-buf-nalus
- sprop-depack-buf-bytes
- sprop-segmentation-id
- sprop-spatial-segmentation-idc

Not usable (when present, they SHOULD be ignored):

- max-lps
- max-lsr
- max-cpb
- max-dpb
- max-br
- max-tr
- max-tc
- max-fps
- max-recv-level-id
- depack-buf-cap
- sub-layer-id
- dec-parallel-cap

o A receiver of the SDP is required to support all parameters and values of the parameters provided; otherwise, the receiver MUST reject (RTSP) or not participate in (SAP) the session. It falls on the creator of the session to use values that are expected to be supported by the receiving application.

7.2.4 Parameter Sets Considerations

If MST is used, the rules on signaling media decoding dependency in SDP as defined in [RFC5583] apply. The rules on "hierarchical or layered encoding" with multicast in Section 5.7 of [RFC4566] do not apply, i.e. the notation for Connection Data "c=" SHALL NOT be used with more than one address. The order of session dependency is given from the RTP stream containing the lowest temporal sub-layer to the RTP stream containing the highest temporal sub-layer.

7.2.5 Dependency Signaling in Multi-Session Transmission

If MST is used, the rules on signaling media decoding dependency in SDP as defined in [RFC5583] apply. The rules on "hierarchical or layered encoding" with multicast in Section 5.7 of [RFC4566] do not

apply, i.e. the notation for Connection Data "c=" SHALL NOT be used with more than one address. The order of session dependency is given from the RTP stream containing the lowest temporal sub-layer to the RTP stream containing the highest temporal sub-layer.

8. Use with Feedback Messages

As specified in section 6.1 of RFC 4585 [RFC4585], payload Specific Feedback messages are identified by the RTCP packet type value PSFB (206). AVPF [RFC4585] defines three payload-specific feedback messages and one application layer feedback message, and CCM [RFC5104] specifies four payload-specific feedback messages.

These feedback messages are identified by means of the feedback message type (FMT) parameter as follows:

Assigned in [RFC4585]:

```
1:     Picture Loss Indication (PLI)
2:     Slice Lost Indication (SLI)
3:     Reference Picture Selection Indication (RPSI)
15:    Application layer FB message
31:    reserved for future expansion of the number space
```

Assigned in [RFC5104]:

```
4:     Full Intra Request (FIR) Command
5:     Temporal-Spatial Trade-off Request (TSTR)
6:     Temporal-Spatial Trade-off Notification (TSTN)
7:     Video Back Channel Message (VBCM)
```

Unassigned:

```
0:      unassigned
8-14:   unassigned
16-30:  unassigned
```

The following subsection defines how to use HEVC with the RPSI message, for the purpose of feedback based reference picture selection for improved error resilience in real-time conversational video applications such as video telephone and video conferencing.

Feedback based reference picture selection has been shown as a powerful tool to stop temporal error propagation for improved error resilience [Girod99][Wang05]. In one approach, the decoder side tracks errors in the decoded pictures and informs to the encoder side that a particular picture that has been decoded relatively earlier is correct and still present in the decoded picture buffer and requests the encoder to use that correct picture for reference when encoding the next picture, so to stop further temporal error propagation. For this approach, the decoder side should use the RPSI feedback message.

Encoders can encode some long-term reference pictures as specified in H.264 or HEVC for purposes described in the previous paragraph without the need of a huge decoded picture buffer. As shown in [Wang05], with a flexible reference picture management scheme as in H.264 and HEVC, even a decoded picture buffer size of two would work for the approach described in the previous paragraph.

8.1 Use of HEVC with the RPSI Feedback Message

The field "Native RPSI bit string defined per codec" is a base16 [RFC4648] representation of the 8 bits consisting of 2 most significant bits equal to 0 and 6 bits of nuh_layer_id, as defined in [HEVC], followed by the 32 bits representing the value of the PicOrderCntVal (in network byte order), as defined in [HEVC], for the picture that is requested to be used for reference when encoding the next picture.

The use of the RPSI feedback message as positive acknowledgement with HEVC is deprecated. In other words, the RPSI feedback message MUST only be used as a reference picture selection request, such that it can also be used in multicast.

9. Security Considerations

RTP packets using the payload format defined in this specification are subject to the security considerations discussed in the RTP specification [RFC3550], and in any applicable RTP profile such as RTP/AVP [RFC3551], RTP/AVPF [RFC4585], RTP/SAVP [RFC3711] or RTP/SAVPF [RFC5124]. However, as "Securing the RTP Protocol Framework: Why RTP Does Not Mandate a Single Media Security

Solution" [I-D.ietf-avt-srtp-not-mandatory] 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 as discussed in "Options for Securing RTP Sessions" [I-D.ietf-avtcore-rtp-security-options].

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 stream that are complex to decode and that cause the receiver to be overloaded. H.265 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 [RFC 3711].

Note that the appropriate mechanism to ensure confidentiality and integrity of RTP packets and their payloads is very dependent on the application and on the transport and signaling protocols employed. Thus, although SRTP is given as an example above, other possible choices exist.

Decoders MUST exercise caution with respect to the handling of user data SEI messages, particularly if they contain active elements, and MUST restrict their domain of applicability to the presentation containing the stream.

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 [RFC 3551]. 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 the RTP flow is achieving. This condition can be satisfied by implementing congestion control mechanisms to adapt the transmission rate, the number of layers subscribed for a layered multicast session, or by arranging for a receiver to leave the session if the loss rate is unacceptably high.

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

However, when pre-encoded content is being transmitted, bandwidth adaptation requires the pre-coded bitstream to be tailored for such adaptivity. The key mechanism available in HEVC is temporal scalability. A media sender can remove NAL units belonging to higher temporal sub-layers (i.e. those NAL units with a high value of TID) until the sending bitrate drops to an acceptable range. HEVC contains mechanisms that allow the lightweight identification of switching points in temporal enhancement layers, as discussed in Section 1.1.2 of this memo. An HEVC media sender can send packets belonging to NAL units of temporal enhancement layers starting from these switching points to probe for available bandwidth and to utilized bandwidth that has been shown to be available.

Above mechanisms 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

media stream. This may be accomplished by using a different RTP payload type.

MANEs MAY remove certain unusable packets from the packet stream when that 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 Consideration

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

12. Acknowledgements

Muhammed Coban and Marta Karczewicz are thanked for discussions on the specification of the use with feedback messages and other aspects in this memo. Jonathan Lennox and Jill Boyce are thanked for their contributions to the PACI design included in this memo. Rickard Sjoberg, Arild Fuldseth, Bo Burman Magnus Westerlund, and Tom Kristensen are thanked for their contributions to parallel processing related signalling. Bernard Aboba, Roni Even, Rickard Sjoberg, Sachin Deshpande, Woo Johnman, Mo Zanaty, and Ross Finlayson made valuable reviewing comments that led to improvements.

This document was prepared using 2-Word-v2.0.template.dot.

13. References

13.1 Normative References

[HEVC] ITU-T Recommendation H.265, "High efficiency video coding", April 2013.

[H.264] ITU-T Recommendation H.264, "Advanced video coding for generic audiovisual services", April 2013.

[RFC5583] Schierl, T. and Wenger, S., "Signaling Media Decoding Dependency in the Session Description Protocol (SDP)", RFC 5583, July 2009.

[RFC6184] Wang, Y.-K., Even, R., Kristensen, T., and R. Jesup, "RTP Payload Format for H.264 Video", RFC 6184, May 2011.

[RFC6190] Wenger, S., Wang, Y.-K., Schierl, T., and A. Eleftheriadis, "RTP Payload Format for Scalable Video Coding", RFC 6190, May 2011.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, October 2006.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and Jacobson, V., "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[RFC4566] Handley, M., Jacobson, V., and Perkins, C., "SDP: Session Description Protocol", RFC 4566, July 2006.

[RFC5576] Lennox, J., Ott, J., and Schierl, T., "Source-Specific Media Attributes in the Session Description Protocol", RFC 5576, June 2009.

[RFC4585] Ott, J., Wenger, S., Sato, N., Burmeister, C., and Rey, J., "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", RFC 4585, July 2006.

[RFC5104] Wenger, S., Chandra, U., Westerlund, M., and Burman, B., "Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)", RFC 5104, February 2008.

13.2 Informative References

[3GPDASH] 3GPP TS 26.247, "Transparent end-to-end Packet-switched Streaming Service (PSS); Progressive Download and Dynamic Adaptive Streaming over HTTP (3GP-DASH)", v12.1.0, December 2013.

[3GPPFF] 3GPP TS 26.244, "Transparent end-to-end packet switched streaming service (PSS); 3GPP file format (3GP)", v12.20, December 2013.

[Girod99] Girod, B. and Faerber, F., "Feedback-based error control for mobile video transmission", Proceedings IEEE, Vol. 87, No. 10, pp. 1707-1723, October 1999.

[I-D.ietf-avt-srtp-not-mandatory] Perkins, C. and M. Westerlund, "Securing the RTP ProtocolFramework: Why RTP Does Not Mandate a Single MediaSecurity Solution", draft-ietf-avt-srtp-not-mandatory-16 (work in progress), January 2014.

[I-D.ietf-avtcore-rtp-security-options] Westerlund, M. and C. Perkins, "Options for Securing RTP Sessions", draft-ietf-avtcore-rtp-security-options-10 (work in progress), January 2014.

[I-D.ietf-avtcore-rtp-multi-stream] Lennox, J., Westerlund, M., Wu, W., and C. Perkins, "Sending Multiple Media Streams in a Single RTP Session", draft-ietf-avtcore-rtp-multi-stream-01 (work in progress), July 2013.

[I-D.ietf-mmusic-sdp-bundle-negotiation] Holmberg, C., Alvestrand, H., and C. Jennings, "Multiplexing Negotiation Using Session Description Protocol (SDP) Port Numbers", draft-ietf-mmusic-sdp-bundle-negotiation-05 (work in progress), October 2013.

[ISOBMFF] IS0/IEC 14496-12 | 15444-12: "Information technology - Coding of audio-visual objects - Part 12: ISO base media file format" | "Information technology - JPEG 2000 image coding system - Part 12: ISO base media file format", 2012.

[JCTVC-J0107] Wang, Y.-K., Chen, Y., Joshi, R., and Ramasubramonian, K., "AHG9: On RAP pictures", JCT-VC document JCTVC-L0107, 10th JCT-VC meeting, July 2012, Stockholm, Sweden.

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

[MPEGDASH] ISO/IEC 23009-1, "Information technology - Dynamic adaptive streaming over HTTP (DASH) - Part 1: Media presentation description and segment formats", 2012.

[RFC5109] Li, A., "RTP Payload Format for Generic Forward Error Correction", RFC 5109, December 2007.

[Wang05] Wang, Y.-K., Zhu, C., and Li, H., "Error resilient video coding using flexible reference fames", Visual Communications and Image Processing 2005 (VCIP 2005), July 2005, Beijing, China.

14. Authors' Addresses

Ye-Kui Wang
Qualcomm Incorporated
5775 Morehouse Drive
San Diego, CA 92121
USA
Phone: +1-858-651-8345
EMail: yekuiw@qti.qualcomm.com

Yago Sanchez
Fraunhofer HHI
Einsteinufer 37
D-10587 Berlin
Germany

Phone: +49-30-31002-227
Email: yago.sanchez@hhi.fraunhofer.de

Thomas Schierl
Fraunhofer HHI
Einsteinufer 37
D-10587 Berlin
Germany
Phone: +49-30-31002-227
Email: ts@thomas-schierl.de

Stephan Wenger
Vidyo, Inc.
433 Hackensack Ave., 7th floor
Hackensack, N.J. 07601
USA
Phone: +1-415-713-5473
EMail: stewe@stewe.org

Miska M. Hannuksela
Nokia Corporation
P.O. Box 1000
33721 Tampere
Finland
Phone: +358-7180-08000
EMail: miska.hannuksela@nokia.com

Payload Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 14, 2014

P. Westin
H. Lundin
M. Glover
J. Uberti
F. Galligan
Google

February 10, 2014

RTP Payload Format for VP8 Video
draft-ietf-payload-vp8-11

Abstract

This memo describes an RTP payload format for the VP8 video codec. The payload format has wide applicability, as it supports applications from low bit-rate peer-to-peer usage, to high bit-rate video conferences.

Status of this Memo

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

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

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

This Internet-Draft will expire on August 14, 2014.

Copyright Notice

Copyright (c) 2014 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 (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

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

Table of Contents

1. Introduction 3
2. Conventions, Definitions and Acronyms 4
3. Media Format Description 5
4. Payload Format 6
4.1. RTP Header Usage 6
4.2. VP8 Payload Descriptor 7
4.3. VP8 Payload Header 11
4.4. Aggregated and Fragmented Payloads 12
4.5. Frame reconstruction algorithm 12
4.5.1. Partition reconstruction algorithm 13
4.6. Examples of VP8 RTP Stream 13
4.6.1. Key frame in a single RTP packet 13
4.6.2. Non-discardable VP8 interframe in a single RTP packet; no PictureID 14
4.6.3. VP8 partitions in separate RTP packets 15
4.6.4. VP8 frame fragmented across RTP packets 16
4.6.5. VP8 frame with long PictureID 18
5. Using VP8 with RPSI and SLI Feedback 19
5.1. RPSI 19
5.2. SLI 19
5.3. Example 20
6. Payload Format Parameters 23
6.1. Media Type Definition 23
6.2. SDP Parameters 24
6.2.1. Mapping of MIME Parameters to SDP 24
6.2.2. Offer/Answer Considerations 25
7. Security Considerations 26
8. Congestion Control 27
9. IANA Considerations 28
10. References 29
Authors' Addresses 30

1. Introduction

This memo describes an RTP payload specification applicable to the transmission of video streams encoded using the VP8 video codec [RFC6386]. The format described in this document can be used both in peer-to-peer and video conferencing applications.

VP8 is based on decomposition of frames into square sub-blocks of pixels, prediction of such sub-blocks using previously constructed blocks, and adjustment of such predictions (as well as synthesis of unpredicted blocks) using a discrete cosine transform (hereafter abbreviated as DCT). In one special case, however, VP8 uses a "Walsh-Hadamard" (hereafter abbreviated as WHT) transform instead of a DCT. An encoded VP8 frame is divided into two or more partitions, as described in [RFC6386]. The first partition (prediction or mode) contains prediction mode parameters and motion vectors for all macroblocks. The remaining partitions all contain the quantized DCT/WHT coefficients for the residuals. There can be 1, 2, 4, or 8 DCT/WHT partitions per frame, depending on encoder settings.

In summary, the payload format described in this document enables a number of features in VP8, including:

- Taking partition boundaries into consideration, to improve loss robustness and facilitate efficient packet loss concealment at the decoder.
- Temporal scalability.
- Advanced use of reference frames to enable efficient error recovery.
- Marking of frames that have no impact on the decoding of any other frame, so that these non-reference frames can be discarded in a server or media-aware network element if needed.

2. Conventions, Definitions and Acronyms

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Media Format Description

The VP8 codec uses three different reference frames for interframe prediction: the previous frame, the golden frame, and the altref frame. The payload specification in this memo has elements that enable advanced use of the reference frames, e.g., for improved loss robustness.

One specific use case of the three reference frame types is temporal scalability. By setting up the reference hierarchy in the appropriate way, up to five temporal layers can be encoded. (How to set up the reference hierarchy for temporal scalability is not within the scope of this memo.)

Another property of the VP8 codec is that it applies data partitioning to the encoded data. Thus, an encoded VP8 frame can be divided into two or more partitions, as described in "VP8 Data Format and Decoding Guide" [RFC6386]. The first partition (prediction or mode) contains prediction mode parameters and motion vectors for all macroblocks. The remaining partitions all contain the transform coefficients for the residuals. The first partition is decodable without the remaining residual partitions. The subsequent partitions may be useful even if some part of the frame is lost. This memo allows the partitions to be sent separately or in the same RTP packet. It may be beneficial for decoder error-concealment to send the partitions in different packets, even though it is not mandatory according to this specification.

The format specification is described in Section 4. In Section 5, a method to acknowledge receipt of reference frames using RTCP techniques is described.

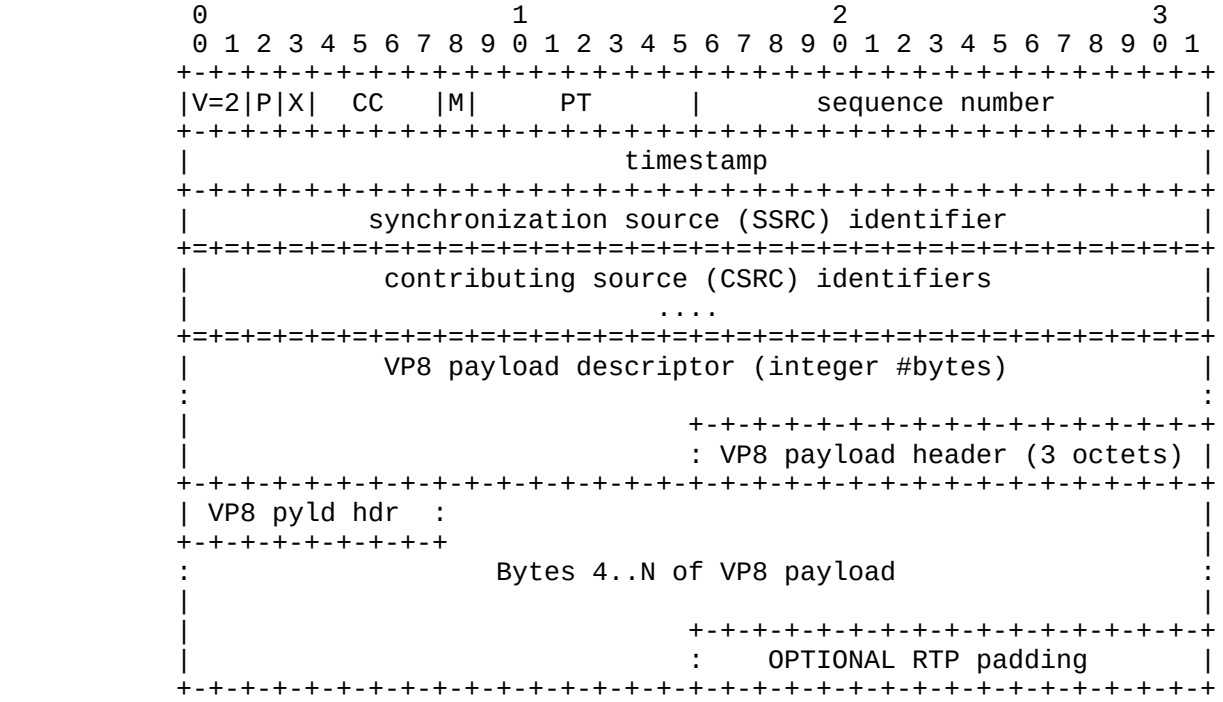
The payload partitioning and the acknowledging method both serve as motivation for three of the fields included in the payload format: the "PID", "1st partition size" and "PictureID" fields. The ability to encode a temporally scalable stream motivates the "TL0PICIDX" and "TID" fields.

4. Payload Format

This section describes how the encoded VP8 bitstream is encapsulated in RTP. To handle network losses usage of RTP/AVPF [RFC4585] is RECOMMENDED. All integer fields in the specifications are encoded as unsigned integers in network octet order.

4.1. RTP Header Usage

The general RTP payload format for VP8 is depicted below.



The VP8 payload descriptor and VP8 payload header will be described in the sequel. OPTIONAL RTP padding MUST NOT be included unless the P bit is set.

Figure 1

Marker bit (M): Set for the very last packet of each encoded frame in line with the normal use of the M bit in video formats. This enables a decoder to finish decoding the picture, where it otherwise may need to wait for the next packet to explicitly know that the frame is complete.

Timestamp: The RTP timestamp indicates the time when the frame was sampled at a clock rate of 90 kHz.

Sequence number: The sequence numbers are monotonically increasing and set as packets are sent.

The remaining RTP header fields are used as specified in [RFC3550].

4.2. VP8 Payload Descriptor

The first octets after the RTP header are the VP8 payload descriptor, with the following structure.

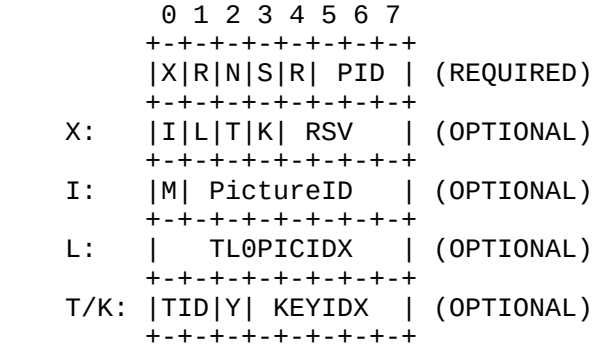


Figure 2

X: Extended control bits present. When set to one, the extension octet MUST be provided immediately after the mandatory first octet. If the bit is zero, all optional fields MUST be omitted.

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

N: Non-reference frame. When set to one, the frame can be discarded without affecting any other future or past frames. If the reference status of the frame is unknown, this bit SHOULD be set to zero to avoid discarding frames needed for reference.

Informative note: This document does not describe how to determine if an encoded frame is non-reference. The reference status of an encoded frame is preferably provided from the encoder implementation.

S: Start of VP8 partition. SHOULD be set to 1 when the first payload octet of the RTP packet is the beginning of a new VP8 partition, and MUST NOT be 1 otherwise. The S bit MUST be set to 1 for the first packet of each encoded frame.

PID: Partition index. Denotes which VP8 partition the first payload octet of the packet belongs to. The first VP8 partition (containing modes and motion vectors) MUST be labeled with PID = 0. PID SHOULD be incremented for each subsequent partition, but MAY be kept at 0 for all packets. PID MUST NOT be larger than 8. If more than one packet in an encoded frame contains the same PID, the S bit MUST NOT be set for any other packet than the first packet with that PID.

When the X bit is set to 1 in the first octet, the OPTIONAL extension bit field MUST be present in the second octet. If the X bit is 0, the extension bit field MUST NOT be present, and all bits below MUST be implicitly interpreted as 0.

I: PictureID present. When set to one, the OPTIONAL PictureID MUST be present after the extension bit field and specified as below. Otherwise, PictureID MUST NOT be present.

L: TL0PICIDX present. When set to one, the OPTIONAL TL0PICIDX MUST be present and specified as below, and the T bit MUST be set to 1. Otherwise, TL0PICIDX MUST NOT be present.

T: TID present. When set to one, the OPTIONAL TID/KEYIDX octet MUST be present. The TID|Y part of the octet MUST be specified as below. If K (below) is set to one but T is set to zero, the TID/KEYIDX octet MUST be present, but the TID|Y field MUST be ignored. If neither T nor K is set to one, the TID/KEYIDX octet MUST NOT be present.

K: KEYIDX present. When set to one, the OPTIONAL TID/KEYIDX octet MUST be present. The KEYIDX part of the octet MUST be specified as below. If T (above) is set to one but K is set to zero, the TID/KEYIDX octet MUST be present, but the KEYIDX field MUST be ignored. If neither T nor K is set to one, the TID/KEYIDX octet MUST NOT be present.

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

After the extension bit field follow the extension data fields that are enabled.

M: The most significant bit of the first octet is an extension flag. The field MUST be present if the I bit is equal to one. If set the PictureID field MUST contain 16 bits else it MUST contain 8 bits including this MSB, see PictureID.

PictureID: 8 or 16 bits including the M bit. This is a running index of the frames. The field MUST be present if the I bit is equal to one. The 7 following bits carry (parts of) the PictureID. If the extension flag is one, the PictureID continues in the next octet forming a 15 bit index, where the 8 bits in the second octet are the least significant bits of the PictureID. If the extension flag is zero, there is no extension, and the PictureID is the 7 remaining bits of the first (and only) octet. The sender may choose 7 or 15 bits index. The PictureID SHOULD start on a random number, and MUST wrap after reaching the maximum ID. The receiver MUST NOT assume that the number of bits in PictureID stay the same through the session.

TL0PICIDX: 8 bits temporal level zero index. The field MUST be present if the L bit is equal to 1, and MUST NOT be present otherwise. TL0PICIDX is a running index for the temporal base layer frames, i.e., the frames with TID set to 0. If TID is larger than 0, TL0PICIDX indicates which base layer frame the current image depends on. TL0PICIDX MUST be incremented when TID is 0. The index SHOULD start on a random number, and MUST restart at 0 after reaching the maximum number 255.

TID: 2 bits temporal layer index. The TID/KEYIDX octet MUST be present when either the T bit or the K bit or both are equal to 1, and MUST NOT be present otherwise. The TID field MUST be ignored by the receiver when the T bit is set equal to 0. The TID field indicates which temporal layer the packet represents. The lowest layer, i.e., the base layer, MUST have TID set to 0. Higher layers SHOULD increment the TID according to their position in the layer hierarchy.

Y: 1 layer sync bit. The TID/KEYIDX octet MUST be present when either the T bit or the K bit or both are equal to 1, and MUST NOT be present otherwise. The Y bit SHOULD be set to 1 if the current frame depends only on the base layer (TID = 0) frame with TL0PICIDX equal to that of the current frame. The Y bit MUST be set to 0 if the current frame depends any other frame than the base layer (TID = 0) frame with TL0PICIDX equal to that of the current frame. If the Y bit is set when the T bit is equal to 0 the current frame MUST only depend on a past base layer (TID=0) key frame as signaled by a change in the KEYIDX field. Additionally this frame MUST NOT depend on any of the three codec buffers (as defined by [RFC6386]) that have been updated since the

last time the KEYIDX field was changed.

Informative note: This document does not describe how to determine the dependence status for a frame; this information is preferably provided from the encoder implementation. In the case of unknown status, the Y bit can safely be set to 0.

KEYIDX: 5 bits temporal key frame index. The TID/KEYIDX octet MUST be present when either the T bit or the K bit or both are equal to 1, and MUST NOT be present otherwise. The KEYIDX field MUST be ignored by the receiver when the K bit is set equal to 0. The KEYIDX field is a running index for key frames. KEYIDX MAY start on a random number, and MUST restart at 0 after reaching the maximum number 31. When in use, the KEYIDX SHOULD be present for both key frames and interframes. The sender MUST increment KEYIDX for key frames which convey parameter updates critical to the interpretation of subsequent frames, and SHOULD leave the KEYIDX unchanged for key frames that do not contain these critical updates. A receiver SHOULD NOT decode an interframe if it has not received and decoded a key frame with the same KEYIDX after the last KEYIDX wrap-around.

Informative note: This document does not describe how to determine if a key frame updates critical parameters; this information is preferably provided from the encoder implementation. A sender that does not have this information may either omit the KEYIDX field (set K equal to 0), or increment the KEYIDX on every key frame. The benefit with the latter is that any key frame loss will be detected by the receiver, which can signal for re-transmission or request a new key frame.

Informative note: Implementations doing splicing of VP8 streams will have to make sure the rules for incrementing TL0PICIDX and KEYIDX are obeyed across the splice. This will likely require rewriting values of TL0PICIDX and KEYIDX after the splice.

4.3. VP8 Payload Header

The beginning of an encoded VP8 frame is referred to as an "uncompressed data chunk" in [RFC6386], and co-serve as payload header in this RTP format. The codec bitstream format specifies two different variants of the uncompressed data chunk: a 3 octet version for interframes and a 10 octet version for key frames. The first 3 octets are common to both variants. In the case of a key frame the remaining 7 octets are considered to be part of the remaining payload in this RTP format. Note that the header is present only in packets which have the S bit equal to one and the PID equal to zero in the payload descriptor. Subsequent packets for the same frame do not carry the payload header.

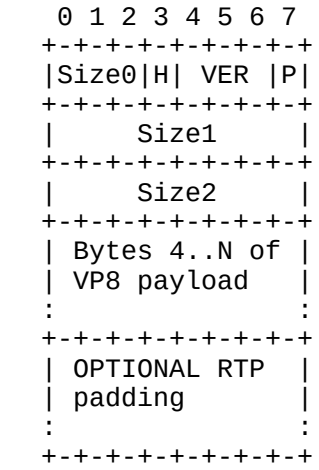


Figure 3

H: Show frame bit as defined in [RFC6386].

VER: A version number as defined in [RFC6386].

P: Inverse key frame flag. When set to 0 the current frame is a key frame. When set to 1 the current frame is an interframe. Defined in [RFC6386]

SizeN: The size of the first partition in bytes is calculated from the 19 bits in Size0, Size1, and Size2 as 1stPartitionSize = Size0 + 8 * Size1 + 2048 * Size2. [RFC6386].

4.4. Aggregated and Fragmented Payloads

An encoded VP8 frame can be divided into two or more partitions, as described in Section 1. One packet can contain a fragment of a partition, a complete partition, or an aggregate of fragments and partitions. In the preferred use case, the S bit and PID fields described in Section 4.2 should be used to indicate what the packet contains. The PID field should indicate which partition the first octet of the payload belongs to, and the S bit indicates that the packet starts on a new partition. Aggregation of encoded partitions is done without explicit signaling. Partitions MUST be aggregated in decoding order. Two fragments from different partitions MAY be aggregated into the same packet. An aggregation MUST have exactly one payload descriptor. Aggregated partitions MUST represent parts of one and the same video frame. Consequently, an aggregated packet will have one or no payload header, depending on whether the aggregate contains the beginning of the first partition of a frame or not, respectively. Note that the length of the first partition can always be obtained from the first partition size parameter in the VP8 payload header.

The VP8 bitstream format [RFC6386] specifies that if multiple DCT/WHT partitions are produced, the location of each partition start is found at the end of the first (prediction/mode) partition. In this RTP payload specification, the location offsets are considered to be part of the first partition.

It is OPTIONAL for a packetizer implementing this RTP specification to pay attention to the partition boundaries within an encoded frame. If packetization of a frame is done without considering the partition boundaries, the PID field MAY be set to zero for all packets, and the S bit MUST NOT be set to one for any other packet than the first.

4.5. Frame reconstruction algorithm

Example of frame reconstruction algorithm.

1: Collect all packets with a given RTP timestamp.

2: Go through packets in order, sorted by sequence numbers, if packets are missing, send NACK as defined in [RFC4585] or decode with missing partitions, see Section 4.5.1 below.

3: A frame is complete if the frame has no missing sequence numbers, the first packet in the frame contains S=1 with partId=0 and the last packet in the frame has the marker bit set.

4.5.1. Partition reconstruction algorithm

Example of partition reconstruction algorithm.

1: Scan for the start of a new partition; S=1.

2: Continue scan to detect end of partition; hence a new S=1 (previous packet was the end of the partition) is found or the marker bit is set. If a loss it detected before the end of the partition, abandon all packets in this partition and continue the scan repeating 1.

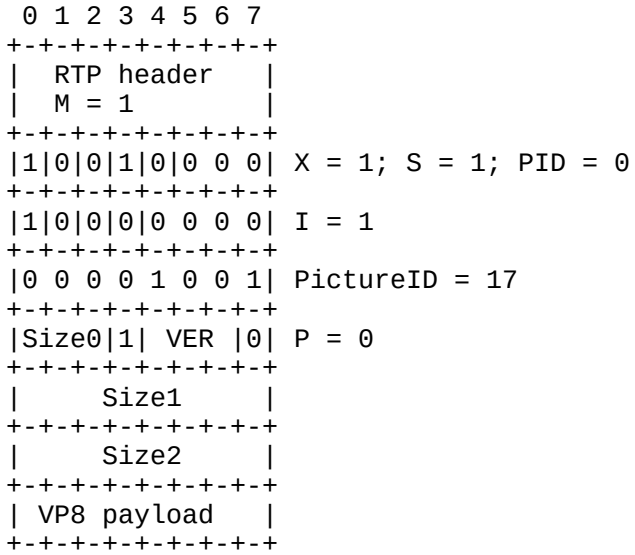
3: Store the packets in the complete partition, continue the scan repeating 1 until end of frame is reached.

4: Send all complete partitions to the decoder. If no complete partition is found discard the whole frame.

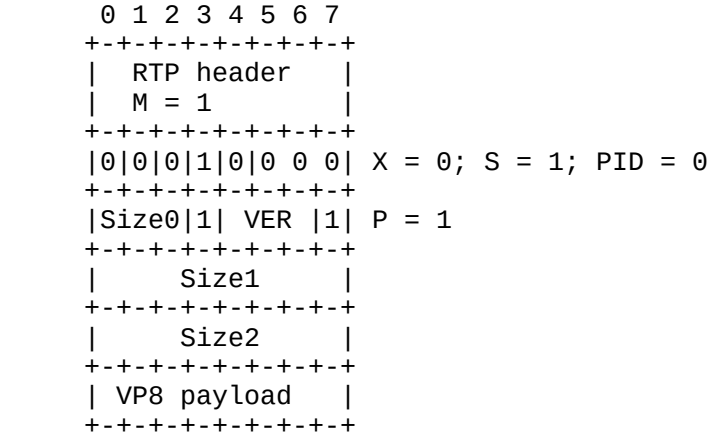
4.6. Examples of VP8 RTP Stream

A few examples of how the VP8 RTP payload can be used are included below.

4.6.1. Key frame in a single RTP packet

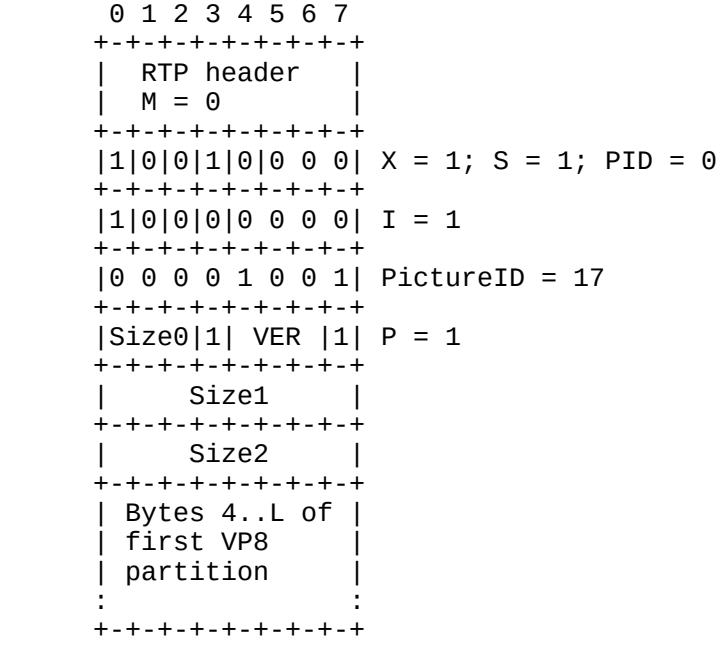


4.6.2. Non-discardable VP8 interframe in a single RTP packet; no PictureID

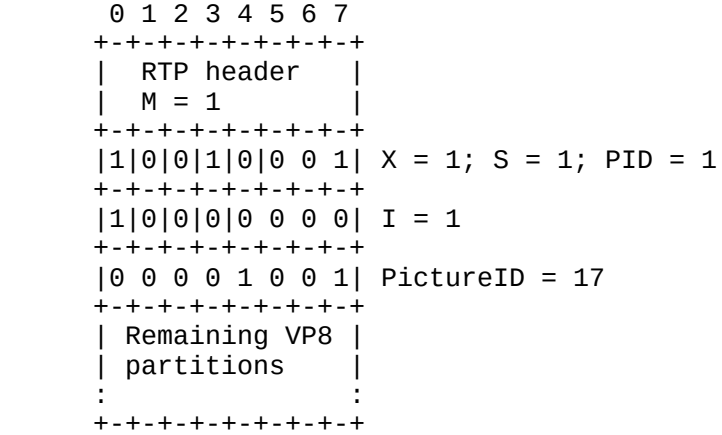


4.6.3. VP8 partitions in separate RTP packets

First RTP packet; complete first partition.

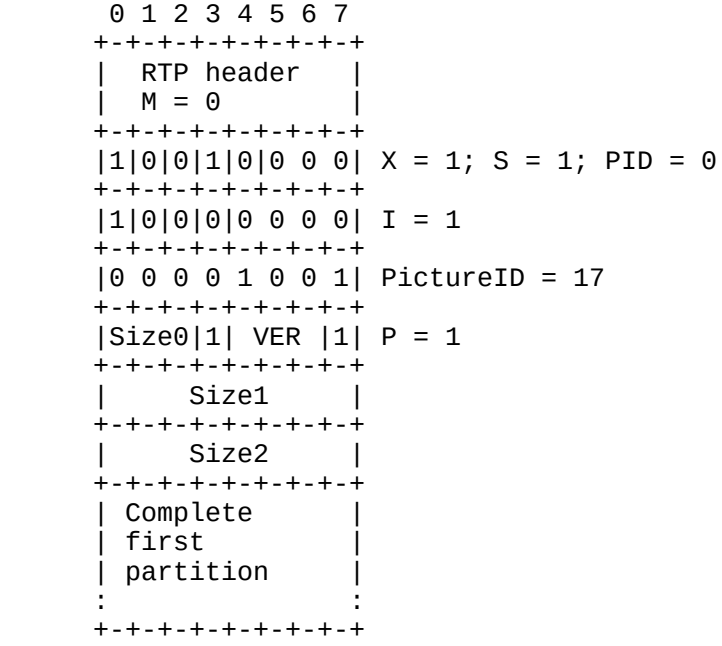


Second RTP packet; complete second partition.

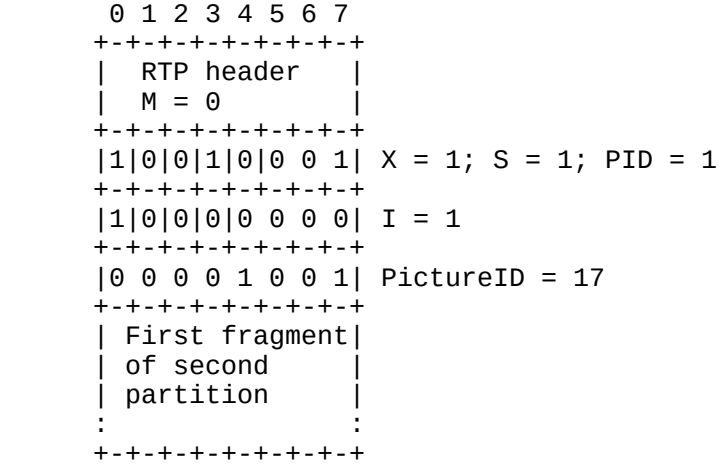


4.6.4. VP8 frame fragmented across RTP packets

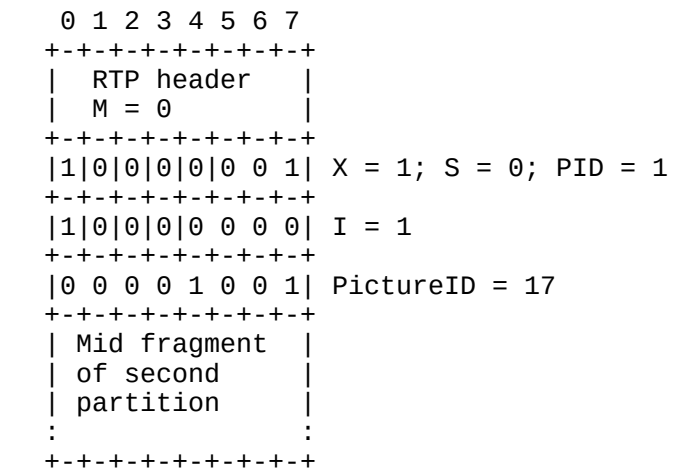
First RTP packet; complete first partition.



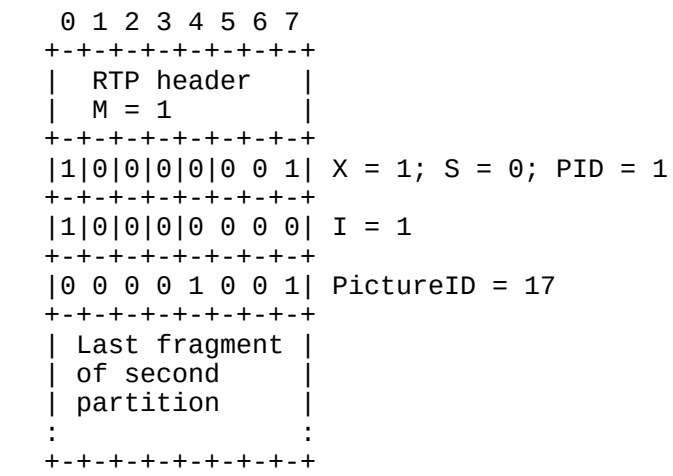
Second RTP packet; first fragment of second partition.



Third RTP packet; second fragment of second partition.

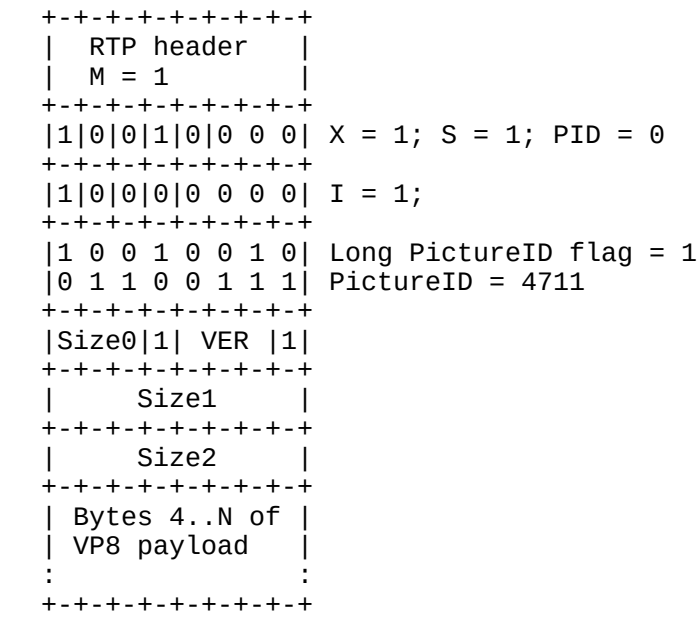


Fourth RTP packet; last fragment of second partition.



4.6.5. VP8 frame with long PictureID

PictureID = 4711 = 001001001100111 binary (first 7 bits: 0010010, last 8 bits: 01100111).



5. Using VP8 with RPSI and SLI Feedback

The VP8 payload descriptor defined in Section 4.2 above contains an optional PictureID parameter. This parameter is included mainly to enable use of reference picture selection index (RPSI) and slice loss indication (SLI), both defined in [RFC4585].

5.1. RPSI

The reference picture selection index is a payload-specific feedback message defined within the RTCP-based feedback format. The RPSI message is generated by a receiver and can be used in two ways. Either it can signal a preferred reference picture when a loss has been detected by the decoder -- preferably then a reference that the decoder knows is perfect -- or, it can be used as positive feedback information to acknowledge correct decoding of certain reference pictures. The positive feedback method is useful for VP8 used as unicast. The use of RPSI for VP8 is preferably combined with a special update pattern of the codec's two special reference frames -- the golden frame and the altref frame -- in which they are updated in an alternating leapfrog fashion. When a receiver has received and correctly decoded a golden or altref frame, and that frame had a PictureID in the payload descriptor, the receiver can acknowledge this simply by sending an RPSI message back to the sender. The message body (i.e., the "native RPSI bit string" in [RFC4585]) is simply the PictureID of the received frame.

5.2. SLI

The slice loss indication is another payload-specific feedback message defined within the RTCP-based feedback format. The SLI message is generated by the receiver when a loss or corruption is detected in a frame. The format of the SLI message is as follows [RFC4585]:

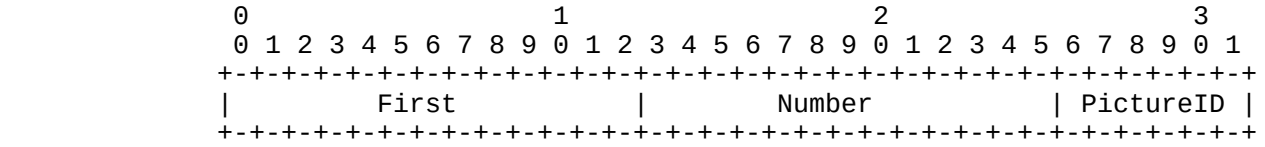


Figure 4

Here, First is the macroblock address (in scan order) of the first lost block and Number is the number of lost blocks. PictureID is the six least significant bits of the codec-specific picture identifier in which the loss or corruption has occurred. For VP8, this codec-specific identifier is naturally the PictureID of the current frame,

as read from the payload descriptor. If the payload descriptor of the current frame does not have a PictureID, the receiver MAY send the last received PictureID+1 in the SLI message. The receiver MAY set the First parameter to 0, and the Number parameter to the total number of macroblocks per frame, even though only parts of the frame is corrupted. When the sender receives an SLI message, it can make use of the knowledge from the latest received RPSI message. Knowing that the last golden or altref frame was successfully received, it can encode the next frame with reference to that established reference.

5.3. Example

The use of RPSI and SLI is best illustrated in an example. In this example, the encoder may not update the altref frame until the last sent golden frame has been acknowledged with an RPSI message. If an update is not received within some time, a new golden frame update is sent instead. Once the new golden frame is established and acknowledge, the same rule applies when updating the altref frame.

Event	Sender	Receiver	Established reference
1000	Send golden frame PictureID = 0		
		Receive and decode golden frame	
1001		Send RPSI(0)	
1002	Receive RPSI(0)		golden
...	(sending regular frames)		
1100	Send altref frame PictureID = 100		
		Altref corrupted or lost	golden
1101		Send SLI(100)	golden
1102	Receive SLI(100)		

1103	Send frame with reference to golden		
		Receive and decode frame (decoder state restored)	golden
...	(sending regular frames)		
1200	Send altref frame PictureID = 200		
		Receive and decode altref frame	golden
1201		Send RPSI(200)	
1202	Receive RPSI(200)		altref
...	(sending regular frames)		
1300	Send golden frame PictureID = 300		
		Receive and decode golden frame	altref
1301		Send RPSI(300)	altref
1302	RPSI lost		
1400	Send golden frame PictureID = 400		
		Receive and decode golden frame	altref
1401		Send RPSI(400)	
1402	Receive RPSI(400)		golden

Table 1: Exemple signaling between sender and receiver

Note that the scheme is robust to loss of the feedback messages. If

the RPSI is lost, the sender will try to update the golden (or altref) again after a while, without releasing the established reference. Also, if an SLI is lost, the receiver can keep sending SLI messages at any interval, as long as the picture is corrupted.

6. Payload Format Parameters

This payload format has two required parameters.

6.1. Media Type Definition

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

Type name: video

Subtype name: VP8

Required parameters:
These parameters MUST be used to signal the capabilities of a receiver implementation. These parameters MUST NOT be used for any other purpose.

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

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

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

Optional parameters: none

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

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

Interoperability considerations: None.

Published specification: VP8 bitstream format [RFC6386] and RFC XXXX.
[RFC Editor: Upon publication as an RFC, please replace "XXXX" with the number assigned to this document and remove this note.]

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

Additional information: None.

Person & email address to contact for further information:
Patrik Westin, patrik.westin@gmail.com

Intended usage: COMMON

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

Author: Patrik Westin, patrik.westin@gmail.com

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

6.2. SDP Parameters

The receiver MUST ignore any parameter unspecified in this memo.

6.2.1. Mapping of MIME Parameters to SDP

The MIME media type video/VP8 string is mapped to fields in the Session Description Protocol (SDP) [RFC4566] as follows:

- The media name in the "m=" line of SDP MUST be video.
- The encoding name in the "a=rtpmap" line of SDP MUST be VP8 (the MIME subtype).
- The clock rate in the "a=rtpmap" line MUST be 90000.
- The parameters "max-fs", and "max-fr", MUST be included in the "a=fmtp" line of SDP. These parameters are expressed as a MIME media type string, in the form of a semicolon separated list of parameter=value pairs.

6.2.1.1. Example

An example of media representation in SDP is as follows:

```
m=video 49170 RTP/AVPF 98
a=rtpmap:98 VP8/90000
a=fmtp:98 max-fr=30; max-fs=3600;
```

6.2.2. Offer/Answer Considerations

The VP8 codec offers a decode complexity that is roughly linear with the number of pixels encoded. The parameters "max-fr" and "max-fs" are defined in Section 6.1, where the macroblock size is 16x16 pixels as defined in [RFC6386], the max-fs and max-fr parameters MUST be used to establish these limits.

NOTE IN DRAFT: If closer control of width and height is desired, the mechanism described in draft-nandakumar-payload-sdp-max-video-resolution is a possible candidate for signalling, but since that document appears to be far from finalization, this document does not make a reference to that document. This note is only intended for facilitating WG discussion, and should be deleted before publication of this document as an RFC.

7. Security Considerations

RTP packets using the payload format defined in this specification are subject to the security considerations discussed in the RTP specification [RFC3550], and in any applicable RTP profile. The main security considerations for the RTP packet carrying the RTP payload format defined within this memo are confidentiality, integrity and source authenticity. Confidentiality is achieved by encryption of the RTP payload. Integrity of the RTP packets through suitable cryptographic integrity protection mechanism. Cryptographic system may also allow the authentication of the source of the payload. A suitable security mechanism for this RTP payload format should provide confidentiality, integrity protection and at least source authentication capable of determining if an RTP packet is from a member of the RTP session or not. Note that the appropriate mechanism to provide security to RTP and payloads following this memo may vary. It is dependent on the application, the transport, and the signaling protocol employed. Therefore a single mechanism is not sufficient, although if suitable the usage of SRTP [RFC3711] is recommended. This RTP payload format and its media decoder do not exhibit any significant non-uniformity in the receiver-side computational complexity for packet processing, and thus are unlikely to pose a denial-of-service threat due to the receipt of pathological data. Nor does the RTP payload format contain any active content.

8. Congestion Control

Congestion control for RTP SHALL be used in accordance with RFC 3550 [RFC3550], and with any applicable RTP profile; e.g., RFC 3551 [RFC3551]. The congestion control mechanism can, in a real-time encoding scenario, adapt the transmission rate by instructing the encoder to encode at a certain target rate. Media aware network elements MAY use the information in the VP8 payload descriptor in Section 4.2 to identify non-reference frames and discard them in order to reduce network congestion. Note that discarding of non-reference frames cannot be done if the stream is encrypted (because the non-reference marker is encrypted).

9. IANA Considerations

The IANA is requested to register the following values:
- Media type registration as described in Section 6.1.

10. References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, July 2003.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

[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, July 2006.

[RFC4855] Casner, S., "Media Type Registration of RTP Payload Formats", RFC 4855, February 2007.

[RFC6386] Bankoski, J., Koleszar, J., Quillio, L., Salonen, J., Wilkins, P., and Y. Xu, "VP8 Data Format and Decoding Guide", RFC 6386, November 2011.

[RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, January 2013.

Authors' Addresses

Patrik Westin
Google, Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043
USA

Email: patrik.westin@gmail.com

Henrik F Lundin
Google, Inc.
Kungsbron 2
Stockholm, 11122
Sweden

Email: hlundin@google.com

Michael Glover
Google, Inc.
5 Cambridge Center
Cambridge, MA 02142
USA

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

Frank Galligan
Google, Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043
USA

Payload Working Group
Internet-Draft
Intended status: Standards Track
Expires: June 10, 2014

S. Nandakumar
Cisco
C. Holmberg
Ericsson
December 7, 2013

Payload specific parameters for specifying video resolution in SDP. draft-nandakumar-payload-sdp-max-video-resolution-00

Abstract

With the rise in realtime communication applications supporting video, there is a need for receivers of the video to setup appropriate expectations on their receive capacity for handling various video image resolutions. Setting up the maximum supported image resolution that could be handled by an Endpoint is important to successfully decode and render the received video streams. This document proposes SDP format specific parameters for specifying the maximum image width and height resolutions along with their behavior under the [RFC3264] Offer/Answer SDP usage.

Status of this Memo

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

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

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

This Internet-Draft will expire on June 10, 2014.

Copyright Notice

Copyright (c) 2013 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 (http://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. Terminology 3
2. Introduction 3
3. Payload Format Parameters 3
3.1. Media Type Registration 3
4. SDP Parameters 4
4.1. Mapping of the parameters to SDP 4
4.2. Usage with SDP offer/answer 4
5. SDP Examples 4
5.1. Successful Scenario 4
5.2. Failure Scenario 5
6. IANA Considerations 5
7. Acknowledgements 6
8. Normative References 6
Authors' Addresses 6

1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

2. Introduction

Off late, multimedia communication sessions are increasingly supporting video by default. This is further fueled with the advent of IETF technology standards such as RTCWeb, CLUE. In order to successfully decode an incoming video stream, the decoder at the Endpoint must be capable of handling the image resolution of the received video streams, amongst other things. This document defines SDP payload format specific paramaters (a=fmtp) to setup an upper limit on the receiver's capability in successfully handling various image resolutions of the incoming video stream. It also describes [RFC3264] Offer/Answer procedures for the same.

Individual RTP Payload specifications that intend to specify limits on the decoder's image resolution handling MUST refer to the parameters defined in this document to achieve the functionality.

3. Payload Format Parameters

The media subtype parameters max-recv-width and max-recv-height specified below MAY be used to signal the capabilities of a receiver implementation. These parameters MUST NOT be used for any other purposes.

3.1. Media Type Registration

New Parameters

1. max-recv-width: The value of max-recv-width is an integer indicating maximum horizontal image range in pixels. When max-recv-width is signaled, the sender MUST NOT send any media with horizontal image resolutions higher than the value requested by the receiver.
2. max-recv-height: The value of max-recv-height is an integer indicating maximum vertical image range in pixels. When max-recv-height is signaled, the sender MUST NOT send any media with vertical image resolutions higher than the value requested by the receiver.

4. SDP Parameters

4.1. Mapping of the parameters to SDP

The parameters max-recv-width, max-recv-height when present, MUST be included in the "a=fmtp" line of SDP. These parameters are expressed as a media type string, in the form of a semicolon separated list of parameter=value pairs.

When signaled, both the attributes MUST be included and they signal the capabilities of a media receiver's implementation. These parameters are implicitly downgradable from the media sender's perspective, i.e, they express the upper limit for a media sender's possible behavior. Thus a media sender MAY select to set its encoder using only lower/lesser or equal values of these parameters when sending media.

4.2. Usage with SDP offer/answer

The interpretation of the parameters max-recv-width and max-recv-height depends on the SDP direction attribute. When the direction is sendrecv or recvonly, the value of this parameter indicates the ranges of horizontal and vertical image resolutions the media receiver is capable of rendering successfully. When the direction is sendonly, these attributes have no interpretation and MUST be ignored by the receiving Endpoint, if present.

If the media sender is not capable of sending any resolution lower than or equal to the values requested by the media receiver, the Offer/Answer procedure is considered as failed.

An SDP Answerer MUST NOT include these parameters in the SDP Answer unless they are specified in the associated SDP Offer.

If the SDP Answer doesn't contain these parameters, the Offerer MUST follow the procedures in the same way as if these parameters were never sent in the first place. This might happen if the Answerer doesn't support/understand these parameters.

5. SDP Examples

5.1. Successful Scenario

The example SDP below shows an Offer from an Endpoint that is capable of receiving up to [720,576] video image resolution for the VP8 codec with Payload Type 96.

```
m=video 62537 RTP/SAVPF 96
a=rtpmap:96 VP8/90000
a=fmtp:96 max-fr=30;max-recv-width=720;max-recv-height=576
a=sendrecv
```

SDP Offer

The example SDP below shows an Answer from an Endpoint that is capable of receiving only up to [640,480] video image resolutions.

```
m=video 62537 RTP/SAVPF 96
a=rtpmap:96 VP8/90000
a=fmtp:96 max-fr=30;max-recv-width=640;max-recv-height=480
a=sendrecv
```

SDP Answer

5.2. Failure Scenario

The example SDP below shows an Offer from an Endpoint that is capable of receiving up to [720,576] video image resolution for the H.264 codec with Payload Type 100.

```
m=video 62537 RTP/SAVPF 100
a=rtpmap:100 H264/90000
a=fmtp:100
  profile-level-id=42800d;max-mbps=40500;max-recv-width=720;max-recv-height=576
a=sendrecv
```

SDP Offer

The example SDP below shows the Answer rejecting the above SDP Offer, since the receiver of the SDP is unable to support the Offerer’s requested image resolutions for sending the media.

```
m=video 0 RTP/SAVPF 100
a=rtpmap:100 H264/90000
```

SDP Answer

6. IANA Considerations

The parameters specified in Section 3 of this document will be registered with the IANA.

7. Acknowledgements

The authors would like to thanks Cullen Jennings, Ali C. Began, Harald Alvestrand and Roni Evens for their review and valuable comments.

8. Normative References

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

Authors' Addresses

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

Email: snandaku@cisco.com

Christer Holmberg
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: christer.holmberg@ericsson.com