Stream: Internet Engineering Task Force (IETF)
RFC: 10034
Category: Standards Track
Published: August 2026
ISSN: 2070-1721
Authors: L. Ilola, *Nokia Technologies*; L. Kondrad, *Nokia Technologies*

RFC 10034
RTP Payload Format for Visual Volumetric Video-Based Coding (V3C)

Abstract

A visual volumetric video-based coding (V3C) ISO/IEC 23090-5 bitstream is composed of V3C units that contain V3C atlas sub-bitstreams, V3C video sub-bitstreams, and a V3C parameter set. This document describes an RTP payload format for V3C atlas sub-bitstreams. The RTP payload format for V3C video sub-bitstreams is defined by relevant IETF RFCs for the applicable video codec. The V3C RTP payload format allows for the packetization of one or more V3C atlas Network Abstraction Layer (NAL) units in an RTP packet payload as well as the fragmentation of a V3C atlas NAL unit into multiple RTP packets. The document also describes the mechanisms for grouping RTP streams of V3C component sub-bitstreams, providing a complete solution for streaming V3C-encoded content.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Volumetric video, similar to conventional 2D video, when uncompressed, is represented by a large amount of data. It enables the three-dimensional (3D) capture and playback of an object or scene, independent from the original capture position(s) or orientation(s). The visual volumetric video-based coding (V3C) specification [ISO.IEC.23090-5] leverages the compression efficiency of existing 2D video codecs to reduce the amount of data needed for storage and transmission of volumetric video. V3C is a generic mechanism for volumetric video coding, and it can be used by applications targeting volumetric content such as point clouds, Video-based Point Cloud Compression (V-PCC) [ISO.IEC.23090-5], and immersive video with depth (MPEG Immersive Video (MIV)) [ISO.IEC.23090-12].

A V3C encoder converts volumetric frames, i.e., 3D volumetric information, into a collection of 2D frames and associated data known as atlas data. The converted 2D frames are subsequently coded using any video or image codec, e.g., ISO/IEC International Standard 14496-10 (Advanced Video Coding, AVC/H.264) [ISO.IEC.14496-10], ISO/IEC International Standard 23008-2 (High Efficiency Video Coding, HEVC/H.265) [ISO.IEC.23008-2], or ISO/IEC International Standard 23090-3 (Versatile Video Coding, VVC/H.266) [ISO.IEC.23090-3]. The atlas data is coded with mechanisms specified in [ISO.IEC.23090-5].

V3C utilizes a high-level syntax (HLS) design, familiar from conventional 2D video codecs, to represent the associated coded data, i.e., atlas data. The coded atlas data is represented by Network Abstraction Layer (NAL) units. Consequently, the RTP payload format for V3C atlas data described in this document shares design philosophy, security, congestion control, and overall implementation complexity with the other NAL unit-based RTP payload formats such as the ones defined in [RFC6184], [RFC6190], and [RFC7798].

2. Conventions

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

All fields defined in this specification related to RTP payload structures **SHALL** be considered in network order.

3. Definitions and Abbreviations

3.1. Abbreviations

ACL: atlas coding layer

AP: aggregation packet

AU: aggregation unit

CVS: coded V3C sequence

DON: decoding order number

DOND: decoding order number difference

DONL: decoding order number least significant bits

IRAP: intra random access point

MTU: maximum transmission unit

NAL: network abstraction layer

NALU: NAL unit

RBSP: raw byte sequence payload

V3C: visual volumetric video-based coding

VPS: V3C parameter set

3.2. Definitions

3.2.1. General

This document uses the definitions of [ISO.IEC.23090-5]. Section 3.2.2 lists relevant definitions from [ISO.IEC.23090-5] for convenience.

3.2.2. Definitions from the V3C Specification

atlas: Collection of 2D bounding boxes and their associated information placed onto a rectangular frame and corresponding to a volume in 3D space on which volumetric data is rendered.

atlas bitstream: Sequence of bits that forms the representation of atlas frames and associated data forming one or more coded atlas sequences.

atlas coding layer NAL unit: Collective term for coded atlas tile layer NAL units and the subset of NAL units that have reserved values of nal_unit_type that are classified as being of type class equal to ACL in this document.

atlas frame: 2D rectangular array of atlas samples onto which patches are projected and additional information related to the patches, corresponding to a volumetric frame.

attribute: Scalar or vector property optionally associated with each point in a volumetric frame such as color, reflectance, surface normal, timestamps, material ID, etc.

coded atlas sequence: Sequence of coded atlas access units that consists, in decoding order, of an IRAP-coded atlas access unit, followed by zero or more coded atlas access units that are not IRAP-coded atlas access units, including all subsequent access units up to but not including any subsequent coded atlas access unit that is an IRAP-coded atlas access unit.

coded atlas access unit: Set of atlas NAL units that are associated with each other according to a specified classification rule, are consecutive in decoding order, and contain all atlas NAL units pertaining to one particular output time.

coded V3C sequence: Sequence of V3C atlas and video sub-bitstream(s) identified and separated by appropriate delimiters, required to start with a VPS, included in at least one V3C unit or provided through external means.

network abstraction layer unit: Syntax structure containing an indication of the type of data to follow and bytes containing that data in the form of an RBSP.

patch: Rectangular region within an atlas associated with volumetric information.

raw byte sequence payload: Syntax structure containing an integer number of bytes that is encapsulated in a NAL unit and that either is empty or has the form of a string of data bits containing syntax elements followed by an RBSP stop bit and zero or more subsequent bits equal to 0.

tile: Independently decodable rectangular region of an atlas frame.

V3C atlas sub-bitstream: Extracted sub-bitstream from the V3C bitstream containing a whole or portion of an atlas bitstream.

V3C video sub-bitstream: Extracted sub-bitstream from the V3C bitstream containing a whole or portion of a video bitstream.

V3C component: Atlas, occupancy, geometry, or attribute of a particular type that is associated with a V3C volumetric content representation.

V3C parameter set: Syntax structure containing syntax elements that apply to zero or more entire CVSs and may be referred to by syntax elements found in the V3C unit header.

volumetric frame: Set of 3D points specified by their Cartesian coordinates and zero or more corresponding sets of attributes at a particular time instance.

4. Media Format Description

4.1. Overview of the V3C Codec (Informative)

V3C encoding of a volumetric frame is achieved through a conversion of the volumetric frame from its 3D representation into multiple 2D representations and a generation of associated data documenting such conversions and transformations. The associated data, also known as the atlas data, provides information on how to reproject the 2D representations back into the 3D volumetric frame.

2D representations, known as V3C video components, of a volumetric frame are encoded using conventional 2D video codecs. A V3C video component may, for example, include occupancy, geometry, or attribute data. The occupancy data informs a V3C decoder which pixels in other V3C video components contribute to reconstructed 3D representation. The geometry data describes information on the position of the reconstructed voxels while attribute data provides additional properties for the voxels, e.g., color or material information. A voxel is the smallest discrete addressable element in a 3D space, analogous to a pixel in 2D.

Atlas data, known as V3C atlas component, provides information to interpret V3C video components and enables the reconstruction from a 2D representation back into a 3D representation of a volumetric frame. Atlas data is composed of a collection of patches. Each patch identifies a region in the V3C video components and provides information necessary to perform the appropriate inverse projection of the indicated region back into a 3D space. The shape of the patch region is determined by a 2D bounding box associated with each patch as well as their coding order. The shape of these patches is also further refined based on occupancy data.

To enable parallelization, random access, as well as a variety of other functionalities, an atlas frame can be divided into one or more rectangular partitions referred to as tiles. Tiles are not allowed to overlap and should be independently decodable. An atlas frame may contain regions that are not associated with any tile or patch.

The binary form of V3C video components, i.e., video bitstream, and V3C atlas components, i.e., atlas bitstream, can be grouped and represented by a single V3C bitstream. The V3C bitstream is composed of a set of V3C units. Each V3C unit has a V3C unit header and a V3C unit payload. The V3C unit header describes the V3C unit type for the payload. The V3C unit payload contains V3C video components, V3C atlas components, or a V3C parameter set. V3C video components, i.e., occupancy, geometry, or attribute components, correspond to video data units (e.g., NAL units defined in [ISO.IEC.23008-2]) that could be decoded by an appropriate video decoder. An example of a V3C bitstream consisting of a V3C parameter set, atlas bitstream, and three video component bitstreams (geometry, occupancy, attribute) is provided in Figure 1.

```
+--------------------+-------------------+-------------------+
| V3C Unit(V3C_VPS) | V3C Unit(V3C_AD) | V3C Unit(V3C_GVD) |
+--------------------+-------------------++------------------++---
| V3C Unit(V3C_OVD) | V3C Unit(V3C_AVD) | V3C Unit(V3C_AD)| ...
+--------------------+--------------------+------------------+----
```

Figure 1: Example of a V3C Bitstream

4.2. V3C Parameter Set (Informative)

This document specifies an encapsulation of V3C atlas data. Aspects related to signaling of V3C parameter set, defined in [ISO.IEC.23090-5], are also considered. A V3C parameter set is encapsulated in its own V3C unit, which allows decoupling the transmission of V3C parameter set from the V3C video and atlas components. The V3C parameter set can be transmitted by external means (e.g., as a result of the capability exchange) or through a (reliable or unreliable) control protocol. Section 9 of this document specifies how a V3C parameter set can be signaled using the Session Description Protocol (SDP).

Generally, it is useful to signal a V3C parameter set out of band, because it describes what overall resources are needed to decode and reconstruct the associated V3C bitstream. Signaling it dynamically as part of an RTP stream might result in undefined behavior when the receiver does not have the required capabilities to decode the received V3C video component sub-bitstreams or when reconstruction process relies on information that the receiver does not support.

4.3. V3C Atlas and Video Components (Informative)

4.3.1. General

In the V3C bitstream, the atlas component is identified by vuh_unit_type equal to V3C_AD, or V3C_CAD in the case of common atlas data, in the V3C unit header. The V3C atlas component consists of atlas NAL units that define header and payload pairs; see Section 4.3.2. V3C video components are identified by vuh_unit_type equal to V3C_OVD, V3C_GVD, V3C_AVD, and V3C_PVD. V3C video components can be further differentiated by other values in the V3C unit header such as vuh_attribute_index, vuh_attribute_partition_index, vuh_map_index, and vuh_auxiliary_video_flag. By mapping the V3C parameter set information to vuh_attribute_index, a V3C decoder identifies which attribute a given V3C video component contains, e.g., color.

The information supplied by a V3C unit header should be provided in one form or another to a V3C decoder, e.g., as part of SDP as described in Section 9. The four-byte V3C unit header syntax and semantics are copied below as defined in [ISO.IEC.23090-5], but the syntax is subject to change. Implementations should always refer to the latest specification of [ISO.IEC.23090-5]. The syntax of four-byte V3C unit header is provided here for informative purposes only. The integers in the parentheses, e.g., unsigned int(5), indicate the number of bits used by the syntax element.

```
v3c_unit_header( ) {
 unsigned int(5) vuh_unit_type;
 if( vuh_unit_type == V3C_AVD || vuh_unit_type == V3C_GVD ||
   vuh_unit_type == V3C_OVD || vuh_unit_type == V3C_AD ||
   vuh_unit_type == V3C_CAD || vuh_unit_type == V3C_PVD ) {
   unsigned int(4) vuh_v3c_parameter_set_id;
  }
  if( vuh_unit_type == V3C_AVD || vuh_unit_type == V3C_GVD ||
    vuh_unit_type == V3C_OVD || vuh_unit_type == V3C_AD ||
    vuh_unit_type == V3C_PVD ) {
    unsigned int(6) vuh_atlas_id;
  }
  if( vuh_unit_type == V3C_AVD ) {
    unsigned int(7) vuh_attribute_index;
    unsigned int(5) vuh_attribute_partition_index;
    unsigned int(4) vuh_map_index;
    unsigned int(1) vuh_auxiliary_video_flag;
  }
  else if( vuh_unit_type == V3C_GVD ) {
    unsigned int(4) vuh_map_index;
    unsigned int(1) vuh_auxiliary_video_flag;
    bit(12) vuh_reserved_zero_12bits;
  }
  else if( vuh_unit_type == V3C_OVD || vuh_unit_type == V3C_AD ||
      vuh_unit_type == V3C_PVD) {
    bit(17) vuh_reserved_zero_17bits;
  }
  else if( vuh_unit_type == V3C_CAD ) {
    bit(23) vuh_reserved_zero_23bits;
  }
  else {
    bit(27) vuh_reserved_zero_27bits;
  }
}
```

vuh_unit_type indicates the V3C unit type for the V3C component as specified in [ISO.IEC.23090-5]. For convenience, the mapping table from vuh_unit_type values to semantics is shown in Table 1.

vuh_unit_type	Identifier	V3C unit type	Description
0	V3C_VPS	V3C parameter set	V3C level parameters
1	V3C_AD	Atlas data	Atlas information
2	V3C_OVD	Occupancy video data	Occupancy information
3	V3C_GVD	Geometry video data	Geometry information

vuh_unit_type	Identifier	V3C unit type	Description
4	V3C_AVD	Attribute video data	Attribute information
5	V3C_PVD	Packed video data	Packing information
6	V3C_CAD	Common atlas data	Information that is common for atlases in a CVS. Specified in ISO/IEC 23090-12.
7...31	V3C_RSVD	Reserved	-

Table 1: V3C Unit Type Semantics

vuh_v3c_parameter_set_id specifies the value of vps_v3c_parameter_set_id for the active V3C VPS.

vuh_atlas_id specifies the ID of the atlas that corresponds to the current V3C unit.

vuh_attribute_index indicates the index of the attribute data carried in the Attribute Video Data unit.

vuh_attribute_partition_index indicates the index of the attribute dimension group carried in the attribute video data unit.

vuh_map_index, when present, indicates the map index of the current geometry or attribute stream. When not present, the map index of the current geometry or attribute sub-bitstream is derived based on the type of the sub-bitstream.

vuh_auxiliary_video_flag equal to 1 indicates that the associated geometry or attribute video sub-bitstream contains only RAW and/or enhanced occupancy map (EOM) coded points.

4.3.2. Atlas NAL Units

The atlas NAL unit (nal_unit(NumBytesInNalUnit)) is a byte-aligned syntax structure defined by [ISO.IEC.23090-5] to carry atlas data. The atlas NAL unit always contains a 16-bit NAL unit header (nal_unit_header()), which indicates the type of the NAL unit (nal_unit_type) among other things. The payload of a NAL unit refers to the NAL unit excluding the NAL unit header. The atlas NAL unit syntax and semantics are copied here as defined in [ISO.IEC.23090-5].

```
nal_unit_header(){
    bit(1) nal_forbidden_zero_bit;
    bit(6) nal_unit_type;
    bit(6) nal_layer_id;
    bit(3) nal_temporal_id_plus1;
}
nal_unit(NumBytesInNalUnit){
    nal_unit_header();
    NumBytesInRbsp = 0;
    for( i = 2; i < NumBytesInNalUnit; i++ )
      bit(8) rbsp_byte[ NumBytesInRbsp++ ];
}
```

nal_forbidden_zero_bit provides means for indicating errors in NAL units.

nal_unit_type indicates the type of the RBSP data structure contained in the NAL unit.

nal_layer_id indicates the identifier of the layer to which an ACL NAL unit belongs or the identifier of a layer to which a non-ACL NAL unit applies.

nal_temporal_id_plus1 minus 1 indicates a temporal identifier for the NAL unit.

4.4. Systems and Transport Interfaces (Informative)

In addition to releasing specifications on V3C applications [ISO.IEC.23090-5] and [ISO.IEC.23090-12], MPEG conducted further systems-level work on file formats to encapsulate compressed V3C content. The seventh edition of the ISO Base Media File Format (ISOBMFF) specification [ISO.IEC.14496-12] introduces a new media handler 'volv', intended to support volumetric visual media. It also specifies other structures to enable development of derived specifications detailing how various volumetric visual media may be stored in ISOBMFF.

One of such derived specifications is [ISO.IEC.23090-10], which defines how V3C content can be stored in a file and streamed over DASH [ISO.IEC.23009-1]. To a large extent, ISO/IEC 23090-10 focuses on describing how ISOBMFF boxes and syntax elements may be used to store volumetric media, but in some cases, new boxes and syntax elements are introduced to accommodate the fundamentally different type of new media. While the specification is not directly relevant for defining RTP payload format for V3C atlas data, it is a useful resource that may be considered especially when designing ingestion of encoded V3C content into RTP streaming pipelines.

5. V3C Atlas RTP Payload Format

5.1. General

This section describes details related to V3C atlas RTP payload format definitions. Aspects related to the RTP header, RTP payload header, and general payload structure are considered. RTP payload format(s) for video components is defined in its respective RTP payload format specifications depending on the video codec used.

5.2. RTP Header

The format of the RTP header is specified in [RFC3550] and replicated in Figure 2 for convenience. V3C RTP payload format uses the fields of the RTP header in a manner consistent with [RFC3550]. Unless contextualized below, the meaning of the fields depicted in Figure 2 is the same as in Section 5.1 of [RFC3550].

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|X|  CC   |M|     PT      |       sequence number         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           timestamp                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           synchronization source (SSRC) identifier            |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|            contributing source (CSRC) identifiers             |
|                             ....                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: RTP Header

Marker bit (M): 1 bit

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

Payload Type (PT): 7 bits

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

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 Supplemental Enhancement Information (SEI) NAL units), the RTP timestamp **MUST** be set to the RTP timestamp of the coded atlas of the access unit in which the NAL unit (according to Section 8.4.5.3 of [ISO.IEC.23090-5]) is included.

Receivers **MUST** use the RTP timestamp for the display process, even when the bitstream contains atlas frame timing SEI messages as specified in [ISO.IEC.23090-5].

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

5.3. RTP Payload Header

The first two bytes of the payload of an RTP packet are referred to as the payload header. The payload header consists of the same fields (F, NUT, NLI, and TID) as the NAL unit header as shown in Section 4.3.2, irrespective of the type of the payload structure. For convenience, the structure of RTP payload header is shown in Figure 3.

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|F|     NUT   |     NLI     | TID |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: RTP Payload Header

F: The nal_forbidden_zero_bit as specified in [ISO.IEC.23090-5] is equal to 0. A value equal to 1 indicates that the payload may contain errors or syntax violations.

Media processing elements in the network that are capable of deep packet inspection **SHOULD** set the F bit to 1 to indicate detected bit errors in the NAL unit(s). A receiver reaction to an RTP payload header in which the F bit is equal to 1 is to discard such RTP packet and to conceal the lost data in the discarded NAL unit(s).

NUT: The nal_unit_type as specified in [ISO.IEC.23090-5] defines the type of the RBSP data structure contained in the NAL unit payload. The NUT value could carry other meaning depending on the RTP packet type.

NLI: The nal_layer_id as specified in [ISO.IEC.23090-5] defines the identifier of the layer to which an ACL NAL unit belongs or the identifier of a layer to which a non-ACL NAL unit applies.

TID: The nal_temporal_id_plus1 minus 1 as specified in [ISO.IEC.23090-5] defines a temporal identifier for the NAL unit. The value of nal_temporal_id_plus1 **MUST NOT** be equal to 0.

5.4. Payload Structures

5.4.1. General

Three different types of RTP packet payload structures are specified. A receiver can identify the payload structure by the first two bytes of the RTP packet payload, which co-serves as the RTP payload header. These two bytes are always structured as a NAL unit header. The NAL unit type field indicates which structure is present in the payload.

The three different payload structures are as follows:

Single NAL Unit Packet:

Contains a single NAL unit in the payload. This payload structure is specified in Section 5.4.2.

Aggregation Packet: Contains multiple NAL units in a single RTP payload. This payload structure is specified in Section 5.4.3.

Fragmentation Unit (FU): Contains a subset of a single NAL unit. This payload structure is specified in Section 5.4.4.

> NOTE (informative): This document does not limit the size of NAL units encapsulated in NAL unit packets and fragmentation units. [ISO.IEC.23090-5] does not restrict the maximum size of a NAL unit directly, either. Instead, a NAL unit sample stream format may be used, which provides flexibility to signal NAL unit size up to UINT64_MAX bytes.
>
> NOTE (informative): Some of the fields described in the payload structures are conditional and their presence is indicated through the relevant parameters as defined in Section 7.3. These parameters are assumed to be made available prior to sending any RTP packets.

5.4.2. Single NAL Unit Packet

A single NAL unit packet contains exactly one NAL unit and consists of an RTP payload header and the following conditional fields: 16-bit DONL and 16-bit v3c-tile-id. The rest of the payload data contains the NAL unit payload data (excluding the NAL unit header). A single NAL unit packet **MUST** only contain atlas NAL units of the types defined in Table 4 of [ISO.IEC.23090-5]. The structure of the single NAL unit packet is shown in Figure 4.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      RTP payload header       |      DONL (conditional)       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
|      v3c-tile-id (cond)       |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
|                                                               |
|                        NAL unit data                          |
|                                                               |
|                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               :...OPTIONAL RTP padding        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: Single NAL Unit Packet

The RTP payload header **MUST** be an exact copy of the NAL unit header of the contained NAL unit.

A NAL unit stream composed by de-packetizing single NAL unit packets in RTP sequence number order **MUST** conform to the NAL unit decoding order when DONL is not present.

The DONL field, when present, specifies the value of the 16-bit decoding order number of the contained NAL unit. The decoding order number indicates the order in which the received NAL units should be reordered to form a bitstream that can be successfully decoded. If sprop-max-don-diff is greater than 0 for any of the RTP streams, the DONL field **MUST** be present; otherwise, the DONL field **MUST NOT** be present.

The v3c-tile-id field, when present, specifies the 16-bit tile identifier for the NAL unit as signaled in the V3C atlas tile header defined in [ISO.IEC.23090-5]. If sprop-v3c-tile-id-pres is equal to 1 and the RTP payload header NUT is in range 0-35 inclusive, the v3c-tile-id field **MUST** be present. Otherwise, the v3c-tile-id field **MUST NOT** be present.

> NOTE (informative): Only values for NAL unit type (NUT) in range 0-35 inclusive are allocated for atlas tile layer data in [ISO.IEC.23090-5].

The presence of the "**OPTIONAL** RTP padding" is indicated by the padding (P) bit in the RTP header. As defined in [RFC3550], the last octet of the padding contains a count of how many padding octets should be ignored, including itself.

5.4.3. Aggregation Packet

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

APs **MAY** be used to wrap multiple NAL units belonging to the same access unit in a single RTP payload. The first two bytes of an AP **MUST** contain the RTP payload header. The NAL unit type (NUT) for the NAL unit header contained in the RTP payload header **MUST** be equal to 56, which falls in the unspecified range of the NAL unit types defined in [ISO.IEC.23090-5]. An AP **MAY** contain a conditional v3c-tile-id field. An AP **MUST** contain two or more AUs. The structure of an AP is shown in Figure 5.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  RTP payload header (NUT=56)  |      v3c-tile-id (cond)       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                  Two or more aggregation units                |
|                                                               |
|                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               :...OPTIONAL RTP padding        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: Aggregation Packet (AP)

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 NUT field **MUST** be equal to 56. The value of NLI **MUST** be equal to the lowest value of NLI of all the aggregated NAL units. The value of TID **MUST** be the lowest value of TID of all the aggregated NAL units.

All ACL NAL units in an aggregation packet have the same TID value since they belong to the same access unit. However, the packet **MAY** contain non-ACL NAL units for which the TID value in the NAL unit header **MAY** be different than the TID value of the ACL NAL units in the same AP.

The v3c-tile-id field, when present, specifies the 16-bit tile identifier for all ACL NAL units in the AP. If sprop-v3c-tile-id-pres is equal to 1, the v3c-tile-id field **MUST** be present. Otherwise, the v3c-tile-id field **MUST NOT** be present.

The presence of the "**OPTIONAL** RTP padding" is indicated by the padding (P) bit in the RTP header. As defined in [RFC3550] the last octet of the padding contains a count of how many padding octets should be ignored, including itself.

An AP **MUST** carry at least two AUs and can carry as many AUs as necessary. However, the total amount of data in an AP **MUST** fit into an IP packet, and the size **SHOULD** be chosen so that the resulting IP packet is smaller than the local MTU size so to avoid IP layer fragmentation. The structure of the AU depends both on the presence of the decoding order number, the sequence order of the AU in the AP, and the presence of v3c-tile-id field. The structure of an AU is shown in Figure 6.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  DOND (cond)  /  DONL (cond)  |      v3c-tile-id (cond)       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
|            NALU size          |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
|                                                               |
|                            NAL unit                           |
|                                                               |
|                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: Aggregation Unit (AU)

If sprop-max-don-diff is greater than 0 for any of the RTP streams, an AU begins with the DOND/DONL field. The first AU in the AP contains DONL field, which specifies the 16-bit value of the decoding order number of the aggregated NAL unit. The variable DON for the aggregated NAL unit is derived as equal to the value of the DONL field. All subsequent AUs in the AP **MUST** contain an (8-bit) DOND field, which 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. 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.

When sprop-max-don-diff is equal to 0 for all the RTP streams, DOND/DONL fields **MUST NOT** be present in an aggregation unit. The aggregation units **MUST** be stored in the aggregation packet so that the decoding order of the containing NAL units is preserved. This means that the first aggregation unit in the aggregation packet **SHOULD** contain the NAL unit that **SHOULD** be decoded first.

If sprop-v3c-tile-id-pres is equal to 2 and the AU NAL unit header type is in range 0-35 inclusive, the 16-bit v3c-tile-id field **MUST** be present in the aggregation unit after the conditional DOND/DONL field; otherwise, the v3c-tile-id field **MUST NOT** be present in the aggregation unit.

The conditional fields of the aggregation unit are followed by a 16-bit NALU size field, which provides the size of the NAL unit (in bytes) in the aggregation unit. The remainder of the data in the aggregation unit **SHOULD** contain the NAL unit (including the unmodified NAL unit header).

5.4.4. Fragmentation Unit

FUs are introduced to enable fragmenting a single NAL unit into multiple RTP packets, possibly without co-operation or knowledge of the 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. Aggregation packets **MUST NOT** be fragmented. FUs **MUST NOT** be nested; i.e., an FU **MUST NOT** contain a subset of another FU. The RTP header timestamp of an RTP packet carrying an FU is set to the NALU-time of the fragmented NAL unit.

An FU consists of an RTP payload header with NUT equal to 57, an 8-bit FU header, a conditional 16-bit DONL field, a conditional 16-bit v3c-tile-id field, and an FU payload. The structure of an FU is illustrated below in Figure 7.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  RTP payload header (NUT=57)  |   FU header   |  DONL (cond)  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
|  DONL (cond)  |    v3c-tile-id (cond)         |               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
|                                                               |
|                          FU payload                          |
|                                                               |
|                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               :...OPTIONAL RTP padding        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7: Fragmentation Unit

The fields in the RTP payload header are set as follows. The NUT field **MUST** be equal to 57. The rest of the fields **MUST** be equal to the fragmented NAL unit.

The presence of the "**OPTIONAL** RTP padding" is indicated by the padding (P) bit in the RTP header. As defined in [RFC3550], the last octet of the padding contains a count of how many padding octets should be ignored, including itself.

The FU header consists of an S bit, an E bit, and a 6-bit fragmentation unit type (FUT) field. The structure of FU header is illustrated in Figure 8.

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

Figure 8: Fragmentation Unit Header

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

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

The field FUT **MUST** be equal to the nal_unit_type of the fragmented NAL unit.

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

The DONL field, when present, specifies the value of the 16-bit decoding order number of the fragmented NAL unit. If sprop-max-don-diff is greater than 0 for any of the RTP streams and the S bit is equal to 1, the DONL field **MUST** be present in the FU, and the variable DON for the fragmented NAL unit is derived as equal to the value of the DONL field. Otherwise (sprop-max-don-diff is equal to 0 for all the RTP streams, or the S bit is equal to 0), the DONL field **MUST NOT** be present in the FU.

The v3c-tile-id field, when present, specifies the 16-bit tile identifier for the fragmented NAL unit. If sprop-v3c-tile-id-pres is equal to 1, FUT is in range 0-35, and the S bit is equal to 1, the v3c-tile-id field **MUST** be present after the conditional DONL field. Otherwise, the v3c-tile-id field **MUST NOT** be present.

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 the F, NLI, and TID fields of the RTP payload headers of the FUs and the FUT field of the FU header. An FU payload **MUST NOT** be empty.

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

5.4.5. Example of Fragmentation Unit (Informative)

This example illustrates how a fragmentation unit may be used to divide one NAL unit into two RTP packets. Figure 9 depicts the structure of the first packet with the first part of the fragmented NAL unit.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|X|  CC   |M|     PT      |       sequence number         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           timestamp                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           synchronization source (SSRC) identifier            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            contributing source (CSRC) identifiers             |
|                             ....                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  RTP payload header (NUT=57)  |1|0|    FUT    |               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
|                                                               |
|                          FU payload                           |
|                                                               |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 9: First Packet of Fragmented NAL Unit

Figure 10 depicts the structure of the second packet with the rest of the fragmented NAL unit.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|X|  CC   |M|     PT      |       sequence number         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           timestamp                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           synchronization source (SSRC) identifier            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            contributing source (CSRC) identifiers             |
|                             ....                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  RTP payload header (NUT=57)  |0|1|    FUT    |               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
|                                                               |
|                          FU payload                           |
|                                                               |
|                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               :...OPTIONAL RTP padding        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 10: Second Packet of Fragmented NAL Unit

5.5. Decoding Order Number

For each atlas NAL unit, the variable AbsDon is derived, representing the decoding order number that is indicative of the NAL unit decoding order. Let NAL unit n be the n-th NAL unit in transmission order within an RTP stream.

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

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

```
If (n == 0)
  AbsDon[n] = DON[0]
Else
  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.

6. Packetization and De-Packetization Rules

The following packetization rules apply for V3C atlas data:

- If sprop-max-don-diff is greater than 0 for any of the RTP streams, the transmission order of NAL units carried in the RTP stream **MAY** be different than the NAL unit decoding order and the NAL unit output order. Otherwise (sprop-max-don-diff is equal to 0 for all the RTP streams), 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-ACL NAL units such as access unit delimiters, parameter sets, or SEI NAL units are typically small and can often be aggregated with ACL NAL units without violating MTU size constraints.
- Each non-ACL NAL unit **SHOULD**, when possible, from an MTU size perspective, be encapsulated in an aggregation packet together with its associated ACL NAL unit, as typically a non-ACL NAL unit would be meaningless without the associated ACL NAL unit being available.
- For carrying exactly one NAL unit in an RTP packet, a single NAL unit packet (Section 5.4.2) **MUST** be used.

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

The de-packetization process is implementation dependent. Therefore, the following de-packetization rules **SHOULD** be taken as an example.

- All normal RTP mechanisms related to buffer management apply. In particular, duplicated or outdated RTP packets (as indicated by the RTP sequence number and the RTP timestamp) are removed. To determine the exact time for decoding, factors such as a possible intentional delay to allow for proper inter-stream synchronization must be factored in.
- NAL units with NAL unit type values in the range of 0 to 55 inclusive may be passed to the decoder. NAL-unit-like structures with NAL unit type values in the range of 56 to 63 inclusive **MUST NOT** be passed to the decoder.

- When sprop-max-don-diff is equal to 0 for the received RTP stream, the NAL units carried in the RTP stream **MAY** be directly passed to the decoder in their transmission order, which is identical to their decoding order.
- When sprop-max-don-diff is greater than 0 for any of the received RTP streams, the received NAL units need to be arranged into decoding order before handing them over to the decoder.
- For further de-packetization examples, the reader is referred to Section 6 of [RFC7798].

Regarding the packetization of V3C video component data, the respective RTP video payload specification(s) define how packetization and de-packetization should be handled.

7. Payload Format Parameters

This section specifies the optional parameters. A mapping of the parameters into the Session Description Protocol (SDP) [RFC8866] is also provided for applications that use SDP. Equivalent parameters could be defined elsewhere for use with control protocols that do not use SDP.

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

7.1. Media Type Registration

See Section 10.1 for information related to media type registration.

7.2. Required Parameters Definition

sprop-v3c-parameter-set:
: sprop-v3c-parameter-set provides V3C parameter set bytes as defined in [ISO.IEC.23090-5]. The value contains a base64-encoded [RFC4648] representation of the v3c_parameter_set() syntax element.

7.3. Definitions of Optional Parameters

sprop-v3c-unit-header:
: sprop-v3c-unit-header provides bytes corresponding to a V3C unit header as defined in [ISO.IEC.23090-5]. The value contains a base64-encoded [RFC4648] representation of the 4 bytes of V3C unit header. V3C unit header indicates the details of which V3C component the media corresponds to.

  sprop-v3c-unit-header contains the same information as sprop-v3c-unit-type, sprop-v3c-vps-id, sprop-v3c-atlas-id, sprop-v3c-attr-idx, sprop-v3c-attr-part-idx, sprop-v3c-map-idx, and sprop-v3c-aux-video-flag combined. To avoid the potential of signaling conflicting information, the separate parameters **MUST NOT** be present when sprop-v3c-unit-header is present.

sprop-v3c-unit-type:
: sprop-v3c-unit-type provides a V3C unit type value corresponding to vuh_unit_type defined in [ISO.IEC.23090-5], i.e., defines a V3C sub-bitstream type such as geometry, occupancy, atlas data, or attribute.

  When sprop-v3c-unit-header is present, sprop-v3c-unit-type **MUST NOT** be present. When present, the value of sprop-v3c-unit-type **SHALL** be in the range of 1 to 31 inclusive.

sprop-v3c-vps-id:
: sprop-v3c-vps-id provides a value corresponding to active vuh_v3c_parameter_set_id defined in [ISO.IEC.23090-5], i.e., defines the value of the active V3C parameter set id.

  When sprop-v3c-unit-header is present, sprop-v3c-vps-id **MUST NOT** be present. When present, the value of sprop-v3c-vps-id **SHALL** be in the range of 0 to 15 inclusive.

sprop-v3c-atlas-id:
: sprop-v3c-atlas-id provides a value corresponding to vuh_atlas_id defined in [ISO.IEC.23090-5]. When a V3C bitstream consists of multiple atlases, this parameter indicates the atlas id for the media component.

  When sprop-v3c-unit-header is present, sprop-v3c-atlas-id **MUST NOT** be present. When present, the value of sprop-v3c-atlas-id **SHALL** be in the range of 0 to 63 inclusive.

sprop-v3c-attr-idx:
: sprop-v3c-attr-idx provides a value corresponding to vuh_attribute_index defined in [ISO.IEC.23090-5]. An attribute in V3C determines a feature of a reconstructed volumetric primitive; for example, this could be texture (color), transparency, reflectance, or normal. The attribute index defines which type of attribute the media corresponds to.

  When sprop-v3c-unit-header is present, sprop-v3c-attr-idx **MUST NOT** be present. When present, the value of sprop-v3c-attr-idx **SHALL** be in the range of 0 to 127 inclusive.

sprop-v3c-attr-part-idx:
: sprop-v3c-attr-part-idx provides a value corresponding to vuh_attribute_partition_index defined in [ISO.IEC.23090-5]. In V3C, an attribute can be partitioned into multiple components. This may for example be useful, when an attribute consists of four dimensions but the video codec only supports coding three channels of data.

  When sprop-v3c-unit-header is present, sprop-v3c-attr-part-idx **MUST NOT** be present. When present, the value of sprop-v3c-attr-part-idx **SHALL** be in the range of 0 to 31 inclusive.

sprop-v3c-map-idx:
: sprop-v3c-map-idx provides a value corresponding to vuh_map_index defined in [ISO.IEC.23090-5]. Maps in V3C allow storing multiple layers of projected volumetric data.

When sprop-v3c-unit-header is present, sprop-v3c-map-idx **MUST NOT** be present. When present, the value of sprop-v3c-map-idx **SHALL** be in the range of 0 to 15 inclusive.

sprop-v3c-aux-video-flag:
: sprop-v3c-aux-video-flag provides a value corresponding to vuh_auxiliary_video_flag defined in [ISO.IEC.23090-5]. Auxiliary video in V3C can be used to pack volumetric data directly in a video frame without projecting it into a 2D plane first.

  When sprop-v3c-unit-header is present, sprop-v3c-aux-video-flag **MUST NOT** be present. When present, the value of sprop-v3c-aux-video-flag **SHALL** be either 0 or 1.

sprop-v3c-tile-id:
: sprop-v3c-tile-id indicates that the RTP stream contains only portion of the tiles in an atlas. The value of sprop-v3c-tile-id contains a comma-separated (',') list of integer values, which indicate the tile ids that are present in the corresponding RTP stream.

  When sprop-v3c-tile-id is not present, the RTP stream is expected to contain all tiles or only consist of a single tile.

sprop-v3c-tile-id-pres:
: sprop-v3c-tile-id-pres indicates that the RTP packets contain v3c-tile-id field.

  When present, the value of sprop-v3c-tile-id-pres **SHALL** be either 0 or 1. When not present, the default value of sprop-v3c-tile-id-pres is 0.

sprop-v3c-atlas-data:
: sprop-v3c-atlas-data **MAY** be used to convey any atlas data NAL units of the V3C atlas sub-bitstream for out-of-band transmission. The value contains a comma-separated (',') list of base64-encoded [RFC4648] representations of the atlas NAL units as specified in [ISO.IEC.23090-5].

  When present, the atlas NAL units stored in the sprop-v3c-atlas-data shall be applied for duration of the entire stream until an in-band atlas NAL unit with the same NAL unit type overrides it.

sprop-v3c-common-atlas-data:
: sprop-v3c-common-atlas-data **MAY** be used to convey common atlas data NAL units of the V3C common atlas sub-bitstream for out-of-band transmission. The value contains a comma-separated (',') list of base64-encoded [RFC4648] representations of the common atlas NAL units (i.e., NAL_CASPS and NAL_CAF_IDR) as specified in [ISO.IEC.23090-5].

  When present, the common atlas NAL units stored in the sprop-v3c-common-atlas-data shall be applied for duration of the entire stream until an in-band common atlas NAL unit with the same NAL unit type overrides it.

sprop-v3c-sei:
sprop-v3c-sei **MAY** be used to convey SEI NAL units of V3C atlas and common atlas sub-bitstreams for out-of-band transmission. The value is a comma-separated (',') list of base64-encoded [RFC4648] representations of SEI NAL units (i.e., NAL_PREFIX_NSEI and NAL_SUFFIX_NSEI, NAL_PREFIX_ESEI, NAL_SUFFIX_ESEI) as specified in [ISO.IEC.23090-5].

When present, the SEI NAL units stored in the sprop-v3c-sei shall be applied for duration of the entire stream until an in-band SEI NAL unit with the same SEI payload type overrides it.

v3c-ptl-level-idc:
v3c-ptl-level-idc provides a value corresponding to ptl_level_idc defined in [ISO.IEC.23090-5]. The value of v3c-ptl-level-idc indicates the level to which the V3C bitstream conforms.

When present, the value of v3c-ptl-level-idc **SHALL NOT** conflict the corresponding value in the sprop-v3c-parameter-set. The value of v3c-ptl-level-idc **SHALL** be in the range of 0 to 255 inclusive.

v3c-ptl-tier-flag:
v3c-ptl-tier-flag provides a value corresponding to ptl_tier_flag defined in [ISO.IEC.23090-5]. The value of v3c-ptl-tier-flag indicates the tier context necessary to interpret the value of v3c-ptl-level-idc.

When present, the value of v3c-ptl-tier-flag **SHALL NOT** conflict the corresponding value in the sprop-v3c-parameter-set. The value of v3c-ptl-tier-flag **SHALL** be either 0 or 1.

v3c-ptl-codec-idc:
v3c-ptl-codec-idc provides a value corresponding to ptl_profile_codec_group_idc defined in [ISO.IEC.23090-5]. The value of v3c-ptl-codec-idc indicates the codec group profile component to which the V3C bitstream conforms.

When present, the value of v3c-ptl-codec-idc **SHALL NOT** conflict the corresponding value in the sprop-v3c-parameter-set. The value of v3c-ptl-codec-idc **SHALL** be in the range of 0 to 127 inclusive.

v3c-ptl-toolset-idc:
v3c-ptl-toolset-idc provides a value corresponding to ptl_profile_toolset_idc defined in [ISO.IEC.23090-5]. The value of v3c-ptl-toolset-idc indicates the toolset combination profile component to which the V3C bitstream conforms.

When present, the value of v3c-ptl-toolset-idc **SHALL NOT** conflict the corresponding value in the sprop-v3c-parameter-set. The value of v3c-ptl-toolset-idc **SHALL** be in the range of 0 to 255 inclusive.

v3c-ptl-rec-idc:
: v3c-ptl-rec-idc provides a value corresponding to ptl_profile_reconstruction_idc as defined in [ISO.IEC.23090-5]. The value of v3c-ptl-rec-idc indicates the reconstruction profile component to which the V3C bitstream is recommended to conform.

  When present, the value of v3c-ptl-rec-idc **SHALL NOT** conflict the corresponding value in the sprop-v3c-parameter-set. The value of v3c-ptl-rec-idc **SHALL** be in the range of 0 to 255 inclusive.

sprop-max-don-diff:
: If the transmission order of NAL units in the RTP stream(s) is the same as the decoding and NAL unit output order, this parameter must be equal to 0.

  Otherwise, if the decoding order of the NAL units of the RTP stream(s) is the same as the NAL unit transmission order but not the same as NAL unit output order, the value of this parameter **MUST** be equal to 1.

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

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

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

7.4. Mapping of Parameters to V3C Syntax

Parameter	Required	V3C Syntax Counterpart	ISO/IEC 23090-5 Section Reference
sprop-v3c-parameter-set	YES	v3c_parameter_set()	8.3.4.1
sprop-v3c-unit-header	NO	v3c_unit_header()	8.3.2.2
sprop-v3c-unit-type	NO	vuh_unit_type	8.4.2.2
sprop-v3c-vps-id	NO	vuh_v3c_parameter_set_id	8.4.2.2
sprop-v3c-atlas-id	NO	vuh_atlas_id	8.4.2.2
sprop-v3c-attr-idx	NO	vuh_attribute_index	8.4.2.2
sprop-v3c-attr-part-idx	NO	vuh_attribute_partition_index	8.4.2.2
sprop-v3c-map-idx	NO	vuh_map_index	8.4.2.2

Parameter	Required	V3C Syntax Counterpart	ISO/IEC 23090-5 Section Reference
sprop-v3c-aux-video-flag	NO	vuh_auxiliary_video_flag	8.4.2.2
sprop-v3c-tile-id	NO	-	-
sprop-v3c-tile-id-pres	NO	-	-
sprop-v3c-atlas-data	NO	nal_unit()	8.3.5.1
sprop-v3c-common-atlas-data	NO	nal_unit()	8.3.5.1
sprop-v3c-sei	NO	nal_unit()	8.3.5.1
v3c-ptl-level-idc	NO	ptl_level_idc	8.4.4.2
v3c-ptl-tier-flag	NO	ptl_tier_flag	8.4.4.2
v3c-ptl-codec-idc	NO	ptl_profile_codec_group_idc	8.4.4.2
v3c-ptl-toolset-idc	NO	ptl_profile_toolset_idc	8.4.4.2
v3c-ptl-rec-idc	NO	ptl_profile_reconstruction_idc	8.4.4.2
sprop-max-don-diff	NO	-	-

Table 2: Mapping of Parameters to V3C Syntax

8. Congestion Control Considerations

Congestion control for RTP **SHALL** be used in accordance with RTP [RFC3550] and with any applicable RTP profile, e.g., AVP [RFC3551]. This section only applies for unicast streaming, leaving considerations for multicast streaming out of scope.

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 that of the RTP flow (see Section 10 of [RFC3550]).

This condition can be satisfied by implementing congestion-control mechanisms to adapt the transmission rate. A simple bitrate adaptation for congestion control can be achieved when real-time coding is used for V3C video components where quality parameters can be adaptively tuned. Video coding specifications **MAY** define further adaptation techniques.

An alternative method is to arrange for a receiver to leave the session if the loss rate is unacceptably high, for example, using a Circuit Breaker [RFC8083] that defines criteria for when one the RTP flow must stop sending RTP Packet Streams.

As an example, both the sender and the receiver may have their own definition for an acceptable packet loss rate. In such a case, the receiver may decide to quit a stream when it finds the packet loss rate too high. Similarly, the sender may decide to drop a receiver when the reports it receives indicate packet loss rates that are too high from its perspective. These decisions can be made independently by either the receiver or the sender.

As an example of a bitrate adaptation technique, a sender or a receiver may adapt bitrates of specific sub-streams. The adaptation should be done in a manner that considers the effects on the quality of the experience as a whole, keeping the subjective quality of experience as high as possible. In an implementation this could mean dropping less important sub-streams fully or reducing the bitrates of the most important sub-streams throughout the session.

9. Session Description Protocol

A new attribute "v3cfmtp" is defined for carrying V3C format media type parameters in the corresponding fields of SDP [RFC8866]. Grouping framework [RFC5888] is used to indicate which media lines (video and application) in the SDP constitute a V3C representation.

9.1. V3C Format Parameters "v3cfmtp" Attribute

This document defines a new attribute for SDP, intended to carry V3C-specific media format parameters. Its functionality is similar to "a=fmtp", with the exception that it **SHALL** be used without the fmt-token and that it can be used also on a session level. The attribute allows V3C-specific media format parameters to be associated with any media line in SDP. The detailed information on the new attribute (a=v3cfmtp) is provided in Section 10.2.

The value of the v3cfmtp attribute is a byte-string, as defined in [RFC8866], which contains at least one V3C-specific media format parameter as a "parameter=value" pair as defined in this document. Multiple semicolon-separated V3C media "parameter=value" pairs can be stored in the byte-string to be conveyed by SDP and given unchanged to the media tool that will use this format. Whitespace in the byte-string is ignored.

An example of the usage of the new attribute is shown below. The first line describes session-level usage of the attribute, signaling a V3C parameter set. The second line describes a media-level attribute, signaling a V3C unit header and profile tier level flag for the associated media line.

```
a=v3cfmtp:sprop-v3c-parameter-set=AUH/AAAP/zwAAAAAACgIAtEAgQLAIAAUQ
BACWAM5QEDgQCAIAAAAABP8CzwAAAAAAAAAQAAAtAE/wLPAAAAAAAg=;

a=v3cfmtp:sprop-v3c-unit-header=CAAAAA==;v3c-ptl-tier-flag=1;
```

9.2. Mapping of Payload Type Parameters to SDP

9.2.1. For V3C Atlas Components

- The media name in the "m=" line of SDP **MUST** be application.
- The encoding name in the "a=rtpmap" line of SDP **MUST** be v3c.
- The clock rate in the "a=rtpmap" line **MUST** be 90000.
- The **OPTIONAL** parameters sprop-v3c-atlas-data, sprop-v3c-common-atlas-data, sprop-v3c-sei, sprop-v3c-tile-id, sprop-v3c-tile-id-pres, 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.
- The **OPTIONAL** parameters sprop-v3c-unit-header, sprop-v3c-unit-type, sprop-v3c-vps-id, sprop-v3c-atlas-id, sprop-v3c-attr-idx, sprop-v3c-attr-part-idx, sprop-v3c-map-idx, sprop-v3c-aux-video-flag, sprop-max-don-diff, sprop-v3c-parameter-set, v3c-ptl-level-idc, v3c-ptl-tier-flag, v3c-ptl-codec-idc, v3c-ptl-toolset-idc, and v3c-ptl-rec-idc, when present, **MUST** be included in the "a=v3cfmtp" line of SDP. This parameter is expressed as a media type string in the form of a semicolon-separated list of parameter=value pairs.

The **OPTIONAL** parameters, when present in the V3C atlas component media line format parameters attribute, specify values that are valid for the coded V3C sequence until a new value is received in-band. Some **OPTIONAL** parameters, like sprop-v3c-parameter-set or sprop-v3c-unit-header, can't be carried in-band in the atlas stream and thus may be considered static for the session. The carriage of V3C payload format parameters in "a=fmtp" and "a=v3cfmtp" attributes is separated by logical context, where "a=fmtp" consists of atlas level media format parameters and "a=v3cfmtp" contains V3C level media format parameters.

An example of media representation corresponding to atlas data component (V3C_AD), where static V3C parameter set and V3C unit header is carried out-of-band in SDP, is as follows:

```
m=application 49170 RTP/AVP 98
a=rtpmap:98 v3c/90000
a=fmtp:98 sprop-v3c-tile-id=0,1
a=v3cfmtp:sprop-v3c-unit-header=CAAAAA==;v3c-ptl-tier-flag=1;
  sprop-v3c-parameter-set=AQD/AAAP/zwAAAAAADwIAQ5BwAAOADjgQAADkA==
```

9.2.2. For V3C Video Components

- The media name in the "m=" line of SDP **MUST** be video.
- The encoding name in the "a=rtpmap" line of SDP can be any video subtype, e.g., H.264, H.265, H.266, etc.
- The clock rate in the "a=rtpmap" line **MUST** be 90000.
- The **OPTIONAL** parameters sprop-v3c-unit-header, sprop-v3c-unit-type, sprop-v3c-vps-id, sprop-v3c-atlas-id, sprop-v3c-attr-idx, sprop-v3c-attr-part-idx, sprop-v3c-map-idx, sprop-v3c-aux-video-flag, sprop-max-don-diff, sprop-v3c-parameter-set, sprop-v3c-atlas-data, sprop-v3c-common-atlas-data, sprop-v3c-sei, sprop-v3c-tile-id, sprop-v3c-tile-id-pres, v3c-ptl-level-idc,

v3c-ptl-tier-flag, v3c-ptl-codec-idc, v3c-ptl-toolset-idc, and v3c-ptl-rec-idc, when present, **MUST** be included in the "a=v3cfmtp" line of SDP. This parameter is expressed as a media type string in the form of a semicolon-separated list of parameter=value pairs.

The **OPTIONAL** parameters, when present in the video media line V3C format parameters ("v3cfmtp") attribute, specify values that are considered static for the session.

An example of media representation corresponding to occupancy video component (V3C_OVD) in SDP is as follows:

```
m=video 49170 RTP/AVP 99
a=rtpmap:99 H265/90000
a=fmtp:99 sprop-max-don-diff=0;
a=v3cfmtp:sprop-v3c-unit-header=EAAAAA==
```

Below is an example of media representation corresponding to packed video component (V3C_PVD), where the static V3C parameter set, atlas data, and common atlas data are carried out-of-band in SDP. The values are considered static for the session, as they can't be signaled in-band in the video stream.

```
m=video 49170 RTP/AVP 99
a=rtpmap:99 H265/90000
a=v3cfmtp:sprop-v3c-unit-header=KAAAAA==;
  sprop-v3c-parameter-set=AUH/AAAP/zwAAAAAACgIAtEAgQLAIAAUQBACWAM5Q
  EDgQCAIAAAAABP8CzwAAAAAAAAAQAAAtAE/wLPAAAAAAAg=;
  sprop-v3c-atlas-data=SAGAFAQBaKjuXgABQEKA,SgHmIA==,LgFoDOAFAABaAA
  AAAAA+;
  sprop-v3c-common-atlas-data=YAEHgFA=,YgEAMAAAC/B0qcvv/Dbr/pTvb8oq
  fhC5JQVS9jn7kAQT/As9EFyrjRBcmxEQe+j5DuGbTT9mZmZAQAAAoA==
```

9.3. Grouping Framework

Different V3C components **MAY** be represented by their own respective RTP streams, whose payload formats are defined in the respective specifications. V3C atlas data RTP payload format is defined in this document, whereas the video component RTP payload formats are defined for example in [RFC6184] or [RFC7798]. A grouping tool, as defined in [RFC5888], is extended to indicate which media lines constitute a V3C representation. Further details on the new grouping type provided in Section 10.3.

The group attribute with V3C type is provided to allow application to identify "m" lines that belong to the same V3C bitstream. Grouping type V3C **MUST** be used with the group attribute. The tokens that follow are mapped to 'mid'-values of individual media lines in the SDP.

```
  a=group:V3C <tokens>
```

The following example shows an SDP including four media lines: three describing V3C video components (PT:96=occupancy, PT:97=geometry, and PT:98=attribute) and one describing a V3C atlas component (PT:100). All of these media lines are grouped under one V3C group. The V3C parameter set is provided via a session-level V3C media format parameter attribute.

```
...
a=group:V3C 1 2 3 4
a=v3cfmtp:sprop-v3c-parameter-set=AQD/AAAP/zwAAAAAADwIAQ5BwAAOADjgQ
  AADkA==
m=video 40000 RTP/AVP 96
a=rtpmap:96 H264/90000
a=v3cfmtp:sprop-v3c-unit-header=EAAAAA==
a=mid:1
m=video 40002 RTP/AVP 97
a=rtpmap:97 H264/90000
a=v3cfmtp:sprop-v3c-unit-header=GAAAAA==
a=mid:2
m=video 40004 RTP/AVP 98
a=rtpmap:98 H264/90000
a=v3cfmtp:sprop-v3c-unit-header=IAAAAA==
a=mid:3
m=application 40008 RTP/AVP 100
a=rtpmap:100 v3c/90000
a=v3cfmtp:sprop-v3c-unit-header=CAAAAA==;
a=mid:4
```

The example below describes how content with two atlases can be signaled as separate streams. The V3C parameter set is carried in a session-level V3C media format parameter attribute and common atlas data are carried as part of the media-level V3C media format parameter attribute corresponding to atlas zero. PT values 96, 97, 98, and 100 correspond to the occupancy, geometry, and attribute video components, as well as the atlas data component, for atlas zero. PT values 101, 102, 103, and 104 correspond to the respective components for atlas one.

```
...
a=group:V3C 1 2 3 4 5 6 7 8
a=v3cfmtp:sprop-v3c-parameter-set=AAUH/AAAP/zwAAABAADwIAWhBwAAOADjg
  QAADgAA8CAFoQcAADgA44EAAA6AkAgABRIA=;
m=video 40000 RTP/AVP 96
a=rtpmap:96 H264/90000
a=v3cfmtp:sprop-v3c-unit-header=EAAAAA==
a=mid:1
m=video 40002 RTP/AVP 97
a=rtpmap:97 H264/90000
a=v3cfmtp:sprop-v3c-unit-header=GAAAAA==
a=mid:2
m=video 40004 RTP/AVP 98
a=rtpmap:98 H264/90000
a=v3cfmtp:sprop-v3c-unit-header=IAAAAA==
a=mid:3
m=application 40008 RTP/AVP 100
a=rtpmap:100 v3c/90000
a=fmtp:100
  sprop-v3c-common-atlas-data=YAEHgFA=,YgEAMAAAa+96Z5v6VP1D+P7LzRsb
  WDJ/yz+ALzMZNfvCg2389Kjd+d6fZyM6QZBfhrDW3K0vaP2Rr8L+gLAq/ny3wAzs9
  veiXEjjS67MfH+H4xV/RgW4fkl/YkINe/OsWCOBwPAVLACCf4FnogwYZKIME6oiD9
  UCodqjLwCCf4FnogxqBiIMZNwiEBpJIduBUoCCf4FnogwOeSIMCaGiEA9VIdtGwwC
  Cf4FnogvB+aILvWIiEBB6IdqobKfmZmZoCmZmefmZmZoCmZmefmZmZoCmZmefmZmZ
  oCmZmdA=
a=v3cfmtp:sprop-v3c-unit-header=CAAAAA==;
a=mid:4
m=video 40010 RTP/AVP 101
a=rtpmap:101 H264/90000
a=v3cfmtp:sprop-v3c-unit-header=EAIAAA==
a=mid:5
m=video 40012 RTP/AVP 102
a=rtpmap:102 H264/90000
a=v3cfmtp:sprop-v3c-unit-header=GAIAAA==
a=mid:6
m=video 40014 RTP/AVP 103
a=rtpmap:103 H264/90000
a=v3cfmtp:sprop-v3c-unit-header=IAIAAA==
a=mid:7
m=application 40018 RTP/AVP 104
a=rtpmap:104 v3c/90000
a=v3cfmtp:sprop-v3c-unit-header=CAIAAA==
a=mid:8
```

9.4. Offer and Answer Considerations

9.4.1. Unicast

This section describes the negotiation of unicast streaming using the offer/answer model as described in [RFC3264]. V3C-coded content consists of an atlas bitstream and one or more video coded bitstreams, together known as V3C components. Atlas and video bitstreams are represented as separate media lines in the SDP.

During the session negotiation the offerer lists all V3C components available and informs the answerer which media lines **SHOULD** be consumed together. The answerer CAN select V3C components as suggested by the offerer, or select a subset of the V3C components by setting the port to zero for the undesired media lines in the answer. This allows the answerer to consume a subset of the V3C components in scenarios where it is fully or partially ignorant of the V3C coding scheme.

The following limitations and rules pertaining to the V3C atlas component media configuration apply:

- The parameters identifying the V3C atlas component media configuration is identified by v3c-ptl-level-idc, v3c-ptl-tier-flag, v3c-ptl-codec-idc, and v3c-ptl-toolset-idc. These media configuration parameters, except level-id, **MUST** be used symmetrically.
- Send only properties, identified by sprop-prefix, are considered declarative and **SHOULD** be omitted in the answers.

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

- maintain all configuration parameters with the values remaining the same as in the offer for the media format (payload type), with the exception that the value of v3c-ptl-level-idc is changeable as long as the highest level indicated by the answer is not higher than that indicated by the offer, or
- reject media line in which one or more of the parameter values are not supported by setting the port to zero in the answer.

The following limitations and rules pertaining to the V3C video component media configuration apply:

- The parameters identifying a video-coded V3C component media configuration format are according to the respective RTP video payload specification.

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

- maintain all configuration parameters with the values remaining the same as in the offer for the media format (payload type), with the exceptions specified in the respective RTP video payload specification, or
- reject the video coded V3C component media line completely when one or more of the parameter values are not supported by setting the port to zero in the answer.

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].

Below is an example of an offer that only sends V3C content. This example contains video components as three different versions (H.264, H.265, and H.266). Further differences between the alternatives would be signaled as part of the media attribute parameters, as is the practice with regular video streams.

```
...
a=group:v3c 1 2 3 4
a=v3cfmtp:v3c-ptl-level-idc=60;
  sprop-v3c-parameter-set=AQD/AAAP/zwAAAAAADwIAQ5BwAAOADjgQAADkA==
m=video 40000 RTP/AVP 96 97 98
a=rtpmap:96 H264/90000
a=rtpmap:97 H265/90000
a=rtpmap:98 H266/90000
a=v3cfmtp:sprop-v3c-unit-type=2;sprop-v3c-vps-id=0;
  sprop-v3c-atlas-id=0
a=sendonly
a=mid:1
m=video 40002 RTP/AVP 99 100 101
a=rtpmap:99 H264/90000
a=rtpmap:100 H265/90000
a=rtpmap:101 H266/90000
a=v3cfmtp:sprop-v3c-unit-type=3;sprop-v3c-vps-id=0;
  sprop-v3c-atlas-id=0
a=mid:2
a=sendonly
m=video 40004 RTP/AVP 102 103 104
a=rtpmap:102 H264/90000
a=rtpmap:103 H265/90000
a=rtpmap:104 H266/90000
a=v3cfmtp:sprop-v3c-unit-type=4;sprop-v3c-vps-id=0;
  sprop-v3c-atlas-id=0
a=mid:3
a=sendonly
m=application 40006 RTP/AVP 105
a=rtpmap:105 v3c/90000
a=v3cfmtp:sprop-v3c-unit-type=1;sprop-v3c-vps-id=0;
  sprop-v3c-atlas-id=0;
a=mid:4
a=sendonly
```

This is an example of an answer that only receives V3C data with the selected versions:

```
...
a=group:v3c 1 2 3 4
m=video 50000 RTP/AVP 96
a=rtpmap:96 H264/90000
a=recvonly
a=mid:1
m=video 50002 RTP/AVP 100
a=rtpmap:100 H265/90000
a=recvonly
a=mid:2
m=video 50004 RTP/AVP 104
a=rtpmap:104 H266/90000
a=recvonly
a=mid:3
m=application 50006 RTP/AVP 105
a=rtpmap:105 v3c/90000
a=recvonly
a=mid:4
```

This is an example of an offer that allows bundling different V3C components into one stream, based on [RFC9143]:

```
...
a=group:BUNDLE 1 2 3 4
a=group:v3c 1 2 3 4
m=video 40000 RTP/AVP 96
a=rtpmap:96 H264/90000
a=v3cfmtp:sprop-v3c-unit-type=2;sprop-v3c-vps-id=0;
  sprop-v3c-atlas-id=0
a=mid:1
a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:mid
m=video 40002 RTP/AVP 97
a=rtpmap:97 H264/90000
a=v3cfmtp:sprop-v3c-unit-type=3;sprop-v3c-vps-id=0;
  sprop-v3c-atlas-id=0;
a=mid:2
a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:mid
m=video 40004 RTP/AVP 98
a=rtpmap:98 H264/90000
a=v3cfmtp:sprop-v3c-unit-type=4;sprop-v3c-vps-id=0;
  sprop-v3c-atlas-id=0
a=mid:3
a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:mid
m=application 40006 RTP/AVP 99
a=rtpmap:99 v3c/90000
a=v3cfmtp:sprop-v3c-unit-type=1;sprop-v3c-vps-id=0;
  sprop-v3c-atlas-id=0;
  sprop-v3c-parameter-set=AQD/AAAP/zwAAAAAADwIAQ5BwAAOADjgQAADkA==
a=mid:4
a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:mid
```

This is an example of an answer that accepts the bundling of different V3C components:

```
a=group:BUNDLE 1 2 3 4
a=group:v3c 1 2 3 4
m=video 50000 RTP/AVP 96
a=rtpmap:96 H264/90000
a=mid:1
a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:mid
m=video 0 RTP/AVP 97
a=rtpmap:97 H264/90000
a=bundle-only
a=mid:2
a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:mid
m=video 0 RTP/AVP 98
a=rtpmap:98 H264/90000
a=bundle-only
a=mid:3
a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:mid
m=application 0 RTP/AVP 99
a=rtpmap:99 v3c/90000
a=bundle-only
a=mid:4
a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:mid
```

9.4.2. Multicast

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

- The atlas V3C component media configuration is identified by v3c-ptl-level-idc, v3c-ptl-tier-flag, v3c-ptl-codec-idc, and v3c-ptl-toolset-idc. These atlas format configuration parameters **MUST** be used symmetrically; that is, the answerer **MUST** either maintain all configuration parameters or reject the media line, including any associated video coded V3C component media lines. This implies that v3c-ptl-level-idc for offer/answer in multicast is not changeable.
- The video-coded V3C component media configuration format is according to the respective RTP video payload specification.
- To simplify the handling and matching of these configurations, the same RTP payload type number used in the offer **MUST** also be used in the answer.
- Parameter sets received **MUST** be associated with the originating source and **MUST** only be used in decoding the incoming bitstream from the same source.

9.5. Declarative SDP Considerations

When V3C content over RTP is offered with SDP in a declarative style, the parameters capable of indicating both bitstream properties as well as answerer capabilities are used to indicate only bitstream properties. For example, in this case, the parameters v3c-ptl-level-idc, v3c-ptl-tier-flag, v3c-ptl-codec-idc, v3c-ptl-toolset-idc, and v3c-ptl-rec-idc declare the values used by the bitstream, not the capabilities for receiving bitstreams.

An answerer of the SDP is required to support all parameters and values of the parameters provided; otherwise, the answerer **MUST** reject or not participate in the session. It falls on the creator of the session to use values that are expected to be supported by the receiving application.

10. IANA Considerations

This document contains three IANA considerations: a new media type, a new SDP attribute, and a new grouping type.

10.1. V3C Media Type Registration

IANA has registered the following media type in the "Media Types" registry.

Type name: application

Subtype name: v3c

Required parameters: sprop-v3c-parameter-set

Optional parameters: sprop-v3c-unit-header, sprop-v3c-unit-type, sprop-v3c-vps-id, sprop-v3c-atlas-id, sprop-v3c-attr-idx, sprop-v3c-attr-part-idx, sprop-v3c-map-idx, sprop-v3c-aux-video-flag, sprop-v3c-tile-id, sprop-v3c-tile-id-pres, sprop-v3c-atlas-data, sprop-v3c-common-atlas-data, sprop-v3c-sei, v3c-ptl-level-idc, v3c-ptl-tier-flag, v3c-ptl-codec-idc, v3c-ptl-toolset-idc, v3c-ptl-rec-idc, and sprop-max-don-diff.

Encoding considerations: framed

Security considerations: See Section 11 of RFC 10034.

Interoperability considerations: N/A

Published specification: RFC 10034

Applications that use this media type: Any application that relies on V3C-based media services over RTP.

Fragment identifier considerations: N/A

Additional information: N/A

Person & email address to contact for further information: Lauri Ilola (lauri.ilola@nokia.com) or Lukasz Kondrad (lukasz.kondrad@nokia.com)

Intended usage: COMMON

Restrictions on usage: This media type depends on RTP framing and, hence, is only defined for transfer via RTP [RFC3550]. Transport within other framing protocols is not defined at this time.

Author: See the Authors' Addresses section of RFC 10034.

Change controller: IETF

10.2. V3C Format Parameters SDP Attribute

IANA has registered the following SDP attribute in the "attribute-name (formerly 'att-field')" registry under the "Session Description Protocol (SDP) Parameters" registry group.

Contact name: See the Authors' Addresses section of RFC 10034.

Contact email address: See the Authors' Addresses section of RFC 10034.

Attribute name: v3cfmtp

Attribute syntax:

```
v3cfmtp-value = byte-string
```

Notes:

- The V3C format parameters are V3C media type parameters and need to reflect their syntax.
- "byte-string" is as defined in [RFC8866].
- ABNF grammar is as defined in [RFC5234].

Attribute semantics: "v3cfmtp-value" is a byte-string, as defined in [RFC8866], that contains at least one V3C-specific media format parameter as a "parameter=value" pair, as defined in this document. Multiple semicolon-separated V3C media "parameter=value" pairs can be stored in the byte-string to be conveyed by SDP and given unchanged to the media tool that will use this format. Whitespace in the byte-string is ignored.

Attribute value: v3cfmtp-value

Usage level: session, media

Charset dependent: No

Purpose: This attribute allows parameters that are specific to a V3C format to be conveyed in a way that SDP does not have to understand them. It allows associating V3C-specific parameters with a session or with any media line. Parameters signaled as part of session-level attributes take effect when conflicting parameters are signaled as media-level attributes.

O/A procedures: The v3cfmtp attribute can be present both in offers and answers.

Mux Category: NORMAL

Reference: RFC 10034

Type	SDP Name	Usage Level	Mux Category	Reference
attribute	v3cfmtp	session, media	NORMAL	RFC 10034

Table 3: The attribute-name (formerly 'att-field') Registry

10.3. V3C Grouping Type Extension

The SDP Group attribute is extended to establish relationships between sub-streams of a V3C representation. IANA has registered the following in the "Semantics for the 'group' SDP Attribute" registry under the "Session Description Protocol (SDP) Parameters" registry group:

Semantics	Token	Mux Category	Reference
V3C grouping	V3C	NORMAL	RFC 10034

Table 4: The Semantics for the 'group' SDP Attribute Registry

11. 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 [RFC7202] discusses, it is not an RTP payload format's responsibility to discuss or mandate what solutions are used to meet the basic security goals like confidentiality, integrity, and source authenticity for RTP in general. This responsibility lies with anyone using RTP in an application. They can find guidance on available security mechanisms and important considerations in [RFC7201].

This document does not mandate a specific security mechanism. Instead, applications are responsible for selecting mechanisms that follow current best practices for confidentiality, integrity, and source authentication and that reflect the evolving security landscape beyond what is covered in [RFC7201]. For modern best practices, applications can consider the following options:

(D)TLS-based protection: For guidance on using TLS 1.3 and DTLS, applications should refer to [BCP195], which provides up-to-date recommendations.

IPsec-based protection: Relevant and current protocol specifications include [RFC4303] (ESP) and [RFC7296] (IKEv2).

The rest of the Security Considerations section discusses the security impacting properties of the payload format itself.

A V3C session can consist of multiple sub-streams carried over different RTP streams. Security considerations such as source authentication **SHOULD** be applied to all its constituent sub-streams. All receivers of V3C data **SHOULD** exercise source caution and only receive data from senders that they can trust. Furthermore, this RTP payload format supports multiple RTP streams for different components necessary to produce the decoded output; thus, it depends on all RTP streams and signaling components, e.g., SDP and RTCP, being authentic to what the sender intended.

This RTP payload format and its media decoder do not exhibit 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. Furthermore, this payload format contains no active content.

Components of a system using this media type **SHALL NOT** construct RTP payloads that contain executable content. The implementer of the RTP payload format **SHALL** guarantee that the received content is properly de-packetized and fed to a V3C standard compliant decoder. What the receiver does with the decoded bitstream is unspecified.

12. References

12.1. Normative References

[ISO.IEC.23090-5] ISO/IEC, "Information technology -- Coded representation of immersive media -- Part 5: Visual volumetric video-based coding (V3C) and video-based point cloud compression (V-PCC)", ISO/IEC 23090-5:2026, 2026, <https://www.iso.org/standard/91546.html>.

[ISO.IEC.23090-12] ISO/IEC, "Information technology -- Coded representation of immersive media -- Part 12: MPEG Immersive video (MIV)", ISO/IEC 23090-12:2025, 2025, <https://www.iso.org/standard/87643.html>.

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

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

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

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <https://www.rfc-editor.org/info/rfc4648>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <https://www.rfc-editor.org/info/rfc5234>.

[RFC5888] Camarillo, G. and H. Schulzrinne, "The Session Description Protocol (SDP) Grouping Framework", RFC 5888, DOI 10.17487/RFC5888, June 2010, <https://www.rfc-editor.org/info/rfc5888>.

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

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

[RFC8866] Begen, A., Kyzivat, P., Perkins, C., and M. Handley, "SDP: Session Description Protocol", RFC 8866, DOI 10.17487/RFC8866, January 2021, <https://www.rfc-editor.org/info/rfc8866>.

[RFC9143] Holmberg, C., Alvestrand, H., and C. Jennings, "Negotiating Media Multiplexing Using the Session Description Protocol (SDP)", RFC 9143, DOI 10.17487/RFC9143, February 2022, <https://www.rfc-editor.org/info/rfc9143>.

12.2. Informative References

[BCP195] Best Current Practice 195, <https://www.rfc-editor.org/info/bcp195>.
At the time of writing, this BCP comprises the following:

Moriarty, K. and S. Farrell, "Deprecating TLS 1.0 and TLS 1.1", BCP 195, RFC 8996, DOI 10.17487/RFC8996, March 2021, <https://www.rfc-editor.org/info/rfc8996>.

Sheffer, Y., Saint-Andre, P., and T. Fossati, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 9325, DOI 10.17487/RFC9325, November 2022, <https://www.rfc-editor.org/info/rfc9325>.

[ISO.IEC.14496-10] ISO/IEC, "Information technology - Coding of audio-visual objects - Part 10: Advanced video coding", ISO/IEC 14496-10:2025, 2025, <https://www.iso.org/standard/87574.html>.

[ISO.IEC.14496-12] ISO/IEC, "Information technology - Coding of audio-visual objects - Part 12: ISO base media file format", ISO/IEC 14496-12:2026, 2026, <https://www.iso.org/standard/85596.html>.

[ISO.IEC.23008-2] ISO/IEC, "Information technology - High efficiency coding and media delivery in heterogeneous environments - Part 2: High efficiency video coding", ISO/IEC 23008-2:2025, 2025, <https://www.iso.org/standard/90502.html>.

[ISO.IEC.23009-1] ISO/IEC, "Information technology - Dynamic adaptive streaming over HTTP (DASH) - Part 1: Media presentation description and segment formats", ISO/IEC 23009-1:2022, 2022, <https://www.iso.org/standard/83314.html>.

[ISO.IEC.23090-3] ISO/IEC, "Information technology - Coded representation of immersive media - Part 3: Versatile video coding", ISO/IEC 23090-3:2024, 2024, <https://www.iso.org/standard/86516.html>.

[ISO.IEC.23090-10] ISO/IEC, "Information technology - Coded representation of immersive media - Part 10: Carriage of visual volumetric video-based coding data", ISO/IEC 23090-10:2022, 2022, <https://www.iso.org/standard/78991.html>.

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

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

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <https://www.rfc-editor.org/info/rfc4303>.

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

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

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

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

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

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

[RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October 2014, <https://www.rfc-editor.org/info/rfc7296>.

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

Authors' Addresses

Lauri Ilola
Nokia Technologies
Hatanpaeaen valtatie 30
FI-33100 Tampere
Finland
Email: lauri.ilola@nokia.com

Lukasz Kondrad
Nokia Technologies
Werinherstrasse 91
D-81541 Munich
Germany
Email: lukasz.kondrad@nokia.com